Universidade do Minho
Escola de Engenharia

Ricardo Sousa Teixeira Pinto

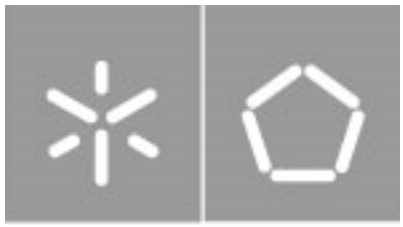
Inteligência de negócio na área de seguros: Seleção de funcionalidades de apólices

Janeiro de 2023

Inteligência de negócio na área de seguros: Seleção de funcionalidades de apólices

Ricardo Pinto

UMinho | 2023



Universidade do Minho
Escola de Engenharia

Ricardo Sousa Teixeira Pinto
A71112

Inteligência de negócio na área de seguros: Seleção de funcionalidades de apólices

Dissertação de Mestrado
Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação do
Professor Doutor Henrique Manuel Dinis Santos

Janeiro de 2023

DIREITOS DE AUTOR

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio, nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Atualmente no setor das seguradoras existem diferentes tipos de seguros, mas nenhuma seguradora decidiu ainda criar um seguro que abranja a perda de informação por parte de uma entidade organizacional. Com a crescente adoção das novas tecnologias por parte das organizações, no âmbito da transação digital, vem também com estas um risco associado à perda ou partilha involuntária de informação por parte destas. O objetivo desta tese passa por desenvolver um modelo que permite calcular o risco de uma organização perder informação que manuseia. Para alcançar este objetivo usamos como parâmetros diferentes propriedades macros referente às organizações.

Foi assim desenvolvido um modelo preditivo de cálculo de risco com base em informações de uma organização. Este projeto foi desenvolvido com base em dados obtidos na plataforma kaggle, aplicando sobre estas técnicas de tratamento que permitam tirar insights dos mesmos e desta forma possam ser usados em algoritmos de machine learning, tais como: Support Vector Machine e Logistic Regression. Foi, ainda, desenvolvida um API que permite usar o modelo gerado.

Os resultados mostram que o modelo preditivo gerado obtém previsões fiáveis tendo em conta os dados usados. É de referir que o número total de dados usados para alimentar o algoritmo foi por volta de mil instâncias, o que tem um impacto bastante significativo no modelo gerado, pois quanto maior o número de linhas maior a acuidade do modelo. Assim, concluímos que o modelo proposto nesta tese é capaz de quantificar o risco associado a determinada organização através de datasets que embora com um pequeno número de dados aplicados a algoritmos de inteligência artificial adequados a quantidades de dados mais pequenas.

Palavras-chave: Aprendizagem máquina; Segurança e Informação; Seguros; Sistemas de apoio à decisão.

ABSTRACT

Nowadays in the insurance sector there are different types of insurance, but no insurance company has yet decided to create an insurance covering the loss of information by an organizational entity. With the growing adoption of new technologies by organizations, there is also a risk associated with the loss or involuntary sharing of information by them. The objective of this thesis is to develop a model that allows calculating the risk of an organization losing information that it handles. To achieve this objective, we use different macro properties referring to organizations as parameters.

A predictive risk calculation model based on information from an organization was thus developed. This project was developed based on data found on the kaggle platform, applying on these treatment techniques that allow taking insights from them and so that they can also be used in certain machine learning algorithms, such as Support Vector Machine and Logistic Regression. Finally, an API was developed that allows using the generated model.

The results show that the generated predictive model obtains reliable predictions considering the data used. It should be noted that the total number of data used to feed the algorithm was around a thousand lines, which has a very significant impact on the generated model, since the greater the number of lines, the more the model's accuracy will decrease. Thus, we conclude that the model proposed in this thesis can quantify the risk associated with a given organization through datasets with a small amount of data.

Keywords: Information Security Risk; Insurance; Support Decision systems; Machine Learning.

ÍNDICE

LISTA DE ABREVIATURAS/SIGLAS

ETL - Extract, Transform and Load

ROC – Receiver operating characteristic

AUC - Area under the ROC Curve

BI – Business Intelligence

DW – Datawarehouse

IoT – Internet of Things

Csv - Comma-separated values

ML – Machine Learning

KNN – K-nearest Neighbors

RF – Random Forest

CART – Classification And Regression Trees

LR – Logistic Regression

RE – Relief

SU – Symmetrical Uncertainty

CFS – Correlation-based Feature Selection

LISTA DE FIGURAS

LISTA DE TABELAS

1. INTRODUÇÃO

Nos tempos que correm é possível fazer seguros para objetos, pessoas e até animais, mas ainda nenhuma seguradora decidiu avançar para a criação de um seguro de informação. Tipicamente, o risco de um seguro é feito com base numa avaliação feita à pessoa, questionando esta perguntas como a idade, estado civil, ocupação, condição de saúde, tempo de carta, o local onde esta habita, entre muitas outras. O risco pode variar muito de caso para caso, pois a alteração de um parâmetro pode significar um aumento ou diminuição do risco aos olhos da seguradora. As seguradoras podem, atualmente, nesta nova era tecnológica, prever se um determinado cliente encaixa num determinado tipo de seguro com base em diferentes parâmetros. Este projeto fez uso de atributos mais macros das organizações para que com bases neles, fosse possível obter uma previsão que permite atribuir um determinado risco e por consequência um determinado prémio de seguro à organização. Atributos como o número de empregados total, lucro anual gerado e até se a organização já sofreu algum tipo de ataque que levou à perda de informação.

Atualmente no setor das seguradoras não existe nenhuma que faça seguros para a perda de informação de uma organização. Com a crescente transição digital por parte das organizações, forçosamente, irá haver também, mais perda de informação por parte destas, seja de forma voluntária ou involuntária. Ao aplicar técnicas de ML é possível extrair e criar um modelo que irá permitir prever a probabilidade de fuga de informação por parte de uma organização.

Este projeto procura prever a probabilidade que uma organização tem de ferir a segurança de informação, quer na confidencialidade, integridade, disponibilidade e Privacidade (RGPD) de informação que lhe pertence, ou está a seu cargo, de forma involuntária. Para cumprir este objetivo, usamos a abordagem de ML e pré-processamento de dados para assim obter o modelo com acuidade.

Esta tese encontra-se organizada por capítulos que serão descritos de seguida.

No capítulo referente à revisão de literatura são abordados estudos no ramo das seguradoras e como é que estas implementaram modelos de machine learning, diferentes abordagens para lidar com o desequilíbrio nos dados e as métricas de desempenho mais adequadas.

De seguida temos o capítulo referente à metodologia de investigação que descreve o contexto em que este projeto se insere, a metodologia usada para elaborar o projeto bem como os passos que foram feitos para a elaboração do mesmo.

No capítulo da implementação explicamos como é que este projeto foi implementado e quais os passos que seguimos, desde o carregamento dos dados na plataforma de computação web até à criação da API. São detalhados os algoritmos de machine learning utilizados e os indicadores de desempenho. Assim como é descrita a forma como aplicamos os algoritmos de machine learning e os resultados obtidos.

Na análise dos resultados foram aplicadas as métricas relativamente aos algoritmos usados, que permitiu destacar o algoritmo que demonstrou um melhor desempenho.

No último capítulo apresentamos as conclusões e contribuições dadas por este projeto, onde delineamos orientações para trabalhos futuros.

2. REVISÃO DE LITERATURA

Para a formulação do problema, bem como para a análise de possíveis soluções procedemos ao levantamento do estado da arte dos seguintes conceitos, de Business Intelligence (BI), BI na área de seguros, Metodologia CRISP DM, Machine Learning (ML) e ML na área de seguros.

2.1 - Gestão de risco de segurança

Segundo a ISO 27001 a gestão de risco consiste em dois elementos fundamentais: a avaliação do risco e o tratamento do risco. A avaliação de risco consiste no processo de identificar os riscos de segurança de informação e determinar a sua probabilidade e impacto na organização, ou seja, a organização tem de perceber quais podem vir a ser os potenciais problemas com a sua informação e qual a probabilidade de se vir a suceder e quais as consequências que daí advém. Já o tratamento do risco consiste em salvaguardar a organização para essas potenciais ocorrências.

2.1.1 - Confidencialidade, Integridade e Disponibilidade

A segurança dos dados é um problema constante na área das tecnologias de informação. Segundo Sun, Y. et al (2014) e Latif R. et al (2014) a segurança dos dados torna-se particularmente mais séria quando falamos na cloud, pois esta significa que os dados estão espalhados por diferentes máquinas e dispositivos de armazenamento, incluindo servidores, computadores e vários equipamentos móveis, como por exemplos os telemóveis.

Existem vários serviços que são fornecidos pela cloud no espectro da computação. Sun, Y. et al (2014) e Kshetri N. (2013) referem que as organizações e empresas estão atualmente a adaptar os seus negócios adotando a computação na cloud para reduzir os seus custos.

Sun, Y. et al (2014) referem também que os principais problemas na segurança dos dados na cloud incluem a privacidade de dados, disponibilidade de dados, proteção de dados, localização de dados e a transmissão segura destes.

Sun, Y. et al (2014) descrevem diferentes técnicas para mitigar a perda de dados, como por exemplo a integridade dos dados, que significa proteger estes da sua total extinção,

modificação ou até a sua fabricação não autorizada e também gerir quais entidades podem ou não ter as permissões e os direitos para manusear os dados de forma a não permitir que estes sejam desviados ou roubados. A confidencialidade dos dados é também um aspeto importante para permitir armazenar dados privados ou confidenciais, aplicando nestas estratégias de autenticação e controlo de acesso. Sun, Y. et al (2014) mencionam diferentes formas para garantir a confidencialidade dos dados, como a Criptografia Homomórfica, criptografia de dados, armazenamento distribuído, técnica híbrida, ocultação de dados e a confirmação de exclusão de dados. Sun, Y. et al (2014) concluem dizendo que ainda existem muitas lacunas a serem preenchidas, de forma a tornar as técnicas mencionadas mais eficazes.

2.1.2 - Ataques e Impactos de Segurança:

A definição de um ciberataque segundo Roscini, "trata-se de Operações, seja em ofensa ou defesa, destinadas a alterar, excluir, corromper ou negar acesso a dados de computador ou software para fins de: (a) propaganda ou engano; e/ou (b) interromper parcial ou totalmente o funcionamento do computador, sistema de computador ou rede de destino e infraestrutura física operada por computador relacionada (se houver); e/ou (c) produzir danos físicos extrínsecos ao computador, sistema informático ou rede." (Roscini, 2014: p. 17)

Um dos ataques mais comum é o DoS, que consiste num grande número de pedidos aos servidores da organização alvo com o objetivo de sobrecarregar estes de forma a tornar o serviço lento/indisponível.

Nos artigos elaborados por Ou et al, (2022) e Ko, M. (2006), estes referem que os três tipos de ataques mais comuns são o Denial of Service (DoS), ataque por vírus e roubo de informações, definindo estes da seguinte forma:

Já o ataque por vírus dá acesso ao programa host por meio de um executável, permitindo assim a quem criou o vírus prosseguir com aquilo que tem planeado, podendo isto ser um grande número variado de coisa, como por exemplo passar uma mensagem, apagar dados ou até mesmo roubar dinheiro.

Por fim ele refere o roubo de informações que consiste como nome indica no acesso não autorizado dos dados de clientes ou outras pessoas de determinada plataforma. Estes dados podem ser registos médicos, nomes, detalhes do cartão de crédito, datas de nascimento,

comportamento de compras online, etc. Estes ataques são considerados como uma violação de confidencialidade.

Os ataques às organizações visam essencialmente, quebrar a continuidade do negócio e exfiltração de dados. Estes ataques são algo que todas as organizações que manuseiam dados estão sujeitas a ser um alvo ou pode apenas ser uma ação involuntária por parte do colaborador da organização. Estes ataques podem ser causados por Hacking & Malware, Inside Job, Divulgação não intencional e outras formas (Mouna Jouini et al, 2014 & Geric S . et al (2007)). Existem outras formas de perder os dados como falha no hardware, falhas elétricas, a falta de uma política de segurança de informação, corrupção de dados e falta de backup por exemplo. Mas neste projeto apenas foram recolhidos dados com as primeiras quatro formas de perda de informação referidas. Estas vulnerabilidades que as organizações podem ser alvo de exploração por não estarem tratadas ou mitigadas, são uma janela de oportunidade para agentes mal-intencionados, pois estes podem facilmente explorar estes pontos em benefício próprio, podendo isto significar perdas monetárias significativas, (Mouna Jouini et al, 2014). Segundo uma pesquisa realizada por McCue, A. (2008), este indica que 70% dos ataques são realizados por “insiders” e não diretamente por criminosos externos, este também refere que 90% das medidas de segurança implementadas estão orientadas para as ameaças externas á organização.

2.2- Business Intelligence na área de seguros:

O Business Intelligence permite que as seguradoras obtenham informações sobre as tendências do mercado e por consequência agir mais rapidamente, permitindo também alcançar serviços ou produtos inovadores. O autor, Ouda, G. K. (2018), citado por SA Ghasemi et al (2022) considera que as seguradoras reconhecem a importância do Business Intelligence e o impacto que estas têm, dando o exemplo dos funcionários que podem facilmente converter o seu conhecimento servindo-se de inteligência analítica ou até a identificação dos clientes mais leais através das razões que levam a essa lealdade e também identificar possíveis novos clientes. Ouda, G. K. (2018) divide BI na indústria das seguradoras em três níveis, sendo que o primeiro corresponde à camada de fonte de dados que representa as diferentes fontes que alimentam uma Datawarehouse e podem estar em diferentes formatos, como por exemplo um ficheiro excel, um csv, uma base de dados relacional ou mesmo outro tipo de

formatos. Estes dados podem ser extraídos de claims, dados sobre os prémios, dados sobre o marketing interno da organização, registos do servidor relativos à navegação do utilizador e até dados com base nos "clicks" efetuados. O segundo nível é o Warehouse layer que consiste numa coleção de informação de forma a auxiliar a organização nas suas decisões. Antes dos dados chegarem ao Data Warehouse estes ainda têm de passar por um processo denominado de Extract, Transform and Load ou ETL, que vai como o próprio nome indica, inicialmente extrair os dados do ficheiro ou da plataforma onde se encontra, de seguida irá transformar estes dados num único formato e por último estes são carregados para o DataWarehouse. O último nível corresponde aos Relatórios/Apresentações, pois é neste nível que os dados são compilados em informação e esta permite uma tomada de decisão mais informada. Pode, ainda, ser possível identificar regras de negócio.

2.3 - Aplicações de BI na área de seguros:

O Business Intelligence é aplicado em muitas áreas relativas a seguradoras, como por exemplo a Customer Relationship Management (CRM), Performance Management (PM), Risk Management (RM) G. K. (2018), entre outras.

Esta revisão de literatura serve para identificar o estado de arte no que toca às tecnologias de BI e ao seu uso na área de seguros e de que forma é que as tecnologias, IA e deep learning tornam possível a melhoria dos diferentes processos das entidades seguradoras. Na área das seguradoras já existem alguns artigos relacionados com a implementação de ferramentas de BI, desde Datawarehouse (DW), algoritmos de machine learning que auxiliam alguns processos das seguradoras. No entanto, ainda não conseguimos identificar nenhum artigo que tenha feito o que este projeto se propõe a fazer que é através de ferramentas de BI identificar perfis de clientes e com base nesta informação atribuir o conjunto de funcionalidades, para assim desenvolver uma solução personalizada da oferta do seguro. Os dados desempenham um papel crucial no que toca a seguros e juntamente com a era do digital tem crescido também a quantidade de dados que é gerada nos últimos anos, sendo que em grande parte é devido à Internet das coisas (IoT). As seguradoras sendo organizações procuram, de forma contínua, melhorar a sua adaptação à evolução das TI, nomeadamente, os dados de forma a tirar o máximo partido destes e criando valor. Ouda, G. K. (2018) e W. McDonald (2015) reconhecem que o Business Intelligence auxilia as organizações a passar de uma análise de

negócio manual para uma análise mais eficiente, rápida e automatizada, através de recursos tecnológicos poderosos que conseguem realizar análises detalhadas, onde o trabalho recai sobre os recursos computacionais como processadores e hardware relativo ao armazenamento dos dados. Desta forma, torna-se mais fácil descobrir informação a partir dos dados, alimentando assim a gestão da organização e, consequentemente, os processos de BI devem ser realizados de forma integrada e próxima das funções de gestão. Caso contrário, todo o output das ferramentas de BI nem sempre são compreendidas no imediato o que pode levar a atrasos na compreensão do conhecimento gerado e por consequência este pode ter um impacto negativo ou inerte na organização. As seguradoras não existem sem os seus clientes e, portanto, é muito importante proceder ao tratamento das informações sobre as preferências dos seus clientes e que consigam integrar estas o mais rapidamente possível naquilo que são os seus processos de forma a dar uma reposta ágil às solicitações que se estão em constante mudança. Na revisão de literatura foi analisado um estudo feito a uma seguradora do Iraque que apresenta problemas, principalmente, no que toca à comunicação entre os seus departamentos e filiais que estão espalhadas por todo o país. Esta seguradora iraquiana não consegue tratar, quer por falta de meios quer pela quantidade, a informação, pois esta é recebida através do correio eletrónico e quando as filiais e/ou departamentos precisam de alguma informação à cerca de algum cliente terá de contactar a empresa mãe que, só passado 7 a 14 dias é que as informações são fornecidas ao departamento ou filial, o que significa que o processo é bastante demorado e além disso a seguradora só consegue fornecer relatórios estáticos com imprecisões e não conseguem proporcionar qualquer tipo de análise preditiva. A solução que é mencionada neste estudo passa por uma solução de Business Intelligence que consiste nas camadas já referidas nesta anteriormente, começando na aquisição de dados, passando por um processo de ETL e finalmente a criação dos relatórios e dashboards através de operações de Drill-down e/ou Roll-up. G. K. (2018). Neste estudo, conclui-se que as ferramentas de BI podem auxiliar muito as seguradoras a gerir o risco, a detetar fraudes, a ganhar maior perceção da sua rentabilidade, a determinar tendências comerciais e sobretudo a tomar decisões atempadas e assertivas.

2.4- Machine Learning na área de seguros:

2.4.1 - Metodologia Crisp-DM :

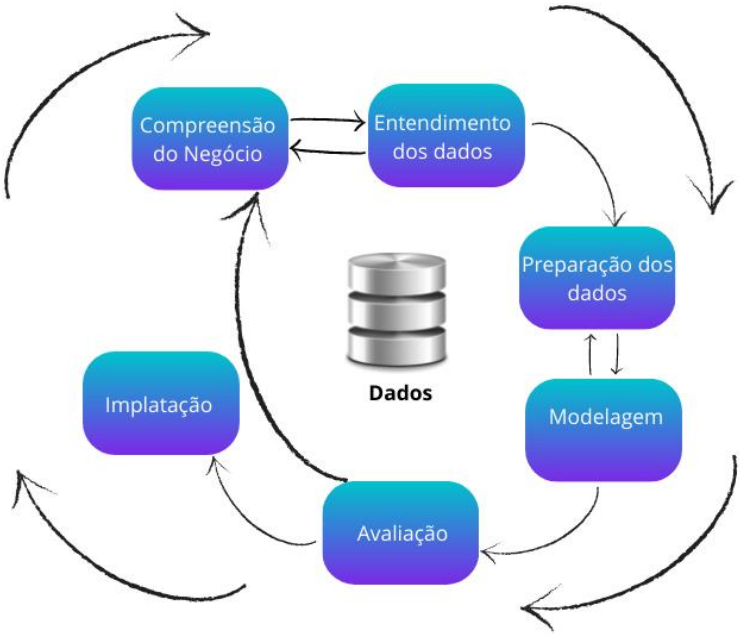


Figura 1: Diagrama de processo CRISP-DM (Fonte Própria)

Segundo Schröer, C. et al (2021), Azevedo, A. et al (2008) e Wirth, R. et al (2000) as fases da metodologia CRISP-DM podem ser descritas da seguinte forma:

Compreensão do Negócio

Esta primeira fase do processo é crucial para todo o projeto. Pois é a partir desta que se estabelece o contexto, o tipo de problema e quais os objetivos.

Compreensão dos dados

Nesta fase os dados são recolhidos, explorados e descritos de forma a perceber em que estado se encontram, permitindo assim também compreender quais é que devem ser usados e quais é devem ser descartados.

Preparação dos dados

Nesta etapa é definido quais as variáveis a utilizar e quais é que devem ser retiradas. Depois é necessário fazer uma limpeza aos restantes dados, onde alguns deles podem vir a sofrer transformações dependendo da situação. Também poderá ser necessário a integração de novos dados juntamente com os existentes de forma a estabelecer uma relação entre estes.

Modelagem e avaliação

A fase de modelagem consiste em selecionar a técnica que vai ser usada. Esta escolha vai depender do objetivo principal e também dos dados selecionados. Já a fase de avaliação consiste em comparar nos resultados obtidos com os objetivos de negócio estabelecidos.

Implementação

Esta fase consiste no planeamento da implantação, monitorização e manutenção do modelo gerado.

2.4.2 - Machine learning :

Existem quatro tipos de algoritmos de machine learning, sendo eles, supervisionado, não supervisionado, semi-supervisionado e por último supervisionado por reforço.
Segundo Mahesh, B. (2020) na aprendizagem supervisionada o algoritmo irá receber os dados etiquetados, ou seja, nós definimos um target, o output e os restantes dados são inputs. Este tipo de algoritmo subdivide-se em classificação e regressão.
Já na aprendizagem não supervisionada Mahesh, B. (2020) refere que os dados não estão etiquetados e, ou seja, não haverão dados de treino para ensinar o modelo nem o objetivo final. Este tipo de aprendizagem é usado para tentar encontrar relacionamentos, regras e padrões nos dados, de forma a tentar encontrar algo implícito nestes. Subdivide-se em agrupamento, associação e redução de dimensão.

Semi-supervisionado é um tipo de aprendizagem semelhante ao supervisionado, mas usa dados etiquetados para treino e não etiquetados para teste, permitindo assim que o algoritmo possa etiquetar os dados que não foram etiquetados, Mahesh, B. (2020).

Por fim temos a aprendizagem por reforço que consiste em treinar modelos de machine learning para tomar uma sequência de decisões, obrigando o computador a tentar e a errar para encontrar uma solução ao problema encontrado. Neste caso a IA irá receber penalizações ou recompensas (Mahesh, B. (2020) de acordo com as decisões que escolhe. Sendo o objetivo final deste tipo de aprendizagem maximizar as recompensas.

Segundo Charbuty, B. et al, (2021) o algoritmo Decision Tree é uma árvore de nós designada para criar uma decisão sobre a afiliação de valores a uma classe ou uma estimativa de um valor numérico. Os nós representam a divisão para outro atributo. Na classificação são separados os diferentes valores relativos a diferentes classes.

Já o Naive Bayes é um classificador de alto viés e baixa variância que pode construir um bom modelo mesmo com um pequeno conjunto de dados. É simples de usar e computacionalmente de baixo custo segundo Webb, G. I. et al,(2010). Normalmente, o seu uso envolve categorização de texto, incluindo deteção de spam, análise de sentimentos e sistemas de recomendação.

O algoritmo de regressão logística é usado para problemas de classificação quando o output é binário, isto é, quando existem só duas opções possíveis.

O Random forest é um algoritmo de supervisão que é bastante usado em problemas tanto de regressão como de classificação. Como o próprio nome indica e segundo Liu, Y., et al (2012) este algoritmo vai gerar um conjunto de árvores de decisão (forest).

2.4.3 - Machine learning na área de seguros:

Num estudo realizado por Hanafy, M., & Ming, R. (2022) foram usados três datasets para analisar claims, onde foram analisados com base num target categórico, o que levou a usarem algoritmos de classificação. O primeiro passo do processo de machine learning é a aquisição dos dados, e neste estudo foram retirados da plataforma Kaggle. Em seguida, os dados foram transformados para que fosse possível serem usados pelos algoritmos, pois a maioria dos algoritmos de ML não consegue processar dados categóricos, sendo que estes dados devem então ser transformados em formato numérico. Como classificadores foi aplicado o K-nearest

Neighbor(KNN), Random Forest(RF), Decision Tree(CART) e por último Logistic Regression (LR).

O K-nearest Neighbor(KNN) é uma maneira fácil e simples de implementar um algoritmo de supervisão que pode ser usado para resolver problemas de regressão e classificação.

Já o algoritmo Random Forest, como o próprio nome indica vai criar várias árvores de decisão de maneira aleatória.

A Decision Tree é dos modelos de machine learning mais úteis e comuns, pois este tipo de modelo permite uma interpretação fácil por parte do utilizador.

Por último temos o Logistic Regression que é uma técnica de Supervised Learning, é usado para prever variáveis categóricas dependentes através de um conjunto de variáveis independentes.

De seguida Hanafy, M., & Ming, R. (2022) aplicou a feature selection de forma a reduzir a redundância nos dados, sendo que o objetivo passava por selecionar um subconjunto dos atributos do conjunto de dados para assim reduzir as suas dimensões. Esta técnica é utilizada para reproduzir a quantidade máxima de variação nos dados, procurando assim aumentar a precisão da classificação. Para aumentar esta precisão foram usados os algoritmos Relief(RE), Symmetrical Uncertainty(SU) e o Correlation-based Feature Selection(CFS). Cada dataset foi dividido em duas partes: treino e teste, 70% para treino dos dados e os restantes 30% para os testar. Por fim os resultados mostraram que todos os algoritmos de classificação utilizados com um cenário inicial apresentaram um baixo desempenho e que a maioria não consegue prever as classes target usando dados sem antes utilizar resampling que consiste em extrair amostras repetidas das amostras de dados originais, sendo um método não paramétrico de inferência estatística e usando também métodos de features selection. Depois de aplicar o resampling e os métodos de features selection(seleção de atributos) os modelos de machine learning melhoraram bastante relativamente aos valores do AUC.

