Mineração de dados em dataset de movimentação de empresas situadas na Rússia pós guerra

ROBSON NOVATO LOBAO, Universidade Federal de Ouro Preto, Brasil
PABLO MARTINS COELHO, Universidade Federal de Ouro Preto, Brasil
RAFAEL AUGUSTO FREITAS OLIVEIRA, Universidade Federal de Ouro Preto, Brasil

Consequências incontáveis podem ser relatadas devido a ocorrência de guerras no mundo todo por várias décadas, entretanto, com a globalização impactando todos os países, inclusive os mais fechados a interferências externas como a Rússia, as consequências de guerras são ainda mais brutais. Com a participação de diversas empresas sediadas nas mais variadas localizações globais, após a guerra contra a Ucrânia em 2022 essas empresas tiveram que se posicionar, principalmente quando grande parte do mundo ocidental é terminantemente contrária a guerra. O objetivo desse trabalho é mostrar o posicionamento das empresas estrangeiras situadas na Rússia, traçar um perfil das que debandaram e criar um modelo de classificação para dizer se uma empresa, de acordo com suas características, sairiam da Rússia ou não. Os dados dos nossos experimentos serão apresentados e debatidos ao longo desse trabalho, demonstrando o passo a passo desde a limpeza e transformação dos dados, até o processo de treinamento do classificador.

CCS Concepts: • **Information systems** → Data mining.

Additional Key Words and Phrases: datasets, aprendizado de máquina, pré-processamento de dados, algoritmos de classificação

ACM Reference Format:
Robson Novato Lobao, Pablo Martins Coelho, and Rafael Augusto Freitas Oliveira. 2023. Mineração de dados em dataset de movimentação de empresas situadas na Rússia pós guerra. 1, 1 (August 2023), 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUÇÃO

Inicialmente, como vivemos em um mundo cada vez mais globalizado e um país que anteriormente era fechado a essa transformação, como a Rússia, assume posição de destaque na geopolítica global [de Souza 2016], suas ações, como o início da guerra com a Ucrânia surtem grande efeito nas multinacionais que se estabeleceram em seu vasto território.

Além de que, como é de interesse internacional o posicionamento dessas empresas, saber suas ações é essencial para a manutenção das relações diplomáticas.

Authors' addresses: Robson Novato Lobao, robson.lobao@aluno.ufop.edu.br, Universidade Federal de Ouro Preto, Rua Professor Paulo Magalhães Gomes, 122 - Bauxita, Ouro Preto, Minas Gerais, Brasil, 35400-000; Pablo Martins Coelho, Universidade Federal de Ouro Preto, Rua Professor Paulo Magalhães Gomes, 122 - Bauxita, Ouro Preto, Minas Gerais, Brasil, 35400-000, pablo.martins@aluno.ufop.edu.br; Rafael Augusto Freitas Oliveira, Universidade Federal de Ouro Preto, Rua Professor Francisco Pignatário - 263 Bauxita, Ouro Preto, Minas Gerais, Brasil, 35402230, rafael.afo@aluno.ufop.edu.br.

XXXX-XXXX/2023/8-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

Tendo isso em vista, utilizando um dataset com diversas empresas, seus setores e ações tomadas pós-guerra, como: abandonar a Rússia, continuar com as operações e etc. Foi possível treinar um modelo para tentar identificar, dadas certas características, se a empresa abandonaria a Rússia em um cenário pós-guerra, ou não.

2 FUNDAMENTAÇÃO TEÓRICA

Primeiramente, cabe situar antes que qualquer pré-processamento, também conhecido como preparação da base de dados, manipula e transforma os dados brutos de maneira que o conhecimento neles contido possa ser mais fácil e corretamente obtido. A melhor maneira de pré-processar os dados depende de três fatores centrais: os problemas (incompletude, inconsistência e ruído) existentes na base bruta; quais respostas pretendem-se obter das bases, ou seja, qual problema deve ser resolvido; e como operam as técnicas de mineração de dados que serão empregadas [de Castro and Ferrari 2016].

Ademais, outros métodos de geração de regras, algoritmos de classificação e itens frequentes foram gerados e serão aprofundados nos próximos tópicos da fundamentação.

2.1 Identificação dos atributos

Inicialmente, em uma mineração de dados, é feito a identificação dos tipos de atributo que o dataset possui, identificar os tipos de atributos em um conjunto de dados é fundamental por várias razões.

Primeiramente, isso possibilita um pré-processamento adequado dos dados. A natureza dos atributos, como categóricos ou numéricos, determina as etapas necessárias, como conversão de categóricos em numéricos ou normalização de valores.

Além disso, a compreensão dos tipos de atributos auxilia na seleção de recursos relevantes. Saber quais variáveis são mais informativas ajuda a melhorar a eficiência da análise ou do modelo de machine learning.

Os tipos de atributos também orientam na escolha dos algoritmos apropriados. Diferentes algoritmos lidam melhor com diferentes tipos de dados. Por exemplo, algoritmos de árvore de decisão podem trabalhar diretamente com atributos categóricos, enquanto algoritmos de aprendizado por distância requerem tratamento especial.

Evitar erros e viés é outra razão importante. Misturar tipos de atributos incorretamente pode levar a resultados distorcidos. A interpretação dos resultados também se beneficia da compreensão dos tipos de atributos, especialmente ao analisar coeficientes ou importâncias de características.

O tratamento de valores ausentes é facilitado ao se conhecer os tipos de atributos. Cada tipo pode demandar uma abordagem diferente para lidar com valores faltantes.

