Abstract

There is a lot of news across various social media platforms and one of the biggest challenges in this era is to tell if the news you read is legitimate or fake. This is due to the complexity of natural language and diversity in the sources of information. We tried to address this problem by applying machine learning algorithms combined with sentiment analysis to classify news statements as true, false, or partially true. We used several classification models including Naive Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbors. There was class imbalance in the dataset which was handled using over-sampling technique SMOTE. The results demonstrated that oversampling improved accuracy slightly and the overall classification performance remained moderate, highlighting the difficulty of detecting the fake news.

Introduction

The rise in the usage of social media and online news outlets has revolutionized the spread of information. However, it also facilitated the rapid spread of misinformation. Identifying fake news manually has become a challenge due to the amount of content that is generated daily. As a result, automated approaches like using machine learning techniques to identify the fake news have become crucial. This project explores the application of machine learning models and sentiment analysis to detect fake news from textual statements.

Background

Existing papers have employed various techniques for detecting the fake news, ranging from feature engineering to complex deep learning models. Classical machine learning classifiers like Naive Bayes, Support Vector Machines, and Decision Trees have been effective for basic text classification tasks. Some papers also considered sentiment analysis, with the hypothesis that fake news often exhibits extreme emotional tones. Nonetheless, challenges like class imbalance, subtle linguistic patterns, and sarcasm detection remain significant obstacles in fake news detection.

Methods and Materials

I. Dataset

The dataset that we used is the LIAR dataset, consisting of short political statements labeled into six categories (‘Pants-On-Fire’, ‘False’, ‘Barely-True’, ‘Half-True’, ‘Mostly-True’, ‘True’). It has 10,000+ rows and 13 columns.

II. Preprocessing

The Liar dataset reflects the real-world data hence it is very noisy and needs processing and cleaning. We labeled all columns after understanding each feature, selected only the ‘Statement’ and ‘Label’ columns, checked for missing values and converted the categorical label into numerical data for further processing. Refer to figure 1

index	Statement	Label
0	Says the Annies List political group supports third-trimester abortions on demand.	false
1	When did the decline of coal start? It started when natural gas took off that started to begin in (President George W.) Bushs administration.	half-true
2	Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on Iran."	mostly-true
3	Health care reform legislation is likely to mandate free sex change surgeries.	false
4	The economic turnaround started at the end of my term.	half-true

Figure 1

III. Feature Extraction

We used Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to convert the textual data into numerical format to apply machine learning models on it.

IV. Sentiment Analysis

We added a feature of sentiment analysis by generating the sentiment score using the TextBlob library of python.

V. Models Applied

We applied the following models

1. Naive Bayes
2. Decision Tree
3. K-Nearest Neighbors (neighbors=5)
4. Logistic Regression

We also explored KMeans clustering to identify the groupings of the statements

VI. Data Balancing

We checked for class distribution of the 'Label' feature and found it was imbalanced and then used oversampling technique SMOTE for the minority classes to make the data more balanced. Refer to figure 2 and 3.

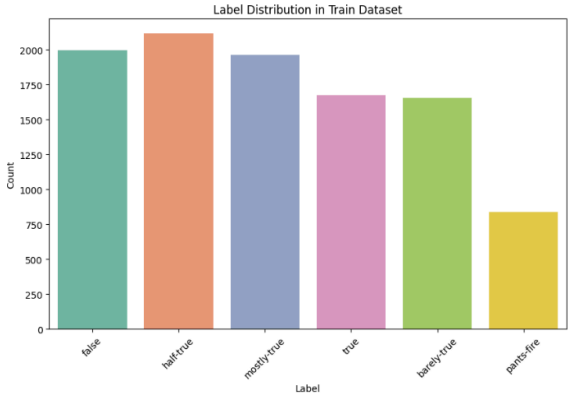

Figure 2: before SMOTE

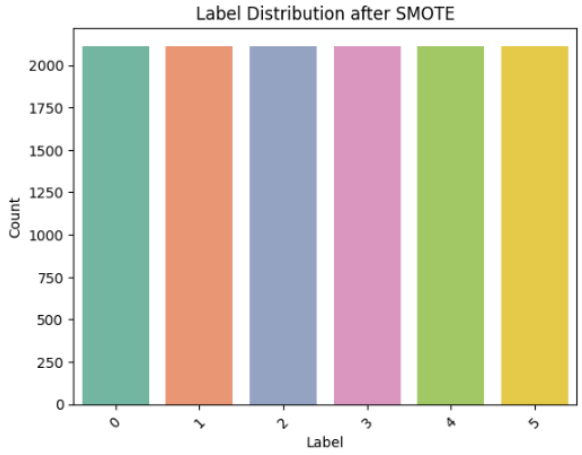

Figure 3: After SMOTE

VII. Evaluation Metrics

For evaluation of the model, we used accuracy, precision, recall, F1-score, and confusion matrix.

