Classificando Fake News no Dataset WELFake

Raphael Lahiry
E-mail: raphaellcmb@al.insper.edu.br

I. Dataset

O WELFake [1] é um conjunto de dados criado para auxiliar no desenvolvimento de soluções baseadas em texto para a detecção de *fake news*. O *dataset* é composto por 72.134 notícias, onde 35.028 são reais e 37.106 são falsas. Para montar esse *dataset*, seus autores combinaram quatro conjuntos de dados populares de notícias (Kaggle, McIntire, Reuters e BuzzFeed Political) para evitar o *overfitting* dos classificadores e fornecer mais dados para um melhor treinamento dos modelos de *machine learning*.

A coluna de interesse deste conjunto de dados é a coluna *label*, onde notícias reais recebem o número 1 e notícias falsas o número 0. Como o tamanho do dataset era muito grande, ele foi reduzido para cerca de 10% do seu tamanho, porém sem perder a representatividade e balanceamento dos dados.

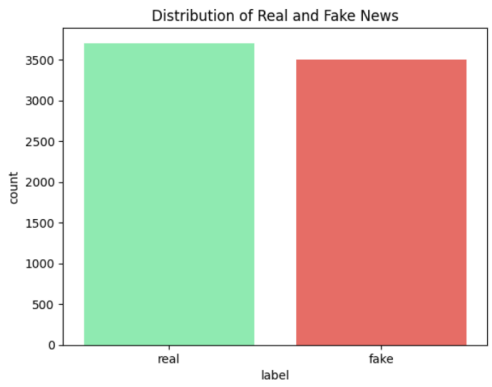

Figura 1: Distribuição das classes na porção reduzida do dataset

II. Pipeline de Classificação

O pipeline de classificação criado, é dividido entre passos de pré-processamento de texto, *feature engineering* e utilização de um modelo de classificação para classificar a notícia como real ou falsa.

Durante o pré-processamento, primeiramente, o texto é submetido a um processo de limpeza, onde são removidos números e caracteres não alfabéticos, uma vez que esses elementos não contribuem para o valor semântico do texto. Após a limpeza, o texto é *tokenizado,* sendo dividido em palavras individuais, onde segue para a remoção de *stopwords,* que são palavras muito comuns da língua (como, "*and*", "*the*", "*in*") e que geralmente também não possuem valor semântico para o texto. Para esse passo foi utilizado uma lista predefinida de *stopwords* do NLTK. O passo seguinte é a lematização, um processo em que os tokens são transformados para sua forma básica (lema), preservando seu significado semântico. Para a lematização foi utilizado o *WordNetLemmatizer* do NLTK. Para finalizar o pré-processamento, os tokens lematizados são reunidos novamente para formar sentenças completas e serem utilizados pelo classificador.

Na etapa de *feature engineering*, com o pré-processamento feito, o texto processado é transformado em uma representação numérica, através do TF-IDF (*Term Frequency-Inverse Document Frequency),* para poder ser utilizado no treinamento dos classificadores de texto.

Para a etapa de classificação foram testados três modelos diferentes, sendo estes o *Support Vector Machine* (SVM), *Random Forest Classifier* e *Logistic Regression.* Ao serem analisadas as métricas da avaliação no subconjunto de teste, os três classificadores tiveram um bom desempenho, apresentando métricas por volta de 0.9, porém o que melhor desempenhou foi o SVM [2], que foi o classificador escolhido.

III. Avaliação do Classificador

O dataset foi dividido nos subconjuntos de treino e teste utilizando uma divisão de 70% para o treino 30% para o teste, uma proporção bastante comum que permite ter um conjunto de treino grande o suficiente para ensinar o modelo, enquanto ainda reservamos uma parte significativa dos dados para a avaliação. Durante a divisão dos dados também foi utilizada a estratificação, garantindo que o modelo tenha exemplos suficientes de cada classe tanto no conjunto de treino quanto no de teste.