Por fim, o conhecimento dos tipos de atributos contribui para otimizar o desempenho do modelo, aplicando as técnicas de pré-processamento mais adequadas.

Em resumo, identificar os tipos de atributos é crucial para orientar todas as etapas da análise de dados e modelagem, evitando erros, viés e garantindo resultados mais confiáveis e eficazes.

2.2 Tratamento dos atributos

Na mineração de dados, o tratamento de atributos é uma etapa fundamental do processo de pré-processamento dos dados. Essa etapa envolve a análise, limpeza, transformação e seleção dos atributos ou características que serão usados para construir e treinar modelos de aprendizado de máquina ou realizar análises estatísticas

Os dados coletados podem conter erros, valores ausentes, ruído ou outliers. O tratamento dos atributos ajuda a identificar e corrigir esses problemas, melhorando a qualidade dos dados usados para treinamento de modelos. Além disso, nem todos os atributos coletados são igualmente relevantes para o problema em questão. A seleção cuidadosa de atributos ajuda a focar nos mais importantes, reduzindo a dimensionalidade e o potencial de sobreajuste dos modelos.

Ademais, muitos algoritmos de aprendizado de máquina trabalham melhor com atributos numéricos, portanto, os atributos categóricos e textuais podem precisar ser convertidos ou processados de maneira adequada.

2.2.1 Algoritmo Z-Score. O Z-score (escore Z) é uma medida estatística que ajuda a identificar outliers em um conjunto de dados. Ele avalia o quão longe um ponto de dados está da média, em termos de desvios padrão. Um valor Z alto indica que o ponto de dados está longe da média, possivelmente sugerindo que seja um outlier.

2.3 Geração de regras de associação

A geração de regras de associação é uma técnica importante na análise de dados e mineração de padrões. Ela é usada para descobrir associações frequentes e padrões interessantes em conjuntos de dados, o que pode ter várias aplicações práticas, permitindo tomadas de decisões mais informadas e aprimoramento de processos em uma variedade de domínios.

2.3.1 Utilização do algoritmo Apriori. O algoritmo Apriori é um dos algoritmos mais conhecidos para mine- ração por regras de associação. O algoritmo emprega busca em profundidade e gera conjuntos de itens candidatos (padrões) de k elementos a partir de conjuntos de itens de k - 1 elementos. Os padrões não frequentes são eliminados. Toda a base de dados é rastreada e os conjuntos de itens frequentes são obtidos a partir dos conjuntos de itens candidatos.

O algoritmo trabalha sobre uma base de transações em busca de itens frequentes, ou seja, aqueles que possuem suporte maior ou igual ao suporte mínimo. Desta forma, como entrada, é necessário fornecer um valor correspondente ao suporte mínimo e outro correspondente à confiança mínima, além de um arquivo de itens e transações.

Algorithm 1 Algoritmo Apriori

$F1 \leftarrow$ {Conjuntos de itens frequentes de tamanho 1} ▷ Na primeira passagem, $k = 1$
for $k = 2$; $F_{k-1} \neq$ vazio; $k++$ **do**
  $C_k \leftarrow$ apriori-gen(F_{k-1}) ▷ Novos candidatos
  for cada transação $t \in T$ **do**
    $C_t \leftarrow$ subconjunto(C_k, t) ▷ Candidatos contidos em t
    for cada candidato $c \in C_t$ **do**
      c.contagem++
    end for
  end for
  $F_k \leftarrow \{c \in C_k \mid c.\text{contagem} \geq \text{MinSup}\}$
end for
Resposta $F \leftarrow$ Reunião de todos os F_k

2.4 Aprendizado de máquina

No campo de aprendizado de máquina, algoritmos de aprendizado indutivo podem ser divididos em Preditivos e Descritivos. Os algoritmos preditivos são aqueles onde um conjunto de dados pré-rotulados são fornecidos ao algoritmo, com o intuito de induzir um modelo preditivo capaz de predizer o valor de um dado novo não rotulado [Faceli et al. 2021]. Esses algoritmos de Aprendizado de Máquina (AM), seguem um paradigma de aprendizado supervisionado, termo este que remete a um supervisor externo, que sabe o verdadeiro rótulo de uma dado, e a partir disso, guia o aprendizado para a geração de um modelo com boa capacidade preditiva. Os algoritmos preditivos podem ser classificados de acordo com o valor do rótulo a ser predito, como discretos, para tarefas de classificação, e contínuos, para tarefas de regressão.

Em uma definição formal, dado um conjunto de objetos de pares D = {(x_i, f(x_i)), i = 1, ..., n}, em que F representa uma função desconhecida, um algoritmo preditivo aprende uma aproximação f' da função f, função essa que permite estimar o valor de f para novas objetos x [Faceli et al. 2021]. A natureza de f pode ser distinguida em classificação e regressão. Na classificação o resultado pode assumir valores em um conjunto discreto, já na classificação, esse conjunto é finito e ordenado.

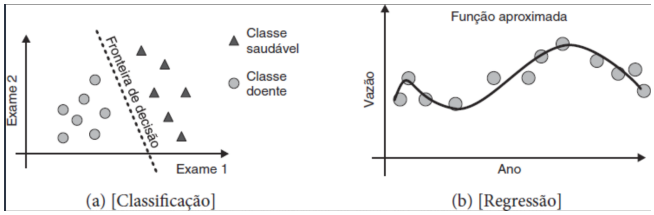

Fig. 1. Gráfico ilustrativo

A figura 1 exemplifica a diferença entre a regressão e a classificação. Na parte (a), de classificação, é ilustrado um cenário com duas classes que possuem dois atributos, que são o resultado de dois exames. Os algoritmos de classificação tem como objetivo encontrar uma fronteira de decisão que separem uma classe da outra. no caso da figura 1(a), os dados possuem dois atributos, podendo

ser plotado em um único plano, possibilitando que um única reta divida as classes. No caso de objetos que tenha mais atributos, será necessários mais planos para a representação e, consequentemente, hiperplanos para a separação [Faceli et al. 2021].

