Celso Gabriel Vieira Ribeiro Lopes

Um estudo comparativo entre bag-of-words e word embedding na classificação de textos
Relatório de Atividades - IC

São José dos Campos - Brasil

Julho de 2020

Celso Gabriel Vieira Ribeiro Lopes

Um estudo comparativo entre bag-of-words e word embedding na classificação de textos
Relatório de Atividades - IC

Relatório final apresentado à Universidade Federal de São Paulo como parte dos requisitos para a bolsa de iniciação científica PIBIC.

Orientador: Profª. Dra. Lilian Berton

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Julho de 2020

Resumo

Atualmente há uma quantidade massiva de dados textuais sendo produzida e armazenada diariamente na forma de e-mails, artigos, notícias e postagens em redes sociais ou blogs. Processar, organizar ou gerenciar essa grande quantidade de dados textuais manualmente exige um grande esforço humano, sendo muitas vezes impossível de ser realizado. Com isso, técnicas computacionais que requerem pouca intervenção humana e que permitem a organização, gerenciamento e extração de conhecimento de grandes quantidades de textos têm ganhado destaque nos últimos anos e vêm sendo aplicadas tanto na academia quanto em empresas. O aprendizado de máquina pode ser empregado para detectar padrões de maneira automática em dados e realizar classificação de textos. Este trabalho objetiva realizar uma avaliação empírica demonstrando o benefício do uso das representações *bag-of-words* e *embeddings* para a classificação de textos, bem como testar diferentes algoritmos de aprendizado de máquina supervisionado.

Palavras-chaves: Aprendizado de máquina. Processamento de linguagem natural. Bag of word. Word embedding.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

A inteligência artificial (IA) pode ser definida como o ramo da ciência da computação que se ocupa da automação do comportamento inteligente. Ela busca que máquinas aprendam com experiências e possam se adequar a entrada de dados, simulando a execução de algumas tarefas como seres humanos (1). Duas subáreas da inteligência artificial são o processamento de linguagem natural (PLN) e o aprendizado de máquina (AM) (2).

O aprendizado de máquina se tornou muito importante nas últimas duas décadas. Com ele é possível dar aos computadores a capacidade de aprender sem serem explicitamente programados. Além disso, os algoritmos acabam aprendendo com seus próprios erros e fazem previsões sobre os dados. Essa facilidade está sendo usada em diversas áreas da tecnologia da informação atualmente.

Enquanto os computadores utilizam linguagens formais para processar informação (linguagens de programação como Python e Java), os humanos utilizam a linguagem natural. O desafio da tecnologia é converter essa linguagem natural para uma linguagem formal, que os computadores entendam. Assim, o processamento de linguagem natural visa oferecer um nível mais alto de compreensão da linguagem natural através do uso de recursos computacionais, com o uso de técnicas para o rápido processamento de texto.

Nesse estudo iremos fazer uma análise para verificar qual o melhor algoritmo de aprendizado de máquina e de extração de características textuais na classificação de textos.

1.1 Objetivos

- Comparar vetores de características textuais tradicionais como *Bag of words* (BOW) com *embeddings* gerados via redes neurais.
- Analisar o impacto dos vetores de características textuais em diferentes algoritmos de classificação.
- Analisar o impacto dos vetores de características textuais em *datasets* desbalanceados e multiclasses.

1.2 Organização do documento

A seguir, a seção 2 apresenta os principais conceitos teóricos relacionados com o trabalho, a seção 3 apresenta os materiais e métodos utilizados no trabalho, como *datasets*,

ferramental teórico e computacional. A seção 4 apresenta os resultados obtidos. E a seção 5 apresenta as considerações finais.

2 Fundamentação teórica

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma área da Inteligência Artificial e Linguística, dedicada a fazer com que os computadores entendam as declarações ou palavras escritas em idiomas humanos. O processamento de linguagem natural surgiu para facilitar o trabalho do usuário e satisfazer o desejo de se comunicar com o computador em linguagem natural (4).

A pesquisa em PLN está voltada a três aspectos da comunicação em língua natural:

- Som: fonologia
- Estrutura: morfologia, léxico e sintaxe
- Significado: semântica e pragmática

A *fonologia* é a parte da linguística que está relacionada ao reconhecimento dos sons que compõem as palavras de uma língua. A *morfologia* é o estudo da estrutura, da formação e da classificação das palavras. O *léxico* é a interpretação do significado da palavra feita pelos seres humanos e por sistemas de PLN. A *sintaxe* define a estrutura gramatical de uma frase, de acordo como as palavras se relacionam nessa frase. A *semântica* associa significado a uma estrutura sintática, determinando os possíveis significados de uma frase. Finalmente, a *pragmática* verifica se o significado dado a uma estrutura sintática é realmente a mais adequada dentro do contexto considerado (5).

Vamos exemplificar com um corpus específico, que trata de uma coleção de texto ou fala legível por computador. Por exemplo, o corpus Brown é uma coleção de milhões de palavras de 500 textos escritos em inglês com diferentes gêneros (jornal, ficção, acadêmico, etc.), reunidos na Brown University em 1963–64. Quantas palavras estão na seguinte frase?

"He stepped out into the hall, was delighted to encounter a water brother".

Essa frase possui 13 palavras se desconsiderarmos os sinais de pontuação como palavras, 15 se contarmos elas. A pontuação é muito importante pois é através dela que identificamos os limites de cada frase dentro de um determinado texto. Para algumas tarefas, como por exemplo, marcação de parte de um discurso ou síntese de fala, acabamos considerando a pontuação como uma palavra.

Porém, não basta apenas saber o que é uma palavra. Também é necessário normalizar o texto que será processado. Agora veremos algumas das técnicas mais comuns utilizadas em PLN.

- Tokenização

  A tokenização tem como objetivo separar palavras ou frases em unidades. Ela marca cada palavra como um token no texto, identificando-a mesmo que ela esteja encostada em uma pontuação. Um exemplo de tokenização é:

  Normalizar o texto é importante.

  [‘Normalizar’, ‘o’, ‘texto’, ‘é’, ‘importante’, ‘.’]

- Normalização das palavras

  A normalização de palavras é a tarefa de colocar palavras/tokens em um formato padrão, escolhendo um único formato normal para palavras com várias formas, como por exemplo “Normalizar” para “normalizar” e “João” para “joão”. Essa padronização pode ser valiosa, apesar das informações de ortografia perdidas no processo de normalização.

- Remoção de *Stopwords*

  Stopwords são palavras muito frequentes, como por exemplo “a”, “de”, “o”, “e”, “da”, entre outras, pois na maioria das vezes são palavras irrelevantes para o processamento do texto. Não existe um conjunto certo de *stopwords*, elas variam de acordo com o texto e a intenção do processo. No caso da análise de sentimentos, não poderia ser retirada a palavra “não” pois ela traz um sentido de negação para a sentença, indicando o sentimento transmitido naquela frase.

- Stemização e Lematização

  O processo de Stemização reduz uma palavra em seu radical. Como por exemplo a palavra “livro” é reduzida em “livr”, assim como “livrinho” e “livreiro”. A lematização é a tarefa de determinar se duas palavras tem a mesma raiz, apesar de suas diferenças. Como por exemplo gato, gata, gatos e gatas são todas formas do mesmo lema: gato.

2.2 Extração de features

2.2.1 Bag of words

Bag-of-words é um método bastante usado para representação de documentos textuais. No caso de texto, este é representado como um vetor, onde cada posição possui

o número de aparições de uma palavra ao longo do conteúdo do documento (6). Este método foca na frequência das palavras, mas o vetor resultante pode ser normalizado ou dimensionado de diversas formas, como por exemplo, com *Term-Frequency-Inverse Document Frequency* (TF-IDF).

Term Frequency (TF)

TF significa frequência de termo. Ela transforma uma base de dados com documentos de textos em uma matriz de contagens de tokens, ou seja, ele conta quantas vezes cada palavra aparece em um determinado texto e coloca o seu valor na matriz.

Considerando t o termo e d o documento, a fórmula utilizada para calcular o TF é:

$$TF(t,d) = 1 \tag{2.1}$$

Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF significa frequência de termo vezes a frequência inversa do documento. Ele pega a matriz de contagem feita no TF e faz uma ponderação. O objetivo de usar TF-IDF é reduzir o impacto de tokens que ocorrem com muita frequência em um determinado corpus.

Considerando t o termo, d o documento, n o número total de documentos e DF(t) a frequência de documentos de t, a fórmula utilizada para calcular o TF-IDF é:

$$IDF(t) = log[n/DF(t)] + 1 \tag{2.2}$$

$$TFIDF(t,d) = TF(t,d) \times IDF(t) \tag{2.3}$$

2.2.2 *Word Embedding*

Word embedding é um conjunto de técnicas que mapeiam a semântica e sintática de uma linguagem natural em um espaço usando estatísticas. Assim, palavras de um conjunto de texto são mapeados para vetores. Esses vetores possibilitam um melhor desempenho nas tarefas do PLN, pois não tratam as palavras como únicas, em vez disso, refletem a similaridade e dissimilaridade entre elas. Onde a troca de uma palavra por um sinônimo não irá interferir na validade da sentença, como por exemplo nas palavras “cão” e “cachorro”, e a sentença continuará válida (7).

Palavras como “Brasília” e “Brasil” são mapeadas em vetores próximos por causa do seu grau de similaridade. Outro exemplo é “a parede é azul” pode ser alterada por “a parede é vermelha”, pois elas são palavras que estão em uma classe similar.

Esse método tem como objetivo aprender uma representação para cada palavra. Onde as palavras são mapeadas para vetores como por exemplo vec(cão) = (0.0, 1.6, -1). O *word embedding* aprende muitas regularidade e padrões linguísticos. A equação abaixo apresenta um caso de relação que pode ser aprendida.

$$rei - homem + mulher \approx rainha \tag{2.4}$$

O *word embedding* é inspirado num modelo de rede neural e vem demonstrando bom desempenho em algumas tarefas de PLN como por exemplo reconhecimento de linguagem natural, similaridade entre palavras, classificação de documentos, entre outros. Um dos modelos que utiliza o *word embedding* é o word2vec, que será utilizado em nossa pesquisa.

2.2.3 Word2vec

O word2vec é um modelo proposto por Mikalov, et al. em 2013 (10). Ela é utilizada para calcular a representação de palavras como vetores via redes neurais. O word2vec se tornou um dos métodos mais usados no pré-processamento de textos. Ele é utilizado em análise de sentimento, reconhecimento de entidades nomeadas, ou até mesmo geração de textos caractere a caractere ou palavra a palavra.

Ele oferece dois modelos de redes neurais em sua arquitetura: continuous bag-of-words (CBOW) e skip-gram.

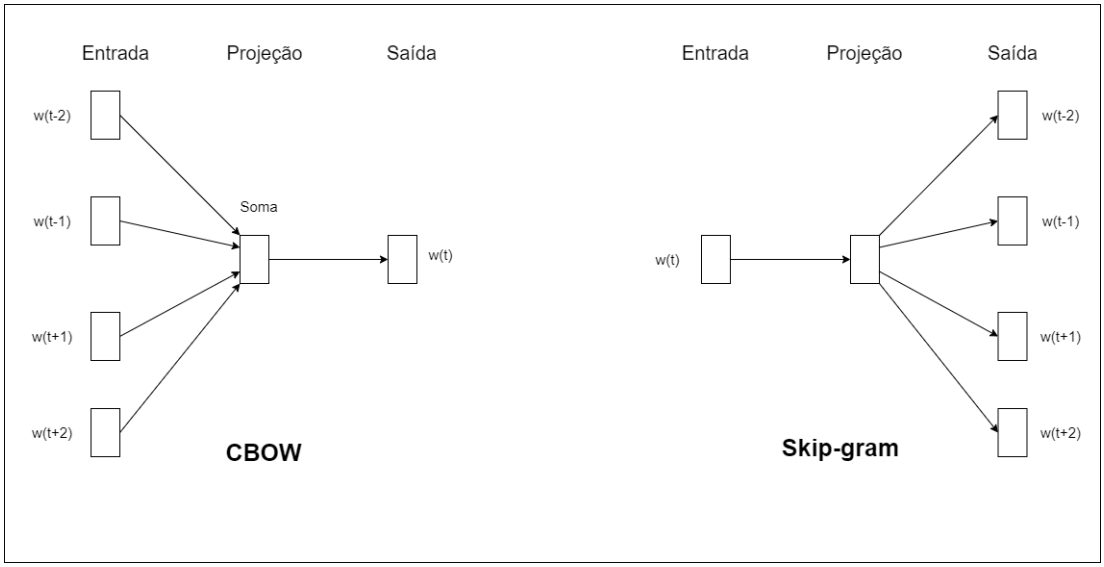
- CBOW

  No modelo CBOW a entrada é um contexto e a saída é uma palavra omitida, para isso se combina as palavras que estão em volta para prever a palavra do meio. O contexto é considerado como as palavras que estão ao redor da palavra alvo. Um exemplo é usar as duas palavras anteriores e as duas palavras posteriores da palavra alvo. No caso da frase A B C D E, o contexto da palavra C é A, B, D e E.

- Skip-gram

  O modelo Skip-gram é o contrário do CBOW. Nesse caso sendo uma palavra a entrada, o algoritmo tenta prever as palavras do contexto como saída.

Figura 1 – Representação da arquitetura CBOW e Skip-gram.

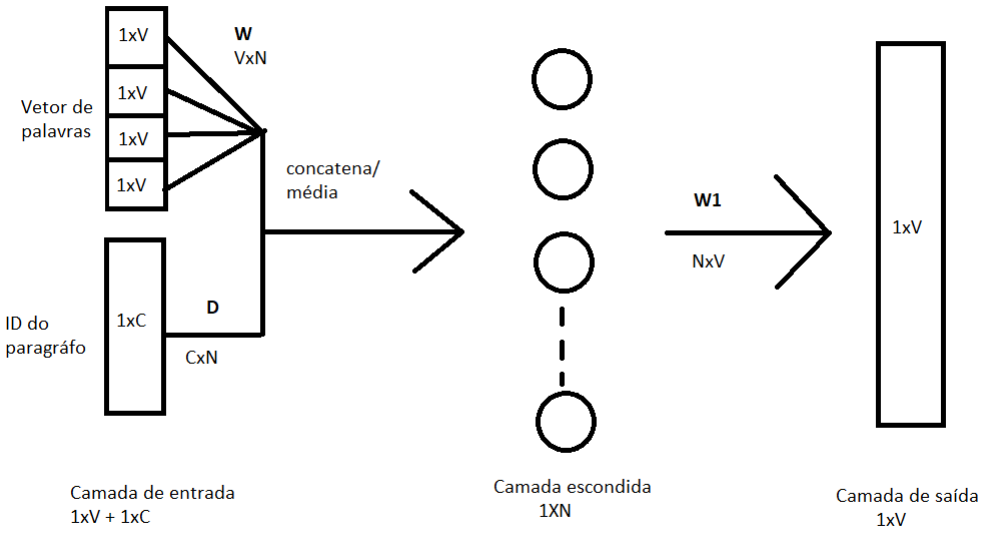


O word2vec funciona lendo um corpus como entrada e devolvendo vetores de palavras como saída. No processo é construído um vocabulário, e ocorre o aprendizado das representações do vocabulário como vetores. Pode ser usado um dos dois modelos da arquitetura. O CBOW é mais rápido e aplicável em grandes bases de dados, já o Skip-gram oferece uma melhor representação das palavras quando a base de dados é menor.

2.2.4 Doc2vec

O conceito do Doc2vec é bem parecido com o modelo do word2vec. A diferença é que o doc2vec utiliza o word2vec e adiciona mais um vetor(ID do parágrafo) à entrada. A arquitetura é mostrada a seguir.