AUC é uma medida de desempenho para os problemas de classificação em várias configurações de limite. ROC é uma curva de probabilidade e AUC representa o grau ou medida de separabilidade. Comunica o quanto o modelo é capaz de distinguir entre as classes. Quanto maior a AUC, melhor o modelo está a prever 0 como 0 e 1 como 1. Por analogia, quanto maior a AUC, melhor é o modelo para distinguir entre pacientes com a doença e sem

doença. Neste estudo Hanafy, M., & Ming, R. (2022) demonstraram que existem classificadores de ML que não apresentam previsões fiáveis sem primeiro ser aplicados métodos de resampling, métodos de feature selection e feature discretization aos diferentes datasets, demonstrando também, que o desempenho dos classificadores vai variando conforme o tipo de dados onde estes são aplicados. Depois de aplicar o teste, Friedman, Hanafy, M., & Ming, R. (2022) concluíram que o modelo de Random Forest é o melhor classificador porque apresenta os resultados de AUC mais precisos para cada dataset referente aos seguros. No estudo realizado por Chang, W. T., & Lai, K. H. (2021), usaram redes neuronais artificiais para prever a tendência dos consumidores a comprar apólices de seguros. Para além disto este estudo também se propôs a identificar diferentes fatores que influenciam a tendência dos clientes a optar por apólices de seguros pela feature selection. Inicialmente foi feita uma análise preliminar aos dados e foi aplicado a estes discretization, que basicamente consiste num processo de inserir valores em intervalos para limitar os estados possíveis. Depois da discretization foi iniciado o processo de feature selection de forma a eliminar features que são redundantes ou que possuem pouca informação preditiva, sendo que neste estudo foram utilizados dois tipos, o método do filtro e o método do wrapper. O método de filtro através de uma métrica escolhida encontra os atributos irrelevantes e filtra também as colunas que sejam redundantes do modelo utilizado, já o método wrapper seleciona um conjunto de atributos, fazendo diferentes combinações, avaliações e comparações com estas. Por fim um modelo preditivo avalia as combinações de atributos atribui uma pontuação com base na precisão do modelo. Assim que aplicados estes dois métodos de feature selection foi aplicado um terceiro método denominado de feature construction ou também conhecido como constructive induction que irá aplicar um conjunto de operadores construtivos a um conjunto de features já existentes, resultando assim na construção de novos atributos. Outro termo que nos deparamos nesta revisão foi o de outlier que consiste num valor extremo que está anormalmente fora do padrão geral da distribuição referente a uma variável segundo Kwak, S. K. et al (2017).Nos datasets usados havia um desequilíbrio de classes, e de forma a resolver este problema foi usado um método externo que consiste na modificação dos dados que iram ser usados para treino de forma a alterar a distribuição de padrões frequentes, normalização, e retirar os outliers para desta forma melhorar a deteção da classe minoritária, sendo esta operação conhecida como sampling. No

estudo realizado por Chang, W. T., & Lai, K. H. (2021) foi usado a técnica de undersampling, pois o oversampling exige mais recursos computacionais e, portanto, implicaria um investimento maior. É também aplicado o downsampling de forma a diminuir a taxa de amostras da classe predominante. Posteriormente foi usado o algoritmo de backpropagation para treinar o modelo e o 10-fold Crossvalidation para avaliar o desempenho do modelo de redes neuronais artificias. Chang, W. T., & Lai, K. H. (2021) concluíram que feature construction e undersampling são úteis para prever a intenção dos consumidores de comprar ou não a apólice de seguro. Assim comprovaram que feature construction é eficaz na resolução de tarefas que requerem classificação. Na revisão de literatura efetuada não foi encontrado nada semelhante ao que este projeto se propõem a fazer, pois já foram usadas ferramentas de Business Intelligence em vários contextos dentro da área de seguros, mas nunca para descobrir antecipadamente as funcionalidades que uma seguradora poderá vir a ter.

2.4.4 - Avaliação da acuidade dos resultados

Depois do algoritmo de machine learning aplicado este irá gerar um modelo que deverá ser avaliado. Ao aplicarmos o algoritmo temos inicialmente de particionar os dados que este vai consumir em partes, onde uma parte irá ser treinada pelo algoritmo e a outra servirá para testar e avaliar o desempenho do modelo gerado, Berrar, D. (2019).

Existem diferentes formas de avaliar diferentes tipos de algoritmos, mas quando falamos em problemas de classificação temos a AUC – curva ROC. Segundo Narkhede S. (2018) esta é uma métrica de avaliação importante quando se fala em avaliar o desempenho de qualquer modelo de classificação. Sendo ROC uma curva de probabilidade e o AUC a capacidade do modelo para distinguir as diferentes classes.

Já a métrica Accuracy segundo Carvalho, D. V (2019) mede a taxa de previsões corretas para todas as classes. Outras métricas muito usuais é a precision e a recall que são baseadas na matriz confusão, a precision consiste são as amostras positivas classificadas corretamente (Positivos verdadeiros), já o recall é a razão de amostras positivas classificadas corretamente sobre o número total de amostras positivas classificadas, ou seja, todas as amostras corretamente classificadas como positivas e as classificadas incorretamente. Por último temos

a métrica f1-score que segundo Lipton, Z. C. (2014) consiste na média harmônica entre a precision e recall.

Muitas vezes o desempenho que os modelos geram não são os desejados e, portanto, é necessário fazer alterações aos dados aplicando a estes diferentes técnicas como por exemplo a discretization que consiste na transformação de um atributo continuo num número finito de intervalos, associando a cada intervalo um valor numérico discreto (Kurgan, L. A. et al (2004)). Existem também técnicas denominadas de sampling e resampling, onde temos o undersampling (também conhecido como downsampling) que consiste em diminuir o número de dados que estão em maior número, ou seja, reduzir os dados da classe maioritária Mohammed, R. (2020). Já o oversampling consiste em aumentar a quantidade de dados da classe minoritária.

Outra das técnicas usadas para aumentar o desempenho do modelo é a remoção ou transformação dos outliers, sendo estes um valor extremo ou inconsistente que está fora do intervalo expectável dentro do conjunto de dados observado Escalante, H. J. (2005).

2.5- Trabalhos de investigação análogos

Existem muitos estudos relacionados ao problema existente da perda de dados em diferentes vertentes, mas pouco são os que estão focados em prever essa perda.
O estudo presente enquadra-se no campo da previsão do risco das organizações sofrerem perda de informação. Segundo o estudo realizado por Barati, M. & Yankson, B. (2022) estes referem que os dados que foram recolhidos por eles demonstram que o número de incidentes referentes á perda de informação aumentou entre os anos de 2005 a 2010, ficando constante até 2014 onde depois começou a diminuir.

O objetivo de Barati, M. & Yankson, B. (2022) é prever a ocorrência de perda de informação e o tamanho dessa perda, ou seja, o número de registos pessoais que foram comprometidos. O dataset usado era constituído pelas seguintes variáveis: data, dia, mês, ano, número de perdas de informação, número total de registos, número mensal e diário de perdas de informação e valores defasados. A variável target definida por Barati, M. & Yankson, B. (2022) é um número inteiro que demonstra o número de incidentes de perda de dados num determinado dia. Neste estudo, Barati, M. & Yankson, B. (2022) usaram dois modelos preditivos, sendo que o primeiro foi o Poisson Autoregressive e o segundo o Negative

Binomial. Segundo Barati, M. & Yankson, B. (2022) ambos os modelos apresentaram indicadores promissores e capazes de prever os incidentes relativos á perda de dados com uma baixa discrepância relativamente aos números reais. Eles sugerem que dados mais granulares sobre incidentes de perda de dados acompanhados de características dessas perdas como o tipo de dados comprometidos, os custos impostos á organização, estratégias de recuperação, resposta e a sua eficácia podem acrescentar mais valor aos modelos preditivos. Barati, M. & Yankson, B. (2022) mencionam o facto que o desenvolvimento de modelos preditivos específicos para cada setor industrial e económico podem melhorar bastante de acordo com as especificidades de cada setor, pois existe uma grande variação de parâmetros importantes para avaliar e gerir os riscos de perdas de informação, permitindo assim a estes modelos obter uma maior precisão.

2.6- Segurança/Gestão de Risco Outros Trabalhos

Os métodos tradicionais de gestão de risco têm dificuldade em analisar, prevenir e supervisionar efetivamente os diferentes riscos que são enfrentados pelas seguradoras.

Liu, Q. (2019) e Samhan, Bahae (2017) referem que novas tecnologias cognitivas, como o big data, machine learning e o processamento de linguagem natural, estão a substituir os métodos tradicionais, o que leva a analisar uma quantidade de dados maiores no mercado de seguros, ajudando assim a encontrar vários indicadores de risco o que significa uma maior eficácia na gestão do risco.

Segundo Liu, Q. (2019) o objetivo da gestão de risco duma seguradora é ajudar esta a evitar riscos, prevenir perdas e reduzir o impacto negativo do risco caso este se concretize. Liu, Q. (2019) refere que a tecnologia de machine learning juntamente com os dados procuram por padrões intrínsecos que iram produzir o resultado ideal. Para finalizar, Liu, Q. (2019) refere que os algoritmos de machine learning são mais eficientes no processo de análise e triagem dos dados dos segurados, mas também eliminam o viés de julgamento causado pela experiência subjetiva humana.

3. METODOLOGIA DE INVESTIGAÇÃO

Este capítulo serve para descrever o tipo de metodologia usada para a elaboração desta investigação e como é que esta foi aplicada no contexto do negócio das seguradoras.

Este projeto de investigação foi elaborado com base na metodologia Design Science Resarch (DSR). Esta metodologia consiste no desenvolvimento de artefactos que expandem os atuais limites das Tecnologias de Informação de maneira a melhorar o funcionamento das empresas e dos seus processos de negócio. Esta é uma metodologia mais frequentemente utilizada na área de sistemas de informação, engenharia, computação e administração. Para elaborar este projeto, inicialmente começamos por identificar um problema para muitas organizações, sendo ele a perda de informação. De forma a mitigar uma possível fuga de informação por parte das organizações, era desejável que houvesse algum tipo de seguro que cobrisse a informação que uma determinada organização utiliza. Os objetivos para enfrentar este problema passam por identificar o risco que uma organização apresenta relativamente à perda de informação, para desta foram, puder atribuir um determinado seguro de acordo com o risco atribuído. De forma a atingir este objetivo teremos de inicialmente identificar fontes de dados com informação referente a diferentes organizações no âmbito de perda de informação. Assim que este forem identificadas, partimos para a análise dos mesmos de forma a escolher entre as fontes as que podem ser úteis para a investigação neste âmbito. De seguida, irá ser feita uma análise dos restantes atributos e valores de forma a a proceder a eventuais transformações, tais como: novas colunas, valores não atribuídos, entre outros. Caso estes não contenham informação suficiente, teremos então de procurar novos datasets de forma que estes possam ser integrados juntamente com o existente. Feito isto será necessário inserir os dados num algoritmo de machine learning para que este permita identificar padrões e tomar decisões puramente com base nos dados introduzidos, de forma a atribuir então um risco a uma organização. Assim que o modelo for gerado é necessário criar uma API que permita utilizar o modelo gerado de uma forma simples e eficaz.

Será possível prever o risco de perda de informação por parte de uma organização só com base em determinadas informações macro referentes a esta?

3.1 - Formulação do problema

Atualmente existem seguros para quase tudo, como seguros para uma habitação, seguro automóvel, seguro de saúde, seguro de vida e até mesmo seguros para animais, sendo este último mais recente. Numa sociedade em que o digital cresce vertiginosamente, cresce também a necessidade de assegurar que a informação transacionada entre as tecnologias deva de alguma forma ser coberta relativamente aos riscos que esta está exposta.

Muitos destes riscos deve-se ao facto de que o hardware (telemóvel, computador, tablet, entre outros) pode ligar-se à rede global denominada de Internet, estabelecendo uma "ponte" entre o mundo virtual e o equipamento, que irá permitir a pessoas de qualquer parte do mundo motivados por uma infinidade de razões, como curiosidade, dinheiro, patriotismo, a aceder a informações sejam elas confidencias, públicas, privadas, pessoais,… denominando-se assim esta prática de hacking, que pode ser feita de várias formas através de um equipamento tecnológico.
Outros riscos que podem ocorrer aquando da troca de informação das/nas organizações é quando deliberadamente ou não alguém que faz parte da organização partilha a informação com outra pessoa sobre um tópico que deveria ser confidencial.
Isto pode ocorrer em conversas em sítios menos formais ou até mesmo nos escritórios da organização ou no café do outro lado da rua.
Outro exemplo, mais comum diríamos nós, seria um engano aquando do envio de um mail, isto é, um mail pode ser enviado para alguém por engano que não deveria ter acesso aquela informação. Todos estes riscos aqui referidos podem por em causa o trabalho a realizar bem como a reputação da organização.
Nos tempos que correm existem várias empresas tecnológicas que parte da sua estratégia de negócio envolve tecnologias de informação, sendo que muitas dessas organizações já não "vivem" sem as tecnologias, pois estas representam uma parte crucial dos seus processos e sem estas todo o negócio teria de ser repensado o que levantariam problemas muitos sérios para as organizações que podiam mesmo deixar de existir, dada a dependência que estas apresentam sobre as tecnologias.

Dito isto, torna-se claro que as tecnologias vieram para ficar, pois estas vêm acrescentar muito valor às entidades que as usam quando estas são bem aplicadas.

Muitas dessas tecnologias trocam sobretudo informação, o que por si só tem muito valor, e como é perceptível, fazem parte dos diferentes processos de uma organização. Tendo em conta o contexto descrito acima esta investigação pretende desenvolver uma solução que é inexistente no mercado atual que passa por criar um modelo que permita determinar o risco de perda de informação de determinada organização com base em ocorrências que no passado aconteceram e que se encontram registadas, o que irá permitir tirar insights das condições em que estas ocorreram. De salientar o decreto-lei 65/2021 que regulamenta a cibersegurança nas infraestruturas críticas e nas entidades públicas de forma a assegurar um nível segurança das redes e sistemas de informação em toda a União europeia. Esta lei remete para a legislação dos requisitos de seguranças das redes e sistemas de informação, bem como normas de comunicação de incidentes que devem ser exercidas por entidades referidas neste decreto-lei.

3.2 - Fontes de dados

Foram usados dois datasets neste projeto, sendo que um deles foi um produto de um estudo realizado suportado pelo artigo “Merging Datasets of CyberSecurity Incidents for Fun and Insight”, já o segundo dataset provém do website kaggle. Este segundo dataset possui dados derivados da lista da Fortune, sendo esta uma revista americana que publica anualmente uma lista com as 500 maiores empresas dos Estados Unidos da América, classificadas por receitas.

3.3 - Contexto de dados

O dataset referente ao artigo “Merging Datasets of CyberSecurity Incidents for Fun and Insight” é constituído por vários atributos, onde o conjunto desses atributos trata-se de registos sobre as causas que levaram a fugas de informação por parte de determinada organização, que ao mesmo tempo são atributos referidos nas normas da indústria onde esta opera. Existem registos entre o ano 2008 e o ano 2018, ou seja, num intervalo de tempo de 10 anos. Já o segundo dataset provém do website kaggle e contém atributos come título, receitas, lucros, número de empregados, sector, indústria, estado onde a organização se localiza sobre as 500 maiores organizações dos Estados Unidos da América.

3.4 - Exploração de dados

Foram usados dois datasets, ambos com ficheiros no formato “valores separados por virgulas” também denominado de csv. De forma a puder manipular os dados foi usado o notebook Jupyter que é um ambiente de desenvolvimento web que permite através da linguagem de programação python fazer a devida preparação dos datasets de forma a ir de encontro às pretensões do projeto. Através desta ferramenta os dois datasets foram analisados separadamente com recurso a bibliotecas como pandas, numpy, search, seaborn de forma a permitir explorar e visualizar os dados, tirando assim insights relevantes que possam de alguma forma ajudar a desenvolver e enriquecer este projeto.

3.5 - Transformação

A transformação dos dados é um passo fundamental num projeto desta natureza, pois podemos ter os dados mais valiosos do mundo, mas se estes não estiverem “legíveis”, não irá ser possível extrair a informação que estes nos podem dar.

Neste projeto fomos obrigados a fazer algumas transformações, pois apesar dos dados já virem estruturados, algumas colunas foram removidas, pois grande parte dos dados estavam em branco o que torna o atributo inviável para estudo, outras foram alteradas pois os tipos de dados não permitiam que estes fossem usados para conceber gráficos ou mesmo para inserir estes em algoritmos de machine learning e, portanto, estes foram transformados de categóricos para numéricos. Por fim foram criadas colunas com base em colunas dos datasets.

3.6 - Performance de ML

Avaliar o desempenho de um modelo gerado por machine learning é muito importante pois quando construímos um modelo queremos que este seja eficaz. A forma de avaliar um modelo de ML pode ser feito através de diferentes métricas. Estas métricas podem ser chamadas de métricas de avaliação ou métricas de desempenho. Através das métricas podemos perceber se o algoritmo escolhido tem um bom ou mau desempenho para os dados que lhe foram inseridos. Existem várias métricas que podem oferecer diferentes perspetivas, algumas métricas de avaliação tipicamente usadas para problemas de classificação como por

exemplo a accuracy ou precisão traduzido para português, não são adequados para imbalanced datasets. Inicialmente começamos por usar uma técnica de oversampling nos dados, mas acabamos por descartar esta técnica pois os valores gerados pelas métricas apresentavam valores que sugeriam que esta técnica não melhorava o modelo gerado e, portanto, optamos por prosseguir por outra abordagem que seria a transformação dos dados. Algumas das métricas que consideramos foram accuracy, precision, recall, f1-score e a curva ROC.

3.7 - Integração de datasets

Inicialmente deparamo-nos com a escassez de atributos que nos pudessem dar contexto das organizações, visto que o dataset inicial não teria variáveis suficientes que permitissem criar um modelo de ML eficaz, neste sentido, tivemos de procurar mais dados referentes às organizações para assim ter mais variáveis que possam reforçar as métricas do modelo gerado. Para podermos estabelecer uma relação com os novos dados, este teria de ter algum atributo que permitisse fazer essa junção com o dataset existente.

3.8 - Distribuição do modelo

Depois do modelo de machine learning ser gerado este já pode ser usado. Para atingir este objetivo optamos por criar uma API onde através desta podemos facilmente inserir novos dados sem ter de alterar qualquer código existente, permitindo assim um fácil uso da modelo gerado.

4. IMPLEMENTAÇÃO

Neste capítulo, descrevemos com mais detalhe todo o processo de implementação, desde a inserção dos dados no Jupyter Notebook até á criação da API para ler o modelo gerado pelo algoritmo de ML.

A metodologia proposta começa por compreender os principais conceitos sobre o mercado alvo, que neste caso é referente ao seguro da informação para as organizações de forma a obter uma melhor compreensão do problema que pretendemos resolver.
Nesta fase, as questões de negócios são traduzidas em objetivos de mineração de dados. Uma vez que os dados foram obtidos na plataforma kaggle, o fluxo de trabalho começa com uma análise prévia para inspecionar os diversos atributos e verificar a qualidade dos dados.

É inicialmente feita uma análise aos dados de forma a perceber quais devem permanecer e quais devem ser descartados de forma a manter só aqueles que podem aportar algo ao âmbito do projeto. Para além disso é necessário identificar aqueles que devem ser transformados.

Os dados estão em dois datasets retirados da plataforma kaggle. Portanto, é importante recolher estes dados e integrá-los em uma única estrutura. Após a integração dos dados, definiram-se alguns critérios para a seleção dos dados. Em seguida, procedeu-se à limpeza dos dados, corrigindo ou removendo registos vazios. Posteriormente, se os dados resultantes não tiverem recursos suficientes, precisamos recorrer a técnicas de engenharia de recursos.

A última fase da metodologia é escolher um modelo adequado ao problema apresentado. O conjunto de dados a ser trabalhado inicialmente encontrava-se desequilibrado e de forma a lidar com este problema aplicamos uma técnica que aumenta o número de registos da classe minoritária, para desta forma equilibrar os dados. Este passo é muito importante, pois queremos que os algoritmos consigam prever ambas as opções com base nos dados fornecidos. Por fim é criado um modelo de classificação com dados de treino e com métricas de desempenho adequadas e avaliado quanto á sua precisão nos dados de teste.

4.1 - Exploração dos dados

Inicialmente apenas começamos com um dataset e a trabalhar sobre o mesmo. O primeiro passo passou por fazer uma análise exploratória aos dados existentes para perceber em que estado é que estes se encontram e se era necessário fazer algumas alterações aos mesmos.

Na tabela 1 encontra-se uma descrição sucinta dos atributos do primeiro dataset, sendo este os dados referentes ao artigo mencionado acima. Essa tabela, contém informação sobre Year, Location, Records, Entity, Industry, Cause, URL e cluster. Seguido dessa primeira tabela temos a tabela 2 que corresponde ao segundo dataset, esta contém informação como rendimentos, lucros, ativos, número de empregados, CEO, indústria onde a organização atua, a cidade, entre outros. A maioria dos atributos são categóricos o que por si só já é uma dificuldade aquando no manuseamento destes.

Tabela 1: Descrição dos atributos referente ao dataset "merged-cleaned.csv"

Atributo	Descrição	Tipo de dados
Year	Ano em que ocorreu o ataque	Long
Location	Localização da organização	String
Records	Número de registos relacionados com o ataque.	Long
Entity	Nome da entidade que sofreu o ataque.	String
Industry	Indústria á qual pertence a entidade atacada.	String
Cause	Tipo de ataque.	String
URL	Link com informação referente ao ataque.	String

Tabela 2: Descrição dos atributos referente ao dataset " fortune1000-final.csv "

Atributo	Descrição	Tipo de dados
rank	Rank actual	Int
Entity	Nome da organização	String
Previous Rank	Rank anterior	Int
Revenues ($M)	Rendimentos da organização	Long
Revenue Change	Indústria á qual pertence a entidade atacada.	String
Profits ($M)	Lucros da organização	Long
Assets ($M)	Ativos da organização	Long
Mkt Value as of 3/29/18 ($M)	Valor de mercado a partir da data 29/03/18	Long
Employees	Número de empregados referente á organização	Long
CEO	Diretor executivo da organização	String
CEO Title	Título do diretor executivo da organização.	String
Sector	Sector á qual a organização pertence.	String
Industry	Indústria á qual a organização pertence.	String
Years on Fortune 500 List	Número de anos na “Fortune 500 List”	Int
City	Cidade referente á organização	String
State	Estado referente á organização	String
Latitude	Latitude onde referente á organização	Float
Longitude	Longitude onde referente á organização	Float

Depois de uma breve transformação sobre os dados da primeira tabela e a aplicação destes em algoritmos de ML percebemos rapidamente que seriam poucos dados para atingir o objetivo estabelecido, pois as métricas resultantes dos algoritmos aplicados estavam bastante aquém do desejado. Quando chegamos a esta conclusão partimos para a procura de mais dados referentes a organizações e encontramos o dataset representado na tabela 2.

A análise exploratória dos dados corresponde á segunda fase da metodologia CRISP-DM, e como é possível ver pela imagem abaixo a maior parte dos dados é referente ao sector das finanças, energia e tecnologia.

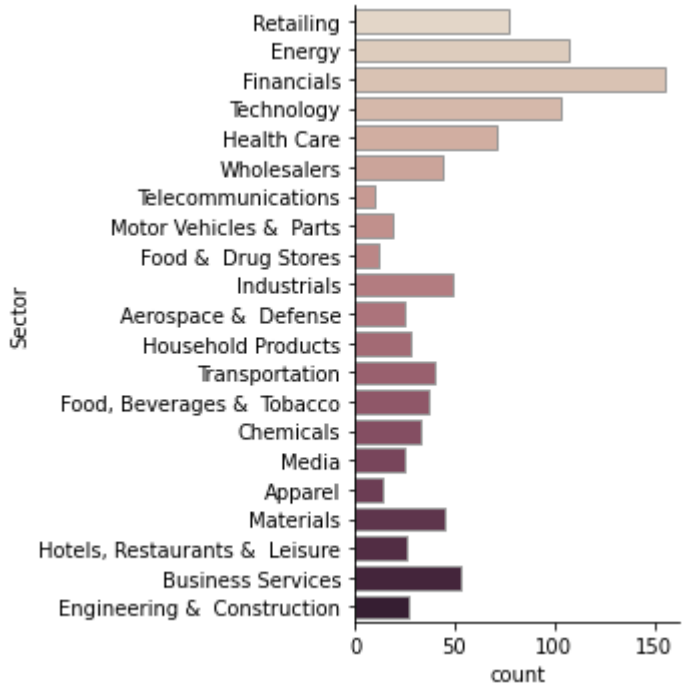


Figura 2: Número de Entidades por Sector (Fonte Própria)

Outro aspeto interessante que os dados demonstram é o facto de que quanto maior o número de empregados de uma organização a causa mais provável para a fuga de informação é a divulgação não intencional.

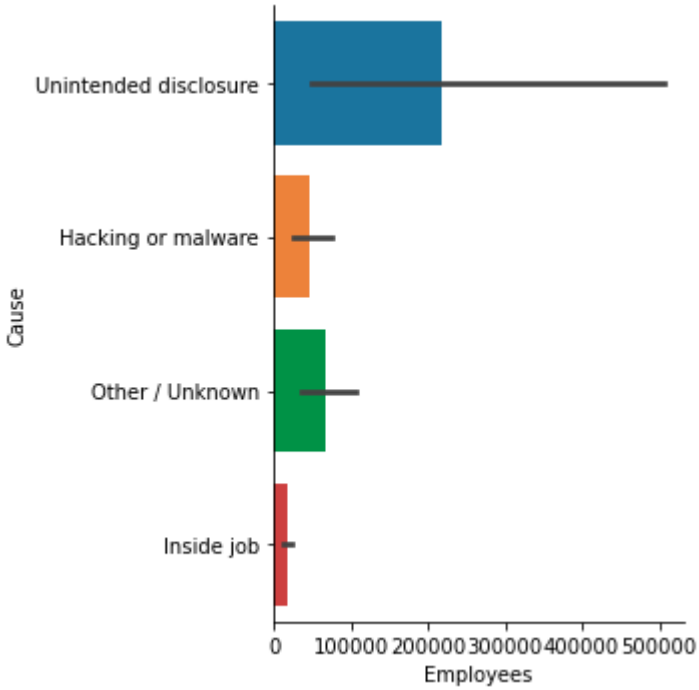


Figura 3: Número de Empregados por Causa (Fonte Própria)

4.2– Transformação de dados

Tivemos de reduzir o conjunto inicial de dados, pois nem todos eram elegíveis, visto que alguns dos registos não se encontravam em ambos os datasets. Construímos novas colunas a partir de outras de forma a podermos obter maior utilidade nas conclusões dos dados, mas também para permitir que estes possam ser usados em determinados algoritmos de machine learning.

Posteriormente os dois datasets foram unidos num só dataset através da coluna “Entity”.

Como é possível ver pelas tabelas acima, o dataset referente à tabela 2, apresenta um número maior e mais significativo de atributos comparativamente aos dados da tabela 1. Visto que os atributos da tabela 2 são mais relevantes que os da tabela 1, juntamos a tabela 1 á tabela 2 transformando os nomes das entidades da tabela 1 igual aos da tabela 2 onde estes eram a mesma entidade, possuindo apenas nomes diferentes. Por exemplo: “The Coca-Cola Company” passou para “Coca-Cola”.

Foram removidas algumas colunas do dataset final, colunas como rank, previous rank, latitude, longitude, years on Fortune 500 List, CEO, CEO Title e State. Todos estes atributos foram removidos pois não foram considerados como relevantes por falta de consistência com o restantes Dataset’s forma de extrair mais informação sobre estes juntamente com outros que permitam tirar aproveitamento para o propósito final do projeto. Por fim algumas colunas sofreram alguns ajustes, por exemplo a atributo Employees possui o número de empregados de uma organização, mas neste dataset ela encontra-se listada como sendo do tipo string e não de um tipo numérico, como float, long ou int. Por esta razão tivemos de remover as virgulas de todos os valores referentes á coluna e de seguida transformar estes para o tipo numérico, para assim ser possível usar este atributo no desenvolvimento de gráficos como o que está ilustrado acima, na Figura 4. De seguida foi verificado que a atributo “Cause” tinha bastantes campos preenchidos a “NaN - not a number” que em python significa que não se trata de números. Isto acontece porque aquando da junção dos dois datasets, nem todas as entidades existentes num dataset vão existir no segundo dataset o que é natural e desejado, pois nem todas as organizações ilustradas sofreram perdas de informação involuntariamente. Para lidar com esta situação criamos um pequeno script que irá preencher todos os campos em branco da coluna “Cause” com a string “No disclosure”, pois estas não sofreram a perda de informação.