Já a figura 1(b) temos uma representação de um caso de regressão em que a meta é aprender uma função que relacione um ano à vazão de água de um dado rio nesse ano, onde temos apenas um atributo de entrada. Podemos ver que a ligação dos pontos formado pelos valores dia e vazão gera uma curva. O objetivo dos algoritmos regressivos é criar uma função que mais se aproxime da função de curva original, podendo prever os próximos pontos [Faceli et al. 2021].

No ramo da mineração de dados o uso de algoritmos descritivos e regressivos é bastante comum. No nosso trabalho usamos 3 modelos que podem ser utilizados tanto para regressão quanto para classificação: SVM (*Support Vector Machine*), Floresta Aleatória (*Random Forest*), Regressão Logística e KNN(*K-Nearest Neighbors*. Todos esses algoritmos podem ser encontrados na biblioteca Scikit-Learn (sklearn) em Python [Scikit-Learn Development Team 2021].

2.4.1 SVM. O Support Vector Machine (SVM) é um algoritmo de aprendizado de máquina usado para tarefas de classificação e regressão. Foi introduzido por Vapnik e Cortes em 1995 [Gunn et al. 1998] e rapidamente se tornou uma técnica popular devido à sua eficácia e capacidade de lidar com dados de alta dimensão.

O SVM é uma abordagem de aprendizado supervisionado que visa encontrar um hiperplano que melhor separa as classes de dados. Para um problema de classificação binária, o SVM busca um hiperplano que maximize a margem entre as classes, ou seja, a distância entre os vetores de suporte mais próximos de cada classe.

A equação do hiperplano para um SVM linear pode ser representada como:

$$f(x) = w \cdot x + b$$

onde w é o vetor de peso, x é o vetor de características da entrada e b é o viés.

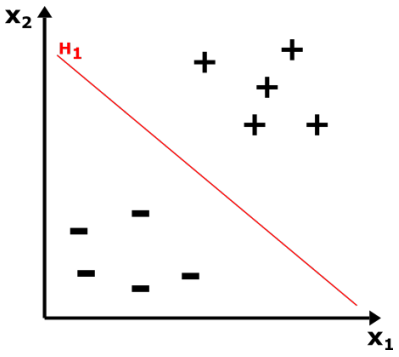

Fig. 2. Plano cortado por um algortimo SVM qualquer

Os vetores de suporte são os pontos de dados mais próximos do hiperplano de separação. A margem é a distância entre o hiperplano e os vetores de suporte. O objetivo do SVM é maximizar essa margem, resultando em um modelo mais robusto e com melhor capacidade de generalização.

Em muitos casos, os dados podem não ser linearmente separáveis no espaço original das características. O SVM pode utilizar o "kernel trick" para mapear os dados para um espaço de dimensão superior, onde a separação pode ser realizada de maneira linear. Alguns exemplos de kernels comuns são o kernel linear, o kernel polinomial e o kernel de função de base radial (RBF).

2.4.2 Random Forest. A Random Forest, ou Floresta Aleatória, é um algoritmo de aprendizado de máquina que pertence à categoria de técnicas de ensemble. Introduzida por Leo Breiman em 2001 [Breiman 2001], a Random Forest é conhecida por sua eficácia em uma variedade de tarefas de classificação e regressão.

Uma Random Forest é composta por um conjunto de árvores de decisão individuais. Cada árvore é treinada em uma amostra aleatória dos dados de treinamento e gera uma previsão. A previsão final é obtida por meio de uma combinação das previsões de todas as árvores, usando técnicas de votação ou média, dependendo do problema.

A característica fundamental da Random Forest é a amostragem aleatória, que introduz diversidade nas árvores individuais. Isso ajuda a reduzir o overfitting e aumentar a capacidade de generalização do modelo para novos dados. Além disso, as variáveis de entrada também são amostradas aleatoriamente para cada divisão na construção das árvores.

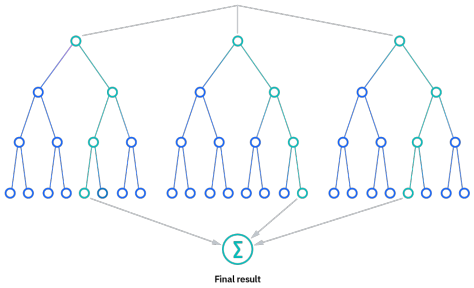

Fig. 3. Representação visual de uma Random Forest

Uma das vantagens da Random Forest é a capacidade de medir a importância relativa das variáveis de entrada. Isso pode ser usado para avaliar a influência de cada variável na previsão final e pode ser útil na seleção de recursos.

A Random Forest é conhecida por sua robustez, tolerância a dados ruidosos e bom desempenho em problemas de alta dimensionalidade. Ela é frequentemente aplicada em áreas como classificação de imagens, análise de dados biológicos, previsão de mercado financeiro e muito mais.