Figura 2 – Representação da arquitetura doc2vec



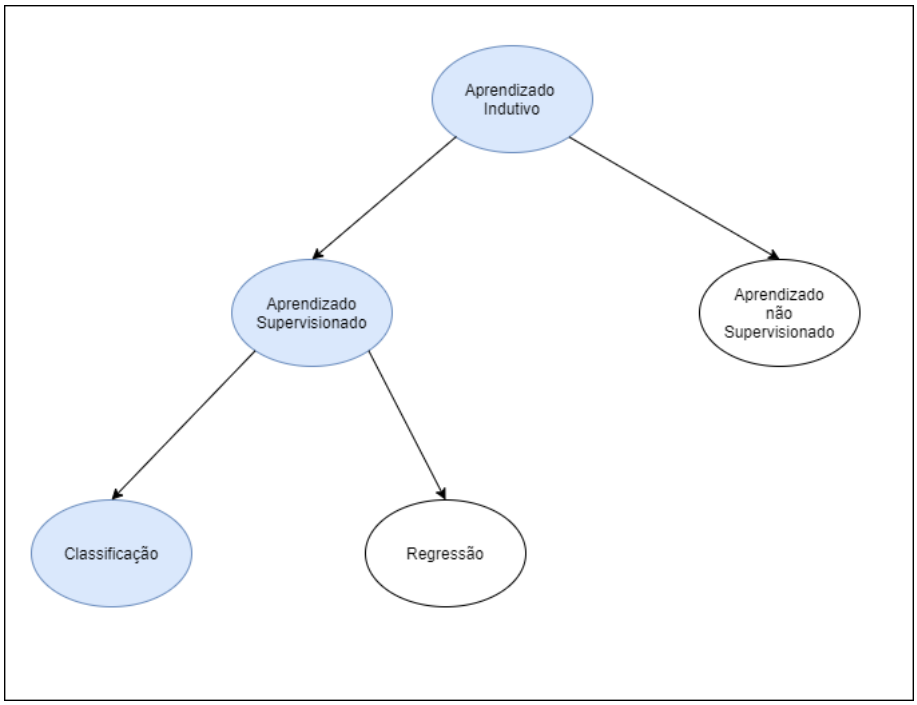
O diagrama acima é baseado no CBOW, mas em vez de usar apenas palavras próximas para prever o contexto, ele também adiciona outro vetor, que é único no documento. Portanto, ao treinar os vetores de palavras W, o vetor D também é treinado e, no fim do treinamento, mantém uma representação numérica do documento.

2.3 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma área de Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Dentro de AM existem técnicas que podem ser divididas em aprendizado supervisionado e não-supervisionado (13).

No caso de aprendizado supervisionado, é dado ao algoritmo um conjunto de exemplos, com atributos de entrada e atributos de saídas (rótulos). O objetivo do algoritmo é construir um classificador que pode determinar corretamente o rótulo de um novo exemplo. Para valores discretos esse problema é conhecido como *classificação*, já para valores contínuos, ele é conhecido como *regressão*. Em contraste, no aprendizado não-supervisionado é dado um conjunto de exemplos apenas com atributos de entrada, onde o algoritmo analisa e tenta determinar se eles podem ser agrupados de alguma maneira. Iremos usar os algoritmos supervisionados e dentre eles usaremos: Nearest Neighbors, Support Vector Machine, Naive Bayes, Decision Tree e Multilayer Perceptron.

Figura 3 – Hierarquia do Aprendizado.



2.3.1 Conceitos Básicos

Com o propósito de facilitar o entendimento de termos utilizados em Aprendizado de Máquina, é apresentado a seguir uma lista dos conceitos mais usados (13).

- **Exemplo:** Um exemplo, também denominado caso, registro ou dado na literatura, é um vetor de valores de atributos. Um exemplo descreve o objeto de interesse, tal como um texto ou dados sobre uma doença.
- **Atributo:** Um atributo descreve alguma característica ou aspecto do exemplo. Normalmente, existem pelo menos dois tipos de atributos: nominal, quando não existe uma ordem entre os valores (por exemplo cor) e contínuo, quando existe uma ordem linear nos valores (por exemplo, altura).
- **Classe:** No aprendizado supervisionado, todo exemplo possui pelo menos um atributo especial denominado rótulo ou classe, que descreve o fenômeno de interesse. No caso em que os exemplos são textos, as classes poderiam ser o gênero de cada texto.
- **Conjunto de Exemplos:** é composto por exemplos contendo valores de atributos bem como a classe associada. Usualmente, um conjunto de exemplos é dividido em dois subconjuntos disjuntos: o **conjunto de treinamento**, utilizado para o aprendizado do conceito e o **conjunto de teste**, utilizado para medir o grau de efetividade do conceito aprendido.
- **Ruído:** é comum, no mundo real, trabalhar com dados imperfeitos. Eles podem ser derivados do próprio processo que gerou os dados, do processo de aquisição de dados, do processo de transformação ou mesmo de classes rotuladas incorretamente.
- **Classificador:** Dado um conjunto de dados para o treinamento, um indutor gera como saída um classificador. Dessa maneira, ao submeter um novo exemplo no classificador, ele retorna com a maior precisão possível a classe à qual aquele exemplo pertence.
- **Erro e precisão:** Uma medida de desempenho muito usada é a taxa de erro de um classificador h, denotada por **err(h)**. A taxa de erro é obtida utilizando (2.5), que compara a classe verdadeira de cada exemplo com o rótulo atribuído pelo classificador induzido. O operador $||E||$ retorna 1 se a expressão E for verdadeira e zero se ela for falsa, e n é o número de exemplos. O complemento da taxa de erro, a precisão do classificador, denotada por $acc(h)$ é dado por (2.6).

$$err(h) = \frac{1}{n}\sum_{i=1}^{n} \|y_i \neq h(x_i)\| \tag{2.5}$$

$$acc(h) = 1 - err(h) \tag{2.6}$$

- **Overfitting:** Ao realizar o treinamento de um classificador, é possível que o conjunto de dados de entrada seja muito específico. Como o conjunto de treinamento é apenas uma amostra de todos os exemplos, é possível induzir casos que melhorem o desempenho do treinamento, enquanto pioram o desempenho de exemplos diferentes daqueles que estavam no conjunto de treinamento.

- **Underfitting:** É possível que poucos exemplos representativos sejam dados ao classificador, ou o usuário defina o tamanho do classificador como muito pequeno ou uma combinação de ambos. Nesse caso, é possível induzir casos que possuam uma melhora de desempenho muito pequena no conjunto de treinamento, assim como em um conjunto de teste.

- **Acurácia:** a taxa de predições corretas (ou incorretas) realizada pelo modelo para um determinado conjunto de dados. A acurácia é, em geral, estimada utilizando um conjunto independente de teste, que não foi usado em nenhum momento durante o processo de aprendizado.

2.3.2 k-Nearest Neighbors (Vizinhos mais próximos)

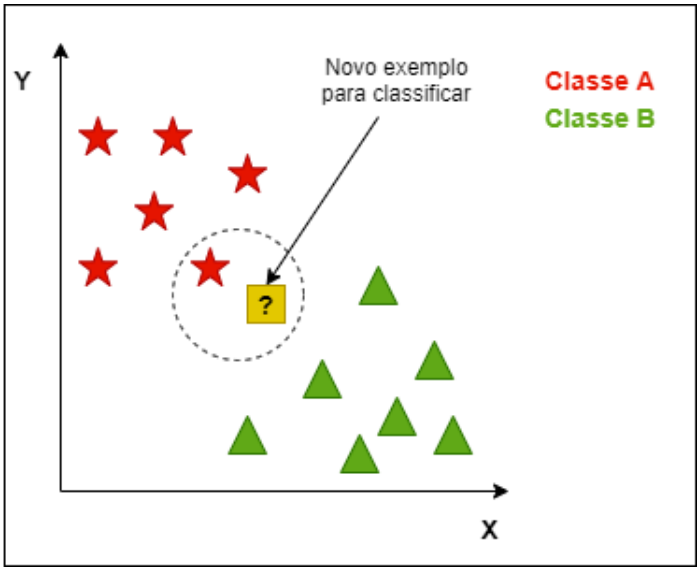
O método dos k-vizinhos mais próximos (kNN, do inglês *k-Nearest Neighbors*) é considerado um dos métodos de classificação mais antigos e simples. Apesar da simplicidade, o método tem alcançado um bom desempenho em vários cenários.

O algoritmo kNN é um algoritmo de aprendizado supervisionado do tipo *lazy*. A ideia geral desse algoritmo consiste em encontrar os k exemplos rotulados mais próximos do exemplo não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado. Os algoritmos da família kNN requerem pouco esforço durante a etapa de treinamento. Em contrapartida, o custo computacional para rotular um novo exemplo é relativamente alto, pois, no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos contidos no conjunto de exemplos de treinamento.

Em relação a escolha do valor k, não existe um valor único para a constante, a mesma varia de acordo com a base de dados. É recomendável sempre utilizar valores ímpares/primos, mas o valor ótimo varia de acordo com a base de dados.

Abaixo está uma figura que representa como o kNN funciona.

Figura 4 – Exemplo de classificação do método k-*Nearest Neighbor*



Calcular a distância é fundamental para o kNN. Existem diversas métricas de distância, e a escolha de qual usar varia de acordo com o problema. A mais utilizada é a distância Euclidiana:

$$D(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.7}$$

Outra distância muito utilizada também é a de Minkowsky:

$$D(p,q) = \left(\sum_{i=1}^{n}|p_i - q_i|^r\right)^{\frac{1}{r}} \tag{2.8}$$

Em ambos os casos o p e q são pontos n-dimensionais, e na equação de Minkowsky r é uma constante que deve ser escolhida previamente.

2.3.3 Naive Bayes

O algoritmo Naive Bayes é do tipo supervisionado, já que são fornecidos ao algoritmo de aprendizado de máquina as instâncias juntamente com as suas classes. O algoritmo faz o uso de fórmulas estatísticas e cálculo de probabilidades para realizar a classificação.

O classificador Naive Bayes se baseia na aplicação do Teorema de Bayes para o cálculo das probabilidades necessárias para a classificação. O Teorema de Bayes é mostrado abaixo já no contexto de aprendizado de máquina, isto é, dada uma nova instância A = a_1, a_2 ... a_n, deseja-se predizer sua classe:

$$P(classe|A) = \frac{P(A|classe) \times P(classe)}{P(A)} \tag{2.9}$$

Como A = a_1, a_2 ... a_n, tem se:

$$P(classe|a_1...a_n) = \frac{P(a_1...a_n|classe) \times P(classe)}{P(a_1...a_n)} \tag{2.10}$$

Para calcular a classe mais provável da nova instância, calcula-se a probabilidades de todas as possíveis classes e, no fim, escolhe-se a classe com a maior probabilidade como rótulo da nova instância. Em termos estatísticos, isso é equivalente a maximizar a P(classe|$a_1...a_n$). Então, deve-se maximizar o valor do numerador P($a_1...a_n$|classe)×P(classe) e minimizar o valor do denominador P($a_1...a_n$). Como o denominador é uma constante, pode-se anulá-lo no Teorema de Bayes, resultando na fórmula abaixo, na qual procura a classe que maximize o valor.

$$argmaxP(classe|a_1...a_n) = argmaxP(a_1...a_n|classe) \times P(classe) \tag{2.11}$$

A suposição "ingênua" que o classificador Naive Bayes faz é que todos os atributos $a_1...a_n$ da instância que se quer classificar são independentes. Dessa maneira, o complexo cálculo do valor do termo p($a_1...a_n$|classe) reduz-se ao simples cálculo P(a_1|classe)×...×P(a_n|classe.). Assim, a fórmula final utilizada pelo classificador é:

$$argmaxP(classe|a_1...a_n) = argmax \prod_i P(a_1...a_n|classe) \times P(classe) \tag{2.12}$$

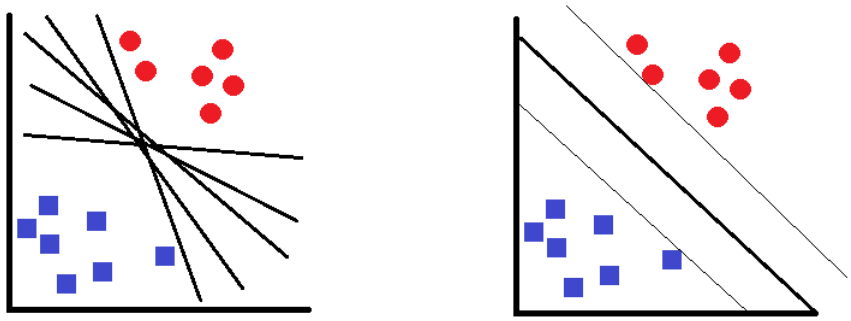
Sabe-se que na maioria dos casos, a suposição da independência dos atributos de uma instância é falsa. Mesmo assim, o classificador Naive Bayes produz resultados bastante satisfatórios.

2.3.4 *Support-vector machine* (Máquina de suporte vetorial)

A *Support-vector machine* (SVM) é uma técnica que cria hiperplanos que separam duas classes, ou mais, construindo uma "fronteira" em uma dimensão do espaço.

Consideremos um exemplo onde existem duas classes de pontos completamente separáveis em um espaço bidimensional, como é mostrado na figura abaixo, em que há duas classes, azul e vermelha. É fácil perceber que existe infinitas possibilidades de se colocar fronteiras capazes de separar as duas classes. Vapnik (15), o criador da SVM, definiu uma medida denominada margem, que auxilia na escolha das fronteiras para separar cada uma das classes. A margem é a distância entre a fronteira e o ponto mais próximo de um conjunto que será treinado.

Figura 5 – Representação de dados perfeitamente separáveis em duas classes



Como se observa na segunda figura acima as linhas mais finas, de ambos os lados da fronteira sólida que separa as duas classes, estão a distância máxima entre a fronteira e o ponto mais próximo que representa cada classe. O ponto e o quadrado que se encontra encostada na linha mais fina, estão equidistantes da fronteira. A margem definida por estes pontos pode ser utilizada para elaborar um modelo de classificação.

A inclinação e intercepto da fronteira que maximizam a distância entre a fronteira e os dados é conhecido como margem máxima de classificação. A margem máxima de classificação determina uma função de decisão D(x) para classificação de amostras de forma que quando D(x) < 0 as amostras classificadas como -1 (azul), e quando D(x) > 0 as amostras são classificadas como 1 (vermelho). Supondo agora uma amostra desconhecida u, a função de decisão pode ser escrita em termos de um intercepto e das inclinações referentes a cada variável de previsão como:

$$D(x) = \beta_0 + \beta^T u \tag{2.13}$$

Uma vez que a função acima é escrita como foco nas variáveis de previsão, podemos transformá-la de maneira que seja escrita em termo de cada ponto da amostra, na seguinte forma:

$$D(u) = \beta_0 + \sum_{i=1}^{P} \beta_j u_j \tag{2.14}$$

$$D(u) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i x_i^T u \tag{2.15}$$

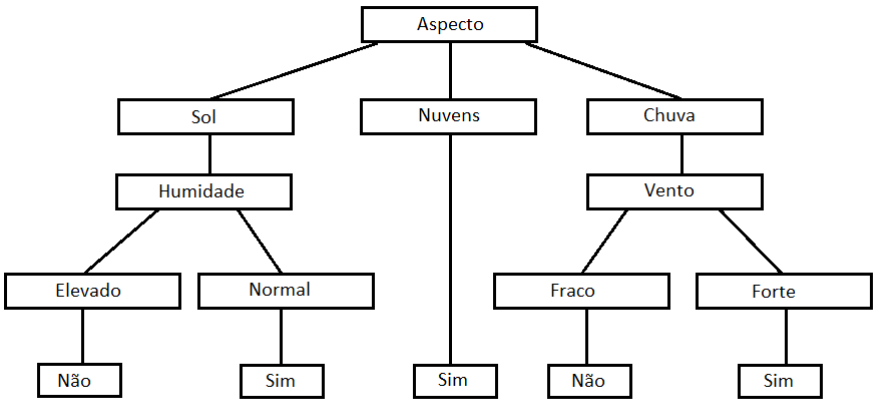
Sendo y_i a classe do ponto i, e $\alpha \geq 0$ um parâmetro calculado a partir do grupo do treinamento para cada ponto que compõe a margem. Se $\alpha = 0$ então os pontos estão fora da margem e se $\alpha > 0$ então os pontos estão sobre os limites da margem. Então, a equação de previsão não é formada por todos os valores observados no conjunto de treino, mas apenas os valores que estão sobre os limites da margem.