Results

a. The distribution of sentiment scores is heavily centered around the neutral mark, indicating that most political statements in the dataset are phrased in a relatively neutral manner, with fewer statements showing extreme positivity or negativity. We observed that there is no correlation between the label of the dataset and the sentiment score. The sentiment score just indicates the tone of the statement.

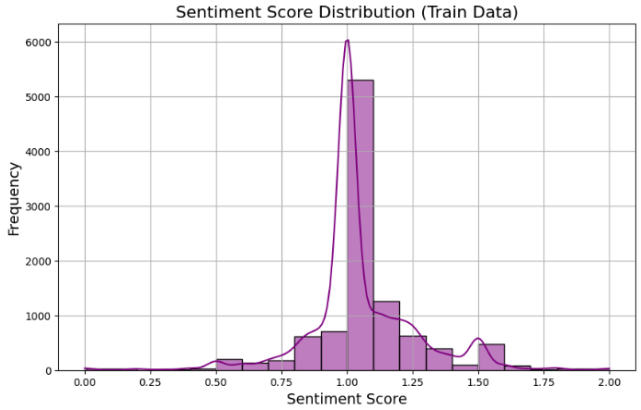

Figure 4

b. Naive Bayes showed an accuracy of approximately 23.68%. Figure 5 shows the confusion matrix of the Naive Bayes model. The square where True Label is equal to the Predicted Label is diagonal and dark it means, correct predictions (high true positives). Off-diagonal dark squares refers to lots of misclassifications (the model often confuses these two labels). *The confusion matrix illustrates the distribution of model predictions versus true labels. Darker squares along the diagonal indicate correct predictions with higher frequency,*

while darker off-diagonal squares highlight frequent misclassifications between similar classes.

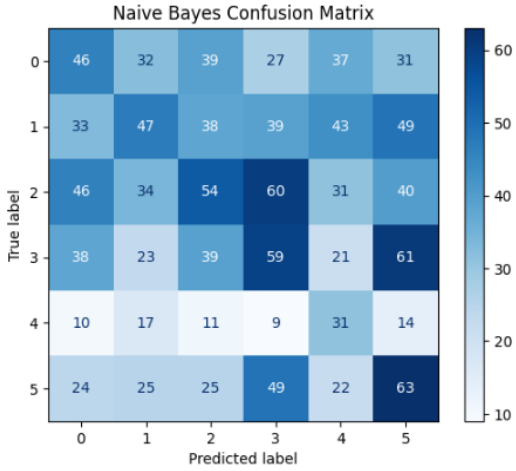

Figure 5

c. Decision tree showed an accuracy of approximately 22% and it seems to overfit generally. In figure 6 you can observe the confusion matrix for Decision Tree

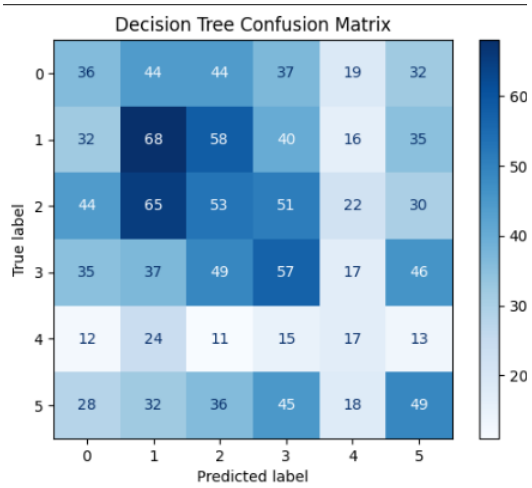

Figure 6

d. K-Nearest Neighbors model with 5 neighbors, exhibited an accuracy of approximately 16%. It performed worse than both Naive Bayes and Decision tree. It worked slowly too. Figure 7 shows the confusion matrix of KNN model

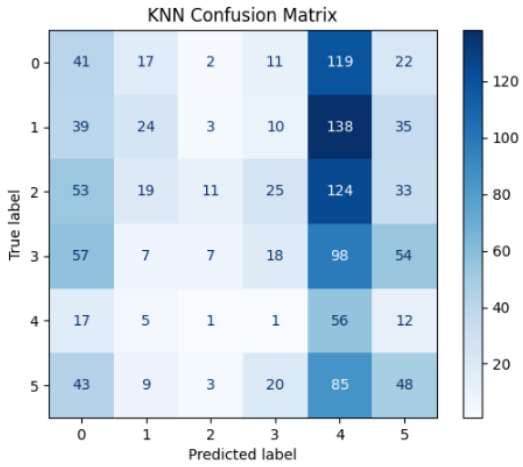

Figure 7

e. Logistic Regression showed an accuracy of 24.15% which is slightly better than Naive Bayes but almost similar. Logistic Regression outperforms Naive Bayes because unlike Naive Bayes, which assumes independence between features, Logistic Regression accounts for possible correlations between features. This makes it more robust when dealing with the complexities in the LIAR dataset. Refer to figure 8 for the visual representation of the confusion matrix.