2.4.3 Regressão Logística. A Regressão Logística é um algoritmo de aprendizado de máquina amplamente utilizado para tarefas de classificação. Apesar do nome, a Regressão Logística é usada para prever a probabilidade de pertencimento a uma determinada classe, sendo especialmente eficaz em problemas de classificação binária.

A Regressão Logística estabelece uma relação linear entre as variáveis independentes e a log-odds da probabilidade de pertencer

a uma classe. A função logística (sigmoid) é usada para mapear a saída linear em uma probabilidade, e a equação pode ser expressa como:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(w \cdot X + b)}}$$

onde $P(Y = 1|X)$ é a probabilidade de pertencer à classe 1 dada a entrada X, w são os coeficientes das variáveis independentes, e b é o viés.

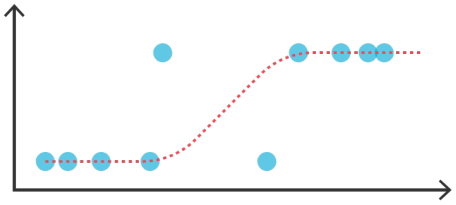

Fig. 4. Representação visual de uma Regressão Logística

O treinamento da Regressão Logística envolve a minimização de uma função de custo, geralmente a Entropia Cruzada, para ajustar os coeficientes w e b. O processo de treinamento é realizado usando algoritmos de otimização, como o Gradiente Descendente, que busca encontrar os valores ideais dos parâmetros.

A Regressão Logística é aplicada em diversas áreas, incluindo medicina, marketing, finanças e muito mais. Exemplos de uso incluem detecção de spam, previsão de risco de crédito e diagnóstico médico.

O capitulo posteriores irão falar sobre as configurações da nossa base dados e como fizemos o pré-processamento dela.

2.4.4 O K-Nearest Neighbors (KNN). O KNN é um algoritmo de aprendizado de máquina que classifica ou prevê novos pontos de dados com base na proximidade de seus vizinhos mais próximos no conjunto de treinamento. O KNN não é expresso usando uma fórmula matemática simples, como no caso da regressão logística. Em vez disso, o KNN é um algoritmo baseado na distância entre pontos de dados.

Dado um conjunto de treinamento com exemplos rotulados, o algoritmo KNN classifica ou prevê novos pontos de dados da seguinte maneira:

(1) **Medição de Distância:** Calcule a distância entre o novo ponto de dados (instância de entrada) e todos os pontos de dados no conjunto de treinamento usando uma métrica de distância, como a distância euclidiana.
(2) **Seleção dos Vizinhos:** Selecione os "K" pontos de dados mais próximos (vizinhos) com base nas distâncias calculadas.
(3) **Decisão de Classificação (ou Previsão):** No caso de classificação, conte quantos dos "K" vizinhos pertencem a cada classe e atribua o novo ponto de dados à classe com mais vizinhos. No caso de regressão, calcule a média (ou mediana) dos valores alvo dos "K" vizinhos e atribua esse valor como a previsão.

3 DESENVOLVIMENTO

A nossa base de dados pode ser encontrada no Kaggle[1]. Ela é composta por dados de empresas que estão sitiadas na Rússia, e como vão seus negócios ao longo da guerra entre Rússia e Ucrânia. A base contem 5 colunas, 'Name' que é o nome da empresa, 'Action' representando a atual situação da empresa na Rússia, 'Industry' que fala o tipo da industria, 'Country' que representa o pais cede/legal daquela industria e 'Grade' que representa uma escala de notas de A a F para a completude da abstinência da companhia. Name, Action, Industry e Country são atributos nominais, já Grade é um atributo ordinal.

Tendo isso em vista, foram identificados da seguinte forma:

(1) **Name**: Nome da companhia, **categórico nominal** pois rotula uma companhia mas sem distinção de hierarquia, apenas nomeia.
(2) **Action**: Ação da empresa, indica o que a empresa anunciou, é um atributo **categórico nominal**, pois rotula o local da empresa, mas não tem uma ordem entre os locais.
(3) **Indutry**: Indústria, indica qual o tipo de indústria é a companhia. É um atributo **categórico nominal**, pois classifica o tipo de indústria e não tem uma ordem específica entre esses tipos.
(4) **Country**: País, representa o endereço legal da empresa. Atributo **categórico nominal**, pois apenas indica o país de origem da empresa.
(5) **Grade**: Nota, representa uma escala de notas de A a F para a completude da abstinência da companhia. Representa um atributo **categórico ordinal**, categórico pois rotula em diferentes classes e ordinal por haver hierarquia entre os dados.

3.1 Bibliotecas necessárias

No projeto, utilizamos várias bibliotecas de Python para análise de dados, pré-processamento e modelagem. Começamos com as básicas, como "numpy" e "pandas", para gerenciar os dados. "Seaborn" e "matplotlib.pyplot" foram usados para criar gráficos claros.

Na preparação dos dados, transformamos e escalamos atributos usando classes como "LabelEncoder", "MinMaxScaler" e "OneHotEncoder". Para processar texto, usamos "TfidfVectorizer". Também aplicamos clustering usando "KMeans".

Para modelagem, importamos algoritmos como "SVC", "LogisticRegression" e "RandomForestClassifier" para classificação. Avaliamos os modelos com a métrica de acurácia.

Na análise de linguagem natural, usamos "nltk" para pré-processar texto, "TextBlob" para análise de sentimentos e "Word2Vec" da biblioteca "gensim" para criar modelos de word embedding.

Exploramos ainda mais com mineração de regras de associação usando "apriori" e continuamos uma abordagem abrangente com várias técnicas e métodos para atingir nossos objetivos.