2.3.5 Decision Tree (Árvore de decisão)

A árvore de decisão é um classificador popular que não requer nenhum conhecimento ou configuração de parâmetro. Ela utiliza um aprendizado supervisionado e tendo os dados para o treinamento, podemos induzi-la. A árvore de decisão utiliza de uma estrutura de árvore usada para classificar classes com base em uma série de perguntas (ou regras) sobre os atributos da classe. Os atributos das classes podem ser qualquer tipo de variável como por exemplo valores binários, nominais, ordinais e quantitativos, enquanto as classes devem ser do tipo qualitativo.

A figura abaixo representa uma árvore de decisão onde cada nó contém um teste para algum atributo, cada ramo corresponde a um possível valor desse atributo, cada folha está associada a uma classe e, cada percurso da árvore, da raiz à folha corresponde uma regra de classificação.

Figura 6 – Exemplo de uma árvore de decisão para sair com o cachorro.



O critério utilizado para criar as partições é o da utilidade do atributo para a classificação. É aplicado, por este critério, um determinado ganho de informação a cada atributo. O atributo escolhido como atributo teste para o corrente nó é aquele que possui o maior ganho de informação. Nos casos em que a árvore é utilizada para classificação, os critérios de partição mais conhecidos são baseados na entropia e índice Gini.

- **Entropia:** é uma forma de medir a pureza de cada subconjunto de uma árvore de decisão. A entropia caracteriza a (im)pureza dos dados: em um conjunto de dados, é uma medida da falta de homogeneidade dos dados de entrada em relação a sua classificação.

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{2.16}$$

Onde p_i é a proporção de dados em S que pertencem à classe i.

- **Índice Gini:** é uma forma de medir o grau de heterogeneidade dos dados. Logo, ele pode ser utilizado para medir a impureza de um nó.

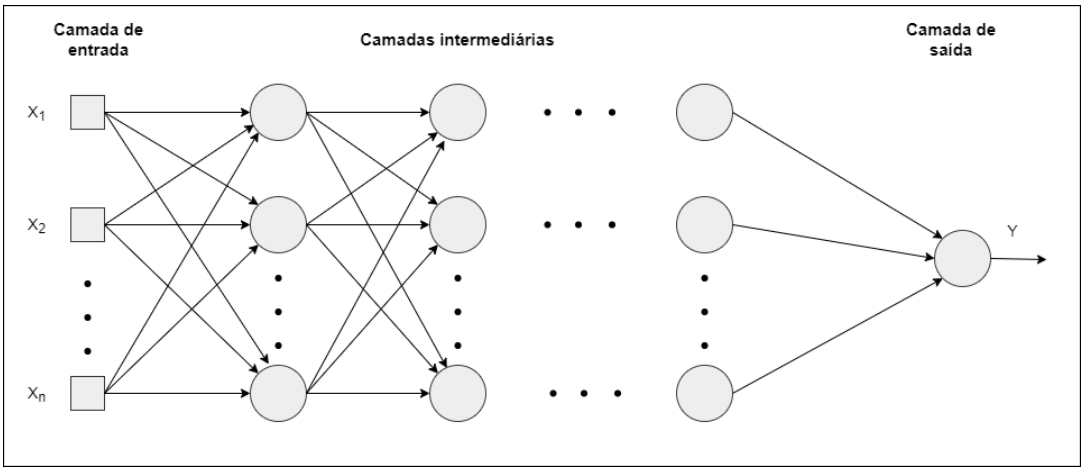
$$IG = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.17}$$

Onde p_i é a frequência relativa de cada classe em cada nó e c é o número de classes.

2.3.6 Multilayer Perceptron

Multilayer Perceptron (MLP) é uma rede neural formada por uma camada de entrada, uma ou mais camadas ocultas (intermediária), com um número indeterminado de neurônios, e uma camada de saída. A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias. A rede neural MLP é totalmente conectada, ou seja, um neurônio artificial em qualquer camada da rede está conectado a todos os outros neurônios artificiais da camada anterior. O fluxo de sinal através da rede, por sua vez, progride para frente, da esquerda para direita e de camada em camada.

Figura 7 – Exemplo de um multilayer perceptron



Fonte: O Autor.

Cada neurônio apresenta um conjunto de sinapses, caracterizada por um peso(w_{ij}) próprio. Sendo assim, um sinal x_p localizado na entrada da sinapse p e conectado ao neurônio n é multiplicado pelo peso sináptico ω_{np}. Todos os sinais de entrada ponderados pelas respectivas sinapses do neurônio são adicionados em seguida por um somador. A saída do somador (υ_n) é a entrada de uma função de ativação ($\varphi(.)$) que, por sua vez, irá produzir a saída do neurônio. A função de ativação ($\varphi(.)$) sigmóide será mostrada a seguir. Através dela é possível realizar o *backpropagation*.

$$y_i = \frac{1}{1 + exp(-v_j)} \tag{2.18}$$

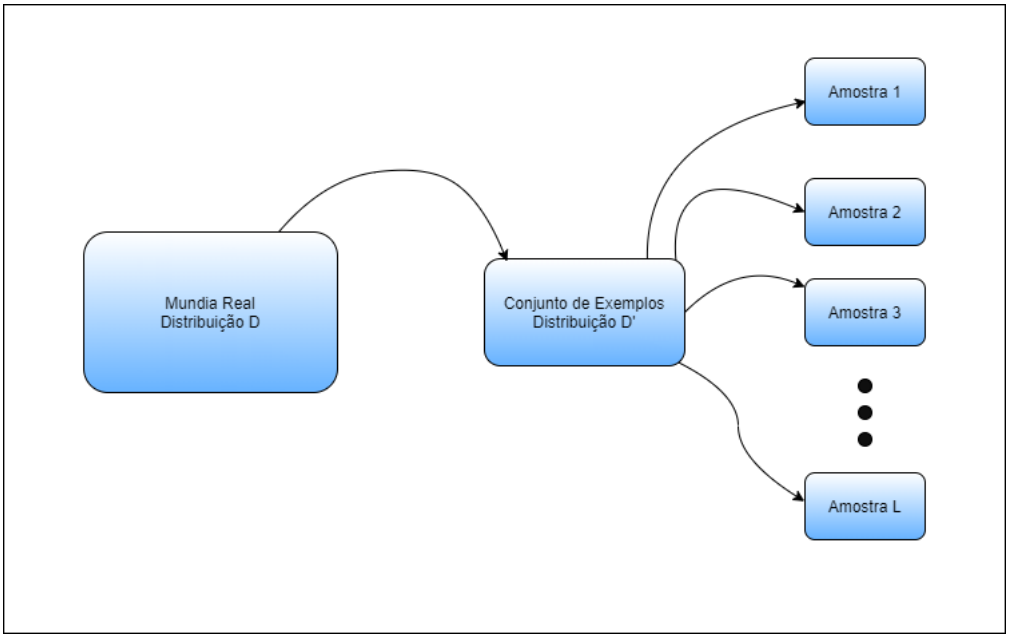
O algoritmo backpropagation (retropropagação de erro) é um algoritmo utilizado em MLP e utiliza funções de transferência diferenciáveis e não-lineares em um treinamento supervisionado.

A aprendizagem por *backpropagation* consiste em dois passos: um passo para frente, que é a propagação, e um passo para trás, que é a retropropagação. No primeiro passo, um vetor é apresentado aos nós de entrada da rede e seu efeito se propaga da esquerda para a direita e de camada em camada. Um conjunto de saídas é produzido como resposta da rede. Durante este processo, os pesos sinápticos de rede são todos fixos. A resposta da rede é subtraída da resposta desejada e, então, determina-se o sinal de erro. Este sinal é propagado para trás através da rede, contra a direção das conexões sinápticas. Os pesos são então ajustados de modo a se minimizar o sinal de erro.

2.4 Avaliação dos classificadores

Dados um conjunto de exemplos de tamanho finito e um indutor, é importante estimar o desempenho futuro do classificador induzido utilizando o conjunto de exemplos. O mundo real possui uma distribuição D desconhecida de exemplos em um dado domínio. Ao retirar alguns exemplos do mundo real, é obtido uma distribuição D', a qual é supostamente similar à distribuição D. Para estimar uma medida, geralmente a precisão ou o erro, de indutores treinados com base na distribuição D', extraem-se amostras a partir de D', treina-se um indutor com essas amostras e testa-se seu desempenho em exemplos de D'. Dessa maneira, é possível simular o processo de amostragem que ocorre no mundo real.

Figura 8 – Técnicas de estimativas baseadas na idéia de amostragem



É importante, ao estimar uma medida verdadeira, que a amostra seja aleatória. Para problemas reais, normalmente é feita uma amostra de tamanho n e o objetivo consiste em estimar uma medida para aquela população em particular. O método utilizado na pesquisa foi o *Cross-Validation*.

Em *k-fold cross-validation* os exemplos são aleatoriamente divididos em k partições mutuamente exclusivas (*folds*) de tamanho aproximadamente igual a n/k exemplos, onde n representa o número de exemplos. Os exemplos nos (k-1) *folds* são usados para treinamento e a hipótese induzida é testada no *fold* remanescente. Este processo é repetido k vezes, cada vez considerando um *fold* diferente para teste. O resultado na *cross-validation* é a média dos resultados calculados em cada um dos k *folds*.

Para se obter os resultados primeiramente é necessário ser feito uma matriz de confusão e através dela é retirado alguns resultados.

- Matriz de confusão

  A matriz de confusão oferece uma medida efetiva do modelo de classificação, ao mostrar o número de classificações corretas versus as classificações preditas para cada classe, em um conjunto de exemplos T. Os resultados são totalizados em duas dimensões: classes verdadeiras e classes preditas, para k classes diferentes $\{C_1, C_2, ..., C_k\}$. Cada elemento M(C_i, C_j) da matriz, $i, j = 1, 2, ..., k$, calculado por 2.19, representa o número de T que realmente pertencem à classe C_i, mas foram classificados como sendo da classe C_j.

Tabela 1 – Matriz de confusão de um classificador

Classe	predita C_1	predita C_2	$\cdots$	predita C_k
verdadeira C_1	$\mathrm{M}(C_1, C_1)$	$\mathrm{M}(C_1, C_2)$	$\cdots$	$\mathrm{M}(C_1, C_k)$
verdadeira C_2	$\mathrm{M}(C_2, C_1)$	$\mathrm{M}(C_2, C_2)$	$\cdots$	$\mathrm{M}(C_2, C_k)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
verdadeira C_k	$\mathrm{M}(C_k, C_1)$	$\mathrm{M}(C_k, C_2)$	$\cdots$	$\mathrm{M}(C_k, C_k)$

$$M\left(C_i, C_j\right) = \sum_{\{\forall(x,y)\in T: y=C_i\}} \|h(x) = C_j\| \tag{2.19}$$

O número de acertos, para cada classe, se localiza na diagonal principal $\mathrm{M}(C_i, C_i)$ da matriz. Os demais elementos $\mathrm{M}(C_i, C_j)$, para $i \neq j$, representam erros na classificação. A matriz de confusão ideal possui todos esses elementos iguais a zero já que ele não comete erros.

Considere um problema com duas classes. Com apenas duas classes, rotularemos como "+" (positivo) e "-" (negativo), as escolhas estão estruturadas para predizer a ocorrência ou não de um evento simples. Nesse caso, os dois erros possíveis são denominados *falso positivo* (F_P) e *falso negativo* (F_N). Na Tabela abaixo essa matriz de confusão é ilustrada para o problema com duas classes, onde V_P é o número de exemplos positivos classificados corretamente e V_N é o número de exemplos negativos classificados corretamente do total de $n = (V_P + V_N + F_P + F_N)$ exemplos.

Tabela 2 – Matriz de confusão para classificação com duas classes

Classe	predita C_+	predita C_-	Taxa de erro da classe	Taxa de erro total
verdadeira C_+	Verdadeiros positivos V_P	Falsos negativos F_N	$\frac{F_N}{V_P+F_N}$	$\frac{F_P+F_N}{n}$
verdadeira C_-	Falsos positivos F_P	Verdadeiros negativos V_N	$\frac{F_P}{F_P+V_N}$	

Através do resultado da matriz de confusão são utilizado 4 fórmulas para realizar a avaliação do classificador. Essas fórmulas são Accuracy, Precision, Recall e F1-Score respectivamente.

$$acc(h) = \frac{V_P + V_N}{n} \tag{2.20}$$

$$prec(h) = \frac{V_P}{V_P + F_P} \tag{2.21}$$

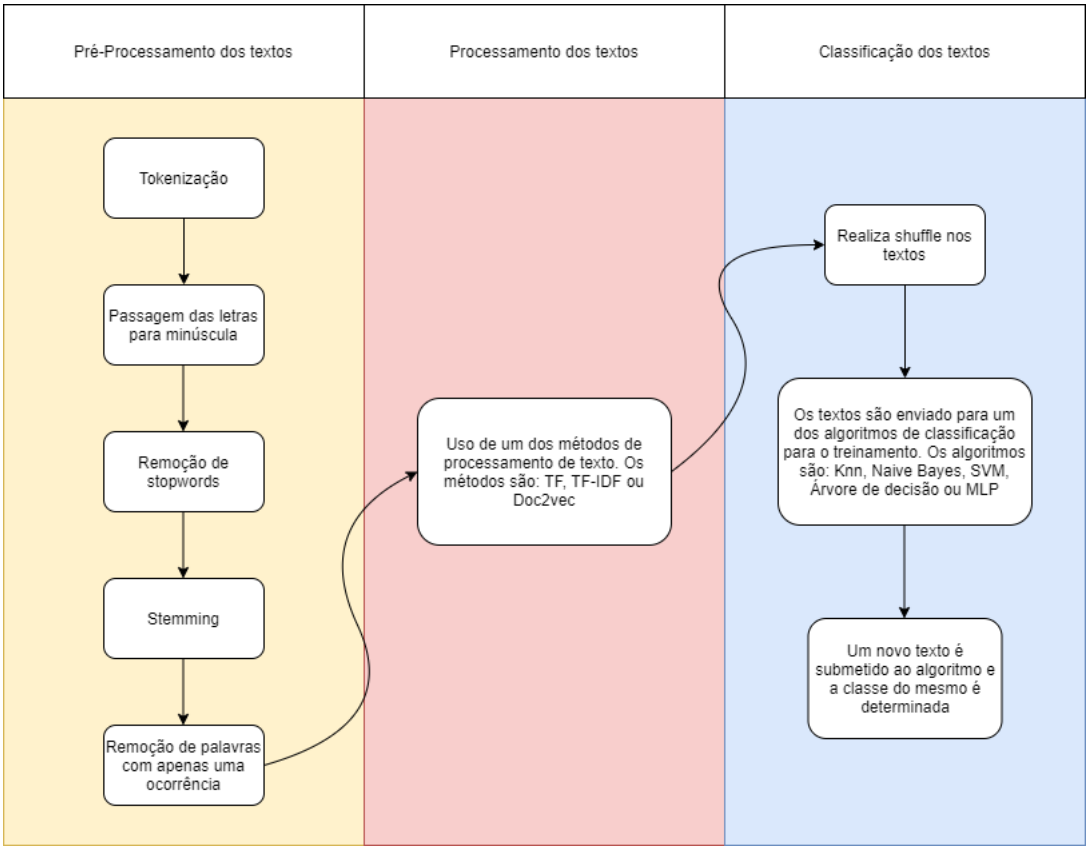
$$rec(h) = \frac{V_P}{V_P + F_N} \tag{2.22}$$

$$f1(h) = \frac{2 \times (prec(h) \times rec(h))}{(prec(h) + rec(h))} \tag{2.23}$$

3 Materiais e métodos

A Figura 9 apresenta as etapas empregadas no trabalho. Para o pré-processamento dos textos os seguintes passos foram efetuados: tokenização, passagem para minúscula, remoção de símbolos e pontuação, *stemming* (método que remove os afixos morfológicos das palavras, deixando apenas a raiz da palavra) e remoção de palavras com apenas uma ocorrência. Para gerar os vetores de atributos foram empregadas as técnicas TF, TF-IDF e Doc2Vec. Para a classificação os seguintes algoritmos foram testados: K-Nearest Neighbors (KNN), Naive Bayes, Support-vector machine (SVM), Decision Tree, Multilayer perceptron (MLP).

