**Relatório Trabalho Final:**

**Análise de Sentimento de “*Tweets*” Referentes a Cantora Anitta**

Aluno: Claudionor Ferreira da Silva Junior

Disciplina: Aprendizado de Máquinas I

Professor: Douglas Rodrigues

**Contexto Geral:**

Anitta é uma cantora, compositora e empresária brasileira conhecida por misturar pop, funk carioca e reggaeton. Além de sua música, ela é uma influente figura nas redes sociais, reconhecida por sua versatilidade e impacto cultural. Este estudo analisou publicações de usuários (ou “*tweets*”) referentes à cantora na plataforma Twitter (atual “X”) para construir uma análise de sentimento e um modelo que classifica esses tweets como positivos ou negativos em relação à celebridade.

**Constituição da Base de Dados Original:**

A base de dados é constituída pelas colunas:

* **user\_id –** Identificação do autor da publicação
* **conversation\_id –** Identificação da conversa da publicação
* **user\_username –** Nome de usuário do autor da publicação
* **text –** Publicação referente a cantora
* **Polaridade –** Identificação se o “*tweet*” é bom ou ruim (ruim = -1 e bom = 1)

**Etapas do Pré-Processamento:**

|  |  |
| --- | --- |
| **Etapa** | **Descrição** |
| **ETAPA 1** | Primeiramente, foi realizada uma divisão da base de dados em treino e teste utilizando a biblioteca **“*caret*”**, na qual todos os pré-processamentos abaixo foram aplicados primeiramente na base treino para fazer o treinamento do modelo, e posteriormente, será aplicado na base teste para verificação das métricas de Acurácia, Sensibilidade e Especificidade. |
| **ETAPA 2** | Após essa separação, foram retiradas as colunas “user\_id”, “conversation\_id” e “user\_username”, utilizando a biblioteca **“*dplyr*”,** para só trabalharmos com as colunas das publicações e identificação. |
| **ETAPA 3** | Nas publicações, foram feitas manipulações para adequação e melhora de desempenho na análise de sentimento e classificação, como a remoção de acentos das palavras, remoção de números e substituição de emojis por palavras que realmente representam o sentimento. Além disso, foi criada uma coluna “RecordID” para identificação de cada publicação. Tudo isso utilizando a biblioteca **“*dplyr*”** |
| **ETAPA 4** | Foi realizado a tokenização da coluna de “tweets”, ou seja, as frases foram separadas em palavras. Além disso, para retirar as chamadas “stopwords”, ou palavras que não agregam na análise de sentimento, como conjunções e conectivos, foi usado um dicionário que contém palavras acrescentadas por mim: <https://raw.githubusercontent.com/Claudionor20/XSentiment/main/stopwords_.txt> |
| **ETAPA 5** | Nas publicações há uma grande variabilidade de “gírias” ou formas diferentes de falar uma palavra, como por exemplo, a palavra “arrasar” foi escrita como “arrasa” ou “arrasouuu”. Com isso, foi feito uma manipulação para converter esse tipo de variações a uma só palavra |
| **ETAPA 6** | Nessa etapa foi feita a chamada “lematização”, ou seja, a transformação de palavras similares por uma só, na maioria por seu infinitivo ou sinônimo. Para isso, foi usado um dicionário com as palavras e seus respectivos sinônimos, também acrescentado por mim: <https://raw.githubusercontent.com/Claudionor20/XSentiment/main/lematizacao_claudio_v2.txt> |
| **ETAPA 7** | Nesse momento foi feita a manipulação com a frequência das palavras, na qual foi retirada as palavras que possuem uma frequência menor que 15 na base de referência. Além disso, para a formação da matriz de termo documento, ou seja, a matriz com a frequência das palavras em cada publicação, foi calculado a frequência percentual de cada palavra em cada publicação. Tudo isso utilizando o pacote **“*dplyr***” e para a criação da matriz termo-documento, foi utilizado a biblioteca **“*maditr*”** |
| **ETAPA 8** | Foram retiradas algumas palavras desnecessárias, como verbos genéricos e palavras que possuem alta correlação entre si. Para detecção da correlação das palavras, foi usado a função a “*findCorrelation”* do pacote**“*caret*”** e estabelecido um parâmetro de exclusão de palavras que possuem uma correlação acima 0.75 entre alguma variável explicativa. Dessa forma, foram excluídas as colunas:  **"V1", "l", "achar", "acreditar", "acontecer", "acompanhar", "antiga","chegar","caro","esquerdo","melhorar","menino","rede",**  **"esquerda", "redar"** |
| **ETAPA 9** | Na base treino, foi detectado um pequeno desbalanceamento na base, contendo mais casos em que as publicações são ruins, ou seja, a classificação -1 associada. Para corrigir isso, foi utilizado o método de rebalanceamento denominado **“*ADASYN*”** da biblioteca **“*SMOTEWB*”.** A ideia desse método é gerar amostras sintéticas da classe minoritária utilizando o **“*KNN*”,** ou seja, o algoritmo de vizinhos próximos. Dessa forma, usado como parâmetro k = 5, foi gerado mais dados de classificação 1 da publicação, ou seja, onde o “tweet” é bom referente a cantora, a partir dos 5 vizinhos mais próximos dessa característica |
| **ETAPA 10** | Em um modelo de classificação de sentimentos, palavras que aparecem frequentemente em muitos tweets geralmente não contribuem para distinguir se a publicação é boa ou ruim, enquanto termos raros tendem a ser mais distintivos. Para lidar com isso, calculamos o IDF (Inverse Document Frequency), uma medida estatística que atribui pesos às frequências das palavras, e multiplicamos pela respectiva frequência da palavra em cada “tweet”. O IDF penaliza palavras comuns e aumenta o peso das palavras raras, tornando-as mais influentes para o modelo de classificação. Isso ajuda a melhorar a precisão do modelo, enfatizando termos que são mais indicativos do sentimento da publicação.  **Fórmula do IDF = Log(Número de “tweets”/(1 + Soma da Frequência da palavra i)** |

