GUSTAVO CASAES ALVES,
Universidade Federal do Tocantins,
Palmas, Tocantins

Análise Comparativa de Random Forest, Regressão Logística e Naive Bayes Gaussiano para tarefas de Classificação de Texto.

RESUMO

Este artigo busca trazer uma análise comparativa da validação de algoritmos de aprendizado de máquina: Random Forest, Regressão Logistíca e Naive Bayes Gaussiano na tarefa de classificação de textos, definindo um valor da avaliação (rating) na seção de comentários do aplicaivo TripAdvisor. Objetivo é investigar o desempenho desses algoritmos na categorização do comentário de acordo com o rating dado nele, classificando o teor da mensagem como positivo, negativo ou neutro.

1. INTRODUÇÃO

A utilização de métodos e algoritmos de aprendizado de máquina para diversas tarefas da área do conhecimento que envolve lidar, processar e extrair informações utilizáveis de obras escritas, o processamento de linguagem natural já é uma área muito bem explorada. Por isso conhecer e entender os algoritmos disponíveis, saber como estruturá-los, alimentá-los e utilizá-los nas atividades adequadas é um processo tão importante quanto o desenvolvimento de modelos e tecnologias.

Dentre essas técnicas que existem na área de processamento de linguagem natural, a classificação de texto é uma atividade que pode ser performada a partir de métodos baseado em regras manualmente escritas, o que pode ser um tarefa árdua e desafiadora, dada a natureza desestruturada nos dados presentes no documento. Por isso, e com a crescente nos últimos anos do número de documentos e textos complexos em níveis industriais e empresariais, a análise e escolha dos métodos automatizados por meio de aprendizagem tem sido cada vez mais requisitados. Para isso, por meio da observação dos dados, a máquina, utilizando um conjunto de dados pré-rotulados como treinamento, irá aprender associações inerentes entre os textos e rótulos dados.

Esclarecido esse tópico, a abordagem desse artigo será trabalhar com o problema de classificação de texto em cinco classes, em três tecnologias diferentes, utilizando uma base de dados dos comentários de usuários de um aplicativo de viagem. As classes são definidas pelo valor atribuído à avaliação de um comentário no aplicativo do TripAdvisor, sendo as de uma até cinco estrelas, e os dados de entrada analisados serão as características (*features*) extraídas do texto a partir do conteúdo dos comentários, os documentos do projeto. Para esse problema de classificação serão utilizados três algoritmos de aprendizado: Random Forest, Regressão Logística e Naive Bayes.

2. METODOLOGIA

Com esse conjunto de dados foi feita uma análise utilizando diferentes algoritmos de aprendizado de máquina na atividade de classificação de textos em conjunto com as ferramentas e

pacotes disponíveis da linguagem de programação Python, incluindo o próprio NLTK e o Scikit-Learn.

Classificação de Textos

Em processamento de linguagem natural a classificação de texto consiste em uma atividade bem conhecida e estudada. Tem como objetivo atribuir rótulos ou categorias a unidades de textos. Esses textos possem ser retirados de diversas fontes, desde *tweets*, *e-mails*, artigos de notícia, atendimento ao cliente, ou como é o caso trabalho, análises (*reviews*) de consumidores, isso pode ser realizado com o propósito de ser solucionar problemas diferentes: Análise de Sentimentos, Análise de Tópicos, Categorização de Notícias, *Question Answering* ou *Natural language inference*. Na *Sentiment Analysis* é buscado encontrar o teor da mensagem com intuito de se reter uma opinião dela, como um posicionamento ou polaridade daquele indivíduo. É uma tarefa que pode ser de binário, ou como é feito no caso deste trabalho, de multi-classificação.Esse problema consiste em quatro fases distintas: *1. Extração de características*, *2. Redução de Dimensionalidade*, *3. Seleção de Classificador* e *4. Validação*. Para isso, temos como entrada um conjunto de textos em documentos, no qual cada documento é composto por uma coletânea de datapoints em segmentos de textos.

No caso desse artigo, o dados foram encontrado a partir de uma base dados do Kaggle, que consiste em uma tabela com um campo de comentários (coluna ***"review_full"***) e o valor do *rating* dado (***"review_rating"***). A primeira coluna possui nosso conteúdo da onde será tirado os dados que alimentaram a máquina no processo de predição de sentimento no texto. A coluna seguinte servira como a variável dependente escolhida para servir de validação da predição elaborada pelo modelo.

Portanto para trabalhar com esse conjunto de documentos, com os comentários dos usuários que registraram a sua avaliação no aplicativo do TripAdvisor, foi necessário fazer um processo de pré-processamento, com as ferramentas da área de processamento de linguagem natural, para transformar os dados no texto da avaliação do usuário em representações numéricas que sirva de entrada para a alimentação do algoritmo de aprendizado de máquina. Assim, foi tratado cada comentário da primeira coluna como um documento que irá compor esse conjunto de texto, o corpus do projeto.

Pré-processamento

Na primeira etapa do pré-processamento foram removidos as amostras nulas no corpus , o que cominou em eliminados apenas dois item da tabela, que foram consideradas vazias, com as entradas no campo de ***"review_full"*** vazia sendo consideradas prejudiciais ao resultado do modelo e por isso foram eliminadas da base de dados.