Figura 9 – Diagrama representando os passos a serem seguidos para a classificação de textos.



Fonte: O Autor

3.1 Bibliotecas utilizadas

Para realizar a análise foi utilizado a linguagem de programação Python, junto com as bibliotecas NLTK, Scikit-learn, Gensim, Pandas, Matplotlib e Numpy.

- NLTK: o Natural Language Toolkit, ou mais comumente o NLTK, é um conjunto de bibliotecas e programas para processamento simbólico e estatístico da linguagem natural para inglês.

- Scikit-learn: é uma biblioteca de aprendizado de máquina de código aberto com algoritmos para o pré-processamento, classificação, regressão e agrupamento de dados.

- Gensim: é uma biblioteca de código aberto para modelagem de tópicos não supervisionados e processamento de linguagem natural, usando o moderno aprendizado de máquina estatística.

- Pandas: é uma biblioteca criada para manipulação e análise de dados. Em particular, oferece estruturas e operações para manipular tabelas numéricas e séries temporais. É software livre sob a licença BSD.

- Numpy: é um pacote que suporta arrays e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas.

- Matplotlib: é uma biblioteca de plotagem para a linguagem de programação Python e sua extensão matemática numérica NumPy.

3.2 Datasets

Foi utilizado os *datasets* Brown e Reuters, disponíveis na biblioteca NLTK.

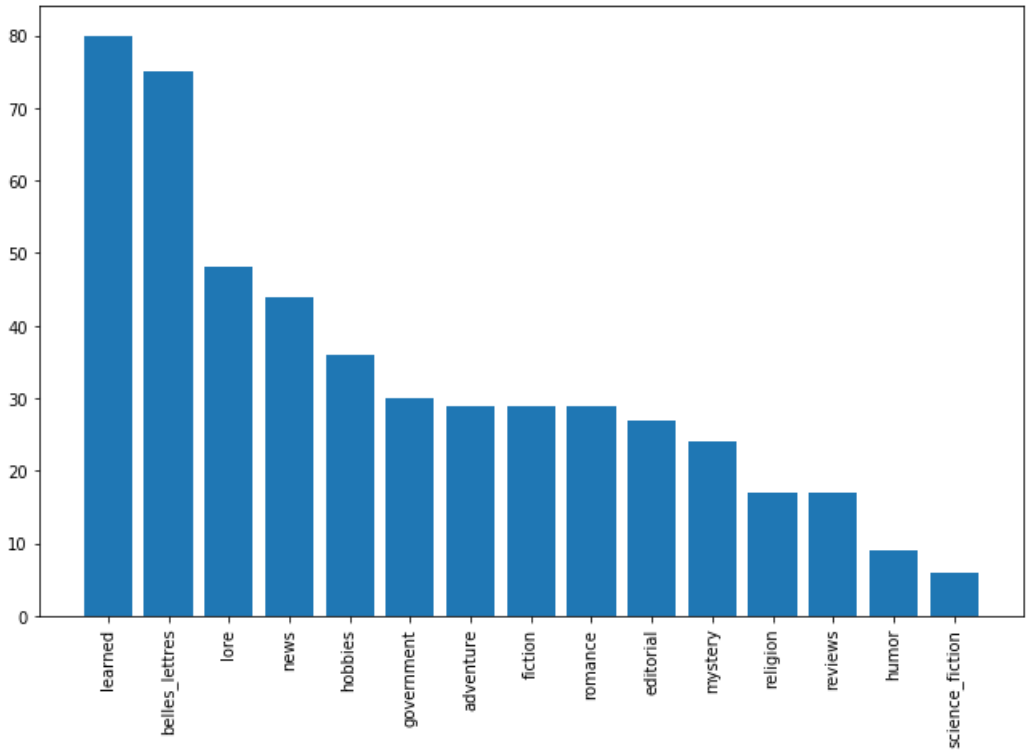
- Brown Corpus: foi o primeiro corpus eletrônico de um milhão de palavras em inglês, criado em 1961 na Brown University. Esse corpus contém texto de 500 fontes e as fontes foram categorizadas por gênero, como notícias, editorial e assim por diante. Existem 15 gêneros (classes) dentro do *dataset* Brown. Abaixo a tabela 3 e a figura 10 mostram a quantidade de textos em cada classe.

Tabela 3 – Quantidade de textos no *dataset* Brown

Classes	**Quantidade de textos**
learned	80
belles lettres	75
lore	48
news	44
hobbies	36
government	30
adventure	29
fiction	29
romance	29
editorial	27
mystery	24
religion	17
reviews	17
humor	9
science fiction	6

Fonte: O Autor

Figura 10 – Divisão de gêneros do *dataset* Brown



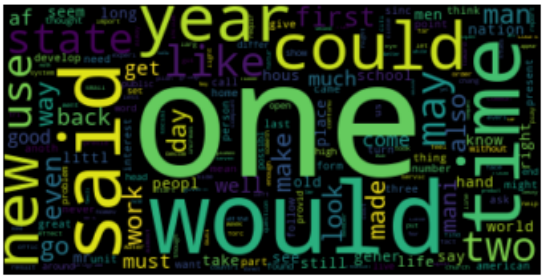
Fonte: O Autor

Em seguida estão as figuras 11, 12 e 13 que representam a quantidade de palavras em forma de nuvens de palavras no *dataset*, na classe “learned” e na classe “belles lettres” respectivamente.

Figura 11 – Nuvem de palavras do *dataset* Brown

Fonte: O Autor

Figura 12 – Nuvem de palavras da classe *learned*



Fonte: O Autor

Figura 13 – Nuvem de palavras da classe *belles lettres*

Fonte: O Autor

As figuras 12 e 13 mostram que muitas palavras estão presente em ambas as classes. Essa repetição de palavras não ocorre só nas duas classes, ela também ocorre em todas as classes. E esse fato de repetição de palavras faz com que a acurácia fique baixa.

- Reuters Corpus: contém 10.788 textos de notícias, totalizando 1,3 milhão de palavras. Os documentos foram classificados em 90 classes e agrupados em dois conjuntos, chamados “training” e “test”. O conjunto “training” possui 7769 textos e o conjunto

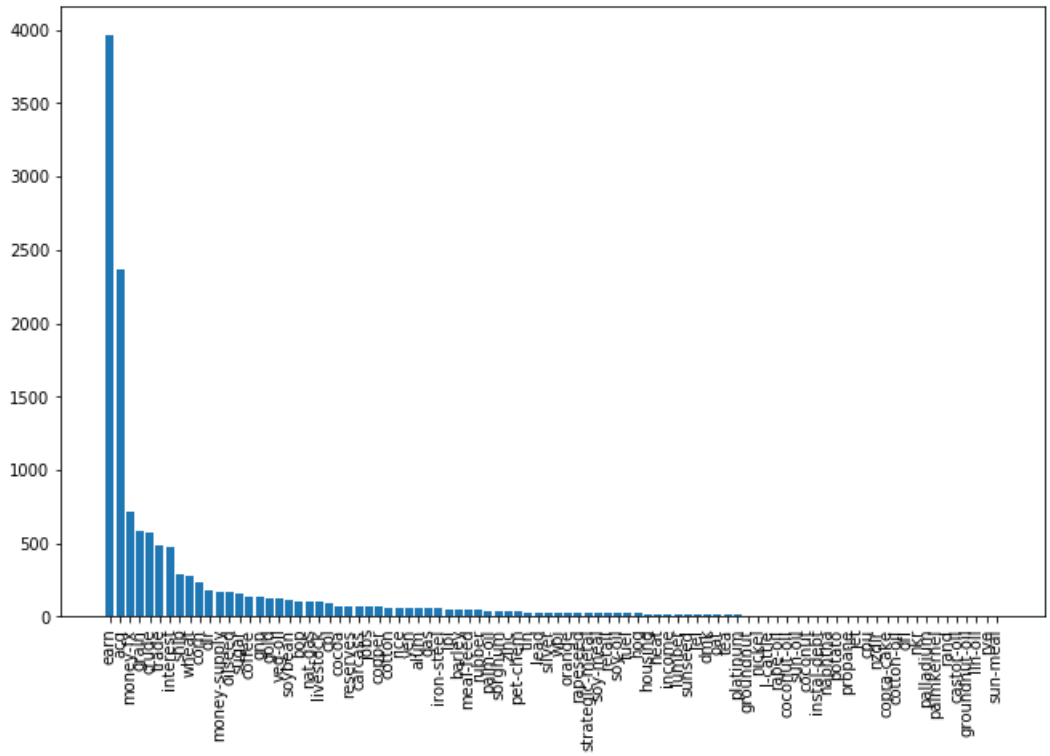
"test" possui 3019 textos. Foi utilizado apenas o conjunto "training" para o treinamento. Abaixo a tabela 4 e a figura 14 mostram a quantidade de textos em cada classe.

Tabela 4 – Quantidade de textos no *dataset* Reuters

Classes	**Quantidade de textos**	**Classes**	**Quantidade de textos**
earn	3964	wpi	29
acq	2369	orange	27
money-fx	717	rapeseed	27
grain	582	strategic-metal	27
crude	578	soy-meal	26
trade	485	retail	25
interest	478	soy-oil	25
ship	286	fuel	23
wheat	283	hog	22
corn	237	housing	20
dlr	175	heat	19
money-supply	174	income	16
oilseed	171	lumber	16
sugar	162	sunseed	16
coffee	139	lei	15
gnp	136	dmk	14
gold	124	oat	14
veg-oil	124	tea	13
soybean	111	platinum	12
bop	105	groundnut	9
nat-gas	105	nickel	9
livestock	99	l-cattle	8
cpi	97	rape-oil	8
cocoa	73	coconut-oil	7
reserves	73	sun-oil	7
carcass	68	coconut	6
jobs	67	instal-debt	6
copper	65	naphtha	6
cotton	59	potato	6
rice	59	propane	6
yen	59	jet	5
alum	58	cpu	4
gas	54	nzdlr	4
iron-steel	54	copra-cake	3
ipi	53	cotton-oil	3
barley	51	dfl	3
meal-feed	49	nkr	3
rubber	49	palladium	3
palm-oil	40	palmkernel	3
sorghum	34	rand	3
zinc	34	castor-oil	2
pet-chem	32	groundnut-oil	2
tin	30	lin-oil	2
lead	29	rye	2
silver	29	sun-meal	2

Fonte: O Autor

Figura 14 – Divisão de tópicos do *dataset* Reuters



Fonte: O Autor

Em seguida estão as figuras 15, 16 e 17 que representam a quantidade de palavras em forma de nuvens de palavras no *dataset*, na classe “earn” e na classe “acq” respectivamente.

Figura 15 – Nuvem de palavras do *dataset* Reuters

Fonte: O Autor

Figura 16 – Nuvem de palavras da classe *Earn*

Fonte: O Autor

Figura 17 – Nuvem de palavras da classe *Acq*

Fonte: O Autor

As figuras 16 e 17 mostram que muitas palavras estão presente em ambas as classes. Novamente essa repetição faz com que a acurácia abaixe nos resultados finais.

4 Resultados

4.1 Configuração dos experimentos

Foi empregado o método de amostragem *10-fold cross validation* em todos os *datasets*.

Foram considerados os seguintes parâmetros nos algoritmos:

- KNN: K = 28; Função de distância: Euclidiana;
- Naive Bayes: alpha = 0.05; Método: Bernoulli Naive Bayes;
- SVM: Kernel: Sigmoid; Gamma: Scale;
- MLP: 10 camadas, 30 neurônios em cada camada; Learning rate = Constant;
- Decision tree: Criterion: gini; Min_samples_split = 2;

4.2 Resultados *dataset* Brown

Os resultados para o *dataset* Brown são apresentados nas tabelas 5, 6 e 7. Em todos os casos o algoritmo Naive Bayes obteve os melhores resultados, e com o descritor word2vec o algoritmo KNN também se destacou. O melhor resultado geral, foi obtido pelo descritor TF e classificador Naive Bayes, com acurácia de 0,52 e f1-score de 0,44.

Tabela 5 – Resultados do *dataset* Brown usando método TF

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	0.07133	0.03916	0.04185	0.05998
SVM	0.40199	0.27266	0.32017	0.30353
Naive Bayes	**0.52**	**0.44693**	**0.49632**	**0.48972**
Decision Tree	0.22599	0.15962	0.17440	0.17928
MLP	0.50600	0.43281	0.45905	0.47077

Fonte: O Autor

4.3 Resultados *dataset* Reuters

Os resultados para o *dataset* Reuters são apresentados nas tabelas 8, 9 e 10. Em todos os casos o algoritmo MLP obteve os melhores resultados, e com o descritor word2vec

Tabela 6 – Resultados do *dataset* Brown usando método TF-IDF

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	0.438	0.33031	0.34857	0.39177
SVM	0.416	0.28555	0.30937	0.33542
Naive Bayes	**0.51**	**0.42641**	**0.46691**	**0.46584**
Decision Tree	0.236	0.17474	0.19901	0.19133
MLP	0.478	0.37166	0.37166	0.39641

Fonte: O Autor

Tabela 7 – Resultados do *dataset* Brown usando método word2vec

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	**0.402**	**0.29985**	**0.35027**	**0.33198**
SVM	0.358	0.18768	0.21376	0.24117
Naive Bayes	**0.366**	**0.29599**	**0.32917**	**0.32969**
Decision Tree	0.281	0.19438	0.21344	0.21141
MLP	0.444	0.33514	0.36913	0.36846

Fonte: O Autor

o algoritmo KNN também se destacou. O melhor resultado geral, foi obtido pelo descritor word2vec e classificador KNN, com acurácia de 0,80 e f1-score de 0,33.

Em anexo está o *classification report* da biblioteca Scikitlearn com os resultados detalhados de cada um dos resultados. Esse método cria um relatório mostrando as principais métricas de classificação.

Tabela 8 – Resultados do *dataset* Reuters usando método TF

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	0.55874	0.21127	0.24463	0.21576
SVM	0.44198	0.04472	0.06375	0.04523
Naive Bayes	0.65776	0.24026	0.26767	0.24382
Decision Tree	0.55958	0.20235	0.22014	0.20579
MLP	**0.63219**	**0.27282**	**0.29794**	**0.27463**

Fonte: O Autor

Tabela 9 – Resultados do *dataset* Reuters usando método TF-IDF

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	0.63918	0.30760	0.329157	0.31766
SVM	0.30018	0.00727	0.00473	0.01577
Naive Bayes	0.61967	0.08884	0.12426	0.09617
Decision Tree	0.57940	0.22992	0.24574	0.23757
MLP	**0.65682**	**0.31762**	**0.33952**	**0.31916**

Fonte: O Autor

Tabela 10 – Resultados do *dataset* Reuters usando método word2vec

Algoritmos	Accuracy	F1 - Score	Precision	Recall
KNN	**0.809**	**0.33845**	**0.37119**	**0.34962**
SVM	0.744	0.17855	0.19488	0.19418
Naive Bayes	0.613	0.23456	0.23538	0.28406
Decision Tree	0.585	0.13691	0.14086	0.14459
MLP	**0.777**	**0.31128**	**0.32649**	**0.32869**

Fonte: O Autor

5 Considerações Finais

Os datasets Brown e Reuters são multiclasse e com classes desbalanceadas, o maior f1-score obtido foi 0,44 em Brown e 0,33 no Reuters, respectivamente. Nesse caso, é esperado que *datasets* multi-classe obtenham resultados menores por serem mais desafiadores. Objetivamos investigar qual descritor de texto fornecia o melhor resultado, considerando essa diversidade de *datasets*.

O melhor resultado para o *dataset* Brown foi obtido pelo método TF e o classificador Naive Bayes. O melhor resultado para o *dataset* Reuters foi obtido pelo método word2vec e o classificador KNN. O classificador MLP também obteve bons resultados para todos os descritores no Reuters.

Concluímos, que nem sempre o descritor word2vec obtém resultados superiores aos métodos tradicionais baseados em *bag of words*. Uma de suas vantagens é a redução no tamanho do vetor e com isso redução do uso de memória e tempo de processamento.

Como trabalho futuro outros métodos de embeddings podem ser testados, especialmente das novas gerações.