Através da coluna “Profits” foi criada a coluna “Risk” que é preenchida por apenas três valores, 1,2 e 3, que representam os lucros abaixo de 5000 (1), acima de 5000 até 26000 (2) e acima de 26000 (3). Todas as colunas em que o tipo de dados é do tipo String foram transformadas em números inteiros, pois os algoritmos onde estes vão ser usados só aceitam colunas cujo tipo de dados é numérico, como por exemplo a coluna “Cause” e “Sector”. Posteriormente adicionamos uma coluna denominada de “Attacked” em que os únicos valores que esta coluna tem é “Yes” ou “No,” isto significa que através desta coluna é possível identificar se uma entidade sofreu perda de informação de forma involuntária ou se esta foi alvo de um ataque. Por fim foi criada a coluna target denominada de “FinalRisk”. Esta coluna foi criada através da coluna “Risk” juntamente com o valor da divisão de número de ataques por setor sobre o número total de registos que existem de determinado setor, permitindo determinar qual a probabilidade que cada setor tem de ser atacado tendo por base apenas os registos existentes neste dataset.

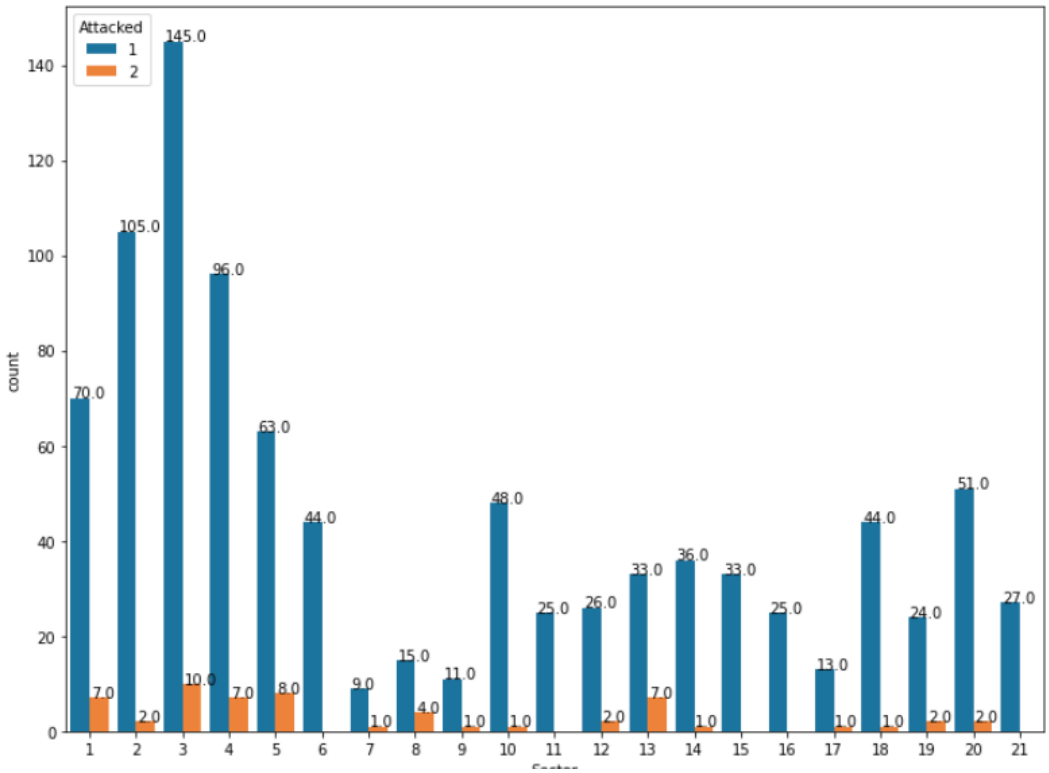


Figura 4: Número de ataques e não ataques por Setor (Fonte Própria)

Depois de obter a percentagem de cada setor, procedemos para a multiplicação desse valor com o respetivo risco de forma a obter um risco mais fiável para cada organização.

4.3 - Algoritmos de Aprendizagem utilizados

Para ter uma melhor noção dos dados e quais os modelos que se ajustam melhor a estes é essencial selecionar diferentes tipos de modelos para obter essa resposta. De forma a encontrar o melhor modelo para os dados existentes, testamos os mesmos nos seguintes algoritmos: Naives Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees e Support Vector Machine.

Todos os algoritmos mencionados acima foram usados com os dados de forma a perceber qual destes seria a melhor opção, optando assim por aquele que apresentar um melhor resultado, quer no custo computacional como na acuidade. Os dados foram usados em todos os algoritmos acima referidos através da plataforma RapidMiner que é um software que fornece um ambiente integrado para preparar dados, fazer machine learning, deep learning, minerar texto e fazer análises preditivas.

O algoritmo que obteve melhor resultados a partir dos dados inseridos foi o algoritmo de regressão logística.

4.4 - Processo de IA desenvolvido

Inicialmente começamos por definir as variáveis em independentes e a dependente (Target) de forma a puder aplicar algoritmos de IA. De seguida importamos train_test_split que é uma função do pacote sklearn.model_selection em python que vai permitir fazer a separação dos dados em treino e teste, sendo que consideramos os de teste 20% do total e os de treino os restantes 80%.

Assim que estes foram divididos importamos o algoritmo SVM e aplicamos este nos dados de treino. A precisão deste modelo foi de 71 % o que não é de todo um mau resultado.

	precision	recall	f1-score	support
High	0.71	0.79	0.75	109
Low	0.71	0.62	0.66	91
accuracy			0.71	200
macro avg	0.71	0.70	0.70	200
weighted avg	0.71	0.71	0.71	200

Figura 5 - Métricas do algoritmo SVM (Fonte Própria)

De seguida corremos o algoritmo de regressão logística com os mesmos dados. Este obteve uma precisão de 73%, mais dois pontos percentuais comparativamente ao algoritmo SVM.

	precision	recall	f1-score	support
High	0.73	0.81	0.77	109
Low	0.73	0.64	0.68	91
accuracy			0.73	200
macro avg	0.73	0.72	0.72	200
weighted avg	0.73	0.73	0.73	200

Figura 6 - Métricas do algoritmo de Regressão Logística (Fonte Própria)

Assim que os modelos foram gerados com base nos dados inseridos começamos por aplicar uma biblioteca que apresenta determinadas métricas relativamente ao modelo criado, a biblioteca denomina-se de classification_report que advém do sklearn.metrics. Depois de aplicar o classification_report começamos a analisar os resultados que podem ser vistos nas figuras 3 e 4.

4.5 - API – Implantação do Modelo de IA

De forma a podermos usufruir dos modelos gerados por estes algoritmos, estes têm de ser expostos de alguma forma. Neste projeto recorremos a uma API RESTful, pois esta permite fornecer serviços através de um conjunto de protocolos e definições, Rodriguez, A. (2008), sendo eles o GET, POST, PUT e DELETE os mais usados com o protocolo HTTP.

Por fim colocamos o modelo disponível para ser usado mais tarde pela API. Fizemos isto através do pickle, que permite serializar e desserializar estruturas de objetos em python, também denominadas de marshalling ou flattening.

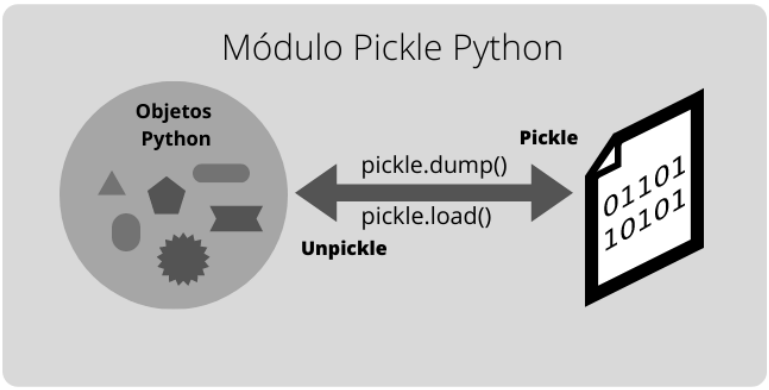


Figura 7- Python Pickle module (Fonte Própria)

A API foi desenvolvida, de forma a expor o modelo gerado, permitindo assim que qualquer pessoa tenho acesso ao trabalho aqui desenvolvido.

Como inicialmente usamos a linguagem de programação python para perceber e tratar os dados, optamos por também usar esta para criar a API. Depois de alguma pesquisa depara-mo-nos com uma framework que permite desenvolver APIs RESTfull em python de uma forma bastante simples e direta. A framework denomina-se de FastAPI e consiste numa web framework rápida, intuitiva, fácil de utilizar, robusta e com pouco código permite desenvolver uma API.

Inicialmente começamos por importar todas a bibliotecas que iríamos necessitar, como a uvicorn, numpy, pickle e pandas, logo de seguida criamos o objeto app para ler o modelo gerado.

Por fim criamos a rota denominada de "/predict" que utiliza o modelo de machine learning carregado, juntamente com oito variáveis, sendo elas a revenues, profits, assets, market value, number of employees, sector, cause e attacked.

Estas oito variáveis vão ser enviadas ao servidor da API conforme os inputs colocados pelo utilizador. É enviado em formato JSON e conforme os inputs inseridos irá gerar um output. O output só pode assumir dois valores, sendo eles "Possível criar seguro, os valores encontram-se dentro dos parâmetros desejados" e "Impossível criar seguro, os valores não se encontram dentro dos parâmetros desejados".

4.6 - Teste da API

Para testar a API começamos por inicialmente criar um conda ambiente que permite separar as dependências de diferentes projetos. Fizemos isso através do comando “conda create –name myenv”, sendo o myenv o nome do ambiente. De seguida ativamos o mesmo através do comando “conda activate myenv” para podermos usá-lo. Por último corremos o comando “uvicorn appPredictRiskOrg:app –reload”, que irá lançar o servidor num url apresentado na consola. O primeiro nome “appPredictRiskOrg” é referente ao ficheiro principal, já o nome depois dos dois pontos é o nome do objeto da FastAPI.

Para podermos usar o modelo gerado teremos de aceder ao url apresentado seguido de /docs. FastAPI permite explorar a API criada através de uma interface web denominada de Swagger UI.

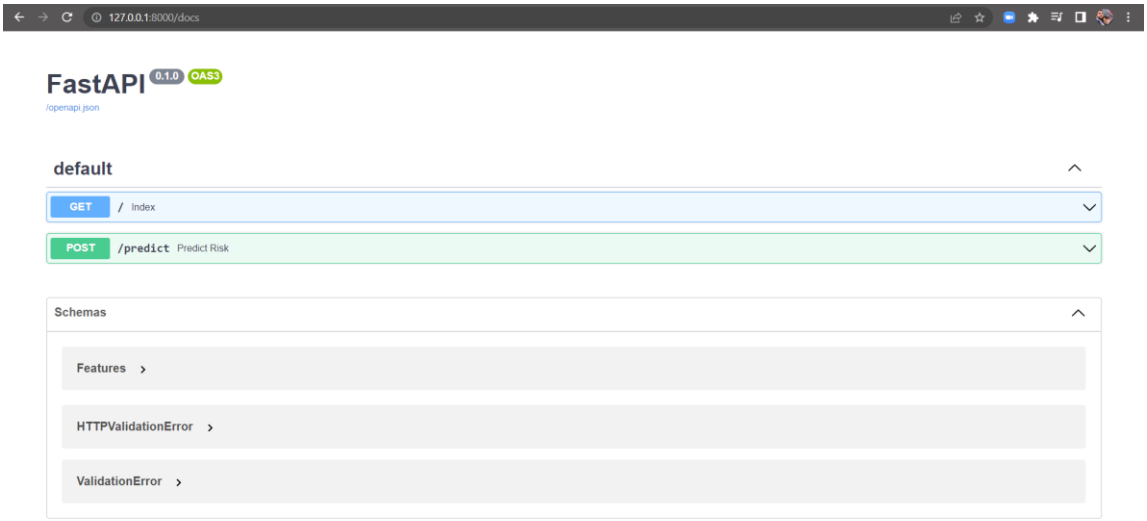


Figura 8 - FastAPI Swagger UI (Fonte Própria)

Por fim inserimos os valores de um row existente dos registos de forma a perceber se o modelo iria dar o output apresentado no dataset para esses dados.

5. RESULTADOS

Começamos por construir modelos de previsão que usam todos os valores iniciais do dataset, composto por 13 atributos numéricos e não numéricos. Os algoritmos descritos acima foram todos aplicados usando o software RapidMiner.
De todos os algoritmos usados o algoritmo regressão logística foi o que se destacou com melhores métricas em relação a todos os outros, pois a precisão deste encontra-se em cerca de 73% o que é uma percentagem considerável e realista. Já o tempo de computação deste foi bastante rápido (menor que um segundo), muito devido á quantidade de dados utilizados.

Como é possível ver pelas figuras 5 e 6, as métricas relativas ao algoritmo de regressão logística apresentam valores mais significativos comparativamente ao algoritmo de SVM.

Com os dados iniciais no seu estado "natural" o desempenho dos algoritmos era bastante pobre em termos de precision, recall e f1-score value. Com a introdução de novas colunas e da transformação de alguns dos dados existentes conseguimos melhorar o desempenho de alguns algoritmos, sobretudo depois de transformarmos a variável target em binária.
Inicialmente começamos por usar uma técnica de oversampling sobre os dados de forma a tentar enviesamento destes para um determinado resultado, mas com o decorrer do tempo e depois de tentar variadas formas de encontrar um modelo fiável, concluímos que o melhor modelo que poderíamos gerar advinha de transformações diretas aos dados sem usar nenhuma das técnicas que usamos anteriormente.

6. CONCLUSÕES E TRABALHO FUTURO

O objetivo desta dissertação era prever o risco de uma organização ser atacada com base em alguns parâmetros de um nível mais macro, isto é, através de variáveis de contexto que qualquer organização no mundo tem, como por exemplo o número de empregados e/ou os lucros que foram gerados, entre outras, para assim perceber se é possível ou não atribuir um seguro de informação. Depois do desenvolvimento deste projeto é possível responder a esta pergunta.

Os valores de F-score relativamente aos algoritmos usados é maior na regressão logística o que também permite validar o algoritmo escolhido. Este modelo vai permitir a uma seguradora prever sobretudo se uma organização apresenta um risco relativamente pequeno sobre a perda de informação.

No entanto este modelo também gera vários casos de falsos positivos, como ilustra a figura 9 abaixo que representa a matriz de confusão, sendo esses valores o 21 e 33 respetivamente.

```
array([[88, 21],
       [33, 58]], dtype=int64)
```

Figura 9 - Matriz de confusão (Fonte Própria)

Durante o desenvolvimento deste projeto fomos encontrando vários obstáculos que tiveram de ser superados. Um dos principais obstáculos que consumiu mais tempo foi logo no início do projeto, pois foi difícil encontrar dados que pudessem ajudar no propósito deste.

Além disso, tivemos de ir alterando a variável target, pois os valores das métricas que obtínhamos referentes a diferentes algoritmos era relativamente baixo.

Em suma, os objetivos inicialmente traçados foram atingidos com sucesso, os obstáculos que foram aparecendo ao longo do projeto também foram superados, mas reconhecemos que este projeto ainda pode melhorar substancialmente, a fim de ser usado em contexto real, mais concretamente por seguradoras.

Após a conclusão do projeto, é possível dizer que o modelo gerado tem potencial para prever sobretudo o baixo risco de uma organização ser atacada. No entanto este modelo pode ser melhorado significativamente, através do aumento dos registos dos atributos atuais, como a

recolha específica de dados de várias organizações de forma a prover o modelo com mais dados, com vista a robustecer o output gerado.

7. REFERÊNCIAS

Amosova, N. A., & Sanakoev, M. Y. (2020, September). Deposit Insurance Systems: Evolution, Risks, Development Trends. In The 3rd International Conference on Economy, Management and Entrepreneurship (ICOEME 2020). –Atlantis Press (pp. 1-10).

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. IADS-DM.

Barati, M., & Yankson, B. (2022). Predicting the Occurrence of a Data Breach. International Journal of Information Management Data Insights, 2(2), 100128.

Berrar, D. (2019). Cross-Validation.

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. Electronics, 8(8), 832.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

Duckett, S., & Nemet, K. (2019). The history and purposes of private health insurance. Grattan Institute.

Elhadad, E. (2021). Insurance Business Enterprises' Intelligence in View of Big Data Analytics. مجلة الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات, 26(26), 13-21

Eling, M., & Jia, R. (2018). Business failure, efficiency, and volatility: Evidence from the European insurance industry. International Review of Financial Analysis, 59, 58-76.

Ernst, C. P., & Geiger, F. (2021, January). Business Intelligence in the Database Marketing–A Case Study of a German Insurance Company. In Proceedings of the 54th Hawaii International Conference on System Sciences (p. 1011).

Escalante, H. J. (2005). A comparison of outlier detection algorithms for machine learning. In Proceedings of the International Conference on Communications in Computing (pp. 228-237).

Fink, L., Yogev, N., & Even, A. (2017). Business intelligence and organizational learning: An empirical investigation of value creation processes. Information & Management, 54(1), 38-56.

Geric S, Hutinski Z. Information system security threats classifications. Journal of Information and Organizational Sciences; 2007. 31: 51

Hanafy, M., & Ming, R. (2021). Machine learning approaches for auto insurance big data. Risks, 9(2), 42.

Hendren, N., Landais, C., & Spinnewijn, J. (2020). Choice in Insurance Markets: A Pigouvian Approach to Social Insurance Design. Annual Review of Economics, 13.

Jouini, M., Rabai, L. B. A., & Aissa, A. B. (2014). Classification of security threats in information systems. Procedia Computer Science, 32, 489-496.

Ko, M., & Dorantes, C. (2006). The impact of information security breaches on financial performance of the breached firms: an empirical investigation. Journal of Information Technology Management, 17(2), 13-22.

Kshetri N. Privacy and security issues in cloud computing: the role of institutions and institutional evolution Telecommunications Policy 2013 37 4-5 372 386 Crossref 2-s2.0-84863096012

Kurgan, L. A., & Cios, K. J. (2004). CAIM discretization algorithm. IEEE transactions on Knowledge and Data Engineering, 16(2), 145-153.

Kwak, S. K., & Kim, J. H. (2017). Statistical data preparation: management of missing values and outliers. Korean journal of anesthesiology, 70(4), 407-411.

Latif R., Abbas H., Assar S., Ali Q. Cloud computing risk assessment: a systematic literature review Future Information Technology 2014 Berlin, Germany Springer 285 295 Crossref.

Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. arXiv preprint arXiv:1402.1892.

Liu, Q. (2019, November). Research on risk management of big data and machine learning insurance based on internet finance. In Journal of Physics: Conference Series (Vol. 1345, No. 5, p. 052076). IOP Publishing.

Liu, Y., Wang, Y., & Zhang, J. (2012, September). New machine learning algorithm: Random forest. In International Conference on Information Computing and Applications (pp. 246-252). Springer, Berlin, Heidelberg.

Ma, Y. L., Zhu, X., Hu, X., & Chiu, Y. C. (2018). The use of context-sensitive insurance telematics data in auto insurance rate making. Transportation Research Part A: Policy and Practice, 113, 243- 258.

Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386

McCue A. Beware the insider security threat, CIO Jury; 2008. http://www.silicon.com/management/cio-insights/2008/04/17/beware-theinsider-security-threat-39188671/

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020, April). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In

2020 11th international conference on information and communication systems (ICICS) (pp. 243-248). IEEE.

Narkhede, S. (2018). Understanding auc-roc curve. Towards Data Science, 26(1), 220-227.

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

Ou, C. X., Zhang, X., Angelopoulos, S., Davison, R. M., & Janse, N. (2022). Security breaches and organization response strategy: Exploring consumers' threat and coping appraisals. International Journal of Information Management, 65, 102498.

Ouda, G. K. (2018). Application of Business Intelligence in Insurance Industry (Iraq). The International Journal of Engineering and Science (IJES), 7(8), 84-92.

Rodriguez, A. (2008). Restful web services: The basics. IBM developerWorks, 33, 18.

Samhan, Bahae. Can cyber risk management insurance mitigate healthcare providers' intentions to resist electronic medical records? [J]. International Journal of Healthcare Management, 2017:1-10.

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. Procedia Computer Science, 181, 526-534.

Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data security and privacy in cloud computing. International Journal of Distributed Sensor Networks, 10(7), 190903.

Verezubova, T. A., Yunxiao, C., Trashchenko, O. L., & Chirich, M. W. (2019). EVOLUTION OF INSURANCE ON THE TERRITORY OF THE REPUBLIC OF BELARUS.
Kancevičienė, N. (2019). Insurance, smart information systems and ethics. ORBIT Journal, 2(2), Article-2.

W. McDonald, Why Business Intelligence Dashboards Are Killing Traditional Reporting, Retrieved 6 June 2015 from: http://www.printaudit.com/premier/, 2015.

Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. Encyclopedia of machine learning, 15, 713-714.

Wickramasekara, N. K. (2018). Reinforcement of business intelligence applications in Sri Lankan life insurance industry (Doctoral dissertation).

Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

Xu, Z., Zhang, M., & Jiang, X. (2005, October). Business intelligence-A case study in life insurance industry. In IEEE International Conference on e-Business Engineering (ICEBE'05) (pp. 129-132). IEEE.

Yu, J., & Yu, J. (2021). Evolution of mariculture insurance policies in China: Review, challenges, and recommendations. Reviews in Fisheries Science & Aquaculture, 29(4), 566-581.

APÊNDICE I

```
import csv
import pandas as pd
import numpy as np
import seaborn as sns
from re import search

df1 = pd.read_csv(r'C:\Users\Ricar\Desktop\Revisao\Datasets\merged-cleaned.csv', sep=';')
df1
```

```
     Year,Location,Records,Entity,Industry,Cause,URL,cluster,,,,,,,,,
0      2013,AMERICA,152000000,Adobe Systems, Inc,Info...
1      2013,AMERICA,360000000,MySpace,Other,Hacking o...
2      2013,AMERICA,10228,Region Ten Community Servic...
3      2013,AMERICA,0,Alabama Title Loans,Financial s...
4      2013,AMERICA,0,American Home Patient Inc,Educa...
...                                      ...
22323  2008,AMERICA,71000,WellCare Health Plans Inc.,...
22324  2008,AMERICA,13000,Pfizer,Other privately-held...
22325  2008,AMERICA,800,Pfizer Inc,Other privately-he...
22326  2008,AMERICA,60000,Harley-Davidson, Inc. (HOG)...
22327  2008,AMERICA,28168,Okemo Mountain Resort,Other...

[22328 rows x 1 columns]
```

```
df1.dtypes
```

```
Year,Location,Records,Entity,Industry,Cause,URL,cluster,,,,,,,,,    object
dtype: object
```

```
df1 = df1.iloc[: , :-1]
print("Modified Dataframe : ")
df1.pop('URL')
df1.pop('Location')
df1.pop('Records')
#df1.pop('Year')
df1
```

```
Modified Dataframe :

      Year                        Entity  \
0     2013                Adobe Systems, Inc
1     2013                         MySpace
2     2013  Region Ten Community Services Board
3     2013                Alabama Title Loans
4     2013           American Home Patient Inc
...    ...                           ...
22323 2008           WellCare Health Plans Inc.
22324 2008                           Pfizer
```

```
22325  2008                    Pfizer Inc
22326  2008       Harley-Davidson, Inc. (HOG)
22327  2008            Okemo Mountain Resort

                    Industry              Cause
0          Information & Technology    Hacking or malware
1                         Other    Hacking or malware
2           Education & Healthcare    Hacking or malware
3              Financial services  Unintended disclosure
4           Education & Healthcare    Hacking or malware
...                          ...               ...
22323       Education & Healthcare  Unintended disclosure
22324  Other privately-held businesses  Unintended disclosure
22325  Other privately-held businesses  Unintended disclosure
22326  Other privately-held businesses  Unintended disclosure
22327  Other privately-held businesses    Hacking or malware

[22328 rows x 4 columns]
```

```
df1.Year.unique()
```

```
array([2013, 2014, 2015, 2016, 2017, 2018, 2005, 2008, 2007, 2006, 2009,
       2010, 2011, 2012], dtype=int64)
```

```
df1_2018 = df1[df1['Year'] == 2018]
```

```
df1_2018 = df1_2018.drop_duplicates(subset=['Entity', 'Industry'], keep='first')
df1_20188 = df1_2018.drop(df1_2018[(df1_2018['Entity'] == 'Facebook') & (df1_2018['Industry'] == '
Standard commercial activities')].index)
```

```
df1_20188
```

```
     Year                                Entity  \
6302   2018                              Facebook
6304   2018                               Exactis
6305   2018                           Under Armour
6306   2018                               Twitter
6307   2018                        Firebase (Google)
...     ...                                   ...
21950  2018  Dallas County Mental Health Mental Retardation...
21951  2018                   CNO Financial Group, Inc.
21952  2018                   CJ Elmwood Partners, L.P.
21956  2018                     Jones Eye Center, P.C.
22145  2018                        SES Environmental

                    Industry              Cause
6302       Information & Technology    Hacking or malware
6304                      Other  Unintended disclosure
6305   Standard commercial activities    Hacking or malware
6306       Information & Technology  Unintended disclosure
6307       Information & Technology  Unintended disclosure
...                         ...               ...
21950       Education & Healthcare    Hacking or malware
```

```
21951        Education & Healthcare  Unintended disclosure
21952        Education & Healthcare     Hacking or malware
21956        Education & Healthcare     Hacking or malware
22145                         Other       Other / Unknown

[1294 rows x 4 columns]
```

```
df1.loc[df1['Entity'] == 'iCloud Apple']
```

```
    Year       Entity              Industry            Cause
116  2014  iCloud Apple  Information & Technology  Hacking or malware
117  2014  iCloud Apple  Information & Technology  Hacking or malware
```

```
df2 = pd.read_csv(r'C:\Users\Ricar\Desktop\Revisao\Datasets\fortune1000-final.csv', encoding= 'uni
code_escape')
df2.rename(columns={"title": "Entity"}, inplace=True)
df2
```

```
     rank                    Entity Previous Rank Revenues ($M)  \
0       1                   Walmart           1     $500,343
1       2               Exxon Mobil           4     $244,363
2       3         Berkshire Hathaway           2     $242,137
3       4                     Apple           3     $229,234
4       5         UnitedHealth Group           6     $201,159
..    ...                       ...         ...          ...
995   996       SiteOne Landscape Supply         NaN      $1,862
996   997  Charles River Laboratories Intl         NaN      $1,858
997   998                 CoreLogic         952      $1,851
998   999              Ensign Group         NaN      $1,849
999  1000                       HCP         798      $1,848

    Revenue Change Profits ($M) Profit Change Assets ($M)  \
0           3.00%   $9,862.00       -27.70%   $204,522
1          17.40%  $19,710.00       151.40%   $348,691
2           8.30%  $44,940.00        86.70%   $702,095
3           6.30%  $48,351.00         5.80%   $375,319
4           8.80%  $10,558.00        50.50%   $139,058
..            ...         ...          ...         ...
995        13.00%     $54.60        78.40%      $911
996        10.50%    $123.40       -20.30%    $2,930
997        -5.20%    $152.20        42.80%    $4,077
998        11.80%     $40.50       -19.00%    $1,102
999       -27.20%    $414.20        -34.00%   $14,089

    Mkt Value as of 3/29/18 ($M)  Employees                  CEO  \
0                 $263,563   2,300,000       C. Douglas McMillon
1                 $316,157      71,200          Darren W. Woods
2                 $492,008     377,000        Warren E. Buffett
3                 $851,318     123,000          Timothy D. Cook
4                 $207,080     260,000        David S. Wichmann
..                     ...         ...                      ...
995                $3,084       3,664             Doug Black
```

```
996              $5,118     11,800          James C. Foster
997              $3,694      5,900         Frank D. Martell
998              $1,354     21,301  Christopher R. Christensen
999             $10,910        190         Thomas M. Herzog

                                CEO Title          Sector  \
0    President, Chief Executive Officer &  Director        Retailing
1              Chairman &  Chief Executive Officer          Energy
2    Chairman, President &  Chief Executive Officer       Financials
3              Chairman &  Chief Executive Officer       Technology
4              Chairman &  Chief Executive Officer      Health Care
..                                       ...             ...
995            Chairman &  Chief Executive Officer      Wholesalers
996            Chairman &  Chief Executive Officer      Health Care
997  President, Chief Executive Officer &  Director  Business Services
998  President, Chief Executive Officer &  Director      Health Care
999  President, Chief Executive Officer &  Director       Financials

                          Industry Years on Fortune 500 List  \
0                    General Merchandisers                  24
1                     Petroleum Refining                    24
2    Insurance: Property and Casualty (Stock)               24
3               Computers, Office Equipment                 24
4     Health Care: Insurance and Managed Care               24
..                                     ...                 ...
995               Wholesalers: Diversified                   -
996  Health Care: Pharmacy and Other Services                -
997                 Financial Data Services                  -
998          Health Care: Medical Facilities                 -
999                        Real estate                       -

           City State   Latitude   Longitude
0     Bentonville    AR  36.372854  -94.208817
1          Irving    TX  32.814018  -96.948894
2           Omaha    NE  41.256537  -95.934503
3       Cupertino    CA  37.322998 -122.032182
4       Minnetonka    MN  44.921184  -93.468749
..             ...   ...        ...        ...
995        Roswell    GA  34.023243  -84.361555
996     Wilmington    MA  42.548171  -71.172447
997          Irvine    CA  33.684567 -117.826505
998  Mission Viejo    CA  33.596891 -117.658156
999          Irvine    CA  33.684567 -117.826505

[1000 rows x 19 columns]
```

```
df1.Entity.unique()
```

```
array(['Adobe Systems, Inc', 'MySpace',
       'Region Ten Community Services Board', ..., 'Pfizer Inc',
       'Harley-Davidson, Inc. (HOG)', 'Okemo Mountain Resort'],
      dtype=object)
```

```
df1['Entity'] = df1['Entity'].str.replace('(', '')
df1['Entity'] = df1['Entity'].str.replace(')', '')
```

```
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/2569765989.py:1: FutureWarning: The defaul
t value of regex will change from True to False in a future version. In addition, single character regula
r expressions will *not* be treated as literal strings when regex=True.
  df1['Entity'] = df1['Entity'].str.replace('(', '')
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/2569765989.py:2: FutureWarning: The defaul
t value of regex will change from True to False in a future version. In addition, single character regula
r expressions will *not* be treated as literal strings when regex=True.
  df1['Entity'] = df1['Entity'].str.replace(')', '')
```

```
df1.Entity.unique()
```

```
array(['Adobe Systems, Inc', 'MySpace',
       'Region Ten Community Services Board', ..., 'Pfizer Inc',
       'Harley-Davidson, Inc. HOG', 'Okemo Mountain Resort'], dtype=object)
```