1. **Normalização**

   Na coluna da variável dependente foi feita uma normalização para valores entre -1 e 1. Isso porque o objetivo final da aplicação deveria ser classificar o sentimento como positivo, negativo ou neutro e não necessariamente acertar o rating da avaliação dada. Dessa forma foi utilizado a fórmula:

$$x'' = 2\frac{x - \min x}{\max x - \min x} - 1$$

2. **Tokenização**

   Depois disso foram feitas uma limpeza, no textos do documento eliminando o prefixo de links, caracteres especiais e transformado o texto em minúsculas, para, em seguida ser feito a tokenização dos textos no documentos. A tokenização consiste em transformar as palavras da sentença, parágrafo ou documento, em tokens, que serve para compor o processo da

transformação das sentenças em valores numéricos. Nisso transformamos as sentenças em uma lista de palavras. Além do processo de tokenização, cada palavra passou por *stemmer*, responsável pela atividade de reduzir palavras diferentes para o mesmo modo uniforme, basicamente transformando flexões da palavra, em tempos ou gêneros diferentes, se reduzirem a um único termo padrão. Isso sendo feito apenas com as palavras que podem ser consideradas significativas no texto, que serviram de dados para o processo de extração de características, excluindo as chamadas stopwords. Com isso, foi eliminado as palavras incluídas no dicionário de Stopwords do pacote de ferramentas de processamento natural da linguagem de programação utilizada, aquelas palavras que possuíam pouco valor significativo no texto, que iria apenas poluir os resultados. Fazendo assim que seja construído um corpus, conjunto de documentos de texto, com os comentários das reviews já pré-processadas e tokenizadas.

Extração de Características

É preciso extrair os dados em representações numéricas a partir do conteúdo do texto. Para isso é preciso que seja contado o número de ocorrências dos tokens em cada documento, criando uma lista de frequência para cada palavra que aparece pelo menos uma vez. Dessa maneira é possível observar nosso conjunto de dados mais facilmente como uma tabela bidimensional, com uma lista de documentos em cada linha e com as palavras como colunas para essas linhas. Uma lista de frequência de termos por documentos. Por mais que já pareça possível extrair informações e características dessa maneira é necessário mais um processo para garantir que apenas conteúdo significativo vai ser, novamente, levado em consideração. É preciso novamente fazer uma normalização dos dados, reduzindo o peso das frequências de palavras mais presentes em diversos documentos, mas sem valor significativo. Agora sim é possível tratar cada token individual como um *feature* e dessa maneira um documento funciona como uma lista de *features*.

Esse processo de transformar as palavras em números é chamado de vetorização e utiliza a estratégia da Sacola de Palavras (***Bag-of-words***). Esse método a palavras são organizadas indiferente da sua gramática e ordenação, é ignorado os relacionamentos e semânticas das palavras e são coletadas apenas sua multiplicidade que define o ponto de foco, os possíveis candidatos para características relevantes. No desenvolvimento desse projeto foi utilizado uma única classe do Scikit-Learn para a implementação da tokenização da contagem de ocorrências, transformando os tokens em valores de frequências.

Redução de Dimensionalidade

Existem diversas técnicas que envolvem reduzir o número de features em documentos, para que sejam menos entradas para o modelo. Isso envolve reduzir a lista de features em conjunto mais compactado para fins de eficiência. Neste trabalho, foi feito reduzido as palavras de cada documento a lista de termos únicos e o valores da variável dependente que busca ser predita foi simplificado para três classes ao invés de cinco. Isso porque os valores originais do rating da análises iam de uma a cinco estrelas, criando uma escala positiva ascendente, mas no trabalho desse artigo foi elaborada uma normalização de -1 a 1 fazendo com que os valores abaixo de três estrelas sejam considerados negativos e os acima ou igual a ele sejam positivos e neutro, respectivamente. Isso para que seja inferido apenas o teor de polaridade daquele comentário, buscando entender o intuito do usuário ao escrevê-lo: elogiar ou criticar algum item disposto no aplicativo.

Seleção de Classificador

Foi utilizado Naive Bayes, Logistic Regression e Random Forest para a solução desse problema de classificação no trabalho deste artigo. Todos implementados por meio das ferramentas e pacotes disponibilizados pelo Scikit-Learn.

Validação

Por se tratar de um problema de classificação foi utilizado da acurácia e de outras métricas da matriz de confusão.

3. RESULTADOS

Os resultados mostraram que o cada um dos algoritmos selecionados possuíam rendimentos relativamente diferentes para essa tarefa de classificação. Dentre as métricas utilizadas, a acurácia, o valor medido pela razão dos valores verdadeiramente preditos com os número total de todas as predições verdadeiras, descreveu muito bem a diferença entre os algoritmos. Uma média de 69%, para os resultados alcançados com Naive Bayes como o mais baixo dos resultados, sendo que os outros algoritmos, Random Forest e Regressão Logistíca ficaram por volta de 86,% 87%, respectivamente. A conclusão é que os algoritmos utilizando apresentam pontos fracos muito claros e conhecidos pela literatura no processo de classificação de texto. E mesmo tendo o Random Forest como um algoritmo mais bem sucedido nesse cenário, a melhor opção seria utilizar outro sistema de aprendizado com reforço como a construção de uma rede neural, como uma rede convolucional para solucionar o problema.

4. REFERÊNCIAS

Text Classification Algorithms: A Survey; KOWSARI, KAMRAM et al.; Information 10, 150; 2019

Deep Learning-based Text Classification: A Comprehensive Review; MINAEE, SHERVIN et al.; ACM Comput, 54 3; 2021

Automatic Text Classification: A Technical Review; DALAL, M K., ZAVERI, M A.; 2011

A novel approach for dimension reduction using word embedding: An enhanced text classification approach; SINGH, K N, et al.; International Journal of Information Management Data Insights 2; 2022

Dataset utilizado: https://www.kaggle.com/code/yemiclaudia/sentiment-analysis-of-2023-tripadvisor-reviews#Importing-Library-&-Dataset