3.2 Estatísticas de alguns atributos

Foi feito então a análise estatística dos atributos nominais e ordinais, a indicação dos possíveis valores e da distribuição de probabilidade dos atributos nominais e ordinais é uma prática essencial na

[1]https://www.kaggle.com/datasets/vadimtynchenko/list-of-companies-leaving-or-staying-in-russia

análise exploratória de dados. Essa abordagem proporciona insights valiosos sobre a composição dos dados e ajuda a compreender a representatividade de cada categoria ou classe.

Além disso, esse procedimento auxilia na detecção de categorias incomuns, anômalas ou pouco frequentes, que podem indicar erros nos dados ou informações relevantes para a análise. A compreensão da popularidade das categorias é fundamental para tomar decisões informadas sobre como tratar essas classes em diferentes etapas da análise de dados e modelagem.

Ao conhecer a distribuição de probabilidade, é possível identificar possíveis vieses nos dados. Isso é especialmente importante em análises sensíveis, onde uma distribuição desigual pode influenciar os resultados de maneira significativa.

Esta foi a identificação para os tipos industriais:

Categoria	Popularidade
Industrials	0.254902
Consumer Discretionary	0.197976
Information Technology	0.123340
Consumer Staples	0.105629
Financials	0.082226
Materials	0.061354
Communication Services	0.046173
Health Care	0.044276
Energy	0.039216
NGO	0.025300
Utilities	0.010753
Real Estate	0.008855

Table 1. Distribuição de Popularidade por categoria industrial

Esta foi a identificação para os países de origem das empresas:

País de Origem	Popularidade
United States	0.289058
Germany	0.092979
United Kingdom	0.080961
France	0.052498
Japan	0.048071
...	...
Uzbekistan	0.000633
Qatar	0.000633
Egypt	0.000633
Syria	0.000633
Panama	0.000633

Table 2. Distribuição de Popularidade por País de Origem pegando os 5 primeiros e 5 últimos

Alguns insights já puderam ser tirados já fazendo a análise estatística que seriam que através da análise dos possíveis valores dos atributos explorados, é possível observar que a maioria das companhias são Industrias, logo após vem outros setores que têm uma menor porcentagem, sendo em sua menor quantidade as companhias Imobiliárias. Com relação às ações anunciadas pelas empresas, uma grande parte dos anúncios é que as empresas ainda estão operando, e em seguida é que as empresas vão suspender as operações na Rússia. É possível ver também que uma boa parte das empresas operando na Rússia são dos Estados Unidos.

Esta é a tabela de grau de completude de paralisação de atividades na Rússia:

Níveis de Completude	Popularidade
A	0.329538
B	0.318153
F	0.146743
D	0.111322
C	0.094244

Table 3. Distribuição de Popularidade por Níveis de Completude de Abstinência

Sobre os níveis de completude de abstinência, é possível notar que a maioria das empresas têm o nível A e o nível B de abstinência (A e B significa que parou totalmente ou quase que totalmente suas ações na Rússia).

Foram aplicados todos os demais passos de tratamento já fundamentos na parte teórica desse artigo.

3.3 Pré-processamento

Após analisar cada dado partimos para o pré processamento em si. Começamos transformando os valores categórico nominais em valores numérico, transformamos o conjunto {F, E, D, C, B, A} em {0, 1, 2, 3, 4, 5}, respectivamente. Após isso, identificamos os valores aberrantes ou inconsistentes, uma forma de identificar valores aberrantes é verificando a média de evasão do país em geral e ver se a empresa correspondente está mais ou menos semelhante, por exemplo, sendo 5 a total evasão da Rússia e 0 a permanência total das atividades a média está em 0.6, ou seja, países chineses que tem valores acima de 4 ou até mesmo 5 podem ser aberrantes.

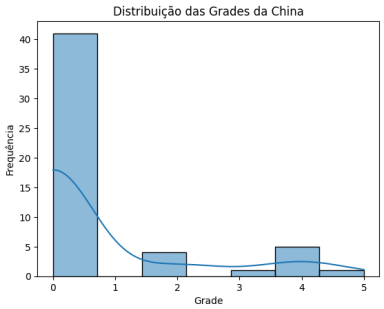

Fig. 5. Relação Grade/Country China

Podemos observar nas figuras 11 e 6 que não contem valores aberrantes, o que faz sentido sabendo que os EUA são contra a Rússia nessa guerra, e a China é uma grande apoiadora da Rússia. Também fizemos uma nova relação de colunas, adicionando o atributo booleano OTAN, que indica se o pais faz parte ou não da OTAN, e podemos perceber essa relação no nível de Grade dos países da

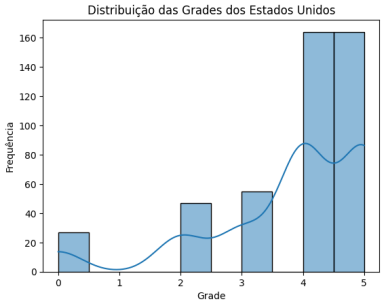

Fig. 6. Relação Grade/Country EUA

participantes da OTAN comparados com os não participante, que confirmaram que os ingressantes da OTAN se encontram com grade próximo a 0, ou seja, quase ou totalmente fora de serviço na Rússia.

Nosso próximo passo foi fazer a identificação de *Outliers*, utilizando *Z-score*, e valores nulos, e por meio de análises feitas, constatamos que não há valores ausentes, valores aberrantes e nem inconsistências, a base é bem concisa.