Check if the entity from the second dataset exists in the first dataset, if it does change the name of the first dataset entity for the second dataset entity

```
for Entity1 in df1_20188.Entity:
   for Entity2 in df2.Entity:
      if isinstance(Entity1, str) and isinstance(Entity2, str):
         if (Entity2 in Entity1) and len(Entity2) > 3 :
            df1_20188.loc[df1_20188["Entity"] == Entity1, "Entity"] = Entity2
            print(Entity1 + '--- > ' + Entity2)
```

```
Facebook--- > Facebook
Under Armour--- > Under Armour
Twitter--- > Twitter
SunTrust Banks--- > SunTrust Banks
Target Direct Marketing/Higher Education Institutions--- > Target
Southwest Airlines Co.--- > Southwest Airlines
FedEx--- > FedEx
Aflac--- > Aflac
Humana--- > Humana
UnitedHealth Group Single Affiliated Covered Entity--- > UnitedHealth Group
Amgen (Willis Towers Watson)--- > Amgen
Hasbro, Inc.--- > Hasbro
Bio-Rad Laboratories, Inc. (Dun & Bradstreet)--- > Bio-Rad Laboratories
Brinker International, Inc. (Chili's)--- > Brinker International
Academy Mortgage Corporation (Workday)--- > Workday
Humana.com and Go365.com--- > Humana
WellCare Health Plans, Inc.--- > WellCare Health Plans
Citizens Financial Group--- > Citizens Financial Group
Maximus Inc. / Business Ink, Co.--- > Maximus
Walmart, Inc.--- > Walmart
Autism Learning Partners Holdings--- > Lear
Bed Bath & Beyond--- > Bed Bath & Beyond
Autoliv ASP--- > Autoliv
Marriott International--- > Marriott International
Massachusetts Mutual Life Insurance Company--- > Massachusetts Mutual Life Insurance
```

Intuit Inc.--- > Intuit
Dollar General Corporation--- > Dollar General
Tesla--- > Tesla
Alabama Ballet--- > Ball
Cary E Williams--- > Williams
Lincoln National Life Insurance Company--- > Lincoln National
Pendleton Square Trust Company--- > Square
Ameriprise Financial--- > Ameriprise Financial
Comcast - Xfinity--- > Comcast
Cigna Corp--- > Cigna
Dover Business Services--- > Dover
Teachers Insurance and Annuity Association (TIAA)--- > TIAA
Western Union--- > Western Union
The Coca-Cola Company--- > Coca-Cola
Ameriprise Financial, Inc.--- > Ameriprise Financial
FedEx--- > FedEx
Southern National Bancorp of Virginia, Inc. d/b/a/ Sonabank--- > Southern
AHM, Inc. on behalf of the Staybridge Suites Lexington & Holiday Inn Express New Buffalo--- > Expres
s
Nuance Communications--- > Nuance Communications
Best Buy--- > Best Buy
Delta Air Lines, Inc.--- > Delta Air Lines
Bed Bath & Beyond, Inc.--- > Bed Bath & Beyond
Hasbro inc--- > Hasbro
Aflac--- > Aflac
Marriott International Inc.--- > Marriott International
Marriott International--- > Marriott International
Dollar General--- > Dollar General
Intuit inc--- > Intuit
Intuit--- > Intuit
Walmart Inc.--- > Walmart
SunTrust Banks, Inc.--- > SunTrust Banks
Facebook, inc.--- > Facebook
Facebook, Inc.--- > Facebook
Humana inc--- > Humana
Kirby and Associates p c--- > Kirby
XPO Logistics inc--- > XPO Logistics
Rite Aid Corporation--- > Rite Aid
Dana incorporated--- > Dana
Coty inc--- > Coty
Macys inc--- > Macys
Aetna inc--- > Aetna
Alphabet, Inc. - Google+--- > Alphabet
Envision Healthcare Corporation--- > Envision Healthcare
Roadrunner Transportation Systems, Inc.--- > Roadrunner Transportation Systems
YRC Worldwide Inc.--- > YRC Worldwide
JPMorgan Chase Bank, N.A.--- > JPMorgan Chase
American Express Travel Related Services Company, Inc.--- > American Express
American Express Travel Related Services Company, Inc.--- > Express
Cigna--- > Cigna
CNO Financial Group, Inc.--- > CNO Financial Group

```
df1_20188.loc[df1_20188['Entity'] == "Facebook"]
```

```
      Year    Entity                    Industry               Cause
6302   2018  Facebook       Information & Technology    Hacking or malware
21497  2018  Facebook  Standard commercial activities  Unintended disclosure
21498  2018  Facebook  Other privately-held businesses    Hacking or malware
```

```
df2.loc[df2['Entity'] == "Facebook"]
```

```
   rank    Entity Previous Rank Revenues ($M) Revenue Change Profits ($M)  \
75    76  Facebook          98      $40,653         47.10%  $15,934.00

  Profit Change Assets ($M) Mkt Value as of 3/29/18 ($M) Employees  \
75       56.00%    $84,524                 $464,190     25,105

           CEO                      CEO Title      Sector  \
75  Mark Zuckerberg  Chairman &  Chief Executive Officer  Technology

                   Industry Years on Fortune 500 List       City  \
75  Internet Services and Retailing                     6  Menlo Park

  State  Latitude   Longitude
75    CA  37.45296 -122.181725
```

```
df_outer = pd.merge(df2, df1_20188, on='Entity', how='outer')
```

```
df_outer
```

```
     rank                                Entity Previous Rank  \
0     1.0                               Walmart           1
1     1.0                               Walmart           1
2     2.0                           Exxon Mobil           4
3     3.0                      Berkshire Hathaway           2
4     4.0                                 Apple           3
...   ...                                   ...         ...
2232  NaN                   Inova Health System         NaN
2233  NaN  Dallas County Mental Health Mental Retardation...         NaN
2234  NaN                CJ Elmwood Partners, L.P.         NaN
2235  NaN                 Jones Eye Center, P.C.         NaN
2236  NaN                     SES Environmental         NaN

     Revenues ($M) Revenue Change Profits ($M) Profit Change Assets ($M)  \
0        $500,343         3.00%   $9,862.00       -27.70%   $204,522
1        $500,343         3.00%   $9,862.00       -27.70%   $204,522
2        $244,363        17.40%  $19,710.00       151.40%   $348,691
3        $242,137         8.30%  $44,940.00        86.70%   $702,095
4        $229,234         6.30%  $48,351.00         5.80%   $375,319
...           ...          ...         ...           ...         ...
2232         NaN          NaN         NaN           NaN         NaN
2233         NaN          NaN         NaN           NaN         NaN
2234         NaN          NaN         NaN           NaN         NaN
2235         NaN          NaN         NaN           NaN         NaN
2236         NaN          NaN         NaN           NaN         NaN
```

```
    Mkt Value as of 3/29/18 ($M)  Employees  ...     Sector  \
0                $263,563   2,300,000  ...   Retailing
1                $263,563   2,300,000  ...   Retailing
2                $316,157     71,200  ...     Energy
3                $492,008    377,000  ...  Financials
4                $851,318    123,000  ...  Technology
...                   ...       ...  ...         ...
2232                  NaN       NaN  ...         NaN
2233                  NaN       NaN  ...         NaN
2234                  NaN       NaN  ...         NaN
2235                  NaN       NaN  ...         NaN
2236                  NaN       NaN  ...         NaN

                        Industry_x Years on Fortune 500 List  \
0                 General Merchandisers                   24
1                 General Merchandisers                   24
2                   Petroleum Refining                   24
3    Insurance: Property and Casualty (Stock)                   24
4              Computers, Office Equipment                   24
...                                   ...                  ...
2232                                  NaN                  NaN
2233                                  NaN                  NaN
2234                                  NaN                  NaN
2235                                  NaN                  NaN
2236                                  NaN                  NaN

          City State   Latitude   Longitude    Year  \
0    Bentonville    AR  36.372854  -94.208817  2018.0
1    Bentonville    AR  36.372854  -94.208817  2018.0
2         Irving    TX  32.814018  -96.948894     NaN
3          Omaha    NE  41.256537  -95.934503     NaN
4      Cupertino    CA  37.322998 -122.032182     NaN
...          ...   ...        ...         ...     ...
2232         NaN   NaN        NaN         NaN  2018.0
2233         NaN   NaN        NaN         NaN  2018.0
2234         NaN   NaN        NaN         NaN  2018.0
2235         NaN   NaN        NaN         NaN  2018.0
2236         NaN   NaN        NaN         NaN  2018.0

              Industry_y              Cause
0    Education & Healthcare  Unintended disclosure
1    Education & Healthcare  Unintended disclosure
2                    NaN                    NaN
3                    NaN                    NaN
4                    NaN                    NaN
...                  ...                    ...
2232  Education & Healthcare       Other / Unknown
2233  Education & Healthcare    Hacking or malware
2234  Education & Healthcare    Hacking or malware
2235  Education & Healthcare    Hacking or malware
```

```
2236                Other       Other / Unknown

[2237 rows x 22 columns]
```

```
df_outer.loc[df_outer['Entity'] == "Facebook"]
```

```
    rank    Entity Previous Rank Revenues ($M) Revenue Change Profits ($M)  \
80  76.0  Facebook          98     $40,653        47.10%  $15,934.00
81  76.0  Facebook          98     $40,653        47.10%  $15,934.00
82  76.0  Facebook          98     $40,653        47.10%  $15,934.00

   Profit Change Assets ($M) Mkt Value as of 3/29/18 ($M) Employees  ... \
80       56.00%   $84,524               $464,190    25,105  ...
81       56.00%   $84,524               $464,190    25,105  ...
82       56.00%   $84,524               $464,190    25,105  ...

      Sector                Industry_x Years on Fortune 500 List  \
80  Technology  Internet Services and Retailing                    6
81  Technology  Internet Services and Retailing                    6
82  Technology  Internet Services and Retailing                    6

        City State  Latitude   Longitude    Year  \
80  Menlo Park    CA  37.45296 -122.181725  2018.0
81  Menlo Park    CA  37.45296 -122.181725  2018.0
82  Menlo Park    CA  37.45296 -122.181725  2018.0

                  Industry_y               Cause
80       Information & Technology    Hacking or malware
81   Standard commercial activities  Unintended disclosure
82  Other privately-held businesses    Hacking or malware

[3 rows x 22 columns]
```

```
df_outer.dropna(subset=['rank'])
```

```
      rank                    Entity Previous Rank Revenues ($M)  \
0      1.0                   Walmart          1    $500,343
1      1.0                   Walmart          1    $500,343
2      2.0                Exxon Mobil         4    $244,363
3      3.0           Berkshire Hathaway         2    $242,137
4      4.0                     Apple          3    $229,234
...    ...                     ...          ...         ...
1012   996.0       SiteOne Landscape Supply        NaN      $1,862
1013   997.0  Charles River Laboratories Intl       NaN      $1,858
1014   998.0                CoreLogic         952      $1,851
1015   999.0             Ensign Group         NaN      $1,849
1016  1000.0                    HCP          798      $1,848

    Revenue Change Profits ($M) Profit Change Assets ($M)  \
0          3.00%   $9,862.00       -27.70%   $204,522
1          3.00%   $9,862.00       -27.70%   $204,522
2         17.40%  $19,710.00        151.40%   $348,691
```

```
3          8.30%  $44,940.00       86.70%   $702,095
4          6.30%  $48,351.00        5.80%   $375,319
...          ...         ...          ...         ...
1012       13.00%     $54.60       78.40%      $911
1013       10.50%    $123.40      -20.30%    $2,930
1014       -5.20%    $152.20       42.80%    $4,077
1015       11.80%     $40.50      -19.00%    $1,102
1016      -27.20%    $414.20       -34.00%   $14,089

    Mkt Value as of 3/29/18 ($M)  Employees  ...          Sector  \
0                 $263,563   2,300,000  ...        Retailing
1                 $263,563   2,300,000  ...        Retailing
2                 $316,157      71,200  ...           Energy
3                 $492,008     377,000  ...       Financials
4                 $851,318     123,000  ...       Technology
...                    ...         ...  ...              ...
1012                $3,084       3,664  ...      Wholesalers
1013                $5,118      11,800  ...       Health Care
1014                $3,694       5,900  ...  Business Services
1015                $1,354      21,301  ...       Health Care
1016               $10,910         190  ...       Financials

                        Industry_x Years on Fortune 500 List  \
0                  General Merchandisers                   24
1                  General Merchandisers                   24
2                   Petroleum Refining                    24
3    Insurance: Property and Casualty (Stock)                  24
4               Computers, Office Equipment                  24
...                                    ...                 ...
1012              Wholesalers: Diversified                   -
1013  Health Care: Pharmacy and Other Services                     -
1014              Financial Data Services                    -
1015         Health Care: Medical Facilities                   -
1016                          Real estate                    -

          City State   Latitude   Longitude    Year  \
0     Bentonville    AR  36.372854  -94.208817  2018.0
1     Bentonville    AR  36.372854  -94.208817  2018.0
2         Irving    TX  32.814018  -96.948894     NaN
3          Omaha    NE  41.256537  -95.934503     NaN
4       Cupertino    CA  37.322998 -122.032182     NaN
...           ...   ...       ...         ...     ...
1012       Roswell    GA  34.023243  -84.361555     NaN
1013    Wilmington    MA  42.548171  -71.172447     NaN
1014        Irvine    CA  33.684567 -117.826505     NaN
1015  Mission Viejo    CA  33.596891 -117.658156     NaN
1016        Irvine    CA  33.684567 -117.826505     NaN

             Industry_y              Cause
0    Education & Healthcare  Unintended disclosure
1    Education & Healthcare  Unintended disclosure
```

```
2                NaN              NaN
3                NaN              NaN
4                NaN              NaN
...              ...              ...
1012             NaN              NaN
1013             NaN              NaN
1014             NaN              NaN
1015             NaN              NaN
1016             NaN              NaN

[1017 rows x 22 columns]
```

```
df_outer.Cause.unique()
```

```
array(['Unintended disclosure', nan, 'Hacking or malware',
       'Other / Unknown', 'Inside job'], dtype=object)
```

```
df_outer['Cause'].value_counts(dropna=False)
```

```
NaN                  943
Hacking or malware   524
Other / Unknown      468
Unintended disclosure  262
Inside job            40
Name: Cause, dtype: int64
```

```
rslt_df = df_outer.loc[df_outer['Cause'] == "Hacking or malware"]
rslt_df
```

```
     rank                                Entity Previous Rank  \
5     5.0                 UnitedHealth Group             6
33   33.0                            Comcast            31
58   56.0                             Humana            53
75   72.0                           Best Buy            72
77   73.0                              Cigna            70
...   ...                                ...           ...
2226  NaN                 James R. Etzkorn, MD           NaN
2231  NaN  Oprex Surgery (Baytown), L.P. d/b/a Altus Bayt...          NaN
2233  NaN  Dallas County Mental Health Mental Retardation...          NaN
2234  NaN                CJ Elmwood Partners, L.P.           NaN
2235  NaN                  Jones Eye Center, P.C.            NaN

     Revenues ($M) Revenue Change Profits ($M) Profit Change Assets ($M)  \
5         $201,159          8.80%  $10,558.00        50.50%   $139,058
33         $84,526          5.10%  $22,714.00       161.20%   $186,949
58         $53,767         -1.10%   $2,448.00       298.70%    $27,178
75         $42,151          7.00%   $1,000.00       -18.60%    $13,049
77         $41,616          4.90%   $2,237.00        19.80%    $61,753
...            ...           ...         ...           ...        ...
2226           NaN           NaN         NaN           NaN        NaN
2231           NaN           NaN         NaN           NaN        NaN
2233           NaN           NaN         NaN           NaN        NaN
2234           NaN           NaN         NaN           NaN        NaN
```

```
2235         NaN          NaN         NaN          NaN        NaN

    Mkt Value as of 3/29/18 ($M) Employees  ...           Sector  \
5                 $207,080    260,000  ...       Health Care
33                $158,703    164,000  ...  Telecommunications
58                 $37,122     45,900  ...       Health Care
75                 $20,460    125,000  ...         Retailing
77                 $40,735     46,000  ...       Health Care
...                    ...        ...  ...              ...
2226                   NaN        NaN  ...              NaN
2231                   NaN        NaN  ...              NaN
2233                   NaN        NaN  ...              NaN
2234                   NaN        NaN  ...              NaN
2235                   NaN        NaN  ...              NaN

                          Industry_x  Years on Fortune 500 List  \
5    Health Care: Insurance and Managed Care                  24
33                   Telecommunications                   23
58   Health Care: Insurance and Managed Care                  24
75              Specialty Retailers: Other                   24
77   Health Care: Insurance and Managed Care                  24
...                                  ...                  ...
2226                                 NaN                  NaN
2231                                 NaN                  NaN
2233                                 NaN                  NaN
2234                                 NaN                  NaN
2235                                 NaN                  NaN

          City State   Latitude  Longitude    Year  \
5      Minnetonka    MN  44.921184 -93.468749  2018.0
33    Philadelphia    PA  39.952584 -75.165222  2018.0
58     Louisville    KY  38.252665 -85.758456  2018.0
75      Richfield    MN  44.883298 -93.283002  2018.0
77     Bloomfield    CT  41.826488 -72.730095  2018.0
...           ...   ...       ...        ...     ...
2226         NaN   NaN       NaN        NaN  2018.0
2231         NaN   NaN       NaN        NaN  2018.0
2233         NaN   NaN       NaN        NaN  2018.0
2234         NaN   NaN       NaN        NaN  2018.0
2235         NaN   NaN       NaN        NaN  2018.0

                  Industry_y            Cause
5          Education & Healthcare  Hacking or malware
33   Standard commercial activities  Hacking or malware
58            Financial services  Hacking or malware
75   Standard commercial activities  Hacking or malware
77         Education & Healthcare  Hacking or malware
...                           ...              ...
2226       Education & Healthcare  Hacking or malware
2231       Education & Healthcare  Hacking or malware
2233       Education & Healthcare  Hacking or malware
```

```
2234        Education & Healthcare  Hacking or malware
2235        Education & Healthcare  Hacking or malware

[524 rows x 22 columns]
```

```
df_outer['Employees'].value_counts(dropna=False)
```

```
NaN       1220
10,000      11
29,000       9
7,000        8
26,000       7
         ...
26,300       1
2,130        1
2,290        1
5,896        1
202,000      1
Name: Employees, Length: 762, dtype: int64
```

```
len(df_outer.index)
```

```
2237
```

```
df_outer.dropna(subset=['Employees'])
```

```
      rank                    Entity Previous Rank Revenues ($M)  \
0      1.0                   Walmart           1    $500,343
1      1.0                   Walmart           1    $500,343
2      2.0                Exxon Mobil          4    $244,363
3      3.0          Berkshire Hathaway          2    $242,137
4      4.0                     Apple           3    $229,234
...    ...                      ...          ...         ...
1012   996.0       SiteOne Landscape Supply         NaN      $1,862
1013   997.0  Charles River Laboratories Intl         NaN      $1,858
1014   998.0                 CoreLogic         952      $1,851
1015   999.0              Ensign Group         NaN      $1,849
1016  1000.0                      HCP         798      $1,848

    Revenue Change Profits ($M) Profit Change Assets ($M)  \
0           3.00%   $9,862.00       -27.70%   $204,522
1           3.00%   $9,862.00       -27.70%   $204,522
2          17.40%  $19,710.00       151.40%   $348,691
3           8.30%  $44,940.00        86.70%   $702,095
4           6.30%  $48,351.00         5.80%   $375,319
...           ...         ...          ...         ...
1012        13.00%     $54.60        78.40%      $911
1013        10.50%    $123.40       -20.30%    $2,930
1014        -5.20%    $152.20        42.80%    $4,077
1015        11.80%     $40.50       -19.00%    $1,102
1016       -27.20%    $414.20        -34.00%   $14,089

    Mkt Value as of 3/29/18 ($M)  Employees  ...           Sector  \
```

```
0              $263,563   2,300,000  ...         Retailing
1              $263,563   2,300,000  ...         Retailing
2              $316,157      71,200  ...            Energy
3              $492,008     377,000  ...        Financials
4              $851,318     123,000  ...        Technology
...                 ...         ...  ...               ...
1012             $3,084       3,664  ...       Wholesalers
1013             $5,118      11,800  ...       Health Care
1014             $3,694       5,900  ...  Business Services
1015             $1,354      21,301  ...       Health Care
1016            $10,910         190  ...        Financials

                      Industry_x Years on Fortune 500 List  \
0                 General Merchandisers                   24
1                 General Merchandisers                   24
2                   Petroleum Refining                   24
3    Insurance: Property and Casualty (Stock)                   24
4              Computers, Office Equipment                   24
...                                 ...                  ...
1012             Wholesalers: Diversified                    -
1013  Health Care: Pharmacy and Other Services                    -
1014              Financial Data Services                    -
1015        Health Care: Medical Facilities                    -
1016                         Real estate                    -

          City State   Latitude   Longitude    Year  \
0     Bentonville    AR  36.372854  -94.208817  2018.0
1     Bentonville    AR  36.372854  -94.208817  2018.0
2          Irving    TX  32.814018  -96.948894     NaN
3           Omaha    NE  41.256537  -95.934503     NaN
4       Cupertino    CA  37.322998 -122.032182     NaN
...           ...   ...        ...         ...     ...
1012       Roswell    GA  34.023243  -84.361555     NaN
1013    Wilmington    MA  42.548171  -71.172447     NaN
1014        Irvine    CA  33.684567 -117.826505     NaN
1015  Mission Viejo    CA  33.596891 -117.658156     NaN
1016        Irvine    CA  33.684567 -117.826505     NaN

             Industry_y              Cause
0    Education & Healthcare  Unintended disclosure
1    Education & Healthcare  Unintended disclosure
2                   NaN                NaN
3                   NaN                NaN
4                   NaN                NaN
...                 ...                ...
1012                NaN                NaN
1013                NaN                NaN
1014                NaN                NaN
1015                NaN                NaN
1016                NaN                NaN
```