**Pré-Processamento na base teste:**

Para adequar a base de teste ao formato da base de treino utilizada para treinar o modelo, foram aplicados os mesmos processamentos, exceto o balanceamento, que é específico da base de treino. Além disso, foi desenvolvida uma função para alinhar as colunas da base de teste com as da base de treino. Isso significa que colunas presentes apenas na base de teste são removidas, enquanto colunas presentes apenas na base de treino são adicionadas na base de teste com valores preenchidos com zero. Esse procedimento garante consistência nos dados utilizados para avaliar o modelo, mantendo o formato e as características do treinamento.

**Treinamento do Modelo:**

Após a conclusão de todo o pré-processamento, deu-se início ao treinamento do modelo de classificação para identificar se um “tweet” é bom ou ruim referente a cantora a partir da frequência da palavra na publicação. Para isso, foi feito validação cruzada utilizando o método **“*k-fold*” com k = 10**, a fim de testar os melhores parâmetros para o modelo “*XGBoost*”. Dessa forma, temos o modelo que melhor desempenhou na base treino com as seguintes especificações:

|  |  |
| --- | --- |
| Modelo | XGBoost |
| Grade de Hiperparâmetros Testados | eta = (0.3, 0.1, 0.3, 0.5) max\_depth = (3, 6, 9,12) nrounds = 1000 early\_stopping\_rounds = 10 |
| Parâmetros | eta = 0.5 max\_depth = 9 nrounds = 1000 early\_stopping\_rounds = 10 |

Vale ressaltar que foi preciso transformar a base treino para o formato de matriz que é aceito pela função de treinamento ***xgboost*** da biblioteca **“*xgboost*”.**

**Resultados do Modelo:**

Após aplicar o mesmo pré-processamento utilizado na base de treino à base de dados de teste, foi comparada as predições do modelo, em comparação com os resultados da base teste. O resultado obtido teve as seguintes métricas:

|  |  |  |  |
| --- | --- | --- | --- |
| Medida | Acurácia | Sensibilidade | Especificidade |
| Valor (%) | 72.73% | 78.70% | 62.39% |

**Conclusão do Estudo:**

Com base nas métricas obtidas, podemos concluir que o modelo apresentou um desempenho robusto na base de teste, indicando sua capacidade de prever com certa precisão o sentimento associado às publicações sobre a cantora Anitta. Isso sugere que o modelo é eficaz e confiável para classificar novas publicações no futuro.