Referências

1 LUGER, G. F. *Inteligência Artificial*. 6a ed. ed. São Paulo, Brasil: Pearson, 2014. Citado na página 9.

2 RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. 3a ed. ed. Rio de Janeiro, Brasil: Elsevier, 2013. Citado na página 9.

3 JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*: An introduction to natural language processing, computational linguistics, and speech recognition. 3th edition. ed. [S.l.], 2019. Nenhuma citação no texto.

4 KHURANA, D. et al. *Natural Language Processing: State of The Art, Current Trends and Challenges*. Faridabad, India. 25 p. Citado na página 11.

5 PEREIRA, S. do L. *Processamento de Linguagem Natural*. São Paulo, Brasil. 9 p. Citado na página 11.

6 MATSUBARA, E. T.; MARTINS, C. A.; MONARD, M. C. *PreTexT: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words*. São Carlos, São Paulo, 2003. 57 p. Citado na página 13.

7 SOUZA, S. de. *Estudo de modelos de word embedding*. Medianeira, Paraná, 2016. 55 p. Citado na página 13.

8 SPIRLING, A.; RODRIGUEZ, P. L. *Word Embeddings*: What works, what doesn't, and how to tell the difference for applied research. New York, USA. 53 p. Nenhuma citação no texto.

9 CARVALHO, M. H. de. *Estudo Comparativo dos Métodos de Word Embedding na Análise de Sentimentos*. Recife, Pernambuco, 2018. 43 p. Nenhuma citação no texto.

10 MIKOLOV, T. et al. *Efficient Estimation of Word Representations in Vector Space*. [S.l.], 2013. 12 p. Disponível em: <https://arxiv.org/pdf/1301.3781.pdf>. Citado na página 14.

11 AGUIAR, E. M. de. *Aplicação do Word2vec e do Gradiente Descendente Estocástico em Tradução Automática*. [S.l.], 2016. 78 p. Nenhuma citação no texto.

12 MIKOLOV, T.; LE, Q. *Distributed Representations of Sentences and Documents*. [S.l.]. 9 p. Disponível em: <https://cs.stanford.edu/~quocle/paragraph_vector.pdf>. Nenhuma citação no texto.

13 MONARD, M. C.; BARANAUSKAS, J. A. *Conceitos sobre Aprendizado de Máquina*: Sistemas inteligentes fundamentos e aplicações. [S.l.], 2003. 18 p. Citado 2 vezes nas páginas 16 e 17.

14 GRUS, J. *Data Science do Zerp*: Primeiras regras com o python. 1a ed. ed. Rio de Janeiro, Brasil: Alta Books, 2016. Nenhuma citação no texto.

15 VAPNIK, V. N. *The nature of statistical learning theory*. [S.l.], 1995. Citado na página 20.

Apêndices

APÊNDICE A – Stop Words

Os *Stop Words* são: my, this, there, yourselves, our, on, other, no, why, only, you'd, after, mustn't, needn, those, needn't, shouldn't, so, themselves, his, couldn't, hers, is, couldn, ll, are, you're, its, to, under, haven, yours, being, having, mustn, as, weren't, don, all, each, hadn, over, by, should've, y, haven't, they, and, at, s, he, hasn, wasn't, she's, above, t, when, below, again, of, wouldn't, weren, aren, these, up, o, how, you've, while, itself, that, does, me, didn, hadn't, am, too, ourselves, myself, shan't, before, both, about, or, such, into, doing, isn, against, who, the, if, an, mightn't, ours, don't, didn't, further, doesn't, your, himself, because, do, then, won, had, ain, in, once, aren't, we, doesn, ma, you'll, through, from, hasn't, isn't, where, it, than, any, but, herself, mightn, out, during, a, shouldn, off, wasn, will, wouldn, nor, for, can, it's, was, which, until, some, yourself, theirs, did, ve, i, been, what, them, m, between, that'll, you, be, not, own, were, whom, here, very, re, has, should, shan, most, with, more, just, now, won't, same, down, d, have, she, her, few, their, him.

Anexos

ANEXO A – Classification report

A.1 Brown

Figura 18 – Dataset Brown usando o método TF e o algoritmo KNN

	precision	recall	f1-score	support
adventure	0.00	0.00	0.00	3
belles_lettres	0.29	0.86	0.43	7
editorial	0.21	0.75	0.33	4
fiction	0.00	0.00	0.00	2
government	0.00	0.00	0.00	3
humor	0.00	0.00	0.00	1
learned	0.00	0.00	0.00	13
lore	0.00	0.00	0.00	1
mystery	0.00	0.00	0.00	2
news	0.67	0.50	0.57	4
religion	0.00	0.00	0.00	3
reviews	1.00	0.50	0.67	2
romance	0.60	0.75	0.67	4
science_fiction	0.00	0.00	0.00	1
accuracy			0.30	50
macro avg	0.20	0.24	0.19	50
weighted avg	0.20	0.30	0.21	50

Fonte: O Autor

Figura 19 – Dataset Brown usando o método TF e o algoritmo SVM

	precision	recall	f1-score	support
adventure	1.00	0.33	0.50	3
belles_lettres	0.29	0.88	0.44	8
editorial	0.00	0.00	0.00	4
fiction	0.33	0.33	0.33	3
government	0.00	0.00	0.00	4
hobbies	0.67	0.50	0.57	4
learned	0.75	0.55	0.63	11
lore	0.00	0.00	0.00	2
news	1.00	0.83	0.91	6
religion	0.00	0.00	0.00	1
reviews	0.00	0.00	0.00	1
romance	0.75	1.00	0.86	3
accuracy			0.50	50
macro avg	0.40	0.37	0.35	50
weighted avg	0.51	0.50	0.47	50

Fonte: O Autor

Figura 20 – Dataset Brown usando o método TF e o algoritmo Naive Bayes

	precision	recall	f1-score	support
adventure	0.20	0.50	0.29	2
belles_lettres	0.25	0.80	0.38	5
editorial	0.67	0.67	0.67	3
fiction	1.00	0.40	0.57	5
government	1.00	0.43	0.60	7
hobbies	1.00	0.50	0.67	2
humor	0.00	0.00	0.00	4
learned	0.33	0.12	0.18	8
lore	0.11	0.33	0.17	3
mystery	1.00	0.50	0.67	2
news	1.00	1.00	1.00	2
religion	0.00	0.00	0.00	3
reviews	0.00	0.00	0.00	1
romance	0.67	0.67	0.67	3
accuracy			0.40	50
macro avg	0.52	0.42	0.42	50
weighted avg	0.53	0.40	0.40	50

Fonte: O Autor

Figura 21 – Dataset Brown usando o método TF e o algoritmo Decision Tree

	precision	recall	f1-score	support
belles_lettres	0.15	0.40	0.22	5
editorial	0.25	0.25	0.25	4
fiction	0.00	0.00	0.00	0
government	0.50	0.20	0.29	5
hobbies	0.00	0.00	0.00	4
humor	0.00	0.00	0.00	1
learned	0.50	0.22	0.31	9
lore	0.25	0.40	0.31	5
mystery	0.00	0.00	0.00	4
news	0.75	0.38	0.50	8
religion	0.00	0.00	0.00	1
reviews	0.00	0.00	0.00	3
romance	0.33	1.00	0.50	1
accuracy			0.24	50
macro avg	0.21	0.22	0.18	50
weighted avg	0.33	0.24	0.25	50

Fonte: O Autor

Figura 22 – Dataset Brown usando o método TF e o algoritmo MLP

	precision	recall	f1-score	support
adventure	0.50	0.50	0.50	2
belles_lettres	0.00	0.00	0.00	4
editorial	0.33	0.20	0.25	5
fiction	0.29	0.50	0.36	4
government	0.75	1.00	0.86	3
hobbies	0.50	0.33	0.40	3
humor	0.00	0.00	0.00	1
learned	0.60	0.75	0.67	8
lore	0.25	0.25	0.25	4
mystery	0.00	0.00	0.00	2
news	0.80	0.80	0.80	5
religion	1.00	0.50	0.67	2
reviews	1.00	0.50	0.67	2
romance	0.40	0.50	0.44	4
science_fiction	0.00	0.00	0.00	1
accuracy			0.46	50
macro avg	0.43	0.39	0.39	50
weighted avg	0.46	0.46	0.45	50

Fonte: O Autor

Figura 23 – Dataset Brown usando o método TF-IDF e o algoritmo KNN

	precision	recall	f1-score	support
adventure	0.00	0.00	0.00	6
belles_lettres	0.58	0.64	0.61	11
editorial	0.00	0.00	0.00	0
fiction	0.00	0.00	0.00	3
government	0.50	1.00	0.67	1
hobbies	0.00	0.00	0.00	2
humor	0.00	0.00	0.00	1
learned	0.80	0.80	0.80	5
lore	0.00	0.00	0.00	6
mystery	0.00	0.00	0.00	4
news	0.83	0.83	0.83	6
religion	0.00	0.00	0.00	1
romance	0.15	1.00	0.26	3
science_fiction	0.00	0.00	0.00	1
accuracy			0.40	50
macro avg	0.20	0.30	0.23	50
weighted avg	0.33	0.40	0.34	50

Fonte: O Autor

Figura 24 – Dataset Brown usando o método TF-IDF e o algoritmo SVM

	precision	recall	f1-score	support
adventure	0.00	0.00	0.00	4
belles_lettres	0.33	0.75	0.46	8
editorial	0.50	1.00	0.67	1
fiction	0.00	0.00	0.00	1
government	1.00	0.40	0.57	5
hobbies	0.00	0.00	0.00	7
humor	0.00	0.00	0.00	1
learned	0.33	0.83	0.48	6
lore	0.00	0.00	0.00	3
mystery	0.00	0.00	0.00	2
news	1.00	0.75	0.86	4
religion	1.00	0.25	0.40	4
reviews	1.00	1.00	1.00	1
romance	0.00	0.00	0.00	3
accuracy			0.38	50
macro avg	0.37	0.36	0.32	50
weighted avg	0.38	0.38	0.32	50

Fonte: O Autor

Figura 25 – Dataset Brown usando o método TF-IDF e o algoritmo Naive Bayes

	precision	recall	f1-score	support
adventure	0.33	1.00	0.50	1
belles_lettres	0.30	0.33	0.32	9
editorial	0.00	0.00	0.00	2
fiction	0.00	0.00	0.00	1
government	0.60	0.75	0.67	4
hobbies	0.67	0.50	0.57	4
humor	0.00	0.00	0.00	1
learned	0.75	0.50	0.60	12
lore	0.14	0.25	0.18	4
mystery	1.00	1.00	1.00	1
news	0.86	0.75	0.80	8
reviews	0.00	0.00	0.00	1
romance	0.50	0.50	0.50	2
accuracy			0.48	50
macro avg	0.40	0.43	0.40	50
weighted avg	0.53	0.48	0.49	50

Fonte: O Autor

Figura 26 – Dataset Brown usando o método TF-IDF e o algoritmo Decision Tree

	precision	recall	f1-score	support
adventure	0.00	0.00	0.00	1
belles_lettres	0.50	0.22	0.31	9
editorial	0.50	0.50	0.50	2
fiction	0.00	0.00	0.00	5
government	0.00	0.00	0.00	1
hobbies	0.20	0.14	0.17	7
humor	0.00	0.00	0.00	0
learned	0.44	0.36	0.40	11
lore	0.00	0.00	0.00	5
mystery	0.00	0.00	0.00	0
news	0.33	0.67	0.44	3
religion	0.00	0.00	0.00	0
reviews	0.33	1.00	0.50	1
romance	0.33	0.20	0.25	5
science_fiction	0.00	0.00	0.00	0
accuracy			0.24	50
macro avg	0.18	0.21	0.17	50
weighted avg	0.30	0.24	0.25	50

Fonte: O Autor

Figura 27 – Dataset Brown usando o método TF-IDF e o algoritmo MLP

	precision	recall	f1-score	support
adventure	1.00	0.25	0.40	4
belles_lettres	0.27	0.86	0.41	7
editorial	0.00	0.00	0.00	1
fiction	0.00	0.00	0.00	4
government	1.00	0.33	0.50	3
hobbies	1.00	0.12	0.22	8
learned	0.43	0.75	0.55	4
lore	0.00	0.00	0.00	6
mystery	0.75	0.75	0.75	4
news	0.67	1.00	0.80	4
religion	0.00	0.00	0.00	1
reviews	0.00	0.00	0.00	4
romance	0.00	0.00	0.00	0
accuracy			0.38	50
macro avg	0.39	0.31	0.28	50
weighted avg	0.49	0.38	0.32	50

Fonte: O Autor

Figura 28 – Dataset Brown usando o método doc2vec e o algoritmo KNN

	precision	recall	f1-score	support
adventure	0.10	1.00	0.18	1
belles_lettres	0.31	0.71	0.43	7
editorial	0.00	0.00	0.00	3
fiction	1.00	0.50	0.67	4
government	0.00	0.00	0.00	3
hobbies	0.33	1.00	0.50	1
humor	0.00	0.00	0.00	1
learned	0.50	0.43	0.46	7
lore	1.00	0.17	0.29	6
mystery	1.00	1.00	1.00	2
news	1.00	0.60	0.75	5
religion	0.00	0.00	0.00	2
reviews	1.00	0.50	0.67	2
romance	0.00	0.00	0.00	4
science_fiction	0.00	0.00	0.00	2
accuracy			0.38	50
macro avg	0.42	0.39	0.33	50
weighted avg	0.50	0.38	0.37	50

Fonte: O Autor

Figura 29 – Dataset Brown usando o método doc2vec e o algoritmo SVM

	precision	recall	f1-score	support
adventure	0.00	0.00	0.00	2
belles_lettres	0.14	1.00	0.24	3
editorial	0.00	0.00	0.00	1
fiction	0.00	0.00	0.00	4
government	1.00	0.50	0.67	2
hobbies	1.00	0.20	0.33	5
learned	0.69	0.75	0.72	12
lore	0.17	0.50	0.25	2
mystery	0.00	0.00	0.00	3
news	0.67	0.50	0.57	4
religion	0.00	0.00	0.00	4
reviews	0.00	0.00	0.00	2
romance	0.50	0.67	0.57	3
science_fiction	0.00	0.00	0.00	3
accuracy			0.38	50
macro avg	0.30	0.29	0.24	50
weighted avg	0.40	0.38	0.34	50

Fonte: O Autor

Figura 30 – Dataset Brown usando o método doc2vec e o algoritmo Naive Bayes

	precision	recall	f1-score	support
adventure	0.50	0.50	0.50	6
belles_lettres	0.43	0.38	0.40	8
editorial	1.00	0.67	0.80	3
fiction	0.00	0.00	0.00	2
government	0.20	1.00	0.33	1
hobbies	0.33	0.25	0.29	4
learned	0.71	0.62	0.67	8
lore	0.00	0.00	0.00	2
mystery	0.50	0.75	0.60	4
news	0.67	0.67	0.67	6
religion	0.00	0.00	0.00	1
reviews	1.00	0.50	0.67	2
romance	0.00	0.00	0.00	2
science_fiction	0.00	0.00	0.00	1
accuracy			0.46	50
macro avg	0.38	0.38	0.35	50
weighted avg	0.49	0.46	0.46	50

Fonte: O Autor

Figura 31 – Dataset Brown usando o método doc2vec e o algoritmo Decision Tree

	precision	recall	f1-score	support
adventure	0.50	0.17	0.25	6
belles_lettres	0.50	0.38	0.43	8
editorial	0.33	0.33	0.33	3
fiction	0.00	0.00	0.00	2
government	0.20	1.00	0.33	1
hobbies	0.17	0.25	0.20	4
humor	0.00	0.00	0.00	0
learned	1.00	0.38	0.55	8
lore	0.00	0.00	0.00	2
mystery	0.00	0.00	0.00	4
news	0.50	0.50	0.50	6
religion	0.00	0.00	0.00	1
reviews	1.00	0.50	0.67	2
romance	0.00	0.00	0.00	2
science_fiction	0.00	0.00	0.00	1
accuracy			0.28	50
macro avg	0.28	0.23	0.22	50
weighted avg	0.44	0.28	0.32	50

