December 28th 2024

Author: Pamela Llerena

Technical Assessment for the Data Scientist Position

1. Introduction and Problem Statement

The objective of this project is to use Natural Language Processing (NPL) techniques to classify real and fake news articles. The main idea is to build an efficient classification model that can accurately distinguish between news articles based on their contextual content. In today's digital age, the growth of social media has led to more misinformation and fake news, making this project highly relevant. NPL plays a great role in transforming raw text data into structured information that can be processed by machine learning models. This report details the process followed for the accomplishment on the assessment. The project was executed on the jupyter notebook LlerenaP_tech_assessment.ipynb.

2. Data Cleaning and Preprocessing

The dataset, sourced from Kaggle, contains around 45000 news articles labeled as real or fake. The first step was to load the text data into Pandas Dataframe and inspect their structure. The different subjects for real and fake news stood out. Also, by plotting the distribution of article lengths in each data set, it was noticed that it was very variable. This was taken into account on the next step.

To prepare the data for later classification, labelling is crucial. Labels allow machine learning models to learn the relationship between features (text) and the target variable (label). In this case, real news was labeled with 1 and fake news with 0. Additionally, both data sets were combined into a unique data frame and shuffled. This ensures consistency and makes data manipulation much easier. The final data frame was then displayed, called *news_df*.
Some further inspection was done, for example checking for missing values, removing duplicates, and plotting the weight of real and fake news on the data set, which is fair to say it was almost even.

For the text preprocessing, the necessary libraries were imported, in particular the Natural Language Toolkit library (*nltk*). The steps in preprocessing involved:

- Convert text to lowercase, which helps consistency in the text.
- Removing special characters, numbers, and stopwords (common words that don't add much meaning).
- Tokenization, which is basically breaking the text into words or tokens.
- Lemmatization, where words were reduced to their “base form” (the word “talking” becomes “talk”, for example). There was also a similar method called “stemming”, but the one used was more accurate and it wasn’t as time consuming as the ladder.

The final processed text was displayed.

3. Text Analysis and Feature Extraction

To explore some text data patterns, Word Cloud and Word Frequency Analysis was implemented for each labeled data, real and fake news. A Word Freq. Analysis counts the occurrences of each word, while the Word Cloud visually represents the most frequent words in the text, where the size of each word corresponds to its frequency. Both data sets had many words in common, like “trump”, “president” and “state”, which makes sense as they are part of “news articles vocabulary”. However, political terms like "Trump" and "Clinton," combined with terms like "people" and "like," could suggest a pattern of sensationalism and bias, especially in fake news.

The next step was to transform text data into a format that can be fed into a machine learning model, called TF-IDF Vectorization. This is a known technique to transform text data into numerical features. It helps reduce the importance of common words and emphasize words that are distinctive to each document. To proper visualization, the vectorized data was transformed into a matrix then to a DataFrame. The output is the first 5 rows of the DataFrame with 1000 columns representing the top 1000 words in the *processed_text* column. Each cell in the DataFrame contained the TF-IDF value of the corresponding word. For example, some non-zero value words were “according" or “year”. To find the top words across the dataset, the total scores for each word were calculated. The top 10 words with the highest scores were presented on a horizontal bar plot.

4. Predictive Modeling with NLP

The first step was to split the data into training and testing sets. Typically, the data would be divided with 80% for training and 20% for testing. However, since it was also necessary to include a validation set, the data was split into 70% for training, 15% for validation, and 15% for testing. To examine the distribution of real and fake news in each set, a pie chart was created. As expected, the proportion of real and fake news in each set was balanced.

The Logistic Regression model was chosen due to its power of classification on binary problems, which is the case. It's a relatively simple algorithm compared to other complex ML models, computationally efficient, and can handle large datasets well. Also, provides clear insights through probabilities and feature coefficients, helping understand what drives predictions, as it will be explained later.

The model was trained using the train set. Then, a prediction was made on the validation set. Finally, the model was evaluated using an accuracy score and the classification report was presented. With an accuracy of 0.9831, the model works well with the data set and it correctly classifies almost all the instances in the validation set. The precision (or recall) for both classes (real and fake news) is above 0.98, meaning the model is accurate when it predicts a class, with very few false positives (or negatives). The F1-Score are also above .98 which suggest the model maintains a strong balance between precision and recall.

To further evaluate the model, a Confusion Matrix was presented. It showed that 50 news were predicted as fake (when they were real) and 63 news were predicted as real (when they were fake). This suggests that the model has more trouble with fake news classification.

In addition, a ROC Curve was presented, which showed the classification power of the model.

Although the classification model was already performing well, hyperparameter tuning was still implemented to further optimize its performance. After finding the best hyperparameter, the model was retrained and evaluated. A final evaluation was done with the test set. The test accuracy improved, now .9895. The precision, recall and F1-score for each type of news improved to 0.99. The confusion matrix showed an improvement also, where now 35 news were predicted as fake (when they were real) and the same number were predicted as real (when they were fake).

5. Model Interpretation

To interpret model predictions, the coefficients were analyzed to identify the top 15 positive and negative features. These features align with intuitive patterns linking names, publication types, or contexts to credibility. For example, "reuters" strongly aligns with real news as it is a reliable news agency [1], while terms like "hillary" and "breitbart" are associated with polarizing political narratives and could be linked to fake news classifications [2].

For a more detailed analysis, SHAP was used to understand how each feature (token) contributes to the model's output. The word "reuters" heavily pushed predictions toward real news, while "gop" and "obama" leaned toward fake news. Interestingly, days of the week like "friday," "monday," and "tuesday" pushed predictions toward real news, likely reflecting associations with timely reporting.

To evaluate potential bias in the model, subsets of the data were analyzed based on the subject mentioned in Task 1, which represented different news categories. The model performed consistently well across all subgroups, achieving similar accuracy (~98-99%) and strong classification metrics (precision, recall, F1-scores close to 0.99), possibly indicating no significant bias.

Finally, feature perturbation further tested the model's sensitivity to specific terms. Reducing the influence of "reuters" lowered accuracy to 0.981, confirming its importance in predictions. In contrast, reducing "obama" slightly increased accuracy to 0.989, suggesting a weaker overall impact on model decisions. Overall, the model performed exceptionally well on the task of real and fake news classification.

To improve the model's performance and robustness, I recommend exploring alternative models like Random Forest, for potentially better results. Expanding the feature set to include metadata (such as author) could enhance classification accuracy. Additionally, ethical considerations, such as reducing bias and ensuring fairness, should be explored to ensure the model is used responsibly when classifying news.

References:

[1] Ad Fontes Media. (n.d.). *Reuters Bias and Reliability*. https://adfontesmedia.com/reuters-bias-and-reliability/

[2] Ad Fontes Media. (2023, January 20). Breitbart Bias and reliability. https://adfontesmedia.com/breitbart-bias-and-reliability/