[1017 rows x 22 columns]

df_outer.dropna(subset=['rank'])

```
      rank                  Entity Previous Rank Revenues ($M)  \
0      1.0                 Walmart           1    $500,343
1      1.0                 Walmart           1    $500,343
2      2.0              Exxon Mobil          4    $244,363
3      3.0        Berkshire Hathaway         2    $242,137
4      4.0                   Apple          3    $229,234
...    ...                    ...         ...         ...
1012   996.0      SiteOne Landscape Supply        NaN      $1,862
1013   997.0  Charles River Laboratories Intl      NaN      $1,858
1014   998.0               CoreLogic         952      $1,851
1015   999.0             Ensign Group        NaN      $1,849
1016  1000.0                   HCP          798      $1,848

   Revenue Change Profits ($M) Profit Change Assets ($M)  \
0          3.00%   $9,862.00       -27.70%   $204,522
1          3.00%   $9,862.00       -27.70%   $204,522
2         17.40%  $19,710.00       151.40%   $348,691
3          8.30%  $44,940.00        86.70%   $702,095
4          6.30%  $48,351.00         5.80%   $375,319
...          ...         ...         ...         ...
1012       13.00%     $54.60       78.40%      $911
1013       10.50%    $123.40       -20.30%    $2,930
1014       -5.20%    $152.20       42.80%     $4,077
1015       11.80%     $40.50       -19.00%    $1,102
1016      -27.20%    $414.20       -34.00%    $14,089

   Mkt Value as of 3/29/18 ($M)  Employees  ...          Sector  \
0                 $263,563   2,300,000  ...        Retailing
1                 $263,563   2,300,000  ...        Retailing
2                 $316,157      71,200  ...           Energy
3                 $492,008     377,000  ...       Financials
4                 $851,318     123,000  ...       Technology
...                    ...         ...  ...              ...
1012                $3,084       3,664  ...       Wholesalers
1013                $5,118      11,800  ...       Health Care
1014                $3,694       5,900  ...  Business Services
1015                $1,354      21,301  ...       Health Care
1016               $10,910         190  ...       Financials

                        Industry_x Years on Fortune 500 List  \
0                  General Merchandisers                  24
1                  General Merchandisers                  24
2                    Petroleum Refining                   24
3    Insurance: Property and Casualty (Stock)                24
4               Computers, Office Equipment                 24
...                                    ...                ...
1012              Wholesalers: Diversified                  -
```

```
1013  Health Care: Pharmacy and Other Services                    -
1014                Financial Data Services                    -
1015          Health Care: Medical Facilities                    -
1016                           Real estate                    -

           City State   Latitude   Longitude    Year  \
0      Bentonville    AR  36.372854  -94.208817  2018.0
1      Bentonville    AR  36.372854  -94.208817  2018.0
2          Irving    TX  32.814018  -96.948894     NaN
3           Omaha    NE  41.256537  -95.934503     NaN
4        Cupertino    CA  37.322998 -122.032182     NaN
...            ...   ...       ...         ...     ...
1012        Roswell    GA  34.023243  -84.361555     NaN
1013     Wilmington    MA  42.548171  -71.172447     NaN
1014         Irvine    CA  33.684567 -117.826505     NaN
1015  Mission Viejo    CA  33.596891 -117.658156     NaN
1016         Irvine    CA  33.684567 -117.826505     NaN

              Industry_y              Cause
0     Education & Healthcare  Unintended disclosure
1     Education & Healthcare  Unintended disclosure
2                    NaN                 NaN
3                    NaN                 NaN
4                    NaN                 NaN
...                  ...                 ...
1012                 NaN                 NaN
1013                 NaN                 NaN
1014                 NaN                 NaN
1015                 NaN                 NaN
1016                 NaN                 NaN

[1017 rows x 22 columns]

df_outer = df_outer[df_outer['rank'].notna()]

df_outer.tail()

      rank                    Entity Previous Rank Revenues ($M)  \
1012   996.0       SiteOne Landscape Supply         NaN      $1,862
1013   997.0  Charles River Laboratories Intl         NaN      $1,858
1014   998.0                  CoreLogic          952      $1,851
1015   999.0               Ensign Group          NaN      $1,849
1016  1000.0                       HCP          798      $1,848

    Revenue Change Profits ($M) Profit Change Assets ($M)  \
1012        13.00%     $54.60        78.40%      $911
1013        10.50%    $123.40       -20.30%    $2,930
1014        -5.20%    $152.20        42.80%    $4,077
1015        11.80%     $40.50       -19.00%    $1,102
1016       -27.20%    $414.20       -34.00%   $14,089

    Mkt Value as of 3/29/18 ($M) Employees  ...           Sector  \
```

```
1012              $3,084     3,664  ...       Wholesalers
1013              $5,118    11,800  ...       Health Care
1014              $3,694     5,900  ...  Business Services
1015              $1,354    21,301  ...       Health Care
1016             $10,910       190  ...       Financials

                      Industry_x Years on Fortune 500 List  \
1012             Wholesalers: Diversified                    -
1013  Health Care: Pharmacy and Other Services               -
1014              Financial Data Services                    -
1015       Health Care: Medical Facilities                   -
1016                      Real estate                       -

          City State   Latitude   Longitude  Year  Industry_y  Cause
1012      Roswell    GA  34.023243  -84.361555   NaN        NaN    NaN
1013    Wilmington    MA  42.548171  -71.172447   NaN        NaN    NaN
1014       Irvine    CA  33.684567 -117.826505   NaN        NaN    NaN
1015  Mission Viejo    CA  33.596891 -117.658156   NaN        NaN    NaN
1016       Irvine    CA  33.684567 -117.826505   NaN        NaN    NaN

[5 rows x 22 columns]
```

```
df_outer['Cause'].value_counts(dropna=False)
```

```
NaN                  943
Hacking or malware    33
Unintended disclosure 22
Other / Unknown       16
Inside job             3
Name: Cause, dtype: int64
```

```
df_outer = df_outer.drop_duplicates(subset=['Entity'], keep='first')
```

```
df_outer.columns
```

```
Index(['rank', 'Entity', 'Previous Rank', 'Revenues ($M)', 'Revenue Change',
       'Profits ($M)', 'Profit Change', 'Assets ($M)',
       'Mkt Value as of 3/29/18 ($M)', 'Employees', 'CEO', 'CEO Title',
       'Sector', 'Industry_x', 'Years on Fortune 500 List', 'City', 'State',
       'Latitude', 'Longitude', 'Year', 'Industry_y', 'Cause'],
      dtype='object')
```

```
df_outer.pop('rank')
df_outer.pop('Previous Rank')
df_outer.pop('Industry_y')
df_outer.pop('Latitude')
df_outer.pop('Longitude')
df_outer.pop('Years on Fortune 500 List')
df_outer.pop('CEO')
df_outer.pop('CEO Title')
```

```
0      President, Chief Executive Officer &  Director
2               Chairman &  Chief Executive Officer
```

```
3      Chairman, President &  Chief Executive Officer
4               Chairman &  Chief Executive Officer
5               Chairman &  Chief Executive Officer
                    ...
1012            Chairman &  Chief Executive Officer
1013            Chairman &  Chief Executive Officer
1014    President, Chief Executive Officer &  Director
1015    President, Chief Executive Officer &  Director
1016    President, Chief Executive Officer &  Director
Name: CEO Title, Length: 1000, dtype: object
```

```
df_outer.pop('Year')
```

```
0      2018.0
2         NaN
3         NaN
4         NaN
5      2018.0
       ...
1012      NaN
1013      NaN
1014      NaN
1015      NaN
1016      NaN
Name: Year, Length: 1000, dtype: float64
```

```
df_outer.pop('State')
```

```
0      AR
2      TX
3      NE
4      CA
5      MN
       ..
1012   GA
1013   MA
1014   CA
1015   CA
1016   CA
Name: State, Length: 1000, dtype: object
```

```
df_outer
```

```
                     Entity Revenues ($M) Revenue Change  \
0                     Walmart     $500,343          3.00%
2                  Exxon Mobil     $244,363         17.40%
3              Berkshire Hathaway     $242,137          8.30%
4                       Apple     $229,234          6.30%
5              UnitedHealth Group     $201,159          8.80%
...                       ...          ...          ...
1012       SiteOne Landscape Supply      $1,862         13.00%
1013  Charles River Laboratories Intl      $1,858         10.50%
1014                  CoreLogic      $1,851         -5.20%
```

```
1015                Ensign Group      $1,849        11.80%
1016                         HCP      $1,848       -27.20%

    Profits ($M) Profit Change Assets ($M) Mkt Value as of 3/29/18 ($M)  \
0     $9,862.00       -27.70%   $204,522                 $263,563
2    $19,710.00       151.40%   $348,691                  $316,157
3    $44,940.00        86.70%   $702,095                  $492,008
4    $48,351.00         5.80%   $375,319                 $851,318
5    $10,558.00        50.50%   $139,058                  $207,080
...         ...          ...        ...                       ...
1012     $54.60        78.40%       $911                 $3,084
1013    $123.40       -20.30%     $2,930                   $5,118
1014    $152.20        42.80%     $4,077                   $3,694
1015     $40.50       -19.00%     $1,102                   $1,354
1016    $414.20       -34.00%    $14,089                   $10,910

     Employees          Sector                        Industry_x  \
0    2,300,000        Retailing                 General Merchandisers
2       71,200          Energy                    Petroleum Refining
3      377,000      Financials  Insurance: Property and Casualty (Stock)
4      123,000      Technology            Computers, Office Equipment
5      260,000      Health Care   Health Care: Insurance and Managed Care
...        ...             ...                               ...
1012     3,664      Wholesalers               Wholesalers: Diversified
1013    11,800      Health Care  Health Care: Pharmacy and Other Services
1014     5,900  Business Services               Financial Data Services
1015    21,301      Health Care         Health Care: Medical Facilities
1016       190      Financials                          Real estate

           City              Cause
0     Bentonville  Unintended disclosure
2           Irving                NaN
3           Omaha                 NaN
4        Cupertino                NaN
5       Minnetonka    Hacking or malware
...            ...              ...
1012       Roswell                NaN
1013    Wilmington                NaN
1014         Irvine               NaN
1015  Mission Viejo                NaN
1016         Irvine               NaN

[1000 rows x 12 columns]
```

```
df_outer.to_excel(r'C:\Users\Ricar\Desktop\Revisao\Datasets\df_outer.xlsx')
```

```
df_outer.Sector.unique()
```

```
array(['Retailing', 'Energy', 'Financials', 'Technology', 'Health Care',
       'Wholesalers', 'Telecommunications', 'Motor Vehicles &  Parts',
       'Food &  Drug Stores', 'Industrials', 'Aerospace &  Defense',
       'Household Products', 'Transportation',
```

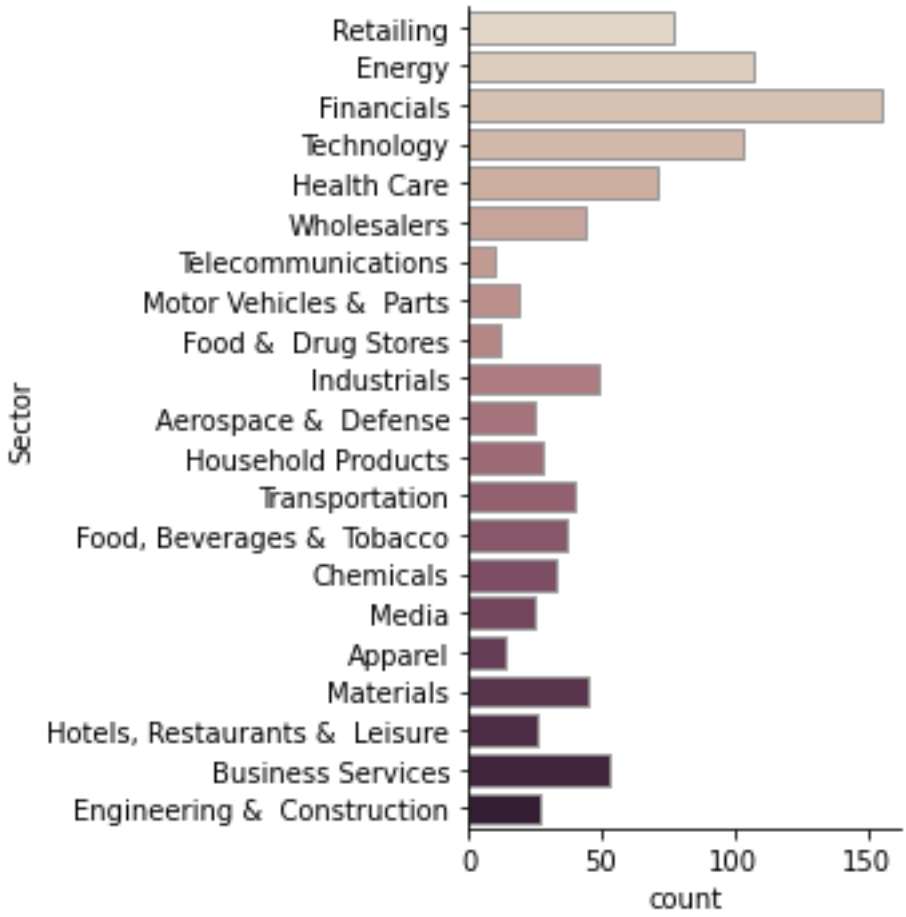
```
      'Food, Beverages &  Tobacco', 'Chemicals', 'Media', 'Apparel',
      'Materials', 'Hotels, Restaurants &  Leisure', 'Business Services',
      'Engineering &  Construction'], dtype=object)
```

```
df_outer.Industry_x.unique()
```

```
array(['General Merchandisers', 'Petroleum Refining',
      'Insurance: Property and Casualty (Stock)',
      'Computers, Office Equipment',
      'Health Care: Insurance and Managed Care',
      'Wholesalers: Health Care',
      'Health Care: Pharmacy and Other Services',
      'Internet Services and Retailing', 'Telecommunications',
      'Motor Vehicles and Parts', 'Food and Drug Stores',
      'Industrial Machinery', 'Commercial Banks',
      'Diversified Financials', 'Specialty Retailers: Other',
      'Aerospace and Defense', 'Computer Software',
      'Information Technology Services',
      'Insurance: Property and Casualty (Mutual)', 'Pharmaceuticals',
      'Household and Personal Products',
      'Insurance: Life, Health (stock)',
      'Mail, Package, and Freight Delivery', 'Food Consumer Products',
      'Semiconductors and Other Electronic Components', 'Chemicals',
      'Food Production', 'Wholesalers: Food and Grocery',
      'Entertainment', 'Network and Other Communications Equipment',
      'Health Care: Medical Facilities', 'Pipelines',
      'Construction and Farm Machinery',
      'Insurance: Life, Health (Mutual)', 'Airlines',
      'Electronics, Electrical Equip.',
      'Wholesalers: Electronics and Office Equipment',
      'Specialty Retailers: Apparel', 'Beverages', 'Apparel', 'Energy',
      'Utilities: Gas and Electric', 'Mining, Crude-Oil Production',
      'Miscellaneous', 'Tobacco', 'Medical Products and Equipment',
      'Packaging, Containers', 'Hotels, Casinos, Resorts',
      'Food Services', 'Automotive Retailing, Services', 'Railroads',
      'Temporary Help', 'Scientific,Photographic and  Control Equipment',
      'Oil and Gas Equipment, Services', 'Metals',
      'Engineering, Construction', 'Financial Data Services',
      'Wholesalers: Diversified', 'Transportation and Logistics',
      'Advertising, marketing', 'Home Equipment, Furnishings',
      'Diversified Outsourcing Services', 'Waste Management',
      'Real estate', 'Homebuilders', 'Securities',
      'Publishing, Printing', 'Trucking, Truck Leasing',
      'Forest and Paper Products', 'Building Materials, Glass',
      'Transportation Equipment', 'Toys, Sporting Goods', 'Education',
      'Shipping'], dtype=object)
```

```
sns.catplot(y="Sector", kind="count", palette="ch:.25", edgecolor=".6", data=df_outer)
```

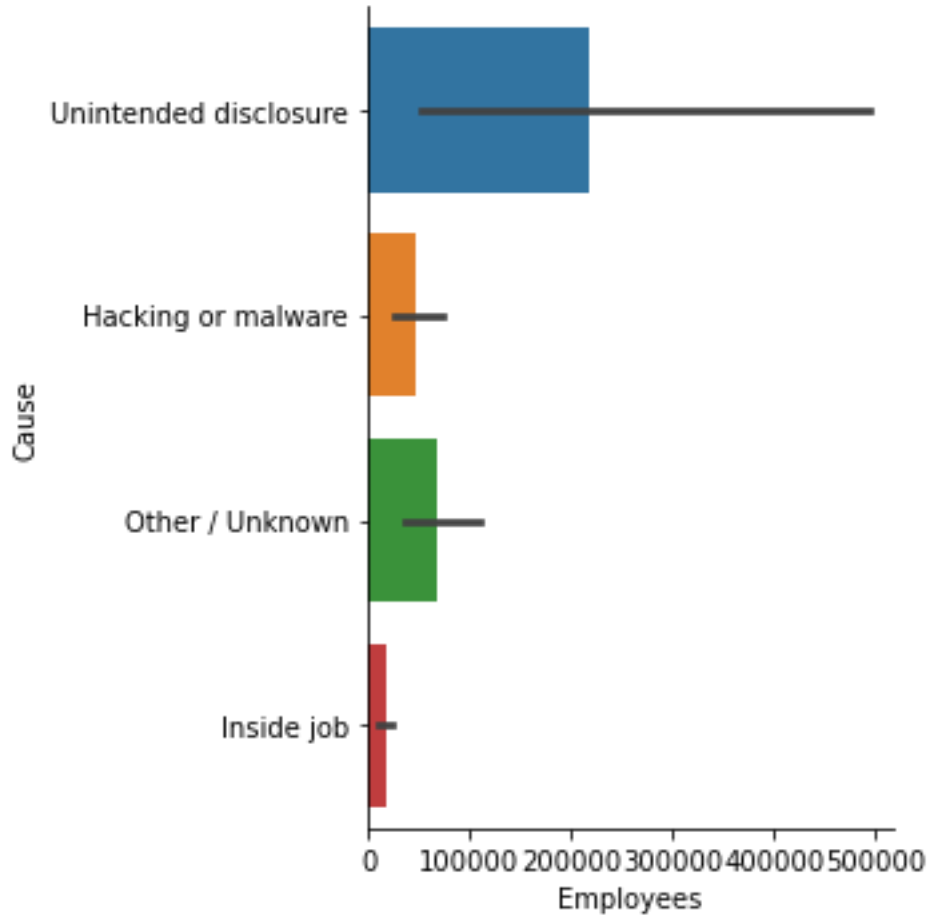
```
<seaborn.axisgrid.FacetGrid at 0x13df2c795e0>
```




```
df_outer["Employees"] = df_outer["Employees"].str.replace(",","").astype(float)
```

```
sns.catplot(y='Cause',
        x='Employees',
        data=df_outer,
        kind='bar')
```

```
<seaborn.axisgrid.FacetGrid at 0x13df32c11c0>
```




```
df_outer['Cause'] = df_outer['Cause'].replace(np.nan, 'No disclosure')

df_outer['Attacked'] = ''

df_outer.assign(Attacked='No')
```

	Entity	Revenues ($M)	Revenue Change
0	Walmart	$500,343	3.00%
2	Exxon Mobil	$244,363	17.40%
3	Berkshire Hathaway	$242,137	8.30%
4	Apple	$229,234	6.30%
5	UnitedHealth Group	$201,159	8.80%
...	...	...	...
1012	SiteOne Landscape Supply	$1,862	13.00%
1013	Charles River Laboratories Intl	$1,858	10.50%
1014	CoreLogic	$1,851	-5.20%
1015	Ensign Group	$1,849	11.80%
1016	HCP	$1,848	-27.20%

	Profits ($M)	Profit Change	Assets ($M)	Mkt Value as of 3/29/18 ($M)
0	$9,862.00	-27.70%	$204,522	$263,563
2	$19,710.00	151.40%	$348,691	$316,157
3	$44,940.00	86.70%	$702,095	$492,008
4	$48,351.00	5.80%	$375,319	$851,318
5	$10,558.00	50.50%	$139,058	$207,080

```
...        ...        ...       ...                    ...
1012     $54.60       78.40%     $911                 $3,084
1013    $123.40      -20.30%    $2,930                 $5,118
1014    $152.20       42.80%    $4,077                 $3,694
1015     $40.50      -19.00%    $1,102                 $1,354
1016    $414.20      -34.00%   $14,089                 $10,910

     Employees          Sector                        Industry_x  \
0    2300000.0       Retailing               General Merchandisers
2      71200.0          Energy                  Petroleum Refining
3     377000.0      Financials  Insurance: Property and Casualty (Stock)
4     123000.0      Technology           Computers, Office Equipment
5     260000.0      Health Care  Health Care: Insurance and Managed Care
...        ...             ...                                 ...
1012    3664.0      Wholesalers              Wholesalers: Diversified
1013   11800.0      Health Care  Health Care: Pharmacy and Other Services
1014    5900.0  Business Services               Financial Data Services
1015   21301.0      Health Care         Health Care: Medical Facilities
1016     190.0      Financials                       Real estate

           City              Cause Attacked
0     Bentonville  Unintended disclosure      No
2          Irving         No disclosure      No
3           Omaha         No disclosure      No
4        Cupertino        No disclosure      No
5       Minnetonka     Hacking or malware      No
...          ...                ...     ...
1012       Roswell        No disclosure      No
1013    Wilmington        No disclosure      No
1014        Irvine        No disclosure      No
1015  Mission Viejo        No disclosure      No
1016        Irvine        No disclosure      No

[1000 rows x 13 columns]
```

```
df_outer['Attacked'] = np.where(df_outer['Cause'] != 'No disclosure', 'Yes', 'No')
```

```
df_outer.Attacked.nunique()
```

```
2
```

```
occur = df_outer.groupby(['Attacked']).size()
display(occur)
```

```
Attacked
No     943
Yes     57
dtype: int64
```

```
df_outer["Revenues ($M)"] = df_outer["Revenues ($M)"].str.replace("$","")
```

```
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/2857820018.py:1: FutureWarning: The defaul
t value of regex will change from True to False in a future version. In addition, single character regula
```

```
r expressions will *not* be treated as literal strings when regex=True.
  df_outer["Revenues ($M)"] = df_outer["Revenues ($M)"].str.replace("$","")

df_outer["Revenues ($M)"] = df_outer["Revenues ($M)"].str.replace(",","").astype(float)

df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace("$","")
df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace(")","")
df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace("(","")
df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace("-","0")
df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace(",","").astype(float)

C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/475581163.py:1: FutureWarning: The default
value of regex will change from True to False in a future version. In addition, single character regular
expressions will *not* be treated as literal strings when regex=True.
  df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace("$","")
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/475581163.py:2: FutureWarning: The default
value of regex will change from True to False in a future version. In addition, single character regular
expressions will *not* be treated as literal strings when regex=True.
  df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace(")","")
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/475581163.py:3: FutureWarning: The default
value of regex will change from True to False in a future version. In addition, single character regular
expressions will *not* be treated as literal strings when regex=True.
  df_outer["Profits ($M)"] = df_outer["Profits ($M)"].str.replace("(","")

df_outer['Revenue Change'] = df_outer['Revenue Change'].str.rstrip("%")

df_outer["Profit Change"] = df_outer["Profit Change"].str.replace("%","")
df_outer

                        Entity  Revenues ($M) Revenue Change  \
0                      Walmart       500343.0          3.00
2                  Exxon Mobil       244363.0         17.40
3              Berkshire Hathaway       242137.0          8.30
4                        Apple       229234.0          6.30
5              UnitedHealth Group       201159.0          8.80
...                          ...          ...          ...
1012       SiteOne Landscape Supply        1862.0         13.00
1013  Charles River Laboratories Intl        1858.0         10.50
1014                   CoreLogic        1851.0         -5.20
1015                Ensign Group        1849.0         11.80
1016                         HCP        1848.0        -27.20

     Profits ($M) Profit Change Assets ($M) Mkt Value as of 3/29/18 ($M)  \
0          9862.0       -27.70   $204,522                 $263,563
2         19710.0       151.40   $348,691                  $316,157
3         44940.0        86.70   $702,095                 $492,008
4         48351.0         5.80   $375,319                 $851,318
5         10558.0        50.50   $139,058                  $207,080
...           ...          ...        ...                       ...
1012         54.6        78.40       $911                  $3,084
1013        123.4       -20.30     $2,930                    $5,118
1014        152.2        42.80     $4,077                    $3,694
1015         40.5       -19.00     $1,102                    $1,354
```

```
1016        414.2       -34.00    $14,089                 $10,910

     Employees           Sector                        Industry_x  \
0     2300000.0         Retailing                 General Merchandisers
2       71200.0           Energy                     Petroleum Refining
3      377000.0        Financials  Insurance: Property and Casualty (Stock)
4      123000.0        Technology            Computers, Office Equipment
5      260000.0       Health Care   Health Care: Insurance and Managed Care
...         ...              ...                                ...
1012     3664.0       Wholesalers               Wholesalers: Diversified
1013    11800.0       Health Care  Health Care: Pharmacy and Other Services
1014     5900.0  Business Services               Financial Data Services
1015    21301.0       Health Care         Health Care: Medical Facilities
1016      190.0        Financials                        Real estate

            City              Cause Attacked
0      Bentonville  Unintended disclosure     Yes
2          Irving         No disclosure      No
3           Omaha         No disclosure      No
4        Cupertino         No disclosure      No
5       Minnetonka     Hacking or malware     Yes
...            ...                ...     ...
1012        Roswell         No disclosure      No
1013     Wilmington         No disclosure      No
1014         Irvine         No disclosure      No
1015   Mission Viejo         No disclosure      No
1016         Irvine         No disclosure      No

[1000 rows x 13 columns]

df_outer.dtypes

Entity                      object
Revenues ($M)              float64
Revenue Change              object
Profits ($M)               float64
Profit Change               object
Assets ($M)                 object
Mkt Value as of 3/29/18 ($M)    object
Employees                  float64
Sector                      object
Industry_x                  object
City                        object
Cause                       object
Attacked                    object
dtype: object

df_outer

                      Entity  Revenues ($M) Revenue Change  \
0                    Walmart      500343.0          3.00
2                 Exxon Mobil     244363.0         17.40
```

```
3              Berkshire Hathaway      242137.0          8.30
4                           Apple      229234.0          6.30
5              UnitedHealth Group      201159.0          8.80
...                            ...          ...          ...
1012        SiteOne Landscape Supply        1862.0         13.00
1013  Charles River Laboratories Intl        1858.0         10.50
1014                    CoreLogic        1851.0         -5.20
1015                 Ensign Group        1849.0         11.80
1016                          HCP        1848.0        -27.20

     Profits ($M) Profit Change Assets ($M) Mkt Value as of 3/29/18 ($M)  \
0          9862.0       -27.70   $204,522                 $263,563
2         19710.0       151.40   $348,691                  $316,157
3         44940.0        86.70   $702,095                  $492,008
4         48351.0         5.80   $375,319                 $851,318
5         10558.0        50.50   $139,058                  $207,080
...           ...          ...        ...                       ...
1012         54.6        78.40       $911                 $3,084
1013        123.4       -20.30     $2,930                   $5,118
1014        152.2        42.80     $4,077                   $3,694
1015         40.5       -19.00     $1,102                  $1,354
1016        414.2       -34.00    $14,089                   $10,910

     Employees          Sector                        Industry_x  \
0    2300000.0       Retailing                 General Merchandisers
2      71200.0          Energy                   Petroleum Refining
3     377000.0      Financials  Insurance: Property and Casualty (Stock)
4     123000.0      Technology            Computers, Office Equipment
5     260000.0      Health Care   Health Care: Insurance and Managed Care
...        ...             ...                                   ...
1012    3664.0      Wholesalers               Wholesalers: Diversified
1013   11800.0      Health Care  Health Care: Pharmacy and Other Services
1014    5900.0  Business Services                Financial Data Services
1015   21301.0      Health Care         Health Care: Medical Facilities
1016     190.0      Financials                          Real estate

            City              Cause Attacked
0     Bentonville  Unintended disclosure     Yes
2          Irving         No disclosure      No
3           Omaha         No disclosure      No
4        Cupertino         No disclosure      No
5       Minnetonka     Hacking or malware     Yes
...            ...                 ...     ...
1012        Roswell         No disclosure      No
1013     Wilmington         No disclosure      No
1014         Irvine         No disclosure      No
1015   Mission Viejo         No disclosure      No
1016         Irvine         No disclosure      No

[1000 rows x 13 columns]
```

```
df_outer["Assets ($M)"] = df_outer["Assets ($M)"].str.replace("$","")
```

```
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/2324635848.py:1: FutureWarning: The defaul
t value of regex will change from True to False in a future version. In addition, single character regula
r expressions will *not* be treated as literal strings when regex=True.
  df_outer["Assets ($M)"] = df_outer["Assets ($M)"].str.replace("$","")
```

```
df_outer["Mkt Value as of 3/29/18 ($M)"] = df_outer["Mkt Value as of 3/29/18 ($M)"].str.replace("$"
,"")
```

```
C:\Users\Ricar\AppData\Local\Temp/ipykernel_10680/2177219366.py:1: FutureWarning: The defaul
t value of regex will change from True to False in a future version. In addition, single character regula
r expressions will *not* be treated as literal strings when regex=True.
  df_outer["Mkt Value as of 3/29/18 ($M)"] = df_outer["Mkt Value as of 3/29/18 ($M)"].str.replace("
$","")
```

```
df_outer.rename(columns = {'Revenues ($M)':'Revenues', 'Profits ($M)':'Profits', 'Assets ($M)':'Assets
', 'Mkt Value as of 3/29/18 ($M)':'Mkt Value as of 3/29/18', 'Industry_x':'Industry'}, inplace = True)
```

```
df_outer
```