No dataset temos o atributo Action, que descreve o que a empresa fez na Rússia após o início da guerra, e podemos notar que 1128 valores únicos criam certa falta de parâmetro para geração de regras posteriormente, dessa forma foi realizada uma padronização do atributo Action, inicialmente listamos o número de valor que Action pode assumir. Uma das primeiras formas que tentamos padronizar o Action é utilizando a semelhança de palavras do *Word2Vec* em que ele é um modelo amplamente utilizado para representação de palavras em forma de vetores em um espaço vetorial contínuo. Essa representação é útil para medir a similaridade semântica entre palavras com base em seus contextos de uso.

```
Classe 1 - Palavras-chave: ['russia', 'in', 'suspend', 'to', 'all']
Classe 2 - Palavras-chave: ['russia', 'in', 'russian', 'suspend', 'to']
   Classe                                             Action
0    Saiu                  continue operating plant in russia
1  N Saiu  russian companies open accounts with the bank;...
2    Saiu                               still flying to russia
3    Saiu                               still flying to russia
4    Saiu                            still operating in russia
5    Saiu                            still operating in russia
6    Saiu                               distributors in russia
7  N Saiu                               not disclosed publicly
8    Saiu                            still operating in russia
9  N Saiu  joint venture with the russian sovereign wealt...
```

Fig. 7. Execução do Word2Vec na nossa base de dados

Analisando os resultados apresentados na figura 7 já conseguimos ver que o resultado obtido não corresponde ao que acontece usando a similaridade de palavras. Tentamos agora utilizar uma abordagem de humor e negatividade da frase, o TextBlob.

A figura 8 mostra que também não tivemos um resultado muito bom, errando com frequência os dados, então para geração dos dados decidimos usar o Grade, invés do Action setando alguns limites, o que gerou um bom resultado e fidedigno as Actions relatadas nos dados que foram observados.

Como a base não tem valores ausentes, não foi necessário fazer o processamento de preenchimento de dados faltantes. Como agora estamos usando a coluna Grade como guia, decidimos verificar se

```
  Classe                                             Action
0   Saiu                  continue operating plant in russia
1   Saiu  russian companies open accounts with the bank;...
2   Saiu                               still flying to russia
3   Saiu                               still flying to russia
4   Saiu                            still operating in russia
5   Saiu                            still operating in russia
6   Saiu                               distributors in russia
7   Saiu                               not disclosed publicly
8   Saiu                            still operating in russia
9   Saiu  joint venture with the russian sovereign wealt...
```

Fig. 8. Execução do TextBlob na nossa base de dados

tinha valores inconsistente na coluna, onde deve ter apenas as letras de A a F, e podemos observar não há valores inconsistentes.

Para a coluna Action, queríamos fazer a vetorização dos dados, pois ela continha muitos valores únicos, e após vários testes com OneHotEncode, StopWords e etc, percebemos que a melhor maneira seria deixar essa coluna como está, pois ou ela ficava com muitas colunas desnecessárias, ou com numerações não muito boas, e acabávamos perdendo desempenho em outros algoritmos que iremos apresentar mais pra frente. Sobre as colunas Industry e Country, aplicamos o LabelEncoder.

Em suma, para o tratamento da nossa base de dados, optamos por transformar alguns atributos em valores numéricos, e priorizar a coluna Grade como um classificador, que demonstraria o quanto uma empresa estaria ligada ou não a Rússia.

3.4 Geração de regras de associação

A mineração de regras de associação é uma técnica crucial na análise de conjuntos de dados. Essa abordagem busca identificar padrões recorrentes entre itens, permitindo a revelação de associações e relações ocultas. Um dos métodos amplamente utilizados para essa tarefa é o algoritmo Apriori. Nesse contexto, uma função de mineração de regras de associação visa encontrar associações frequentes entre conjuntos de itens, utilizando métricas como suporte e lift para avaliar a força e importância dessas associações. Ao aplicar esse processo, é possível obter insights valiosos sobre a interdependência entre diferentes atributos, possibilitando tomadas de decisão informadas e estratégias direcionadas com base nas descobertas obtidas.

A função "regras" desempenha um papel fundamental na análise de dados por meio da mineração de regras de associação. Com uma abordagem sistemática e configurável, essa função permite a descoberta e avaliação de associações significativas entre itens dentro de um conjunto de dados.

O parâmetro de suporte mínimo estabelece o suporte mínimo necessário para considerar uma associação como relevante. Ele determina a proporção de transações nas quais um conjunto de itens ocorre em relação ao total de transações. Já o parâmetro de confidence mínima define o nível mínimo de confiança exigido para uma regra ser considerada válida. A confiança mede a probabilidade condicional de que um consequente ocorra, dado um antecedente.

Por fim, o parâmetro "consequente" permite direcionar a busca por regras, focando em um conjunto específico de itens como consequente. Isso possibilita investigações direcionadas em relação a um conjunto pré-definido de interesse.

Em resumo, a função "regras" proporciona uma ferramenta poderosa para a mineração de regras de associação, permitindo aos analistas explorar interações e padrões em dados, com base em critérios personalizáveis. Sua capacidade de descobrir associações relevantes e significativas é crucial para a obtenção de insights valiosos e embasados, contribuindo para a tomada de decisões informadas em diversas áreas de aplicação.