Fonte: O Autor

Figura 32 – Dataset Brown usando o método doc2vec e o algoritmo MLP

	precision	recall	f1-score	support
adventure	1.00	0.50	0.67	2
belles_lettres	0.67	1.00	0.80	4
editorial	1.00	0.33	0.50	3
fiction	0.00	0.00	0.00	2
government	0.00	0.00	0.00	2
hobbies	0.22	0.67	0.33	3
humor	0.00	0.00	0.00	1
learned	0.75	0.43	0.55	14
lore	0.00	0.00	0.00	5
mystery	0.00	0.00	0.00	1
news	0.60	0.50	0.55	6
religion	0.00	0.00	0.00	1
reviews	0.00	0.00	0.00	2
romance	0.80	1.00	0.89	4
accuracy			0.42	50
macro avg	0.36	0.32	0.31	50
weighted avg	0.51	0.42	0.43	50

Fonte: O Autor

A.2 Reuters

Figura 33 – Dataset Reuters usando o método TF e o algoritmo KNN

	precision	recall	f1-score	support
acq	0.78	0.72	0.75	178
alum	0.17	0.20	0.18	5
barley	0.00	0.00	0.00	1
bop	0.29	0.25	0.27	8
carcass	0.20	0.33	0.25	3
cocoa	0.62	1.00	0.77	5
coconut-oil	0.00	0.00	0.00	1
coffee	0.54	0.64	0.58	11
copper	0.19	1.00	0.32	3
corn	0.12	0.21	0.16	14
cotton	0.00	0.00	0.00	8
cpi	0.30	0.67	0.41	9
crude	0.31	0.45	0.37	47
dlr	0.13	0.15	0.14	13
dmk	0.00	0.00	0.00	2
earn	0.92	0.90	0.91	287
gas	0.00	0.00	0.00	6
gnp	0.60	0.46	0.52	13
gold	0.33	0.40	0.36	5
grain	0.19	0.18	0.18	45
groundnut	0.00	0.00	0.00	1
groundnut-oil	0.00	0.00	0.00	0
heat	0.00	0.00	0.00	0
hog	0.00	0.00	0.00	0
housing	1.00	1.00	1.00	2
income	0.00	0.00	0.00	0
instal-debt	0.00	0.00	0.00	1
interest	0.35	0.48	0.41	29
ipi	0.67	0.50	0.57	4
iron-steel	0.00	0.00	0.00	1
jobs	1.00	1.00	1.00	3
lei	1.00	0.33	0.50	3
lin-oil	0.00	0.00	0.00	1
livestock	0.67	0.29	0.40	7
lumber	0.00	0.00	0.00	1
meal-feed	0.00	0.00	0.00	2
money-fx	0.38	0.35	0.37	57
money-supply	0.20	0.58	0.30	12
nat-gas	0.00	0.00	0.00	8
nickel	0.00	0.00	0.00	1
oat	0.00	0.00	0.00	1
oilseed	0.11	0.07	0.08	15
orange	0.00	0.00	0.00	1
palladium	0.00	0.00	0.00	1
palm-oil	0.00	0.00	0.00	4
pet-chem	0.00	0.00	0.00	2
platinum	0.00	0.00	0.00	0
potato	0.00	0.00	0.00	1
rape-oil	0.00	0.00	0.00	0
rapeseed	0.00	0.00	0.00	1
reserves	1.00	0.33	0.50	9
retail	1.00	0.50	0.67	2
rice	0.00	0.00	0.00	1
rubber	1.00	0.33	0.50	3
ship	0.32	0.35	0.33	20
silver	0.00	0.00	0.00	1
soy-meal	0.00	0.00	0.00	2
soybean	0.00	0.00	0.00	9
strategic-metal	1.00	0.50	0.67	2
sugar	0.71	0.50	0.59	10
sunseed	0.00	0.00	0.00	1
tin	1.00	1.00	1.00	3
trade	0.64	0.47	0.55	38
veg-oil	0.25	0.25	0.25	8
wheat	0.07	0.06	0.06	17
wpi	0.00	0.00	0.00	0
yen	0.00	0.00	0.00	6
zinc	0.00	0.00	0.00	3
accuracy			0.57	958
macro avg	0.27	0.24	0.23	958
weighted avg	0.59	0.57	0.57	958

Fonte: O Autor

Figura 34 – Dataset Reuters usando o método TF e o algoritmo SVM

	precision	recall	f1-score	support
acq	0.53	0.61	0.57	154
alum	0.00	0.00	0.00	4
barley	0.00	0.00	0.00	6
bop	0.00	0.00	0.00	7
carcass	0.00	0.00	0.00	4
cocoa	0.00	0.00	0.00	3
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.00	0.00	0.00	10
copper	0.00	0.00	0.00	8
corn	0.00	0.00	0.00	19
cotton	0.00	0.00	0.00	7
cpi	0.00	0.00	0.00	10
crude	0.44	0.30	0.36	40
dlr	0.00	0.00	0.00	12
dmk	0.00	0.00	0.00	1
earn	0.43	0.97	0.60	301
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	4
gnp	0.00	0.00	0.00	9
gold	0.00	0.00	0.00	11
grain	0.26	0.17	0.20	42
groundnut	0.00	0.00	0.00	1
heat	0.00	0.00	0.00	3
hog	0.00	0.00	0.00	2
housing	0.00	0.00	0.00	2
instal-debt	0.00	0.00	0.00	1
interest	0.50	0.04	0.07	25
ipi	0.00	0.00	0.00	2
iron-steel	0.00	0.00	0.00	5
jobs	0.00	0.00	0.00	4
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	2
livestock	0.00	0.00	0.00	4
lumber	0.00	0.00	0.00	1
meal-feed	0.00	0.00	0.00	2
money-fx	0.39	0.20	0.26	46
money-supply	0.00	0.00	0.00	14
naphtha	0.00	0.00	0.00	1
nat-gas	0.00	0.00	0.00	13
nickel	0.00	0.00	0.00	1
oilseed	0.00	0.00	0.00	8
orange	0.00	0.00	0.00	2
palladium	0.00	0.00	0.00	2
palm-oil	0.00	0.00	0.00	4
pet-chem	0.00	0.00	0.00	3
platinum	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	1
reserves	0.00	0.00	0.00	11
retail	0.00	0.00	0.00	1
rice	0.00	0.00	0.00	4
rubber	0.00	0.00	0.00	4
ship	0.00	0.00	0.00	26
silver	0.00	0.00	0.00	1
sorghum	0.00	0.00	0.00	3
soy-meal	0.00	0.00	0.00	1
soy-oil	0.00	0.00	0.00	2
soybean	0.00	0.00	0.00	9
strategic-metal	0.00	0.00	0.00	3
sugar	1.00	0.07	0.13	14
sunseed	0.00	0.00	0.00	1
tea	0.00	0.00	0.00	4
trade	0.63	0.34	0.44	35
veg-oil	0.00	0.00	0.00	5
wheat	0.00	0.00	0.00	18
wpi	0.00	0.00	0.00	4
yen	0.00	0.00	0.00	1
zinc	0.00	0.00	0.00	5
accuracy			0.45	958
macro avg	0.06	0.04	0.04	958
weighted avg	0.32	0.45	0.34	958

Fonte: O Autor

Figura 35 – Dataset Reuters usando o método TF e o algoritmo Naive Bayes

```
[9584 rows x 17936 columns]
                precision    recall  f1-score   support

           acq       0.87      0.87      0.87       185
          alum       0.00      0.00      0.00         2
        barley       0.00      0.00      0.00         3
           bop       0.38      0.27      0.32        11
       carcass       0.00      0.00      0.00         4
         cocoa       1.00      0.56      0.71         9
       coconut       0.00      0.00      0.00         2
        coffee       0.56      0.45      0.50        11
        copper       1.00      0.12      0.22         8
          corn       0.00      0.00      0.00        13
        cotton       0.14      0.50      0.22         2
           cpi       0.22      1.00      0.36         2
         crude       0.63      0.48      0.54        46
           dfl       0.00      0.00      0.00         1
           dlr       0.11      0.20      0.14        10
           dmk       0.00      0.00      0.00         1
          earn       0.76      0.90      0.83       290
          fuel       0.00      0.00      0.00         2
           gas       0.00      0.00      0.00         4
           gnp       0.21      0.33      0.26         9
          gold       0.73      0.67      0.70        12
         grain       0.06      0.06      0.06        36
     groundnut       0.00      0.00      0.00         2
          heat       0.00      0.00      0.00         2
           hog       0.00      0.00      0.00         2
       housing       0.00      0.00      0.00         1
        income       0.00      0.00      0.00         1
      interest       0.41      0.41      0.41        29
           ipi       0.33      0.25      0.29         4
    iron-steel       0.40      0.67      0.50         3
          jobs       0.67      1.00      0.80         2
     l-cattle       0.00      0.00      0.00         1
          lead       0.00      0.00      0.00         1
     livestock       0.14      0.14      0.14         7
        lumber       0.00      0.00      0.00         2
     meal-feed       0.00      0.00      0.00         1
      money-fx       0.32      0.14      0.20        42
  money-supply       0.53      0.56      0.54        18
       nat-gas       0.08      0.14      0.10         7
        nickel       0.00      0.00      0.00         1
           oat       0.00      0.00      0.00         1
       oilseed       0.00      0.00      0.00        10
      palm-oil       0.00      0.00      0.00         4
      pet-chem       0.00      0.00      0.00         3
      platinum       0.00      0.00      0.00         0
          rand       0.00      0.00      0.00         1
     rapeseed       0.00      0.00      0.00         2
     reserves       0.33      0.33      0.33         3
       retail       0.00      0.00      0.00         2
         rice       0.00      0.00      0.00         2
       rubber       0.00      0.00      0.00         2
         ship       0.62      0.42      0.50        19
       silver       0.00      0.00      0.00         1
      sorghum       0.00      0.00      0.00         4
     soy-meal       0.00      0.00      0.00         2
      soy-oil       0.00      0.00      0.00         2
      soybean       0.00      0.00      0.00        10
strategic-metal       0.00      0.00      0.00         1
        sugar       0.58      0.64      0.61        11
      sunseed       0.00      0.00      0.00         0
          tea       0.00      0.00      0.00         2
          tin       0.00      0.00      0.00         2
        trade       0.63      0.52      0.57        42
      veg-oil       0.40      0.31      0.35        13
        wheat       0.11      0.14      0.12        21
          wpi       0.00      0.00      0.00         2
          yen       0.00      0.00      0.00         6
         zinc       0.00      0.00      0.00         1

     accuracy                           0.58       958
    macro avg       0.18      0.18      0.16       958
 weighted avg       0.57      0.58      0.57       958
```

Fonte: O Autor

Figura 36 – Dataset Reuters usando o método TF e o algoritmo Decision Tree

	precision	recall	f1-score	support
acq	0.76	0.83	0.79	173
alum	0.33	0.50	0.40	2
barley	0.00	0.00	0.00	4
bop	0.00	0.00	0.00	7
carcass	0.00	0.00	0.00	3
cocoa	0.50	0.38	0.43	8
coconut	0.00	0.00	0.00	0
coconut-oil	0.00	0.00	0.00	0
coffee	0.38	0.27	0.32	11
copper	0.33	0.33	0.33	9
corn	0.00	0.00	0.00	13
cotton	0.00	0.00	0.00	6
cpi	0.67	0.44	0.53	9
crude	0.41	0.46	0.43	39
dlr	0.00	0.00	0.00	14
dmk	0.00	0.00	0.00	1
earn	0.92	0.88	0.90	285
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	4
gnp	0.00	0.00	0.00	16
gold	0.67	0.33	0.44	6
grain	0.08	0.10	0.09	41
groundnut	0.00	0.00	0.00	2
heat	0.00	0.00	0.00	0
hog	0.00	0.00	0.00	1
housing	0.00	0.00	0.00	0
income	0.00	0.00	0.00	0
interest	0.30	0.41	0.34	32
ipi	1.00	0.25	0.40	4
iron-steel	0.00	0.00	0.00	3
jobs	0.60	0.75	0.67	4
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	1
livestock	0.00	0.00	0.00	5
meal-feed	0.00	0.00	0.00	3
money-fx	0.25	0.20	0.22	65
money-supply	0.42	0.71	0.53	7
nat-gas	0.25	0.09	0.13	11
nickel	0.00	0.00	0.00	1
oat	0.00	0.00	0.00	2
oilseed	0.07	0.09	0.08	11
orange	0.00	0.00	0.00	3
palm-oil	0.00	0.00	0.00	3
platinum	0.00	0.00	0.00	2
rand	0.00	0.00	0.00	0
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	2
reserves	0.20	0.25	0.22	4
retail	0.50	1.00	0.67	2
rice	0.00	0.00	0.00	2
rubber	1.00	0.71	0.83	7
ship	0.33	0.40	0.36	10
silver	0.00	0.00	0.00	2
sorghum	0.00	0.00	0.00	3
soy-meal	0.00	0.00	0.00	1
soy-oil	0.00	0.00	0.00	1
soybean	0.00	0.00	0.00	7
strategic-metal	0.00	0.00	0.00	1
sugar	0.78	0.64	0.70	11
sun-oil	0.00	0.00	0.00	1
tea	0.00	0.00	0.00	1
tin	1.00	1.00	1.00	2
trade	0.47	0.34	0.39	41
veg-oil	0.20	0.11	0.14	9
wheat	0.00	0.00	0.00	25
wpi	1.00	0.50	0.67	2
yen	0.00	0.00	0.00	6
zinc	1.00	0.25	0.40	4
accuracy			0.53	958
macro avg	0.21	0.18	0.18	958
weighted avg	0.55	0.53	0.53	958

Fonte: O Autor

Figura 37 – Dataset Reuters usando o método TF e o algoritmo MLP

	precision	recall	f1-score	support
acq	0.89	0.93	0.91	159
alum	0.67	1.00	0.80	2
barley	0.17	0.50	0.25	2
bop	0.50	0.33	0.40	9
carcass	0.25	0.50	0.33	2
cocoa	1.00	0.80	0.89	5
coconut-oil	0.00	0.00	0.00	1
coffee	0.70	0.64	0.67	11
copper	0.25	0.25	0.25	4
corn	0.00	0.00	0.00	23
cotton	0.00	0.00	0.00	2
cpi	0.50	0.50	0.50	4
cpu	0.00	0.00	0.00	1
crude	0.61	0.57	0.59	44
dlr	0.00	0.00	0.00	19
dmk	0.00	0.00	0.00	2
earn	0.98	0.97	0.97	315
fuel	1.00	1.00	1.00	1
gas	0.33	0.33	0.33	6
gnp	1.00	0.33	0.50	9
gold	0.50	0.67	0.57	6
grain	0.20	0.21	0.20	43
groundnut	0.00	0.00	0.00	0
heat	0.00	0.00	0.00	0
hog	0.00	0.00	0.00	2
housing	0.00	0.00	0.00	0
income	0.00	0.00	0.00	1
interest	0.39	0.50	0.44	30
ipi	0.67	0.80	0.73	5
iron-steel	0.80	1.00	0.89	4
jobs	0.75	0.86	0.80	7
l-cattle	0.00	0.00	0.00	0
lead	0.00	0.00	0.00	0
lei	0.67	1.00	0.80	2
livestock	0.00	0.00	0.00	5
meal-feed	0.25	0.25	0.25	4
money-fx	0.21	0.25	0.23	48
money-supply	0.67	0.73	0.70	11
nat-gas	0.00	0.00	0.00	8
nickel	0.00	0.00	0.00	3
nkr	0.00	0.00	0.00	0
oat	0.00	0.00	0.00	2
oilseed	0.05	0.08	0.06	13
orange	0.00	0.00	0.00	2
palladium	0.00	0.00	0.00	0
palm-oil	0.00	0.00	0.00	2
pet-chem	0.00	0.00	0.00	1
platinum	0.00	0.00	0.00	2
rand	0.00	0.00	0.00	1
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	3
reserves	0.80	0.57	0.67	7
retail	1.00	0.20	0.33	5
rice	0.00	0.00	0.00	3
rubber	0.40	1.00	0.57	2
ship	0.40	0.25	0.31	16
silver	0.00	0.00	0.00	2
sorghum	0.00	0.00	0.00	2
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	2
soybean	0.00	0.00	0.00	9
sugar	0.90	0.69	0.78	13
sun-oil	0.00	0.00	0.00	1
sunseed	0.00	0.00	0.00	1
trade	0.58	0.63	0.60	30
veg-oil	0.09	0.17	0.12	6
wheat	0.00	0.00	0.00	21
wpi	0.75	1.00	0.86	3
yen	0.25	0.20	0.22	5
zinc	0.00	0.00	0.00	1
accuracy			0.64	958
macro avg	0.27	0.28	0.26	958
weighted avg	0.64	0.64	0.64	958