	Entity	Revenues	Revenue Change	Profits	Profit Change	Assets	Mkt Value as of 3/29/18	Employees	Sector	Industry
0	Walmart	500343.0	3.00	9862.0	-27.70	204,522	263,563	2300000.0	Retailing	General Merchandisers
2	Exxon Mobil	244363.0	17.40	19710.0	151.40	348,691	316,157	71200.0	Energy	Petroleum Refining
3	Berkshire Hathaway	242137.0	8.30	44940.0	86.70	702,095	492,008	377000.0	Financials	Insurance: Property and Casualty (Stock)
4	Apple	229234.0	6.30	48351.0	5.80	375,319	851,318	123000.0	Technology	Computers, Office Equipment
5	UnitedHealth Group	201159.0	8.80	10558.0	50.50	139,058	207,080	260000.0	Health Care	Health Care: Insurance and Managed Care
...	...	...	...	...	...	...	...	...	...	...
1012	SiteOne Landscape Supply	1862.0	13.00	54.6	78.40	911	3,084	3664.0		
1013	Charles River Laboratories Intl	1858.0	10.50	123.4	-20.30	2,930	5,118	11800.0		
1014	CoreLogic	1851.0	-5.20	152.2	42.80	4,077	3,694	5900.0		
1015	Ensign Group	1849.0	11.80	40.5	-19.00	1,102	1,354	21301.0		
1016	HCP	1848.0	-27.20	414.2	-34.00	14,089	10,910	190.0		

```
...              ...                                ...
1012       Wholesalers                Wholesalers: Diversified
1013       Health Care  Health Care: Pharmacy and Other Services
1014  Business Services                Financial Data Services
1015       Health Care          Health Care: Medical Facilities
1016        Financials                             Real estate

            City                Cause Attacked
0      Bentonville  Unintended disclosure      Yes
2          Irving         No disclosure       No
3           Omaha         No disclosure       No
4        Cupertino         No disclosure       No
5       Minnetonka     Hacking or malware      Yes
...           ...                 ...     ...
1012       Roswell         No disclosure       No
1013    Wilmington         No disclosure       No
1014        Irvine         No disclosure       No
1015  Mission Viejo         No disclosure       No
1016        Irvine         No disclosure       No

[1000 rows x 13 columns]
```

```
df_outer["Assets"] = df_outer["Assets"].str.replace(",","")
df_outer['Assets'] = df_outer['Assets'].str.rstrip("%").astype(float)

df_outer["Mkt Value as of 3/29/18"] = df_outer["Mkt Value as of 3/29/18"].str.replace(",","")
df_outer["Mkt Value as of 3/29/18"] = df_outer["Mkt Value as of 3/29/18"].str.replace("-","0")
df_outer['Mkt Value as of 3/29/18'] = df_outer['Mkt Value as of 3/29/18'].str.rstrip("%").astype(float)

df_outer['Mkt Value as of 3/29/18'].unique()
```

```
array([2.63563e+05, 3.16157e+05, 4.92008e+05, 8.51318e+05, 2.07080e+05,
       2.90670e+04, 6.31140e+04, 7.00668e+05, 2.18946e+05, 5.09720e+04,
       4.42440e+04, 1.89380e+04, 2.17845e+05, 1.97260e+04, 8.26870e+04,
       1.97497e+05, 2.08280e+04, 1.17054e+05, 6.49240e+04, 3.75043e+05,
       1.63300e+03, 7.19124e+05, 2.06272e+05, 3.06618e+05, 3.87910e+04,
       2.55556e+05, 1.92539e+05, 4.47300e+04, 5.61820e+04, 7.02760e+05,
       3.99770e+04, 1.72822e+05, 1.58703e+05, 1.41335e+05, 0.00000e+00,
       3.43780e+05, 8.78000e+02, 3.74090e+04, 7.28120e+04, 3.46830e+04,
       1.99865e+05, 4.75720e+04, 9.01560e+04, 1.54933e+05, 2.43109e+05,
       1.48186e+05, 2.42380e+04, 5.52290e+04, 6.41610e+04, 1.00667e+05,
       4.36860e+04, 3.12940e+04, 1.51029e+05, 3.71220e+04, 2.11115e+05,
       3.58930e+04, 9.65890e+04, 4.91450e+04, 1.87040e+04, 2.06623e+05,
       3.41650e+04, 1.53350e+04, 8.80780e+04, 9.66880e+04, 9.54630e+04,
       2.45840e+04, 2.04600e+04, 4.07350e+04, 8.09540e+04, 3.87460e+04,
       4.64190e+05, 1.08149e+05, 1.46862e+05, 3.34780e+04, 2.92330e+04,
       1.97780e+04, 1.86766e+05, 3.24900e+03, 5.15710e+04, 8.02340e+04,
       1.85207e+05, 1.08094e+05, 1.53860e+04, 1.66100e+03, 3.76440e+04,
       1.79300e+03, 6.96410e+04, 1.30550e+05, 7.37580e+04, 6.58450e+04,
       4.65840e+04, 5.02910e+04, 8.04000e+02, 5.29040e+04, 3.76910e+04,
       2.72430e+04, 1.54514e+05, 6.79690e+04, 1.50180e+05, 1.04640e+05,
       3.54780e+04, 6.75000e+03, 7.59200e+04, 6.32800e+03, 9.82980e+04,
```

6.20660e+04, 6.07780e+04, 6.22650e+04, 9.06500e+03, 7.06500e+03,
8.33670e+04, 2.51410e+04, 2.20630e+04, 5.42760e+04, 4.50240e+04,
4.85310e+04, 5.00900e+03, 8.45420e+04, 1.22842e+05, 1.24244e+05,
8.13700e+04, 8.20270e+04, 2.25230e+04, 3.75200e+03, 9.90100e+03,
3.40380e+04, 4.29600e+03, 3.81700e+03, 1.08240e+04, 1.04261e+05,
3.36780e+04, 7.61400e+03, 8.29520e+04, 1.03415e+05, 4.10680e+04,
2.45200e+03, 1.24530e+04, 2.67570e+04, 6.04700e+04, 1.94320e+04,
4.86700e+03, 8.00600e+03, 1.18436e+05, 2.32980e+04, 1.83920e+04,
1.10210e+04, 2.74560e+04, 5.03500e+03, 4.54000e+02, 2.70222e+05,
6.83850e+04, 3.85530e+04, 5.67100e+03, 7.13230e+04, 1.75780e+04,
7.68950e+04, 2.26640e+04, 4.70200e+03, 8.62200e+03, 3.10400e+03,
2.88000e+02, 2.54900e+04, 1.12020e+04, 5.19190e+04, 2.54390e+04,
1.31830e+04, 4.33590e+04, 1.20010e+04, 5.85000e+02, 1.21470e+04,
2.67160e+04, 8.12300e+03, 6.26070e+04, 3.37660e+04, 1.22100e+04,
6.39200e+03, 1.67340e+04, 1.07170e+04, 3.68990e+04, 2.78240e+04,
1.02135e+05, 1.31530e+04, 1.63720e+04, 4.73380e+04, 1.23630e+04,
1.96680e+04, 4.64700e+03, 5.14440e+04, 9.72500e+03, 5.84290e+04,
3.63720e+04, 1.76820e+04, 5.31420e+04, 1.59460e+04, 8.65000e+03,
1.60420e+04, 1.52930e+04, 2.98000e+04, 1.76430e+04, 1.64800e+04,
4.19240e+04, 1.67250e+04, 7.51000e+03, 3.95240e+04, 1.61340e+04,
3.32230e+04, 1.61750e+04, 4.97380e+04, 1.29190e+04, 9.10560e+04,
1.33200e+03, 6.71020e+04, 1.11240e+04, 2.25320e+04, 5.67210e+04,
2.36100e+04, 1.00459e+05, 1.87380e+04, 1.06550e+04, 1.87600e+04,
4.53440e+04, 9.93000e+03, 9.43000e+02, 1.84161e+05, 8.70650e+04,
1.03300e+04, 6.00410e+04, 8.09800e+03, 2.94030e+04, 5.03020e+04,
2.07410e+04, 5.79300e+04, 6.20000e+03, 9.81000e+02, 2.98900e+03,
2.33730e+04, 4.04540e+04, 5.76950e+04, 2.16130e+04, 1.48650e+04,
2.41920e+04, 2.27550e+04, 3.11310e+04, 5.50640e+04, 3.66880e+04,
4.49550e+04, 1.28167e+05, 8.37200e+03, 2.54310e+04, 4.14340e+04,
4.94280e+04, 2.31070e+04, 1.05300e+04, 1.11630e+04, 9.69800e+03,
6.09130e+04, 2.93510e+04, 1.36600e+03, 1.42980e+04, 1.62380e+04,
1.62830e+04, 1.39200e+04, 1.74500e+04, 2.44000e+03, 1.98440e+04,
1.19500e+03, 7.09000e+03, 3.49790e+04, 4.12260e+04, 8.50740e+04,
1.64820e+04, 1.58230e+04, 1.25870e+04, 1.27100e+04, 8.77200e+03,
7.33000e+03, 9.91800e+03, 2.36770e+04, 2.51500e+03, 1.67640e+04,
2.18320e+04, 8.38000e+03, 1.17670e+04, 1.17500e+04, 1.05860e+04,
1.09830e+04, 3.18630e+04, 2.04330e+04, 6.24400e+03, 1.40112e+05,
8.68600e+03, 1.18220e+04, 4.07260e+04, 1.50730e+04, 1.04440e+04,
2.74600e+03, 1.72830e+04, 5.28100e+03, 8.77000e+03, 9.61000e+02,
2.16500e+03, 2.82200e+03, 1.47760e+04, 1.27900e+03, 1.81740e+04,
2.49570e+04, 5.82000e+02, 3.18930e+04, 2.53590e+04, 3.76880e+04,
2.06070e+04, 7.03130e+04, 5.22000e+02, 1.16490e+04, 3.79200e+03,
9.62900e+03, 1.66200e+03, 6.81600e+03, 3.76500e+03, 8.43900e+03,
3.45100e+03, 5.03300e+03, 3.48180e+04, 6.17000e+02, 5.15700e+03,
1.41300e+03, 3.92000e+03, 9.27200e+03, 6.20000e+02, 1.43850e+04,
2.05670e+04, 3.31050e+04, 7.64600e+03, 7.93000e+03, 2.52200e+03,
1.30210e+04, 8.85400e+03, 1.51830e+04, 2.21510e+04, 7.87000e+02,
5.37900e+03, 1.86710e+04, 1.18060e+04, 2.18120e+04, 1.36190e+04,
4.54900e+03, 8.83600e+03, 2.92000e+03, 1.37230e+04, 1.97840e+04,
1.25900e+04, 2.87200e+04, 1.38160e+04, 2.07740e+04, 1.96350e+04,
1.15390e+04, 9.35800e+03, 1.40870e+04, 3.41600e+03, 2.08530e+04,

4.43790e+04, 3.86700e+03, 1.39930e+04, 1.06413e+05, 1.83200e+03,
7.80700e+03, 6.06900e+03, 3.74100e+03, 2.64630e+04, 1.28580e+04,
1.05310e+04, 1.60090e+04, 2.10580e+04, 7.19100e+03, 2.26400e+03,
5.11770e+04, 6.54000e+03, 2.63110e+04, 8.60000e+01, 9.59500e+03,
6.12000e+02, 1.91490e+04, 1.10870e+04, 1.11560e+04, 3.53800e+03,
1.86300e+03, 2.05700e+04, 9.92300e+03, 5.76800e+03, 2.21080e+04,
8.36000e+02, 1.25200e+03, 2.12780e+04, 6.40730e+04, 2.69790e+04,
9.08800e+03, 3.32800e+03, 3.94000e+02, 1.45370e+04, 4.12000e+03,
9.36100e+03, 3.87400e+03, 1.27930e+04, 7.04000e+03, 1.30190e+04,
3.73700e+03, 6.63800e+03, 1.41200e+03, 2.04740e+04, 1.06330e+04,
9.46000e+03, 1.46780e+04, 2.28900e+03, 4.80300e+03, 1.91330e+04,
8.98200e+03, 1.70270e+04, 1.02620e+04, 2.22600e+04, 2.15900e+03,
5.08100e+03, 1.69900e+03, 5.77800e+03, 8.84000e+03, 3.90000e+01,
1.37400e+04, 1.25120e+04, 1.36110e+04, 7.16200e+03, 7.68500e+03,
3.56800e+03, 9.01800e+03, 2.08610e+04, 1.42800e+03, 4.76430e+04,
1.62640e+04, 1.44000e+02, 3.92300e+03, 2.73790e+04, 1.72250e+04,
2.41600e+03, 9.88200e+03, 1.24590e+04, 2.83070e+04, 3.70870e+04,
2.89000e+02, 2.60500e+03, 4.21320e+04, 9.31300e+03, 1.14110e+04,
6.56600e+03, 4.99200e+03, 5.61700e+03, 6.51100e+03, 1.21850e+04,
1.25100e+03, 2.32640e+04, 2.94660e+04, 7.19800e+03, 1.27030e+04,
2.10400e+03, 4.64900e+03, 1.04060e+04, 4.78610e+04, 8.85700e+03,
1.65280e+04, 7.22400e+03, 2.92540e+04, 2.20000e+03, 8.92500e+03,
1.81650e+04, 1.59480e+04, 1.38220e+04, 1.11600e+03, 2.41000e+02,
3.58600e+03, 9.74000e+03, 4.05240e+04, 4.44100e+03, 8.71700e+03,
1.83000e+03, 7.19400e+03, 4.96800e+03, 1.04680e+04, 5.01700e+03,
3.45600e+03, 4.43970e+04, 2.66700e+03, 1.37070e+04, 1.92910e+04,
3.37000e+04, 3.11500e+03, 2.94000e+02, 4.82900e+03, 8.36400e+03,
2.39400e+03, 6.75900e+03, 6.75600e+03, 6.37200e+03, 1.56760e+04,
5.18800e+03, 2.99000e+02, 4.52300e+03, 8.06800e+03, 7.83800e+03,
3.71880e+04, 1.95200e+03, 1.39640e+04, 2.64090e+04, 7.66500e+03,
1.09800e+03, 1.62090e+04, 5.97700e+03, 1.60900e+03, 1.38390e+04,
9.05400e+03, 3.12200e+03, 1.17000e+03, 2.88620e+04, 4.12000e+02,
7.67000e+03, 1.49880e+04, 1.49300e+03, 9.02500e+03, 2.15740e+04,
3.33200e+03, 5.50000e+03, 2.31300e+03, 1.50000e+03, 2.59800e+03,
5.35600e+03, 4.75400e+03, 1.57030e+04, 4.61000e+03, 2.57800e+03,
3.60900e+03, 2.46800e+03, 3.31280e+04, 3.03000e+03, 4.54520e+04,
3.72700e+03, 6.75200e+03, 1.61500e+03, 2.02460e+04, 5.94100e+03,
1.75900e+04, 3.63000e+03, 3.08400e+03, 5.51000e+03, 1.49130e+04,
2.25600e+03, 3.08260e+04, 1.60670e+04, 2.42500e+03, 6.23100e+03,
2.27100e+03, 1.21490e+04, 9.58300e+03, 1.27960e+04, 1.33600e+03,
8.80500e+03, 6.40800e+03, 3.07400e+03, 4.00200e+03, 1.41340e+04,
1.43900e+03, 2.59150e+04, 6.86000e+02, 8.36700e+03, 1.14390e+04,
5.41000e+02, 1.77890e+04, 5.61100e+03, 1.43610e+04, 5.84200e+03,
1.98600e+03, 2.05800e+03, 4.60400e+03, 3.60000e+02, 2.63900e+03,
1.51100e+04, 2.61000e+03, 7.82200e+03, 1.82400e+03, 9.37500e+03,
8.41500e+03, 1.49900e+03, 3.54000e+03, 1.22940e+04, 2.77400e+03,
3.25700e+03, 4.06200e+03, 3.93400e+03, 7.41200e+03, 3.36050e+04,
2.14500e+03, 6.67800e+03, 4.92100e+03, 2.84800e+03, 5.66800e+03,
6.28300e+03, 1.82640e+04, 5.49400e+03, 1.51700e+03, 5.50580e+04,
2.63600e+03, 9.60000e+02, 4.00100e+03, 6.08000e+02, 5.88700e+03,
6.88000e+02, 5.00700e+03, 9.09000e+02, 8.93200e+03, 2.47840e+04,