Com as seguintes configurações:

```
resultadoSaiu = regras(0.09, 0.1, 1, 'Grade:1')
resultadoSaiu.head(len(resultadoSaiu))
```

Encontramos essas regras que tem como consequente se a empresa saiu:

Antecedentes	Suporte	Confiança	Lift
(Country:64)	0.103732	0.358862	1.088985
(Industry:6)	0.095509	0.374690	1.137015
(OTAN:1)	0.234662	0.337273	1.023471
(OTAN:1, Country:64)	0.103732	0.358862	1.088985
(OTAN:1)	0.103732	0.149091	1.437273
(Country:64)	0.103732	0.358862	1.529275

Table 4. Tabela de Associações

E com o consequente da empresa que não saiu:

Antecedentes	Suporte	Confiança	Lift
(Industry:1)	0.142315	0.718850	1.072171
(OTAN:0)	0.209361	0.688150	1.026382
(Country:64)	0.185326	0.641138	1.390451
(OTAN:1)	0.185326	0.266364	1.437273
(Industry:1, OTAN:1)	0.105629	0.729258	1.087695
(Industry:1)	0.105629	0.533546	1.157115
(OTAN:1)	0.105629	0.151818	1.066776

Table 5. Tabela de Associações

3.5 Classificação

Sobre a classificação, como foi informado na parte de fundamentação teórica utilizamos o SVM, Regressão Logística, Random Forest e KNN e para cada um desses esses foram os parâmetros testados no Grid Search:

Para o SMV testamos com os param_grid 'C': [200, 300] 'gamma': [0.0095, 0.0096] e kernel='rbf'. O hiperparâmetro C controla a penalidade por classificações erradas no SVM. É um parâmetro de regularização que controla a suavização do hiperplano de decisão. O hiperparâmetro gamma (γ) controla o alcance da influência de um único ponto de treinamento. Para kernels não lineares, como o kernel RBF (Radial Basis Function), gamma determina o quão "flexível" ou "complexo" é o modelo. E o kernel é uma função que transforma os dados de entrada em um espaço de maior dimensão, muitas vezes infinitamente dimensional, permitindo assim a separação de dados não linearmente separáveis em dimensões mais altas.

Utilizando a função *grid_search.best_params_* encontramos que os melhores parâmetros foram {'C': 300, 'gamma': 0.0096} e obtivemos uma acurácia de 67%, F1-Score de 0.176 e AUC-ROC de 0.515.

Para a Regressão Logística testamos os param_grid 'C': [150, 200], 'solver': ['lbfgs', 'liblinear']}. O parâmetro "C" controla a força da regularização no modelo de Regressão Logística. A regularização é uma técnica que ajuda a evitar o overfitting (sobreajuste) do modelo aos dados de treinamento, tornando-o mais geral e melhor em generalizar para dados não vistos. O parâmetro "solver" determina o algoritmo usado para otimizar os parâmetros do modelo durante o treinamento da Regressão Logística. A escolha do solver pode afetar a velocidade da convergência e a eficiência geral do treinamento. Alguns dos valores possíveis para o parâmetro "solver" incluem:

'newton-cg': Algoritmo de otimização baseado no método de Newton-Conjugado.

'lbfgs': Algoritmo de otimização baseado no método Quasi-Newton Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS).

'liblinear': Algoritmo especialmente adequado para conjuntos de dados pequenos. Usa uma abordagem de programação linear.

'sag': Stochastic Average Gradient descent, um método estocástico para grandes conjuntos de dados.

'saga': Uma versão melhorada do "sag" que suporta regularização L1.

Utilizando a função *grid_search.best_params_* encontramos que os melhores parâmetros foram {'C': 150, 'solver': 'lbfgs'} e obtivemos uma acurácia de 70%, F1-Score de 0.0 e AUC-ROC de 0.5, o que indica ineficiência total do modelo.

Para a Random Forest testamos os param_grid {'n_estimators': [100, 200], 'max_depth': [None, 10], 'min_samples_split': [5, 10], 'min_samples_leaf': [1, 2]}. O n_estimators é o número de árvores na floresta aleatória. Quanto mais árvores, mais robusto o modelo tende a ser, reduzindo o risco de overfitting. No entanto, um grande número de árvores também aumenta o tempo de treinamento e a previsão. O max_depth define a profundidade máxima de cada árvore na floresta. Uma árvore com profundidade maior pode se ajustar aos detalhes dos dados de treinamento, mas também pode ser mais suscetível a overfitting. Definir um limite de profundidade pode ajudar a limitar o crescimento da árvore e melhorar a generalização para dados não vistos. O min_samples_split define o número mínimo de amostras necessárias em um nó para que ele possa ser dividido em dois nós filhos. Isso ajuda a controlar o crescimento das árvores. Um valor maior para este parâmetro pode levar a árvores mais profundas e detalhadas, enquanto um valor menor pode levar a árvores mais rasas. O minz_samplesz_leaf define o número mínimo de amostras necessárias em uma folha. Ter folhas com poucas amostras pode levar a um ajuste excessivo. Aumentar esse valor resulta em folhas maiores e pode ajudar a evitar overfitting.

Utilizando a função *grid_search.best_params_* encontramos que os melhores parâmetros foram {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200} e obtivemos uma acurácia de 67%, F1-Score de 0.383 e AUC-ROC de 0.582.

Para o KNN utilizamos os param_grid {'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'], 'p': [1, 2]} em que o primeiro representa o número de vizinhos mais próximos que o algoritmo KNN usará para fazer uma previsão ou classificação, o weights especifica como os vizinhos devem ser ponderados ao fazer uma

previsão ou classificação e o 'p' define o parâmetro de potência usado na métrica de distância. Isso afeta a maneira como as distâncias entre pontos são calculadas. Alcançando uma acurácia de 68.5%, F1-Score de 0.479 e AUC-ROC de 0.627.