Fonte: O Autor

Figura 38 – Dataset Reuters usando o método TF-IDF e o algoritmo KNN

	precision	recall	f1-score	support
acq	0.91	0.81	0.85	178
alum	1.00	1.00	1.00	1
barley	0.00	0.00	0.00	4
bop	0.22	0.29	0.25	7
carcass	0.17	0.25	0.20	4
cocoa	0.50	1.00	0.67	4
coconut	0.00	0.00	0.00	1
coffee	0.53	0.82	0.64	11
copper	0.40	1.00	0.57	2
corn	0.14	0.18	0.15	17
cotton	0.33	0.50	0.40	2
cpi	0.56	0.71	0.63	7
crude	0.55	0.69	0.61	39
dfl	0.00	0.00	0.00	1
dlr	0.16	0.31	0.21	13
earn	0.90	0.97	0.94	309
fuel	1.00	0.25	0.40	4
gas	0.00	0.00	0.00	5
gnp	0.00	0.00	0.00	1
gold	0.50	1.00	0.67	7
grain	0.04	0.08	0.06	24
groundnut	0.00	0.00	0.00	1
heat	0.00	0.00	0.00	1
hog	0.00	0.00	0.00	2
housing	0.50	1.00	0.67	1
instal-debt	1.00	1.00	1.00	1
interest	0.68	0.50	0.58	30
ipi	1.00	1.00	1.00	6
iron-steel	0.33	0.33	0.33	3
jobs	1.00	0.80	0.89	5
l-cattle	0.00	0.00	0.00	0
lead	0.00	0.00	0.00	3
lei	1.00	1.00	1.00	1
livestock	0.40	0.25	0.31	8
meal-feed	0.00	0.00	0.00	4
money-fx	0.53	0.52	0.53	63
money-supply	0.89	0.89	0.89	18
naphtha	0.00	0.00	0.00	1
nat-gas	0.00	0.00	0.00	7
nzdlr	0.00	0.00	0.00	1
oat	0.00	0.00	0.00	2
oilseed	0.18	0.17	0.17	12
orange	1.00	1.00	1.00	1
palm-oil	0.00	0.00	0.00	1
pet-chem	1.00	0.25	0.40	4
platinum	0.00	0.00	0.00	2
rape-oil	0.00	0.00	0.00	2
rapeseed	0.00	0.00	0.00	2
reserves	0.60	0.50	0.55	6
retail	1.00	0.67	0.80	3
rice	0.00	0.00	0.00	4
rubber	1.00	0.50	0.67	6
ship	0.55	0.43	0.48	14
silver	0.00	0.00	0.00	1
sorghum	0.00	0.00	0.00	3
soy-meal	0.00	0.00	0.00	1
soy-oil	0.00	0.00	0.00	2
soybean	0.00	0.00	0.00	6
strategic-metal	0.00	0.00	0.00	1
sugar	1.00	0.53	0.70	15
sunseed	0.00	0.00	0.00	1
tea	0.00	0.00	0.00	1
tin	1.00	1.00	1.00	1
trade	0.62	0.58	0.60	31
veg-oil	0.25	0.14	0.18	7
wheat	0.07	0.06	0.06	17
wpi	1.00	0.50	0.67	2
yen	0.00	0.00	0.00	10
zinc	0.00	0.00	0.00	3
accuracy			0.67	958
macro avg	0.36	0.34	0.33	958
weighted avg	0.67	0.67	0.66	958

Fonte: O Autor

Figura 39 – Dataset Reuters usando o método TF-IDF e o algoritmo SVM

	precision	recall	f1-score	support
acq	0.00	0.00	0.00	182
alum	0.00	0.00	0.00	2
barley	0.00	0.00	0.00	2
bop	0.00	0.00	0.00	12
carcass	0.00	0.00	0.00	4
cocoa	0.00	0.00	0.00	6
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	2
coffee	0.00	0.00	0.00	8
copper	0.00	0.00	0.00	8
corn	0.00	0.00	0.00	18
cotton	0.00	0.00	0.00	2
cpi	0.00	0.00	0.00	7
cpu	0.00	0.00	0.00	1
crude	0.00	0.00	0.00	44
dlr	0.00	0.00	0.00	7
earn	0.29	1.00	0.45	280
fuel	0.00	0.00	0.00	2
gas	0.00	0.00	0.00	1
gnp	0.00	0.00	0.00	11
gold	0.00	0.00	0.00	5
grain	0.00	0.00	0.00	37
groundnut	0.00	0.00	0.00	1
heat	0.00	0.00	0.00	3
housing	0.00	0.00	0.00	1
interest	0.00	0.00	0.00	40
ipi	0.00	0.00	0.00	6
iron-steel	0.00	0.00	0.00	8
jobs	0.00	0.00	0.00	6
lead	0.00	0.00	0.00	4
livestock	0.00	0.00	0.00	10
meal-feed	0.00	0.00	0.00	3
money-fx	0.00	0.00	0.00	47
money-supply	0.00	0.00	0.00	11
nat-gas	0.00	0.00	0.00	6
nickel	0.00	0.00	0.00	2
nkr	0.00	0.00	0.00	1
oat	0.00	0.00	0.00	1
oilseed	0.00	0.00	0.00	16
orange	0.00	0.00	0.00	2
palm-oil	0.00	0.00	0.00	2
palmkernel	0.00	0.00	0.00	1
platinum	0.00	0.00	0.00	2
potato	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	1
reserves	0.00	0.00	0.00	7
retail	0.00	0.00	0.00	2
rice	0.00	0.00	0.00	4
rubber	0.00	0.00	0.00	3
ship	0.00	0.00	0.00	14
silver	0.00	0.00	0.00	1
sorghum	0.00	0.00	0.00	3
soy-oil	0.00	0.00	0.00	3
soybean	0.00	0.00	0.00	4
strategic-metal	0.00	0.00	0.00	1
sugar	0.00	0.00	0.00	6
sunseed	0.00	0.00	0.00	2
tea	0.00	0.00	0.00	2
tin	0.00	0.00	0.00	2
trade	0.00	0.00	0.00	44
veg-oil	0.00	0.00	0.00	10
wheat	0.00	0.00	0.00	27
wpi	0.00	0.00	0.00	1
yen	0.00	0.00	0.00	4
zinc	0.00	0.00	0.00	1
accuracy			0.29	958
macro avg	0.00	0.02	0.01	958
weighted avg	0.09	0.29	0.13	958

Fonte: O Autor

Figura 40 – Dataset Reuters usando o método TF-IDF e o algoritmo Naive Bayes

	precision	recall	f1-score	support
acq	0.89	0.82	0.85	173
alum	1.00	0.33	0.50	3
barley	0.00	0.00	0.00	3
bop	0.50	0.75	0.60	8
carcass	0.22	0.33	0.27	6
cocoa	1.00	0.12	0.22	8
coconut	0.00	0.00	0.00	1
coffee	0.67	0.22	0.33	9
copper	0.25	0.25	0.25	4
corn	0.00	0.00	0.00	16
cotton	0.00	0.00	0.00	3
cpi	0.50	0.50	0.50	4
crude	0.50	0.51	0.51	39
dlr	0.19	0.20	0.19	15
dmk	0.00	0.00	0.00	1
earn	0.81	0.90	0.85	290
fuel	0.00	0.00	0.00	0
gas	0.33	0.67	0.44	3
gnp	0.30	0.60	0.40	10
gold	0.64	0.82	0.72	11
grain	0.15	0.29	0.19	31
groundnut	0.00	0.00	0.00	1
heat	0.00	0.00	0.00	1
hog	0.00	0.00	0.00	1
housing	1.00	0.33	0.50	3
instal-debt	0.00	0.00	0.00	1
interest	0.38	0.52	0.44	29
ipi	0.75	0.38	0.50	8
iron-steel	0.25	0.50	0.33	2
jobs	0.75	0.50	0.60	6
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	3
livestock	0.22	0.15	0.18	13
lumber	0.00	0.00	0.00	1
meal-feed	0.00	0.00	0.00	3
money-fx	0.48	0.28	0.36	53
money-supply	0.33	0.89	0.48	9
nat-gas	0.17	0.11	0.13	9
nickel	0.00	0.00	0.00	2
oilseed	0.10	0.10	0.10	10
orange	1.00	0.33	0.50	3
palm-oil	0.00	0.00	0.00	1
palmkernel	0.00	0.00	0.00	1
pet-chem	0.00	0.00	0.00	2
potato	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	2
reserves	0.67	0.33	0.44	6
retail	0.00	0.00	0.00	1
rice	0.00	0.00	0.00	1
rubber	1.00	0.33	0.50	3
ship	0.60	0.45	0.51	20
silver	0.00	0.00	0.00	4
sorghum	0.00	0.00	0.00	3
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	1
soybean	0.00	0.00	0.00	9
strategic-metal	0.00	0.00	0.00	2
sugar	0.46	0.50	0.48	12
tea	0.00	0.00	0.00	0
tin	0.00	0.00	0.00	0
trade	0.47	0.41	0.43	37
veg-oil	0.36	0.33	0.35	12
wheat	0.13	0.07	0.09	29
wpi	0.50	0.50	0.50	2
yen	0.50	0.14	0.22	7
zinc	0.00	0.00	0.00	3
accuracy			0.58	958
macro avg	0.27	0.22	0.22	958
weighted avg	0.59	0.58	0.57	958

Fonte: O Autor

Figura 41 – Dataset Reuters usando o método TF-IDF e o algoritmo Decision Tree

	precision	recall	f1-score	support
acq	0.80	0.82	0.81	156
alum	0.00	0.00	0.00	2
barley	0.00	0.00	0.00	2
bop	0.15	0.40	0.22	5
carcass	0.00	0.00	0.00	3
cocoa	0.75	0.60	0.67	5
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	0
coffee	0.61	0.85	0.71	13
copper	0.60	0.43	0.50	7
corn	0.00	0.00	0.00	23
cotton	0.00	0.00	0.00	6
cotton-oil	0.00	0.00	0.00	0
cpi	0.45	0.83	0.59	6
cpu	0.00	0.00	0.00	1
crude	0.44	0.50	0.47	34
dlr	0.00	0.00	0.00	12
dmk	0.00	0.00	0.00	1
earn	0.95	0.90	0.92	314
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	7
gnp	0.33	0.33	0.33	9
gold	0.47	0.67	0.55	12
grain	0.07	0.15	0.09	20
groundnut-oil	0.00	0.00	0.00	0
heat	0.00	0.00	0.00	2
hog	0.00	0.00	0.00	1
housing	1.00	1.00	1.00	1
income	0.00	0.00	0.00	1
interest	0.37	0.45	0.41	31
ipi	0.00	0.00	0.00	1
iron-steel	0.25	0.33	0.29	3
jobs	1.00	1.00	1.00	3
l-cattle	0.00	0.00	0.00	1
lead	0.00	0.00	0.00	0
lei	0.00	0.00	0.00	2
lin-oil	0.00	0.00	0.00	1
livestock	0.00	0.00	0.00	8
lumber	0.00	0.00	0.00	1
meal-feed	0.00	0.00	0.00	4
money-fx	0.23	0.17	0.19	53
money-supply	0.71	0.88	0.79	17
nat-gas	0.00	0.00	0.00	8
nickel	0.00	0.00	0.00	1
nzdlr	0.00	0.00	0.00	1
oilseed	0.13	0.15	0.14	13
orange	1.00	0.33	0.50	3
palm-oil	0.17	0.12	0.14	8
pet-chem	0.33	0.33	0.33	3
platinum	0.00	0.00	0.00	1
rand	0.00	0.00	0.00	1
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	2
reserves	0.80	0.80	0.80	5
retail	1.00	0.50	0.67	2
rice	0.00	0.00	0.00	2
rubber	0.75	1.00	0.86	3
ship	0.30	0.44	0.36	18
silver	0.00	0.00	0.00	2
sorghum	0.00	0.00	0.00	2
soy-meal	0.00	0.00	0.00	4
soy-oil	0.00	0.00	0.00	4
soybean	0.00	0.00	0.00	8
strategic-metal	0.00	0.00	0.00	1
sugar	0.75	0.82	0.78	11
sun-oil	0.00	0.00	0.00	1
sunseed	0.00	0.00	0.00	2
tea	0.00	0.00	0.00	1
tin	1.00	0.50	0.67	2
trade	0.55	0.38	0.45	42
veg-oil	0.00	0.00	0.00	12
wheat	0.00	0.00	0.00	16
wpi	0.50	0.50	0.50	2
yen	0.00	0.00	0.00	6
zinc	0.00	0.00	0.00	0
accuracy			0.58	958
macro avg	0.22	0.22	0.21	958
weighted avg	0.59	0.58	0.58	958

Fonte: O Autor

Figura 42 – Dataset Reuters usando o método TF-IDF e o algoritmo MLP

	precision	recall	f1-score	support
acq	0.90	0.96	0.93	159
alum	0.75	1.00	0.86	3
barley	0.33	0.25	0.29	4
bop	0.18	0.50	0.27	4
carcass	0.00	0.00	0.00	8
cocoa	1.00	0.83	0.91	6
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	1.00	0.80	0.89	10
copper	0.38	1.00	0.55	3
copra-cake	0.00	0.00	0.00	1
corn	0.00	0.00	0.00	16
cotton	0.43	0.75	0.55	4
cpi	1.00	0.62	0.77	8
crude	0.55	0.53	0.54	32
dlr	0.00	0.00	0.00	14
dmk	0.00	0.00	0.00	1
earn	0.97	0.97	0.97	279
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	5
gnp	0.40	0.50	0.44	8
gold	0.67	0.67	0.67	6
grain	0.11	0.11	0.11	44
groundnut	0.00	0.00	0.00	1
heat	0.50	0.25	0.33	4
hog	0.00	0.00	0.00	3
housing	1.00	1.00	1.00	1
income	0.50	1.00	0.67	1
interest	0.53	0.53	0.53	32
ipi	0.75	1.00	0.86	3
iron-steel	1.00	0.33	0.50	9
jobs	1.00	0.80	0.89	5
l-cattle	0.00	0.00	0.00	1
lead	0.00	0.00	0.00	2
lei	1.00	1.00	1.00	2
lin-oil	0.00	0.00	0.00	1
livestock	0.29	0.33	0.31	12
lumber	1.00	1.00	1.00	1
meal-feed	0.33	0.33	0.33	3
money-fx	0.34	0.31	0.32	52
money-supply	1.00	0.73	0.85	15
naphtha	0.00	0.00	0.00	0
nat-gas	0.44	0.40	0.42	10
nickel	0.00	0.00	0.00	4
oat	0.00	0.00	0.00	2
oilseed	0.05	0.06	0.06	16
orange	1.00	1.00	1.00	2
palm-oil	0.00	0.00	0.00	3
platinum	0.00	0.00	0.00	2
propane	0.00	0.00	0.00	1
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	2
reserves	0.88	0.78	0.82	9
retail	1.00	1.00	1.00	2
rice	0.00	0.00	0.00	1
rubber	1.00	1.00	1.00	2
ship	0.38	0.69	0.49	13
silver	0.00	0.00	0.00	1
sorghum	0.00	0.00	0.00	6
soy-meal	0.00	0.00	0.00	1
soy-oil	0.00	0.00	0.00	1
soybean	0.00	0.00	0.00	20
strategic-metal	0.00	0.00	0.00	1
sugar	0.80	0.86	0.83	14
sun-meal	0.00	0.00	0.00	0
sunseed	0.00	0.00	0.00	2
tin	1.00	1.00	1.00	1
trade	0.57	0.61	0.59	33
veg-oil	0.11	0.12	0.12	8
wheat	0.00	0.00	0.00	26
wpi	1.00	1.00	1.00	1
yen	0.00	0.00	0.00	1
zinc	0.50	0.17	0.25	6
accuracy			0.64	958
macro avg	0.36	0.37	0.35	958
weighted avg	0.64	0.64	0.63	958