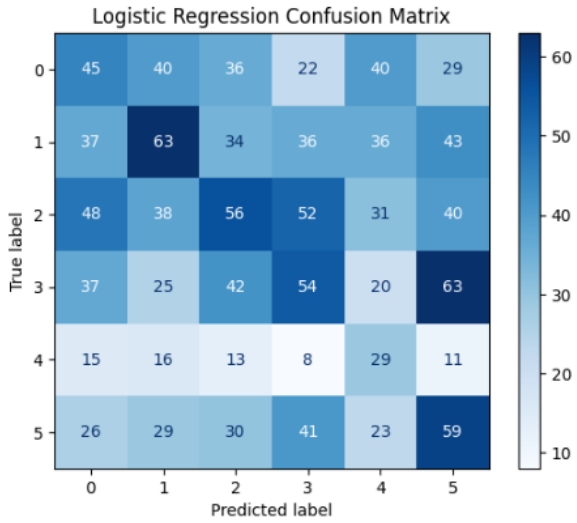

Figure 8

f. We explored the unsupervised algorithm of clustering too. We chose 7 clusters and checked the homogeneity score which was 0.0183. This indicated that the clustering doesn't separate the data well. We tried to reduce the number of clusters or increase it but 7 seems to be the most optimal number of clusters. The result is as low as this because text data is tricky to cluster since the context and meaning behind the words are not captured well by just TF-IDF alone. Refer to figure 9 for the confusion matrix.

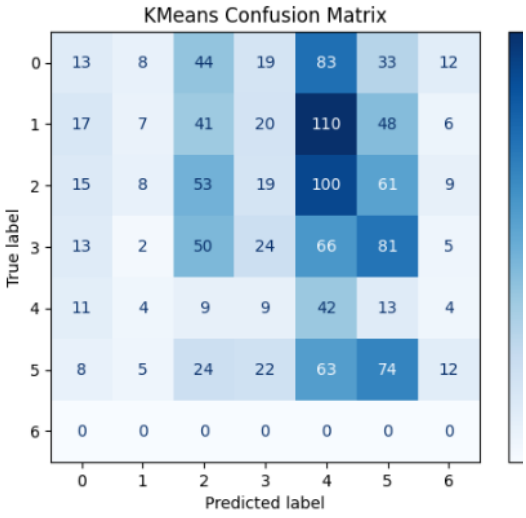

Figure 9

g. Figure 10 shows the comparison of model performance scores for Naive Bayes, Decision Tree, Logistic Regression, K-Nearest Neighbors (KNN) with 5 neighbors, and KMeans Clustering. The scores represent classification accuracy for supervised models and homogeneity score for the unsupervised KMeans model. Naive Bayes achieved the highest accuracy among the models evaluated.

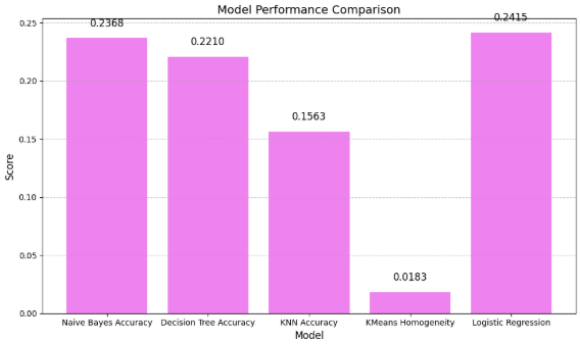

Figure 10

Conclusion

This project explored the application of machine learning algorithms to classify the textual statements into how truthful they are. Various algorithms like Naive Bayes, Decision Tree, K-Nearest Neighbors (KNN) were implemented and evaluated. Unsupervised algorithm K-Means Clustering was also explored. The dataset was balanced using the oversampling techniques to address class imbalance and improve the model performance.

Among all the models applied, Logistic Regression achieved the highest accuracy. Logistic Regression tends to perform better when the feature relationships are non-linear, as it uses the logistic function to map predictions to probabilities. On the other hand, Naive Bayes, while simple and fast, may struggle with the assumption of feature independence in datasets. However, overall performance remained moderate due to the complexity and subtlety of the natural language data. A sentiment analysis extension was also incorporated in this project to further enrich the understanding of the textual statements by evaluating the tone of the statements.

While the models provided meaningful insights, there were various limitations of the study.
1. Simpler models were unable to capture the complex linguistic patterns

2. Sentiment analysis contributed only modest gains, indicating that emotional tone alone isn't sufficient for fake news detection
3. Despite the use of oversampling techniques, some minority classes remained difficult to classify.

Future work could involve exploring transformer-based models (like BERT), ensemble learning, or sarcasm detection modules to further enhance fake news detection capabilities.

References

[1] *W. Y. Wang, "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.*