2.03800e+03, 6.64900e+03, 1.60210e+04, 2.97400e+03, 7.49000e+02,
5.86800e+03, 1.65100e+03, 1.70730e+04, 1.10200e+03, 5.25100e+03,
7.28500e+03, 3.40000e+03, 1.02970e+04, 2.93720e+04, 2.14100e+04,
6.76700e+03, 2.53800e+03, 3.15000e+03, 4.90200e+03, 1.08040e+04,
3.26600e+03, 9.45300e+03, 1.25650e+04, 3.24040e+04, 1.31400e+03,
3.03000e+02, 1.41520e+04, 3.22500e+03, 1.21070e+04, 1.46220e+04,
4.11900e+03, 2.71800e+03, 3.34100e+03, 1.06840e+04, 1.29260e+04,
1.23400e+03, 2.92100e+03, 1.65620e+04, 3.24500e+03, 2.45900e+03,
4.34300e+03, 1.83900e+03, 2.83800e+03, 5.66400e+03, 2.54200e+03,
9.82500e+03, 2.21280e+04, 1.67300e+03, 2.40700e+03, 1.12600e+03,
3.40900e+03, 6.44000e+03, 4.66750e+04, 2.21190e+04, 6.95900e+03,
3.74300e+03, 1.02690e+04, 9.34800e+03, 1.03310e+04, 5.31600e+03,
8.91200e+03, 1.58000e+03, 2.64400e+03, 3.55600e+03, 1.09080e+04,
2.54000e+03, 4.25600e+03, 2.61600e+04, 4.38800e+03, 2.22800e+03,
3.35700e+03, 2.75800e+03, 7.41600e+03, 4.88100e+03, 3.82100e+03,
6.16000e+02, 1.49820e+04, 6.88200e+03, 6.71000e+02, 8.09000e+02,
5.17800e+03, 3.43700e+03, 1.15700e+03, 5.97000e+03, 1.26350e+04,
3.45900e+03, 1.11980e+04, 2.43700e+03, 5.42000e+02, 1.49100e+03,
3.33300e+03, 3.03800e+03, 8.22000e+02, 1.43800e+03, 2.48700e+03,
1.20950e+04, 1.84900e+03, 3.14000e+03, 5.93500e+03, 3.71900e+03,
1.79800e+03, 2.07300e+03, 5.43500e+03, 5.93100e+03, 6.18000e+02,
1.00600e+04, 5.54700e+03, 1.44400e+03, 2.46100e+03, 2.73400e+03,
3.47540e+04, 3.32200e+03, 5.75700e+03, 5.44000e+02, 1.03620e+04,
3.18500e+03, 4.17300e+03, 1.46350e+04, 1.23800e+04, 2.93300e+03,
3.44400e+03, 6.85600e+03, 4.61000e+02, 1.08500e+03, 3.71300e+03,
3.48850e+04, 1.58200e+03, 5.31000e+03, 8.92000e+03, 1.10400e+03,
2.57700e+03, 4.27200e+03, 5.90200e+03, 3.35780e+04, 3.61400e+03,
6.68200e+03, 2.68200e+03, 1.90180e+04, 2.81900e+03, 3.31200e+03,
2.24090e+04, 1.13200e+04, 1.56700e+03, 7.48000e+03, 1.75300e+03,
3.74200e+03, 2.45600e+03, 3.25900e+03, 9.36000e+02, 1.00000e+00,
8.28000e+02, 2.01200e+03, 1.06300e+03, 1.33150e+04, 6.62100e+03,
1.20690e+04, 2.94900e+03, 4.13790e+04, 2.26870e+04, 3.56600e+03,
5.60000e+03, 1.45500e+03, 5.35100e+03, 1.27000e+03, 5.80000e+02,
2.17010e+04, 3.23000e+02, 1.46920e+04, 2.71000e+02, 4.35300e+03,
2.17840e+04, 7.84000e+02, 2.05000e+03, 3.06600e+03, 8.19900e+03,
2.64640e+04, 2.89100e+03, 3.58900e+03, 1.28700e+03, 6.05100e+03,
5.85100e+03, 1.50400e+03, 2.90700e+03, 1.66700e+03, 3.89200e+03,
2.75300e+03, 1.84080e+04, 1.27860e+04, 6.89600e+03, 1.28200e+03,
2.06700e+03, 1.95810e+04, 6.02100e+03, 1.91300e+03, 1.25400e+03,
1.55530e+04, 2.51300e+03, 1.69350e+04, 1.09160e+04, 4.96200e+03,
1.15200e+03, 3.88800e+03, 1.90700e+03, 5.97600e+03, 3.51300e+03,
1.34100e+03, 1.50500e+03, 2.53100e+03, 5.40400e+03, 6.54400e+03,
2.21300e+03, 5.29200e+03, 2.47300e+03, 4.05800e+03, 1.81860e+04,
1.75500e+03, 1.03150e+04, 2.52900e+03, 1.28600e+04, 4.47100e+03,
8.52300e+03, 4.12300e+03, 4.59200e+03, 1.95010e+04, 4.43500e+03,
8.47400e+03, 4.73100e+03, 9.80000e+01, 1.56800e+03, 4.16500e+03,
1.44200e+03, 7.44300e+03, 2.27110e+04, 4.97000e+02, 1.53800e+03,
4.79700e+03, 2.60300e+03, 1.71620e+04, 4.25300e+03, 2.69470e+04,
1.12170e+04, 5.64000e+02, 8.90100e+03, 2.64500e+03, 6.18800e+03,
4.39100e+03, 8.94300e+03, 4.42800e+03, 1.69200e+03, 5.47400e+03,
2.47000e+02, 1.21500e+03, 7.90400e+03, 1.40740e+04, 5.70000e+02,

```
       1.14000e+02, 6.39000e+02, 3.27600e+03, 2.74180e+04, 1.48100e+03,
       2.09900e+03, 1.22200e+03, 5.06100e+03, 1.26940e+04, 2.79900e+03,
       7.78000e+02, 2.17400e+03, 2.71400e+03, 8.66000e+02, 2.05500e+03,
       5.79100e+03, 1.66630e+04, 3.53900e+03, 8.88000e+02, 1.51200e+03,
       1.04200e+03, 3.53500e+03, 2.09100e+03, 1.35300e+03, 7.74000e+02,
       1.04150e+04, 4.62600e+03, 5.59000e+02, 1.04410e+04, 2.89040e+04,
       3.34200e+03, 8.99000e+02, 4.80600e+03, 1.87500e+03, 1.82700e+03,
       8.92400e+03, 3.08600e+03, 5.36000e+02, 6.30700e+03, 8.96200e+03,
       2.72500e+03, 7.02500e+03, 2.11300e+03, 1.30000e+03, 1.69800e+03,
       2.33000e+03, 3.54400e+03, 3.20400e+03, 5.11800e+03, 3.69400e+03,
       1.35400e+03, 1.09100e+04])
```

```
df_outer['Revenue Change'].unique()
```

```
array(['3.00', '17.40', '8.30', '6.30', '8.80', '3.10', '4.10', '30.80',
       '-2.00', '-5.50', '3.30', '4.30', '25.10', '6.90', '8.70', '0.00',
       '6.40', '-3.50', '0.70', '8.00', '4.90', '22.80', '6.70', '7.00',
       '-0.20', '3.80', '-1.20', '26.50', '6.10', '5.40', '26.00', '6.80',
       '5.10', '-1.00', '21.40', '2.90', '13.70', '3.40', '5.50', '21.00',
       '-7.70', '4.20', '8.20', '1.20', '5.70', '30.20', '-2.40', '-4.10',
       '19.80', '4.50', '1.50', '1.60', '9.90', '-0.90', '-1.10', '-0.50',
       '7.90', '0.80', '-5.40', '19.30', '-2.50', '6.50', '26.60',
       '18.00', '9.60', '15.00', '11.40', '3.70', '12.00', '5.00',
       '43.40', '4.00', '47.10', '3.20', '1.80', '40.20', '-2.90', '8.10',
       '5.20', '-15.40', '42.50', '24.70', '-11.40', '33.80', '10.60',
       '9.00', '11.60', '99.40', '27.00', '4.60', '-42.40', '7.70',
       '10.10', '31.30', '14.50', '12.50', '29.90', '-14.10', '-0.10',
       '5.30', '-3.70', '10.50', '-0.80', '15.80', '34.10', '-12.80',
       '7.80', '-0.60', '-7.30', '7.40', '36.80', '33.00', '-4.00',
       '-0.30', '6.20', '2.60', '29.80', '-2.20', '10.30', '16.70',
       '63.90', '25.00', '11.80', '2.50', '14.20', '2.20', '46.90',
       '3.90', '-13.60', '21.70', '-7.90', '0.30', '9.80', '1.10',
       '-3.00', '21.20', '19.50', '-24.60', '10.40', '6.00', '-19.60',
       '-8.70', '-5.70', '1.70', '-5.80', '1.40', '26.40', '-1.40',
       '11.90', '13.10', '1.00', '11.10', '2.30', '220.40', '8.40',
       '1.30', '34.30', '-4.70', '35.30', '15.90', '14.40', '-3.10',
       '4.80', '27.70', '20.80', '10.90', '-0.70', '12.90', '11.70',
       '15.50', '0.50', '18.60', '7.20', '8.60', '16.00', '-3.60',
       '14.10', '19.40', '0.90', '36.50', '5.90', '51.30', '10.70',
       '68.00', '32.40', '10.00', '2.70', '7.30', '46.50', '-3.40',
       '2.10', '125.20', '2.40', '12.70', '13.90', '-20.90', '9.70',
       '24.90', '2.80', '-2.30', '23.70', '-40.10', '44.40', '16.20',
       '14.70', '-8.60', '41.60', '40.60', '-10.40', '22.70', '20.60',
       '-4.20', '9.50', '-34.70', '-21.60', '-1.30', '17.20', '31.70',
       '-7.10', '30.30', '-2.80', '465.30', '12.10', '58.40', '7.10',
       '36.10', '16.60', '15.20', '0.20', '0.40', '4.70', '75.90',
       '12.20', '14.60', '43.90', '35.70', '-12.30', '58.20', '-8.90',
       '5.80', '38.90', '9.30', '11.50', '0.60', '-7.80', '13.50',
       '26.70', '5.60', '29.70', '-6.40', '22.20', '-', '-10.20', '-4.90',
       '15.30', '28.70', '-13.20', '18.20', '-0.40', '20.00', '0.10',
       '-14.80', '41.20', '58.70', '-1.50', '-9.60', '71.40', '20.70',
       '-2.10', '3.50', '336.50', '18.30', '41.90', '1.90', '42.70',
```

```
      '-1.80', '-6.30', '10.80', '49.30', '-6.20', '7.50', '17.50',
      '9.10', '-10.50', '8.50', '10.20', '-4.30', '-4.60', '25.30',
      '39.70', '24.80', '38.00', '21.60', '-7.40', '13.30', '21.50',
      '-25.30', '16.30', '37.00', '46.30', '21.80', '21.90', '-24.70',
      '16.90', '43.00', '11.30', '13.20', '6.60', '17.90', '-11.50',
      '75.30', '9.40', '54.00', '-3.30', '-6.10', '61.60', '-20.20',
      '-8.30', '109.50', '11.00', '15.10', '12.30', '19.90', '241.30',
      '2.00', '-20.40', '-9.10', '57.60', '110.00', '17.70', '23.20',
      '18.70', '35.50', '17.60', '-11.60', '-34.10', '-24.90', '31.50',
      '16.10', '-1.90', '15.70', '-12.00', '13.40', '18.90', '22.60',
      '-15.70', '-11.10', '-9.00', '12.40', '613.20', '16.80', '20.10',
      '23.40', '-8.40', '12.80', '8.90', '22.40', '-8.80', '15.40',
      '65.90', '25.70', '14.00', '7.60', '137.40', '21.10', '4.40',
      '-23.70', '3.60', '-5.00', '46.20', '-6.80', '-10.90', '31.00',
      '14.30', '120.40', '116.90', '19.20', '22.50', '157.30', '-6.60',
      '-2.70', '22.90', '217.00', '-3.80', '18.80', '29.60', '27.80',
      '-19.00', '27.90', '14.90', '36.60', '38.30', '56.30', '48.00',
      '-33.30', '18.40', '9.20', '23.30', '25.20', '39.00', '-7.00',
      '-5.30', '30.60', '52.60', '47.90', '19.10', '29.20', '13.00',
      '-5.20', '-27.20'], dtype=object)

df_outer

                      Entity  Revenues Revenue Change  Profits  \
0                    Walmart  500343.0          3.00   9862.0
2                Exxon Mobil  244363.0         17.40  19710.0
3          Berkshire Hathaway  242137.0          8.30  44940.0
4                      Apple  229234.0          6.30  48351.0
5          UnitedHealth Group  201159.0          8.80  10558.0
...                      ...      ...           ...      ...
1012     SiteOne Landscape Supply    1862.0         13.00     54.6
1013  Charles River Laboratories Intl    1858.0         10.50    123.4
1014                 CoreLogic    1851.0         -5.20    152.2
1015              Ensign Group    1849.0         11.80     40.5
1016                      HCP    1848.0        -27.20    414.2

     Profit Change    Assets  Mkt Value as of 3/29/18  Employees  \
0         -27.70  204522.0              263563.0  2300000.0
2         151.40  348691.0              316157.0    71200.0
3          86.70  702095.0              492008.0   377000.0
4           5.80  375319.0              851318.0   123000.0
5          50.50  139058.0              207080.0   260000.0
...          ...      ...                   ...       ...
1012       78.40    911.0                3084.0    3664.0
1013      -20.30   2930.0                5118.0   11800.0
1014       42.80   4077.0                3694.0    5900.0
1015      -19.00   1102.0                1354.0   21301.0
1016      -34.00  14089.0                10910.0     190.0

            Sector                  Industry  \
0        Retailing          General Merchandisers
2           Energy            Petroleum Refining
```

```
3          Financials  Insurance: Property and Casualty (Stock)
4          Technology             Computers, Office Equipment
5         Health Care   Health Care: Insurance and Managed Care
...               ...                                ...
1012       Wholesalers               Wholesalers: Diversified
1013       Health Care  Health Care: Pharmacy and Other Services
1014  Business Services                Financial Data Services
1015       Health Care          Health Care: Medical Facilities
1016        Financials                           Real estate

            City               Cause Attacked
0      Bentonville  Unintended disclosure      Yes
2          Irving         No disclosure       No
3           Omaha         No disclosure       No
4        Cupertino         No disclosure       No
5       Minnetonka     Hacking or malware      Yes
...            ...                ...      ...
1012       Roswell         No disclosure       No
1013     Wilmington         No disclosure       No
1014        Irvine         No disclosure       No
1015  Mission Viejo         No disclosure       No
1016        Irvine         No disclosure       No

[1000 rows x 13 columns]
```

```
df_outer.to_csv(r'C:\\Users\\Ricar\\Desktop\\Revisao\\Datasets\\df_outer.csv')
```

```
import pandas as pd
import numpy as np

df=pd.read_csv(r'C:\\Users\\Ricar\\OneDrive\\Desktop\\Revisao\\Datasets\\df_outer.csv')

df.head()
```

```
  Unnamed: 0            Entity  Revenues Revenue Change  Profits  \
0          0           Walmart  500343.0             3   9862.0
1          2       Exxon Mobil  244363.0          17.4  19710.0
2          3  Berkshire Hathaway  242137.0           8.3  44940.0
3          4             Apple  229234.0           6.3  48351.0
4          5  UnitedHealth Group  201159.0           8.8  10558.0

  Profit Change    Assets  Mkt Value as of 3/29/18  Employees       Sector  \
0        -27.7  204522.0                 263563    2300000    Retailing
1        151.4  348691.0                 316157     71200       Energy
2         86.7  702095.0                 492008     377000   Financials
3          5.8  375319.0                 851318     123000   Technology
4         50.5  139058.0                 207080     260000  Health Care

  Nsector                        Industry        City  \
0     NaN               General Merchandisers  Bentonville
1     5.0                Petroleum Refining       Irving
```

```
2     7.0  Insurance: Property and Casualty (Stock)       Omaha
3    18.0              Computers, Office Equipment    Cupertino
4     NaN   Health Care: Insurance and Managed Care   Minnetonka

                Cause Attacked
0  Unintended disclosure     Yes
1          No disclosure      No
2          No disclosure      No
3          No disclosure      No
4     Hacking or malware     Yes
```

```
df['Profits'].max()
```

```
48351.0
```

```
df['Profits'].mean()
```

```
1279.7730000000015
```

```
df['Profits'].min()
```

```
0.0
```

```
# create a function
def riskFun(row):
    if row <= 5000:
        return 1
    if row > 5000 and row <= 26000:
        return 2
    else:
        return 3
# create a new column based on condition
df['Risk'] = df['Profits'].apply(riskFun)
```

```
df
```

```
    Unnamed: 0                    Entity  Revenues Revenue Change  \
0            0                   Walmart  500343.0            3
1            2                Exxon Mobil  244363.0         17.4
2            3          Berkshire Hathaway  242137.0          8.3
3            4                     Apple  229234.0          6.3
4            5          UnitedHealth Group  201159.0          8.8
..         ...                       ...      ...          ...
995       1012       SiteOne Landscape Supply    1862.0           13
996       1013  Charles River Laboratories Intl    1858.0         10.5
997       1014                   CoreLogic    1851.0         -5.2
998       1015                Ensign Group    1849.0         11.8
999       1016                         HCP    1848.0        -27.2

    Profits Profit Change    Assets  Mkt Value as of 3/29/18  Employees  \
0    9862.0        -27.7  204522.0               263563    2300000
1   19710.0        151.4  348691.0                316157     71200
2   44940.0         86.7  702095.0               492008    377000
```

```
3   48351.0         5.8  375319.0              851318    123000
4   10558.0        50.5  139058.0              207080    260000
..      ...         ...      ...                 ...       ...
995    54.6        78.4    911.0                3084      3664
996   123.4       -20.3   2930.0                5118     11800
997   152.2        42.8   4077.0                3694      5900
998    40.5         -19   1102.0                1354     21301
999   414.2         -34   14089.0               10910       190

            Sector  Nsector                      Industry  \
0         Retailing      NaN            General Merchandisers
1            Energy      5.0                 Petroleum Refining
2        Financials      7.0  Insurance: Property and Casualty (Stock)
3        Technology     18.0          Computers, Office Equipment
4       Health Care      NaN  Health Care: Insurance and Managed Care
..              ...      ...                              ...
995      Wholesalers     21.0            Wholesalers: Diversified
996      Health Care     10.0  Health Care: Pharmacy and Other Services
997  Business Services      3.0              Financial Data Services
998      Health Care     10.0        Health Care: Medical Facilities
999       Financials      7.0                          Real estate

          City              Cause Attacked  Risk
0     Bentonville  Unintended disclosure      Yes     2
1         Irving         No disclosure       No     2
2          Omaha         No disclosure       No     3
3       Cupertino        No disclosure       No     3
4      Minnetonka     Hacking or malware      Yes     2
..          ...                ...     ...   ...
995       Roswell         No disclosure       No     1
996    Wilmington         No disclosure       No     1
997        Irvine         No disclosure       No     1
998  Mission Viejo        No disclosure       No     1
999        Irvine         No disclosure       No     1

[1000 rows x 16 columns]

df.dtypes

Unnamed: 0               int64
Entity                  object
Revenues                float64
Revenue Change           object
Profits                 float64
Profit Change            object
Assets                  float64
Mkt Value as of 3/29/18     int64
Employees                int64
Sector                  object
Nsector                 float64
Industry                 object
City                    object
```

```
Cause                  object
Attacked               object
Risk                   int64
dtype: object
```

```
df_f = df[['Revenues', 'Profits', 'Assets', 'Mkt Value as of 3/29/18', 'Employees', 'Sector', 'Cause', 'Atta
cked', 'Risk']]
```

```
df_f
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  \
0   500343.0   9862.0  204522.0                263563    2300000
1   244363.0  19710.0  348691.0                 316157     71200
2   242137.0  44940.0  702095.0                 492008    377000
3   229234.0  48351.0  375319.0                 851318    123000
4   201159.0  10558.0  139058.0                 207080    260000
..       ...     ...     ...                    ...       ...
995   1862.0    54.6    911.0                  3084      3664
996   1858.0   123.4   2930.0                  5118     11800
997   1851.0   152.2   4077.0                  3694      5900
998   1849.0    40.5   1102.0                 1354     21301
999   1848.0   414.2  14089.0                  10910       190

             Sector              Cause Attacked  Risk
0         Retailing  Unintended disclosure     Yes     2
1            Energy         No disclosure      No     2
2         Financials         No disclosure      No     3
3         Technology         No disclosure      No     3
4        Health Care     Hacking or malware      Yes     2
..               ...                   ...     ...   ...
995       Wholesalers         No disclosure      No     1
996       Health Care         No disclosure      No     1
997  Business Services         No disclosure      No     1
998       Health Care         No disclosure      No     1
999         Financials         No disclosure      No     1

[1000 rows x 9 columns]
```

```
df_f.loc[df_f["Cause"] == "Unintended disclosure", "Cause"] = 1
df_f.loc[df_f["Cause"] == "Hacking or malware", "Cause"] = 2
df_f.loc[df_f["Cause"] == "Inside job", "Cause"] = 3
df_f.loc[df_f["Cause"] == "Other / Unknown", "Cause"] = 4
df_f.loc[df_f["Cause"] == "No disclosure", "Cause"] = 5
```

```
df_f.loc[df_f["Attacked"] == "No", "Attacked"] = 1
df_f.loc[df_f["Attacked"] == "Yes", "Attacked"] = 2
#y=y.astype('int')
```

```
df_f["Attacked"] = df_f["Attacked"].astype(str).astype(int)
df_f["Cause"] = df_f["Cause"].astype(str).astype(int)
```

```
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\811749788.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f["Attacked"] = df_f["Attacked"].astype(str).astype(int)
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\811749788.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f["Cause"] = df_f["Cause"].astype(str).astype(int)

df_f.loc[df_f["Sector"] == "Retailing", "Sector"] = 1
df_f.loc[df_f["Sector"] == "Energy", "Sector"] = 2
df_f.loc[df_f["Sector"] == "Financials", "Sector"] = 3
df_f.loc[df_f["Sector"] == "Technology", "Sector"] = 4
df_f.loc[df_f["Sector"] == "Health Care", "Sector"] = 5
df_f.loc[df_f["Sector"] == "Wholesalers", "Sector"] = 6
df_f.loc[df_f["Sector"] == "Telecommunications", "Sector"] = 7
df_f.loc[df_f["Sector"] == "Motor Vehicles &  Parts", "Sector"] = 8
df_f.loc[df_f["Sector"] == "Food &  Drug Stores", "Sector"] = 9
df_f.loc[df_f["Sector"] == "Industrials", "Sector"] = 10
df_f.loc[df_f["Sector"] == "Aerospace &  Defense", "Sector"] = 11
df_f.loc[df_f["Sector"] == "Household Products", "Sector"] = 12
df_f.loc[df_f["Sector"] == "Transportation", "Sector"] = 13
df_f.loc[df_f["Sector"] == "Food, Beverages &  Tobacco", "Sector"] = 14
df_f.loc[df_f["Sector"] == "Chemicals", "Sector"] = 15
df_f.loc[df_f["Sector"] == "Media", "Sector"] = 16
df_f.loc[df_f["Sector"] == "Apparel", "Sector"] = 17
df_f.loc[df_f["Sector"] == "Materials", "Sector"] = 18
df_f.loc[df_f["Sector"] == "Hotels, Restaurants &  Leisure", "Sector"] = 19
df_f.loc[df_f["Sector"] == "Business Services", "Sector"] = 20
df_f.loc[df_f["Sector"] == "Engineering &  Construction", "Sector"] = 21
df_f["Sector"] = df_f["Sector"].astype(str).astype(int)

C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2009240665.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f["Sector"] = df_f["Sector"].astype(str).astype(int)

df_f["Sector"].nunique()

21

df_f.iloc[:, 0:7]

    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                263563    2300000       1
1   244363.0  19710.0  348691.0                 316157     71200       2
```

```
2   242137.0  44940.0  702095.0            492008    377000     3
3   229234.0  48351.0  375319.0            851318    123000     4
4   201159.0  10558.0  139058.0            207080    260000     5
..       ...     ...      ...               ...       ...    ...
995   1862.0    54.6    911.0              3084      3664      6
996   1858.0   123.4   2930.0              5118     11800      5
997   1851.0   152.2   4077.0              3694      5900     20
998   1849.0    40.5   1102.0              1354     21301      5
999   1848.0   414.2  14089.0              10910      190      3

    Cause
0       1
1       5
2       5
3       5
4       2
..    ...
995     5
996     5
997     5
998     5
999     5

[1000 rows x 7 columns]
```

df_f

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0              263563    2300000      1
1   244363.0  19710.0  348691.0              316157     71200      2
2   242137.0  44940.0  702095.0              492008    377000      3
3   229234.0  48351.0  375319.0              851318    123000      4
4   201159.0  10558.0  139058.0              207080    260000      5
..       ...     ...      ...               ...       ...    ...
995   1862.0    54.6    911.0              3084      3664      6
996   1858.0   123.4   2930.0              5118     11800      5
997   1851.0   152.2   4077.0              3694      5900     20
998   1849.0    40.5   1102.0              1354     21301      5
999   1848.0   414.2  14089.0              10910      190      3

    Cause  Attacked  Risk
0       1         2     2
1       5         1     2
2       5         1     3
3       5         1     3
4       2         2     2
..    ...       ...   ...
995     5         1     1
996     5         1     1
997     5         1     1
998     5         1     1
999     5         1     1
```

[1000 rows x 9 columns]

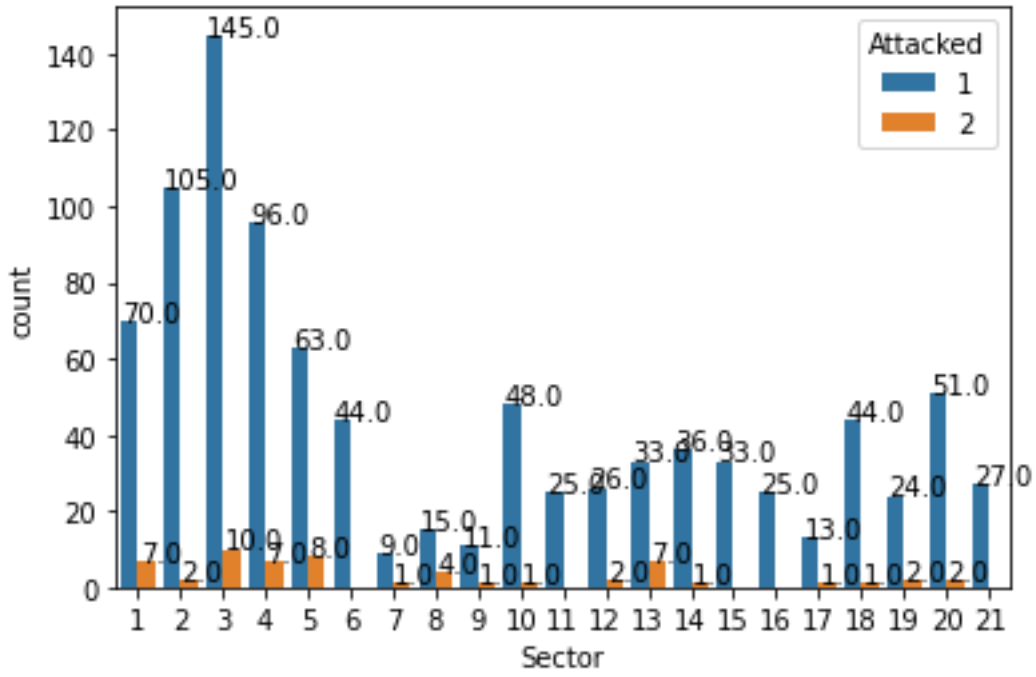
```
df_f['Sector'].value_counts()
```

```
3    155
2    107
4    103
1     77
5     71
20    53
10    49
18    45
6     44
13    40
14    37
15    33
12    28
21    27
19    26
16    25
11    25
8     19
17    14
9     12
7     10
Name: Sector, dtype: int64
```

```
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.countplot(x="Sector", hue="Attacked", data=df_f)
plt.rcParams["figure.figsize"] = [10.00, 7.50]
plt.rcParams["figure.autolayout"] = True

for p in ax.patches:
    ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.025, p.get_height()+0.01))

plt.show()
```



df_f['Employees'].value_counts().sort_index()

```
126       1
152       1
190       1
205       1
392       1
         ..
413000    1
449000    1
450000    1
566000    1
2300000   1
Name: Employees, Length: 754, dtype: int64
```

df_f['Employees'].unique()

```
array([2300000,   71200,  377000,  123000,  260000,   64500,  203000,
       566000,  254000,  180000,  202000,   19500,   51900,   40400,
       182000,  155400,  449000,  313000,  290000,  252539,    7200,
        80110,  413000,  209376,   26600,  262700,  140800,   14600,
        56000,  124000,   10015,  209000,  164000,  397800,  145000,
        65664,  134000,    6165,  345000,  255000,   43800,   95000,
        49000,  346415,  263000,  102700,   98000,   31300,   47950,
       357000,  204700,   49705,  273000,   66500,  199000,   45900,
        90200,  100000,   49800,   33700,   72900,  221491,   29486,
        98400,   33135,   57633,   50000,   11114,   36600,  126600,
       125000,   46000,   94800,   86564,   25105,  131000,   69000,
        42680,  122000,   89800,  138000,   14000,   16829,  249000,
        55000,   61800,  193000,   74400,   14300,    5000,   34621,
        11811,   70430,   11400,   11626,   91536,   26000,   98600,
        32705,   49300,   60476,    1607,    5437,    7000,   30800,
```

66000, 80600, 21700, 29000, 99000, 33656, 18800,
39000, 4850, 10000, 83000, 70000, 64000, 130000,
25204, 72402, 129000, 29060, 31344, 177000, 15700,
40655, 20800, 235000, 277000, 33800, 116200, 3165,
89034, 11318, 92000, 41992, 56110, 66100, 23700,
111980, 165000, 58600, 34100, 25100, 20000, 56706,
8300, 25000, 16400, 85000, 67629, 170000, 86000,
15000, 67000, 42000, 87000, 51632, 51000, 23000,
113600, 8900, 89000, 16000, 24344, 52500, 25200,
48000, 76500, 74500, 135000, 38000, 76000, 35900,
17666, 77300, 8726, 52695, 47200, 29714, 15074,
44800, 14715, 57750, 21900, 215000, 18400, 42300,
17000, 9047, 3522, 80000, 37000, 82700, 15378,
7735, 3414, 10500, 48400, 18100, 10897, 15617,
11000, 11200, 18700, 2700, 7467, 41500, 32944,
50500, 57765, 22900, 9111, 59200, 10200, 16200,
2640, 13400, 13900, 22000, 9283, 33000, 12700,
58000, 12521, 7300, 29200, 8413, 65000, 2470,
36484, 41933, 13312, 13015, 15591, 56690, 4400,
36643, 37543, 5100, 16500, 26461, 24006, 11105,
9400, 72300, 5940, 2664, 16046, 14108, 13504,
17200, 18300, 70035, 6900, 68500, 45000, 47000,
27110, 60000, 25050, 28255, 67352, 15050, 36100,
31000, 46200, 12899, 22615, 35000, 49750, 9750,
7900, 43000, 24525, 23785, 14900, 11528, 6300,
7977, 14100, 11307, 7635, 3578, 3200, 38800,
32800, 55500, 8600, 9700, 32000, 12600, 39500,
20200, 163000, 22736, 53000, 12945, 60365, 17600,
2000, 28050, 26300, 2130, 2290, 5896, 24342,
30300, 4810, 3622, 2650, 15150, 20462, 3500,
11950, 8799, 5425, 23156, 81900, 18000, 50200,
18200, 32175, 8084, 18125, 7722, 9100, 8129,
5227, 4393, 5260, 16135, 12512, 8500, 7140,
3941, 12569, 8700, 31605, 17973, 21000, 17800,
30100, 9300, 24681, 178729, 450000, 2360, 9800,
17424, 126, 15500, 42700, 26500, 41250, 18415,
7600, 64200, 6697, 4752, 18250, 17113, 40000,
14800, 25463, 30000, 17300, 7887, 34000, 67200,
8000, 17594, 4402, 3356, 31400, 14175, 9386,
5200, 13489, 6400, 8615, 26783, 7592, 13000,
11800, 21714, 9200, 20400, 15600, 19611, 90000,
16354, 8100, 1184, 24200, 6200, 12979, 10700,
2226, 4952, 39200, 4500, 23300, 18705, 4925,
24000, 5800, 1230, 7500, 7100, 4150, 11500,
10100, 3836, 140000, 2075, 205, 7800, 68700,
30500, 19350, 23870, 11410, 5400, 4600, 6700,
8200, 2300, 15300, 9900, 9381, 8797, 18500,
11350, 14318, 13800, 15116, 28000, 5441, 8175,
8800, 2489, 11700, 6881, 10061, 62550, 15770,
7751, 6825, 13700, 27000, 14236, 7570, 12450,
2298, 68890, 13500, 15391, 10140, 36050, 6370,

```
 5228,  20565,  19600,  54500,   5406,   1915,   7273,
18600,    392,  16900,   3300,  10227,   3736,   2277,
94000,  11896,  29300,   8150,  13100,   3102,  21100,
 2950,  12515,  16582,   4377,   3560,  17250,  16100,
 1600,  30935,   8406,   4734,  22200,  21755,  18592,
  515,   8287,   1800,    954,  11450,  24100,   4700,
 7850,   1819,  10412,  17700,  15605,    593,   8400,
 1273,   1870,   3580,   2830,  16330,   9000,   1427,
 4300,  23200,   4520,    493,   6292,   3600,   2525,
14700,   4167,   3834,  14309,   7700,  12500,  22500,
 5990,   3140,  15488,  12200,  12656,  20376,   4621,
20900,  11900,   3989,   2067,   2589,  10300,  23600,
19183,   9238,  62150,   4139,  14669,  15131,  14631,
 2900,   7999,   6500,   1605,  16300,   6800,   1575,
26047,   1079,  57906,  18754,  19000,   4900,   4444,
 1127,  19468,   4565,   6233,  44900,  12650,  15006,
 7903,   4570,  19915,  14365,   4100,  73000,   3517,
 4025,   2505,   2244,   6563,   3338,   1047,  34050,
 6600,   6672,    981,   9204,   7825,   3089,    544,
 1251,    152,   9961,   2039,   1630,   5737,  12000,
115000,    520,   4000,  12300,  14200,  10600,  10690,
20500,  10083,  11686,   5550,   2709,   6660,   5600,
 5500,   9523,   4463,   6100,   3800,   1565,    773,
10340,    740,   5554,  16153,   3693,   1203,   7130,
 1491,   2205,  15377,   3880,   5283,   7771,   2699,
 1802,   6779,   7650,  10675,   2683,   2260,  13200,
 6188,   4200,   1429,   3607,   1905,   3372,    905,
18900,   4750,   8737,  25300,  19707,   7102,   8950,
 1586,   3700,   3831,   8493,   2938,   1700,   6099,
 3790,   4464,   5965,   6000,   7040,    926,   7167,
12350,  48700,   7478,   1065,   1052,   4125,  18512,
 2413,  39100,  15800,   7890,   3190,    889,   5900,
 1128,  29524,   5775,   2338,   3794,   2223,   4118,
 2388,   9350,    450,  11917,   3123,    853,   2870,
10615,   9852,  14400,   7192,   7784,   9576,  12154,
  930,   6829,   4366,   4645,   1220,  10800,  11250,
 7532,   6080,   7324,   3782,  10744,   9471,   4050,
 1947,   2100,   2438,   7150,   2906,   1450,  13255,
 2980,   2790,   6259,   5886,   1238,   1931,   5960,
 6620,   2453,   7214,  11600,   6222,   3891,   1228,
 2096,    910,   1100,   4923,  20150,   1300,   8491,
 5157,  13375,   3664,  21301,    190], dtype=int64)
```