Fonte: O Autor

Figura 43 – Dataset Reuters usando o método doc2vec e o algoritmo KNN

	precision	recall	f1-score	support
acq	0.96	0.99	0.98	155
alum	0.00	0.00	0.00	3
barley	0.00	0.00	0.00	2
bop	0.50	0.43	0.46	7
carcass	0.00	0.00	0.00	5
cocoa	0.27	0.75	0.40	4
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.36	0.67	0.47	6
copper	0.50	0.33	0.40	6
corn	0.71	0.48	0.57	21
cotton	0.67	0.50	0.57	4
cpi	0.67	0.25	0.36	8
crude	0.73	0.88	0.80	40
dlr	0.50	0.75	0.60	8
dmk	1.00	1.00	1.00	1
earn	0.99	0.96	0.98	310
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	3
gnp	0.24	1.00	0.38	5
gold	0.83	0.38	0.53	13
grain	0.83	1.00	0.91	34
heat	0.00	0.00	0.00	1
hog	1.00	0.33	0.50	3
housing	0.00	0.00	0.00	2
interest	0.87	0.92	0.89	37
ipi	0.50	0.50	0.50	4
iron-steel	0.00	0.00	0.00	2
jet	0.00	0.00	0.00	1
jobs	0.00	0.00	0.00	2
l-cattle	0.00	0.00	0.00	1
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	1
livestock	0.50	0.42	0.45	12
lumber	0.00	0.00	0.00	2
meal-feed	0.00	0.00	0.00	5
money-fx	0.66	0.98	0.79	43
money-supply	0.86	0.43	0.57	14
nat-gas	0.33	0.75	0.46	4
nickel	0.00	0.00	0.00	2
oat	0.00	0.00	0.00	1
oilseed	0.50	0.62	0.55	13
orange	0.00	0.00	0.00	1
palladium	0.00	0.00	0.00	1
palm-oil	0.50	0.25	0.33	4
palmkernel	0.00	0.00	0.00	1
pet-chem	0.33	0.50	0.40	2
platinum	0.00	0.00	0.00	1
potato	0.00	0.00	0.00	2
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	0
reserves	0.57	0.50	0.53	8
retail	0.00	0.00	0.00	4
rice	0.33	0.25	0.29	4
rubber	0.50	0.50	0.50	4
ship	0.73	0.92	0.81	26
silver	0.00	0.00	0.00	0
sorghum	0.00	0.00	0.00	2
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	5
soybean	0.33	0.12	0.18	8
strategic-metal	0.00	0.00	0.00	2
sugar	0.50	0.75	0.60	12
sun-oil	0.00	0.00	0.00	2
tin	1.00	0.50	0.67	2
trade	0.75	0.97	0.85	37
veg-oil	0.75	0.50	0.60	6
wheat	0.92	0.89	0.91	27
yen	0.67	0.50	0.57	4
zinc	0.00	0.00	0.00	1
accuracy			0.81	958
macro avg	0.33	0.32	0.31	958
weighted avg	0.79	0.81	0.79	958

Fonte: O Autor

Figura 44 – Dataset Reuters usando o método doc2vec e o algoritmo SVM

	precision	recall	f1-score	support
acq	0.86	0.97	0.91	155
alum	0.00	0.00	0.00	3
barley	0.00	0.00	0.00	2
bop	0.33	0.14	0.20	7
carcass	0.00	0.00	0.00	5
cocoa	1.00	0.25	0.40	4
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.30	0.50	0.37	6
copper	0.00	0.00	0.00	6
corn	0.52	0.81	0.63	21
cotton	0.00	0.00	0.00	4
cpi	1.00	0.12	0.22	8
crude	0.78	0.90	0.84	40
dlr	0.57	0.50	0.53	8
dmk	0.00	0.00	0.00	1
earn	0.94	0.98	0.96	310
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	3
gnp	0.00	0.00	0.00	5
gold	0.71	0.38	0.50	13
grain	0.71	0.88	0.79	34
heat	0.00	0.00	0.00	1
hog	0.00	0.00	0.00	3
housing	0.00	0.00	0.00	2
interest	0.59	0.81	0.68	37
ipi	0.00	0.00	0.00	4
iron-steel	0.00	0.00	0.00	2
jet	0.00	0.00	0.00	1
jobs	0.00	0.00	0.00	2
l-cattle	0.00	0.00	0.00	1
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	1
livestock	0.00	0.00	0.00	12
lumber	0.00	0.00	0.00	2
meal-feed	0.00	0.00	0.00	5
money-fx	0.62	0.93	0.74	43
money-supply	0.31	0.29	0.30	14
nat-gas	0.00	0.00	0.00	4
nickel	0.00	0.00	0.00	2
oat	0.00	0.00	0.00	1
oilseed	0.34	0.77	0.48	13
orange	0.00	0.00	0.00	1
palladium	0.00	0.00	0.00	1
palm-oil	0.00	0.00	0.00	4
palmkernel	0.00	0.00	0.00	1
pet-chem	0.00	0.00	0.00	2
platinum	0.00	0.00	0.00	1
potato	0.00	0.00	0.00	2
rape-oil	0.00	0.00	0.00	1
reserves	1.00	0.50	0.67	8
retail	0.00	0.00	0.00	4
rice	1.00	0.25	0.40	4
rubber	0.00	0.00	0.00	4
ship	0.59	0.88	0.71	26
sorghum	0.00	0.00	0.00	2
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	5
soybean	0.00	0.00	0.00	8
strategic-metal	0.00	0.00	0.00	2
sugar	0.10	0.08	0.09	12
sun-oil	0.00	0.00	0.00	2
tin	0.00	0.00	0.00	2
trade	0.58	0.86	0.70	37
veg-oil	0.00	0.00	0.00	6
wheat	0.66	0.70	0.68	27
yen	0.00	0.00	0.00	4
zinc	0.00	0.00	0.00	1
accuracy			0.75	958
macro avg	0.20	0.18	0.17	958
weighted avg	0.68	0.75	0.70	958

Fonte: O Autor

Figura 45 – Dataset Reuters usando o método doc2vec e o algoritmo Naive Bayes

	precision	recall	f1-score	support
acq	0.97	0.75	0.84	155
alum	0.08	0.33	0.12	3
barley	0.00	0.00	0.00	2
bop	0.07	0.14	0.09	7
carcass	0.00	0.00	0.00	5
cocoa	0.11	0.25	0.15	4
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.29	0.33	0.31	6
copper	0.22	0.33	0.27	6
corn	0.57	0.38	0.46	21
cotton	0.20	0.50	0.29	4
cotton-oil	0.00	0.00	0.00	0
cpi	0.43	0.38	0.40	8
crude	0.77	0.68	0.72	40
dlr	0.19	0.62	0.29	8
dmk	0.50	1.00	0.67	1
earn	0.99	0.79	0.88	310
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	3
gnp	0.05	0.40	0.09	5
gold	0.59	0.77	0.67	13
grain	0.79	0.65	0.71	34
groundnut-oil	0.00	0.00	0.00	0
heat	0.00	0.00	0.00	1
hog	0.33	0.33	0.33	3
housing	0.50	0.50	0.50	2
interest	0.86	0.51	0.64	37
ipi	0.57	1.00	0.73	4
iron-steel	0.00	0.00	0.00	2
jet	0.00	0.00	0.00	1
jobs	0.14	0.50	0.22	2
l-cattle	0.00	0.00	0.00	1
lead	1.00	1.00	1.00	1
lei	0.00	0.00	0.00	1
livestock	0.67	0.67	0.67	12
lumber	0.00	0.00	0.00	2
meal-feed	0.00	0.00	0.00	5
money-fx	0.76	0.67	0.72	43
money-supply	0.62	0.57	0.59	14
nat-gas	0.17	0.50	0.25	4
nickel	0.00	0.00	0.00	2
oat	0.00	0.00	0.00	1
oilseed	0.20	0.23	0.21	13
orange	0.00	0.00	0.00	1
palladium	0.00	0.00	0.00	1
palm-oil	0.00	0.00	0.00	4
palmkernel	0.00	0.00	0.00	1
pet-chem	0.50	0.50	0.50	2
platinum	0.00	0.00	0.00	1
potato	0.00	0.00	0.00	2
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	0
reserves	1.00	0.50	0.67	8
retail	0.00	0.00	0.00	4
rice	0.40	0.50	0.44	4
rubber	0.25	0.25	0.25	4
ship	0.85	0.42	0.56	26
silver	0.00	0.00	0.00	0
sorghum	0.00	0.00	0.00	2
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	5
soybean	0.07	0.12	0.09	8
strategic-metal	0.33	0.50	0.40	2
sugar	0.32	0.50	0.39	12
sun-oil	0.00	0.00	0.00	2
sunseed	0.00	0.00	0.00	0
tea	0.00	0.00	0.00	0
tin	0.20	0.50	0.29	2
trade	0.68	0.46	0.55	37
veg-oil	0.07	0.17	0.10	6
wheat	0.61	0.41	0.49	27
wpi	0.00	0.00	0.00	0
yen	0.12	0.25	0.17	4
zinc	0.12	1.00	0.22	1
accuracy			0.61	958
macro avg	0.24	0.28	0.24	958
weighted avg	0.75	0.61	0.66	958

Fonte: O Autor

Figura 46 – Dataset Reuters usando o método doc2vec e o algoritmo Decision Tree

	precision	recall	f1-score	support
acq	0.85	0.85	0.85	155
alum	0.00	0.00	0.00	3
barley	0.00	0.00	0.00	2
bop	0.00	0.00	0.00	7
carcass	0.00	0.00	0.00	5
cocoa	0.09	0.25	0.13	4
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.29	0.33	0.31	6
copper	0.00	0.00	0.00	6
corn	0.42	0.24	0.30	21
cotton	0.25	0.25	0.25	4
cpi	0.25	0.12	0.17	8
cpu	0.00	0.00	0.00	0
crude	0.64	0.45	0.53	40
dlr	0.15	0.25	0.19	8
dmk	0.00	0.00	0.00	1
earn	0.88	0.87	0.88	310
fuel	0.00	0.00	0.00	1
gas	0.00	0.00	0.00	3
gnp	0.12	0.40	0.19	5
gold	0.44	0.31	0.36	13
grain	0.73	0.71	0.72	34
groundnut-oil	0.00	0.00	0.00	0
heat	0.00	0.00	0.00	1
hog	0.00	0.00	0.00	3
housing	0.00	0.00	0.00	2
interest	0.56	0.49	0.52	37
ipi	0.20	0.25	0.22	4
iron-steel	0.11	0.50	0.18	2
jet	0.00	0.00	0.00	1
jobs	0.00	0.00	0.00	2
l-cattle	0.00	0.00	0.00	1
lead	0.00	0.00	0.00	1
lei	0.00	0.00	0.00	1
livestock	0.00	0.00	0.00	12
lumber	0.00	0.00	0.00	2
meal-feed	0.00	0.00	0.00	5
money-fx	0.58	0.70	0.63	43
money-supply	0.31	0.29	0.30	14
nat-gas	0.11	0.25	0.15	4
nickel	0.00	0.00	0.00	2
oat	0.00	0.00	0.00	1
oilseed	0.27	0.23	0.25	13
orange	0.00	0.00	0.00	1
palladium	0.00	0.00	0.00	1
palm-oil	0.00	0.00	0.00	4
palmkernel	0.00	0.00	0.00	1
pet-chem	0.00	0.00	0.00	2
platinum	0.00	0.00	0.00	1
potato	0.00	0.00	0.00	2
propane	0.00	0.00	0.00	0
rape-oil	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	0
reserves	0.20	0.12	0.15	8
retail	0.00	0.00	0.00	4
rice	0.00	0.00	0.00	4
rubber	0.25	0.25	0.25	4
ship	0.25	0.23	0.24	26
silver	0.00	0.00	0.00	0
sorghum	0.20	0.50	0.29	2
soy-meal	0.00	0.00	0.00	2
soy-oil	0.00	0.00	0.00	5
soybean	0.29	0.25	0.27	8
strategic-metal	0.00	0.00	0.00	2
sugar	0.12	0.17	0.14	12
sun-oil	0.00	0.00	0.00	2
sunseed	0.00	0.00	0.00	0
tin	0.00	0.00	0.00	2
trade	0.56	0.54	0.55	37
veg-oil	0.11	0.17	0.13	6
wheat	0.45	0.37	0.41	27
yen	0.00	0.00	0.00	4
zinc	0.00	0.00	0.00	1
accuracy			0.59	958
macro avg	0.13	0.14	0.13	958
weighted avg	0.60	0.59	0.59	958

Fonte: O Autor

Figura 47 – Dataset Reuters usando o método doc2vec e o algoritmo MLP

	precision	recall	f1-score	support
acq	0.97	0.94	0.95	165
alum	0.50	0.50	0.50	2
barley	0.00	0.00	0.00	2
bop	0.17	0.25	0.20	4
carcass	0.33	0.50	0.40	4
castor-oil	0.00	0.00	0.00	1
cocoa	0.17	0.33	0.22	3
coconut	0.00	0.00	0.00	1
coconut-oil	0.00	0.00	0.00	1
coffee	0.56	0.62	0.59	16
copper	1.00	0.12	0.22	8
copra-cake	0.00	0.00	0.00	1
corn	0.52	0.65	0.58	17
cotton	0.00	0.00	0.00	2
cpi	0.44	0.57	0.50	7
cpu	0.00	0.00	0.00	1
crude	0.83	0.85	0.84	40
dlr	0.38	0.30	0.33	10
dmk	0.00	0.00	0.00	0
earn	0.97	0.97	0.97	294
fuel	0.00	0.00	0.00	3
gas	0.33	0.25	0.29	4
gnp	0.56	0.45	0.50	11
gold	0.50	0.38	0.43	8
grain	0.73	0.87	0.80	38
groundnut	0.00	0.00	0.00	1
heat	0.00	0.00	0.00	1
hog	0.00	0.00	0.00	1
housing	0.50	0.33	0.40	3
income	0.00	0.00	0.00	1
instal-debt	0.00	0.00	0.00	1
interest	0.76	0.85	0.80	33
ipi	0.67	0.29	0.40	7
iron-steel	0.11	0.33	0.17	3
jobs	0.75	0.43	0.55	7
lead	0.00	0.00	0.00	1
livestock	0.50	0.57	0.53	7
lumber	0.00	0.00	0.00	1
meal-feed	0.00	0.00	0.00	1
money-fx	0.87	0.90	0.89	60
money-supply	0.83	0.59	0.69	17
nat-gas	0.33	0.33	0.33	6
nickel	0.00	0.00	0.00	1
oat	0.00	0.00	0.00	1
oilseed	0.60	0.67	0.63	9
orange	0.50	0.25	0.33	4
palm-oil	0.33	0.33	0.33	3
pet-chem	0.25	0.50	0.33	2
platinum	0.00	0.00	0.00	1
rapeseed	0.00	0.00	0.00	0
reserves	0.75	0.67	0.71	9
retail	0.50	0.25	0.33	4
rice	0.50	0.17	0.25	6
rubber	0.17	1.00	0.29	1
ship	0.78	0.67	0.72	21
silver	0.00	0.00	0.00	1
sorghum	0.50	0.50	0.50	2
soy-meal	0.00	0.00	0.00	0
soy-oil	0.00	0.00	0.00	1
soybean	0.50	0.71	0.59	7
strategic-metal	0.00	0.00	0.00	1
sugar	0.45	0.50	0.48	10
sunseed	1.00	0.50	0.67	2
trade	0.84	0.89	0.86	35
veg-oil	0.75	0.25	0.38	12
wheat	0.48	0.70	0.57	20
wpi	1.00	1.00	1.00	1
yen	0.25	0.12	0.17	8
zinc	0.33	0.50	0.40	2
accuracy			0.78	958
macro avg	0.36	0.34	0.33	958
weighted avg	0.79	0.78	0.78	958

Fonte: O Autor