Para todos esses modelos utilizamos o cv=10 e scoring='accuracy'. A validação cruzada (cv) é uma técnica usada para avaliar o desempenho de um modelo de aprendizado de máquina. Ela envolve dividir o conjunto de dados em várias partes (chamadas de "dobras" ou "folds"), treinar o modelo em parte dos dados (conjunto de treinamento) e avaliá-lo na parte restante (conjunto de validação). Esse processo é repetido várias vezes, alternando as dobras de treinamento e validação. A validação cruzada ajuda a estimar quão bem o modelo pode generalizar para novos dados, reduzindo o risco de overfitting e fornecendo uma avaliação mais robusta do desempenho do modelo. Já o "scoring" se refere à métrica usada para avaliar o desempenho do modelo. Ao treinar um modelo, você deseja avaliar quão bem ele está se ajustando aos dados e fazendo previsões precisas. Existem várias métricas de avaliação, dependendo do tipo de problema (classificação, regressão, clustering, etc.). Exemplos comuns de métricas de classificação incluem "accuracy" (acurácia), "precision", "recall", "F1-score" e "ROC AUC". Para problemas de regressão, métricas como "mean squared error" (erro quadrático médio), "mean absolute error" (erro absoluto médio) e "R-squared" são comuns. Ao usar o GridSearchCV ou outras técnicas de busca de hiperparâmetros, você pode especificar a métrica de "scoring" que deseja otimizar para encontrar a melhor combinação de parâmetros.

4 AVALIAÇÃO EXPERIMENTAL E ANÁLISE DOS RESULTADOS

Conseguimos os seguintes gráficos plotando a acurácia, F1-Score e AUC-ROC:

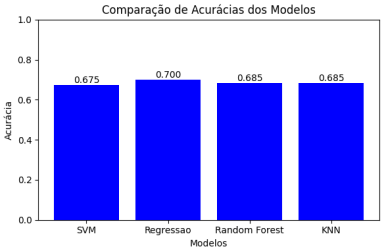

Fig. 9. Comparação de Acurácias dos Modelos

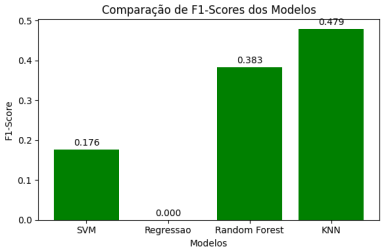

Fig. 10. Comparação de F1-Scores dos Modelos

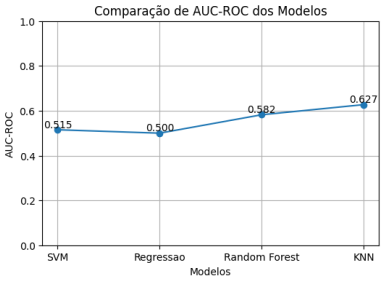

Fig. 11. Comparação de AUC-ROC dos Modelos

Em que podemos notar uma leve vantagem do KNN e descredibilizar a Regressão Logística já que o AUC-ROC = 0.5 o modelo não tem poder de discriminação e está apenas fazendo previsões aleatórias.

5 CONCLUSÃO

Em conclusão, o artigo abordou de maneira abrangente e sistemática a análise exploratória de uma base de dados que retrata as ações tomadas por empresas situadas na Rússia durante o conflito com a Ucrânia. Através de uma série de etapas de pré-processamento, análise estatística e aplicação de técnicas de mineração de dados, o estudo proporcionou insights relevantes sobre as dinâmicas e tendências presentes nos dados coletados.

Através da análise estatística, observou-se a predominância de indústrias de consumo primário e secundário, além da notável representatividade de empresas originárias dos Estados Unidos. A análise das ações anunciadas pelas empresas demonstrou que muitas optaram por manter suas operações na Rússia, seguida da decisão de suspender as atividades.

Através da mineração de regras de associação, foi possível identificar padrões recorrentes entre os atributos, enquanto os modelos de classificação SVM, Regressão Logística, Random Forest e KNN permitiram prever e analisar as ações das empresas com base em suas características, com melhor resultado para o KNN e descredibilizando a Regressão Logística.

Portanto, o artigo representa uma contribuição significativa para a compreensão das implicações das ações empresariais em contextos geopolíticos complexos, demonstrando o poder da mineração de dados e da aprendizagem de máquina na análise de conjuntos de dados desafiadores. Os insights obtidos têm o potencial de influenciar estratégias empresariais e a tomada de decisões informadas em cenários semelhantes no futuro.

REFERENCES

Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

LN de Castro and DG Ferrari. 2016. Introdução à Mineração de Dados: Conceitos Básicos. *Algoritmos e Aplicações, Saraiva* (2016).

Danilo Rogério de Souza. 2016. A nova geopolítica russa e o Eurasianismo. *Revista de Geopolítica* 3, 2 (2016), 61–70.

Katti Faceli, Ana Carolina Lorena, João Gama, Tiago Agostinho de Almeida, and André C. P. L. F. de Carvalho. 2021. *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina.* https://integrada.minhabiblioteca.com.br/reader/books/9788521637509

Steve R Gunn et al. 1998. Support vector machines for classification and regression. *ISIS technical report* 14, 1 (1998), 5–16.

Scikit-Learn Development Team. 2021. *Scikit-Learn: Machine Learning in Python.* URL: https://scikit-learn.org/.