O classificador SVM apresentou uma acuraria balanceada de 0.928, indicando um bom desempenho do modelo, sugerindo que ele é eficaz em identificar corretamente tanto notícias reais, quanto notícias falsas.

	Balanced Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.910084	0.911342	0.910731	0.910640
Random Forest	0.900414	0.900551	0.900555	0.900548
Support Vector Machine	0.928008	0.928405	0.928307	0.928284

Figura 2: Tabela de métricas dos classificadores

As palavras mais importantes para classificar uma notícia real foram "hillary", "featured", "com", "image" e "via", e as que mais indicaram notícias falsas foram "reuters", "said", "follow", "breitbart" e "twitter"

IV. Avaliação do Tamanho do Dataset

Para entender a performance do classificador de acordo com o tamanho do dataset, foi realizada uma avaliação utilizando a curva de aprendizado, que ilustra como a precisão do modelo varia com diferentes tamanhos de dados de treinamento. Os tamanhos de subconjuntos utilizados na avaliação foram de 10%, 30%, %50%, 70% e 100%.

Foi observado que o classificador SVM tem um desempenho melhor com mais dados de treinamento, reduzindo o overfitting, e que o aumento adicional de dados pode melhorar um pouco mais o desempenho do modelo no teste, embora a taxa de melhoria começa a diminuir a partir de subconjunto tamanhos maiores.

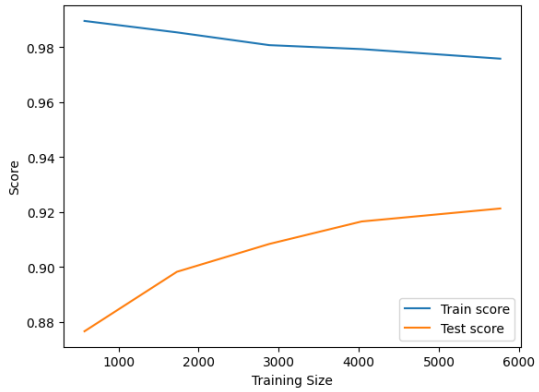

Figura 3: Curva de aprendizado com diferentes tamanhos de treinamento

V. Análise por Tópicos

Para realizar a análise por tópicos foi utilizado o Latent Dirichlet Allocation (LDA), que permite com que tópicos sejam descobertos em um conjunto de documentos ao identificar padrões de coocorrência de palavras. Para entender se existiam alguns tópicos nos quais a classificação era mais efetiva, foi criado um classificador de duas camadas, no qual primeiramente o documento tinha seu tópico identificado. Esse modelo revelou alguns tópicos principais, cada um com um conjunto distinto de palavras que ajudaram a definir seu tema. Os principais tópicos encontrados na análise foram:

- Questões internacionais: “zika”, “assange”, “rohingya”
- Política americana: “trump”, “clinton”, “president”
- Cultura e mídia: “comment”, “funny”, “http”
- Conflitos e segurança: “taliban”, “kabul”, “broadcast”
- Oriente médio: “iraq”, “kurd”, “Baghdad”

Em seguida, para cada tópico, foi treinado um classificador SVM focado. Para todos os principais tópicos extraídos, a acurácia dos classificadores foi de 100% exceto para o tópico de cultura e mídia, que foi de 92,75%. Isso sugere que para os tópicos relacionados a questões humanitárias, culturais e de segurança, a classificação foi extremamente eficaz.

O classificador de duas camadas não apenas melhorou a precisão da classificação ao conseguir segmentar os documentos por tópicos, mas também permitiu uma análise mais profunda do desempenho da classificação em diferentes temas de notícias.

Referências

[1] P. K. Verma, P. Agrawal, I. Amorim and R. Prodan, "WELFake: Word Embedding Over Linguistic Features for Fake News Detection," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 881-893, Aug. 2021, doi: 10.1109/TCSS.2021.3068519.

[2] Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026683