```
df_f_sec1 = df_f.loc[df_f['Sector'] == 1]
df_f_sec2 = df_f.loc[df_f['Sector'] == 2]
df_f_sec3 = df_f.loc[df_f['Sector'] == 3]
df_f_sec4 = df_f.loc[df_f['Sector'] == 4]
df_f_sec5 = df_f.loc[df_f['Sector'] == 5]
df_f_sec6 = df_f.loc[df_f['Sector'] == 6]
df_f_sec7 = df_f.loc[df_f['Sector'] == 7]
df_f_sec8 = df_f.loc[df_f['Sector'] == 8]
```

```
df_f_sec9 = df_f.loc[df_f['Sector'] == 9]
df_f_sec10 = df_f.loc[df_f['Sector'] == 10]
df_f_sec11 = df_f.loc[df_f['Sector'] == 11]
df_f_sec12 = df_f.loc[df_f['Sector'] == 12]
df_f_sec13 = df_f.loc[df_f['Sector'] == 13]
df_f_sec14 = df_f.loc[df_f['Sector'] == 14]
df_f_sec15 = df_f.loc[df_f['Sector'] == 15]
df_f_sec16 = df_f.loc[df_f['Sector'] == 16]
df_f_sec17 = df_f.loc[df_f['Sector'] == 17]
df_f_sec18 = df_f.loc[df_f['Sector'] == 18]
df_f_sec19 = df_f.loc[df_f['Sector'] == 19]
df_f_sec20 = df_f.loc[df_f['Sector'] == 20]
df_f_sec21 = df_f.loc[df_f['Sector'] == 21]

df_f['Risk'].value_counts()
```

```
1    950
2     46
3      4
Name: Risk, dtype: int64
```

```
r = 0
listaS1 = []
for risk in df_f_sec1['Risk']:
    r = risk * 0.0909 * 10000
    listaS1.append(r)
df_f_sec1['FinalRisk'] = listaS1
####################################
r = 0
listaS2 = []
for risk in df_f_sec2['Risk']:
    r = risk * 0.0187 * 10000
    listaS2.append(r)
df_f_sec2['FinalRisk'] = listaS2
####################################
r = 0
listaS3 = []
for risk in df_f_sec3['Risk']:
    r = risk * 0.0645 * 10000
    listaS3.append(r)
df_f_sec3['FinalRisk'] = listaS3
####################################
r = 0
listaS4 = []
for risk in df_f_sec4['Risk']:
    r = risk * 0.0680 * 10000
    listaS4.append(r)
df_f_sec4['FinalRisk'] = listaS4
####################################
r = 0
listaS5 = []
for risk in df_f_sec5['Risk']:
```

```
    r = risk * 0.1127 * 10000
    listaS5.append(r)
df_f_sec5['FinalRisk'] = listaS5
####################################
# Sec6 # 0
listaS6 = []
for risk in df_f_sec6['Risk']:
    r = risk * 0 * 10000
    listaS6.append(r)
df_f_sec6['FinalRisk'] = listaS6
####################################
# Sec7
r = 0
listaS7 = []
for risk in df_f_sec7['Risk']:
    r = risk * 0.1 * 10000
    listaS7.append(r)
df_f_sec7['FinalRisk'] = listaS7
####################################
# Sec8
r = 0
listaS8 = []
for risk in df_f_sec8['Risk']:
    r = risk * 0.2105 * 10000
    listaS8.append(r)
df_f_sec8['FinalRisk'] = listaS8
####################################
# Sec9
r = 0
listaS9 = []
for risk in df_f_sec9['Risk']:
    r = risk * 0.0833 * 10000
    listaS9.append(r)
df_f_sec9['FinalRisk'] = listaS9
####################################
# Sec10
r = 0
listaS10 = []
for risk in df_f_sec10['Risk']:
    r = risk * 0.0204 * 10000
    listaS10.append(r)
df_f_sec10['FinalRisk'] = listaS10
####################################
# Sec11
listaS11 = []
for risk in df_f_sec11['Risk']:
    r = risk * 0 * 10000
    listaS11.append(r)
df_f_sec11['FinalRisk'] = listaS11
```

```
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec1['FinalRisk'] = listaS1
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec2['FinalRisk'] = listaS2
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec3['FinalRisk'] = listaS3
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:27: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec4['FinalRisk'] = listaS4
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:34: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec5['FinalRisk'] = listaS5
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:41: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec6['FinalRisk'] = listaS6
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:49: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec7['FinalRisk'] = listaS7
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:57: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec8['FinalRisk'] = listaS8
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:65: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec9['FinalRisk'] = listaS9
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:73: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec10['FinalRisk'] = listaS10
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\4007250398.py:80: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec11['FinalRisk'] = listaS11
```

```
####################################
# Sec12
r = 0
listaS12 = []
for risk in df_f_sec12['Risk']:
    r = risk * 0.0714 * 10000
    listaS12.append(r)
df_f_sec12['FinalRisk'] = listaS12
####################################
# Sec13
r = 0
listaS13 = []
for risk in df_f_sec13['Risk']:
    r = risk * 0.175 * 10000
    listaS13.append(r)
df_f_sec13['FinalRisk'] = listaS13
####################################
# Sec14
r = 0
listaS14 = []
for risk in df_f_sec14['Risk']:
    r = risk * 0.027 * 10000
    listaS14.append(r)
df_f_sec14['FinalRisk'] = listaS14
```

```
###################################
# Sec15 # 0
listaS15 = []
for risk in df_f_sec15['Risk']:
    r = risk * 0 * 10000
    listaS15.append(r)
df_f_sec15['FinalRisk'] = listaS15
###################################
# Sec16 # 0
listaS16 = []
for risk in df_f_sec16['Risk']:
    r = risk * 0 * 10000
    listaS16.append(r)
df_f_sec16['FinalRisk'] = listaS16
###################################
# Sec17
r = 0
listaS17 = []
for risk in df_f_sec17['Risk']:
    r = risk * 0.0714 * 10000
    listaS17.append(r)
df_f_sec17['FinalRisk'] = listaS17
###################################
# Sec18
r = 0
listaS18 = []
for risk in df_f_sec18['Risk']:
    r = risk * 0.0222 * 10000
    listaS18.append(r)
df_f_sec18['FinalRisk'] = listaS18
###################################
# Sec19
r = 0
listaS19 = []
for risk in df_f_sec19['Risk']:
    r = risk * 0.0769 * 10000
    listaS19.append(r)
df_f_sec19['FinalRisk'] = listaS19
###################################
# Sec20
r = 0
listaS20 = []
for risk in df_f_sec20['Risk']:
    r = risk * 0.0377 * 10000
    listaS20.append(r)
df_f_sec20['FinalRisk'] = listaS20
###################################
# Sec21 # 0
listaS21 = []
for risk in df_f_sec21['Risk']:
    r = risk * 0 * 10000
```

```
    listaS21.append(r)
df_f_sec21['FinalRisk'] = listaS21
```

```
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec12['FinalRisk'] = listaS12
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec13['FinalRisk'] = listaS13
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec14['FinalRisk'] = listaS14
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:31: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec15['FinalRisk'] = listaS15
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:38: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec16['FinalRisk'] = listaS16
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:46: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec17['FinalRisk'] = listaS17
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:54: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
```

```
  df_f_sec18['FinalRisk'] = listaS18
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:62: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec19['FinalRisk'] = listaS19
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:70: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec20['FinalRisk'] = listaS20
C:\Users\Ricar\AppData\Local\Temp\ipykernel_7536\2956356683.py:77: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy
  df_f_sec21['FinalRisk'] = listaS21

df_f_sec3

     Revenues  Profits     Assets  Mkt Value as of 3/29/18  Employees  Sector  \
2    242137.0  44940.0   702095.0                   492008     377000       3
19   113899.0  24441.0  2533600.0                   375043     252539       3
20   112394.0   2463.0  3345529.0                     1633       7200       3
23   100264.0  18232.0  2281234.0                   306618     209376       3
25    97741.0  22183.0  1951757.0                   255556     262700       3
..        ...     ...        ...                      ...        ...     ...
964    1956.0    48.7     1406.0                     1042       5960       3
977    1927.0    76.0    61162.0                     4806       1228       3
985    1907.0    25.9    41843.0                        0       1300       3
987    1881.0   399.6     5748.0                     7025       8491       3
999    1848.0   414.2    14089.0                    10910        190       3

     Cause  Attacked  Risk  FinalRisk
2        5         1     3     1935.0
19       4         2     2     1290.0
20       5         1     1      645.0
23       5         1     2     1290.0
25       5         1     2     1290.0
..     ...       ...   ...        ...
964      5         1     1      645.0
977      5         1     1      645.0
985      5         1     1      645.0
987      5         1     1      645.0
999      5         1     1      645.0

[155 rows x 10 columns]
```

```
result = pd.concat([df_f_sec1, df_f_sec2, df_f_sec3, df_f_sec4, df_f_sec5,
                    df_f_sec6, df_f_sec7, df_f_sec8, df_f_sec9, df_f_sec10,
                    df_f_sec11, df_f_sec12, df_f_sec13, df_f_sec14,
                    df_f_sec15, df_f_sec16, df_f_sec17, df_f_sec18,
                    df_f_sec19, df_f_sec20, df_f_sec21], ignore_index=True)

display(result.dtypes)
```

```
Revenues                float64
Profits                 float64
Assets                  float64
Mkt Value as of 3/29/18     int64
Employees                 int64
Sector                    int32
Cause                     int32
Attacked                  int32
Risk                      int64
FinalRisk               float64
dtype: object
```

```
#result['FinalRisk'] = result['FinalRisk'].apply(np.int64)

result['FinalRisk'] = result['FinalRisk'] / 10000

result
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                  263563    2300000       1
1   177866.0   3033.0  131310.0                  700668     566000       1
2   129025.0   2679.0   36347.0                   82687     182000       1
3   100904.0   8630.0   44529.0                  206272     413000       1
4    71879.0   2934.0   38999.0                   37409     345000       1
..       ...     ...      ...                     ...        ...     ...
995   2452.0    332.2    1901.0                     271       1905      21
996   2380.0     72.4    1256.0                    1287       7102      21
997   1962.0     72.1    1864.8                     888       1238      21
998   1916.0     31.8    2221.0                     536       1100      21
999   1906.0    158.1    1750.0                    2725       8400      21

    Cause  Attacked  Risk  FinalRisk
0       1         2     2     0.1818
1       5         1     1     0.0909
2       5         1     1     0.0909
3       5         1     2     0.1818
4       1         2     1     0.0909
..    ...       ...   ...        ...
995     5         1     1     0.0000
996     5         1     1     0.0000
997     5         1     1     0.0000
998     5         1     1     0.0000
999     5         1     1     0.0000

[1000 rows x 10 columns]
```

 # shuffle rows
#result = result.sample(frac=1).reset_index(drop=True)

result

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                 263563    2300000       1
1   177866.0   3033.0  131310.0                 700668     566000       1
2   129025.0   2679.0   36347.0                  82687     182000       1
3   100904.0   8630.0   44529.0                 206272     413000       1
4    71879.0   2934.0   38999.0                  37409     345000       1
..       ...     ...      ...                    ...        ...     ...
995   2452.0    332.2    1901.0                    271       1905      21
996   2380.0     72.4    1256.0                   1287       7102      21
997   1962.0     72.1    1864.8                    888       1238      21
998   1916.0     31.8    2221.0                    536       1100      21
999   1906.0    158.1    1750.0                   2725       8400      21

    Cause  Attacked  Risk  FinalRisk
0       1         2     2     0.1818
1       5         1     1     0.0909
2       5         1     1     0.0909
3       5         1     2     0.1818
4       1         2     1     0.0909
..    ...       ...   ...        ...
995     5         1     1     0.0000
996     5         1     1     0.0000
997     5         1     1     0.0000
998     5         1     1     0.0000
999     5         1     1     0.0000

[1000 rows x 10 columns]
```

```
result["FinalRisk"].unique()
```

```
array([0.1818, 0.0909, 0.0374, 0.0187, 0.1935, 0.129 , 0.0645, 0.204 ,
       0.136 , 0.068 , 0.2254, 0.1127, 0.    , 0.3   , 0.2   , 0.1   ,
       0.2105, 0.421 , 0.0833, 0.0408, 0.0204, 0.1428, 0.0714, 0.175 ,
       0.35  , 0.027 , 0.054 , 0.0222, 0.0769, 0.1538, 0.0377, 0.0754])
```

```
result["FinalRisk"].value_counts().sort_index()
```

```
0.0000    154
0.0187    102
0.0204     48
0.0222     45
0.0270     34
0.0374      5
0.0377     52
0.0408      1
0.0540      3
0.0645    144
0.0680     94
```

```
0.0714    41
0.0754     1
0.0769    25
0.0833    12
0.0909    75
0.1000     6
0.1127    67
0.1290    10
0.1360     8
0.1428     1
0.1538     1
0.1750    37
0.1818     2
0.1935     1
0.2000     2
0.2040     1
0.2105    18
0.2254     4
0.3000     2
0.3500     3
0.4210     1
Name: FinalRisk, dtype: int64
```

```
FinalResult = result.drop(['Risk'], axis=1)
#FinalResult1 = FinalResult.drop(['Sector'], axis=1)
```

```
FinalResult
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                   263563    2300000       1
1   177866.0   3033.0  131310.0                   700668     566000       1
2   129025.0   2679.0   36347.0                    82687     182000       1
3   100904.0   8630.0   44529.0                   206272     413000       1
4    71879.0   2934.0   38999.0                    37409     345000       1
..       ...     ...      ...                      ...        ...     ...
995   2452.0    332.2    1901.0                      271       1905      21
996   2380.0     72.4    1256.0                     1287       7102      21
997   1962.0     72.1    1864.8                      888       1238      21
998   1916.0     31.8    2221.0                      536       1100      21
999   1906.0    158.1    1750.0                     2725       8400      21

    Cause  Attacked  FinalRisk
0       1         2     0.1818
1       5         1     0.0909
2       5         1     0.0909
3       5         1     0.1818
4       1         2     0.0909
..    ...       ...        ...
995     5         1     0.0000
996     5         1     0.0000
997     5         1     0.0000
998     5         1     0.0000
```

```
999     5       1    0.0000

[1000 rows x 9 columns]
```

```
corr = FinalResult["Revenues"].corr(FinalResult["FinalRisk"])
corr
```

```
0.26149852575096166
```

```
corr = FinalResult["Profits"].corr(FinalResult["FinalRisk"])
corr
```

```
0.3437223616130861
```

```
corr = FinalResult["Assets"].corr(FinalResult["FinalRisk"])
corr
```

```
0.14671444338341502
```

```
corr = FinalResult["Mkt Value as of 3/29/18"].corr(FinalResult["FinalRisk"])
corr
```

```
0.24406817293108238
```

```
corr = FinalResult["Employees"].corr(FinalResult["FinalRisk"])
corr
```

```
0.23465103579952581
```

```
corr = FinalResult["Attacked"].corr(FinalResult["FinalRisk"])
corr
```

```
0.19839131811626623
```

```
FinalResult.iloc[:, 0:8]
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                   263563    2300000       1
1   177866.0   3033.0  131310.0                   700668     566000       1
2   129025.0   2679.0   36347.0                    82687     182000       1
3   100904.0   8630.0   44529.0                   206272     413000       1
4    71879.0   2934.0   38999.0                    37409     345000       1
..       ...     ...      ...                      ...        ...     ...
995   2452.0    332.2    1901.0                      271       1905      21
996   2380.0     72.4    1256.0                     1287       7102      21
997   1962.0     72.1    1864.8                      888       1238      21
998   1916.0     31.8    2221.0                      536       1100      21
999   1906.0    158.1    1750.0                     2725       8400      21

    Cause  Attacked
0       1         2
1       5         1
2       5         1
3       5         1
4       1         2
```

```
..    ...     ...
995    5       1
996    5       1
997    5       1
998    5       1
999    5       1

[1000 rows x 8 columns]
```

```
FinalResult["FinalRisk"].nunique()
```

```
32
```

```
FinalResult["FinalRisk"].value_counts().sort_index()
```

```
0.0000   154
0.0187   102
0.0204    48
0.0222    45
0.0270    34
0.0374     5
0.0377    52
0.0408     1
0.0540     3
0.0645   144
0.0680    94
0.0714    41
0.0754     1
0.0769    25
0.0833    12
0.0909    75
0.1000     6
0.1127    67
0.1290    10
0.1360     8
0.1428     1
0.1538     1
0.1750    37
0.1818     2
0.1935     1
0.2000     2
0.2040     1
0.2105    18
0.2254     4
0.3000     2
0.3500     3
0.4210     1
Name: FinalRisk, dtype: int64
```

```
FinalResult["FinalRisk"].mean()
```

```
0.06079940000000007
```

```
r = ''
listaFR = []
for risk in FinalResult['FinalRisk']:
   if risk < 0.06079940000000007:
      r = 'Low'
      listaFR.append(r)
#    elif risk > 0.0540 and risk <= 0.0680:
 #      r = 'Medium'
  #     listaFR.append(r)
   else:
      r = 'High'
      listaFR.append(r)
FinalResult['FinalRiskLevel'] = listaFR

FinalResult["FinalRiskLevel"].value_counts()
```

```
High    556
Low     444
Name: FinalRiskLevel, dtype: int64
```

```
FinalResultWithoutFinalRisk = FinalResult.drop(['FinalRisk'], axis=1)

FinalResultWithoutFinalRisk
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                 263563    2300000       1
1   177866.0   3033.0  131310.0                 700668     566000       1
2   129025.0   2679.0   36347.0                  82687     182000       1
3   100904.0   8630.0   44529.0                 206272     413000       1
4    71879.0   2934.0   38999.0                  37409     345000       1
..       ...     ...      ...                    ...        ...     ...
995   2452.0    332.2    1901.0                    271       1905      21
996   2380.0     72.4    1256.0                   1287       7102      21
997   1962.0     72.1    1864.8                    888       1238      21
998   1916.0     31.8    2221.0                    536       1100      21
999   1906.0    158.1    1750.0                   2725       8400      21

    Cause  Attacked FinalRiskLevel
0       1         2           High
1       5         1           High
2       5         1           High
3       5         1           High
4       1         2           High
..    ...       ...            ...
995     5         1            Low
996     5         1            Low
997     5         1            Low
998     5         1            Low
999     5         1            Low

[1000 rows x 9 columns]
```

```
FinalResultWithoutFinalRisk.to_csv(r'C:\Users\Ricar\Desktop\Tese123\orangeTestLEVEL.csv'
, index=False)
```

```
FinalResultWithoutFinalRisk.iloc[:, 0:8]
```

```
    Revenues  Profits    Assets  Mkt Value as of 3/29/18  Employees  Sector  \
0   500343.0   9862.0  204522.0                 263563    2300000       1
1   177866.0   3033.0  131310.0                 700668     566000       1
2   129025.0   2679.0   36347.0                  82687     182000       1
3   100904.0   8630.0   44529.0                 206272     413000       1
4    71879.0   2934.0   38999.0                  37409     345000       1
..       ...     ...      ...                    ...        ...     ...
995   2452.0    332.2    1901.0                    271       1905      21
996   2380.0     72.4    1256.0                   1287       7102      21
997   1962.0     72.1    1864.8                    888       1238      21
998   1916.0     31.8    2221.0                    536       1100      21
999   1906.0    158.1    1750.0                   2725       8400      21

    Cause  Attacked
0       1         2
1       5         1
2       5         1
3       5         1
4       1         2
..    ...       ...
995     5         1
996     5         1
997     5         1
998     5         1
999     5         1

[1000 rows x 8 columns]
```

```
# variaveis independentes
#x = np.asarray(df_f)
x = np.asarray(FinalResultWithoutFinalRisk.iloc[:, 0:8])

# variavel dependente
y = np.asarray(FinalResultWithoutFinalRisk["FinalRiskLevel"])

y[0:9]
```

```
array(['High', 'High', 'High', 'High', 'High', 'High', 'High', 'High',
       'High'], dtype=object)
```

```
FinalResultWithoutFinalRisk["Attacked"].unique()
```

```
array([2, 1])
```

```
FinalResultWithoutFinalRisk.dtypes
```

```
Revenues              float64
Profits               float64
```

```
Assets                  float64
Mkt Value as of 3/29/18     int64
Employees                int64
Sector                  int32
Cause                   int32
Attacked                int32
FinalRiskLevel           object
dtype: object
```

FinalResultWithoutFinalRisk

	Revenues	Profits	Assets	Mkt Value as of 3/29/18	Employees	Sector	Cause	Attacked	FinalRiskLevel
0	500343.0	9862.0	204522.0	263563	2300000	1	1	2	High
1	177866.0	3033.0	131310.0	700668	566000	1	5	1	High
2	129025.0	2679.0	36347.0	82687	182000	1	5	1	High
3	100904.0	8630.0	44529.0	206272	413000	1	5	1	High
4	71879.0	2934.0	38999.0	37409	345000	1	1	2	High
..	...	...	...	...	...	...	...	...	...
995	2452.0	332.2	1901.0	271	1905	21	5	1	Low
996	2380.0	72.4	1256.0	1287	7102	21	5	1	Low
997	1962.0	72.1	1864.8	888	1238	21	5	1	Low
998	1916.0	31.8	2221.0	536	1100	21	5	1	Low
999	1906.0	158.1	1750.0	2725	8400	21	5	1	Low

[1000 rows x 9 columns]

from imblearn import over_sampling

from imblearn.over_sampling import RandomOverSampler ros = RandomOverSampler(random_state = 0) X_resampled, Y_resampled = ros.fit_resample(x,y) print(Y_resampled)
from sklearn.model_selection **import** train_test_split

 # Without over samping
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=4)

 # With over sampling
#X_train, X_test, y_train, y_test = train_test_split(X_resampled, Y_resampled, test_size=0.2, random_state=4)

```
X_train.shape
```

```
(800, 8)
```

```
#Modelling WITH SVM
from sklearn import svm
from sklearn import preprocessing

classifier = svm.SVC(kernel="linear", gamma="auto", C=2)

#lab_enc = preprocessing.LabelEncoder()
#_training_scores_encoded = lab_enc.fit_transform(y_train)
#print(y_training_scores_encoded)

classifier.fit(X_train, y_train)

y_predict = classifier.predict(X_test)

#Modelling WITH LOGISTIC REGRESSION
from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(solver='lbfgs', max_iter=1000)

classifier.fit(X_train, y_train)

y_predict = classifier.predict(X_test)

print(X_test.shape)
print(y_test.shape)
```

```
(200, 8)
(200,)
```

```
### Check Accuracy
from sklearn.metrics import accuracy_score
score=accuracy_score(y_test,y_predict)

score
```

```
0.73
```

```
#Avaliacao dos resultados
from sklearn.metrics import classification_report

print(classification_report(y_test, y_predict))
```

```
              precision    recall  f1-score   support

        High       0.73      0.81      0.77       109
         Low       0.73      0.64      0.68        91

    accuracy                           0.73       200
   macro avg       0.73      0.72      0.72       200
```

```
weighted avg       0.73      0.73      0.73       200
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_predict, labels=["High", "Low"])
```

```
array([[88, 21],
       [33, 58]], dtype=int64)
```

```
import pickle
with open('model_TESTFINAL', 'wb') as f:
   pickle.dump(classifier, f)
```

```
with open('model_TESTFINAL', 'rb') as f:
   mp = pickle.load(f)
```

```
## Prediction
mp.predict(X_test)
```

```
array(['Low', 'High', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low',
       'High', 'Low', 'High', 'Low', 'High', 'High', 'Low', 'High', 'Low',
       'High', 'Low', 'Low', 'Low', 'High', 'High', 'High', 'High', 'Low',
       'High', 'Low', 'Low', 'High', 'High', 'Low', 'High', 'High',
       'High', 'High', 'Low', 'High', 'High', 'Low', 'High', 'High',
       'Low', 'Low', 'Low', 'Low', 'High', 'High', 'High', 'High', 'Low',
       'Low', 'High', 'High', 'High', 'High', 'Low', 'Low', 'Low', 'High',
       'High', 'High', 'High', 'High', 'Low', 'High', 'Low', 'High',
       'Low', 'High', 'High', 'High', 'Low', 'High', 'High', 'High',
       'Low', 'Low', 'High', 'High', 'High', 'High', 'High', 'High',
       'High', 'High', 'High', 'High', 'High', 'High', 'Low', 'High',
       'Low', 'High', 'High', 'High', 'Low', 'High', 'High', 'High',
       'High', 'Low', 'High', 'Low', 'High', 'High', 'High', 'High',
       'Low', 'Low', 'High', 'Low', 'High', 'Low', 'High', 'High', 'Low',
       'High', 'High', 'High', 'Low', 'High', 'High', 'Low', 'High',
       'High', 'Low', 'High', 'Low', 'High', 'High', 'High', 'High',
       'Low', 'High', 'Low', 'Low', 'High', 'High', 'High', 'Low', 'High',
       'High', 'Low', 'Low', 'Low', 'High', 'Low', 'High', 'High', 'High',
       'Low', 'Low', 'Low', 'Low', 'High', 'Low', 'High', 'High', 'High',
       'Low', 'High', 'Low', 'Low', 'High', 'High', 'Low', 'High', 'High',
       'Low', 'High', 'Low', 'High', 'High', 'High', 'High', 'High',
       'High', 'High', 'Low', 'High', 'High', 'High', 'Low', 'High',
       'Low', 'High', 'Low', 'Low', 'High', 'Low', 'High', 'High', 'Low',
       'High', 'Low', 'Low', 'Low', 'High'], dtype=object)
```

```
mp.predict([[500343.0,9862.0,204522.0,263563,2300000,1,1,2]])
```

```
array(['High'], dtype=object)
```

FinalResultWithoutFinalRisk

	Revenues	Profits	Assets	Mkt Value as of 3/29/18	Employees	Sector	\
0	500343.0	9862.0	204522.0	263563	2300000	1	
1	177866.0	3033.0	131310.0	700668	566000	1	
2	129025.0	2679.0	36347.0	82687	182000	1	

```
3   100904.0   8630.0   44529.0             206272    413000      1
4    71879.0   2934.0   38999.0             37409    345000      1
..       ...     ...      ...                 ...       ...    ...
995    2452.0    332.2    1901.0              271      1905     21
996    2380.0     72.4    1256.0             1287      7102     21
997    1962.0     72.1    1864.8              888      1238     21
998    1916.0     31.8    2221.0              536      1100     21
999    1906.0    158.1    1750.0             2725      8400     21

    Cause  Attacked FinalRiskLevel
0       1       2          High
1       5       1          High
2       5       1          High
3       5       1          High
4       1       2          High
..    ...     ...           ...
995     5       1           Low
996     5       1           Low
997     5       1           Low
998     5       1           Low
999     5       1           Low

[1000 rows x 9 columns]
```