

Fake News Detection with Machine Learning







VANCE AI



Team Members Information

- Sujoy Paul Dakkumalla, 700744252.
- Venkata Sai Varun Mooraboina, 700744268.
- Pranavi Guttikonda, 700744258.
- Sai Kaushik peesari, 700744275.

Roles and Responsibilities

-  **Sujoy Paul Dakkumalla, 700744252:**
 - **Worked on Dataset, Data Preprocessing, stemming and lemmatization, KNN and Documentation.**
-  **Venkata Sai Varun Mooraboina, 700744268:**
 - **Worked on Dataset, Data Visualization, tf-idf creation, SDG Classifier and Documentation.**
-  **Pranavi Guttikonda, 700744258:**
 - **Worked on POS tagging and tf-idf_ngrams creation, and SVM model building.**
-  **Sai Kaushik peesari, 700744275:**
 - **Worked in Naive Bayes, Random Forest and Logistic Regression Algorithm.**

Introduction

Fake news has quickly become a society problem, being used to propagate false or rumour information in order to change people's behaviour. It has been shown that propagation of fake news has had a non-negligible influence of 2016 US presidential elections. A few facts on fake news in the United States:

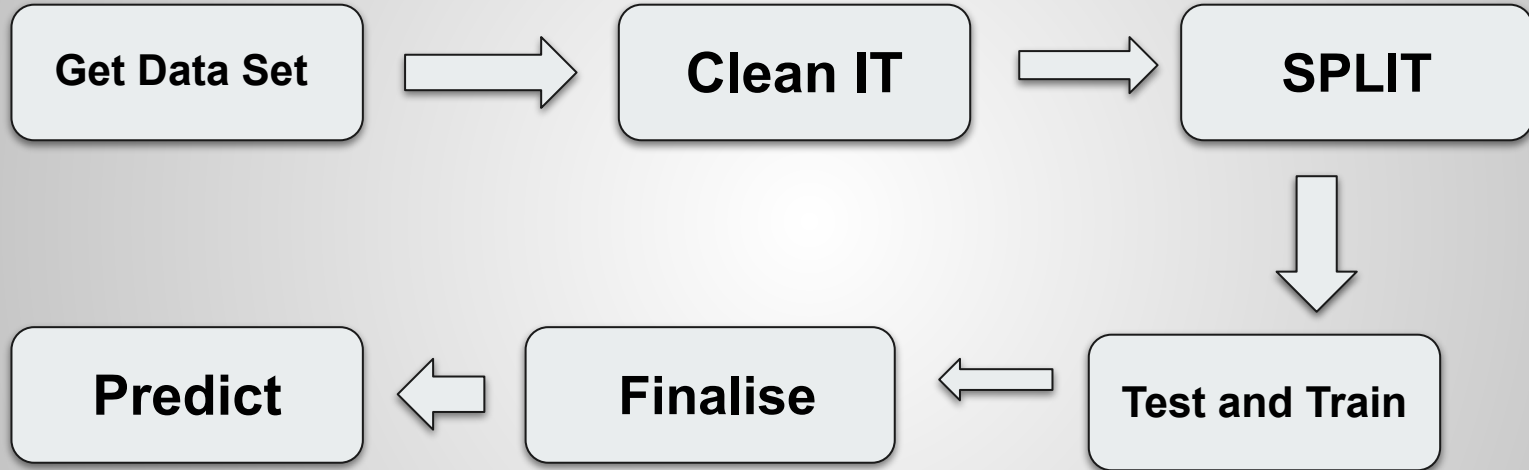
- 62% of US citizens get their news for social medias.
- Fake news had more influence on Social Media than mainstream news.

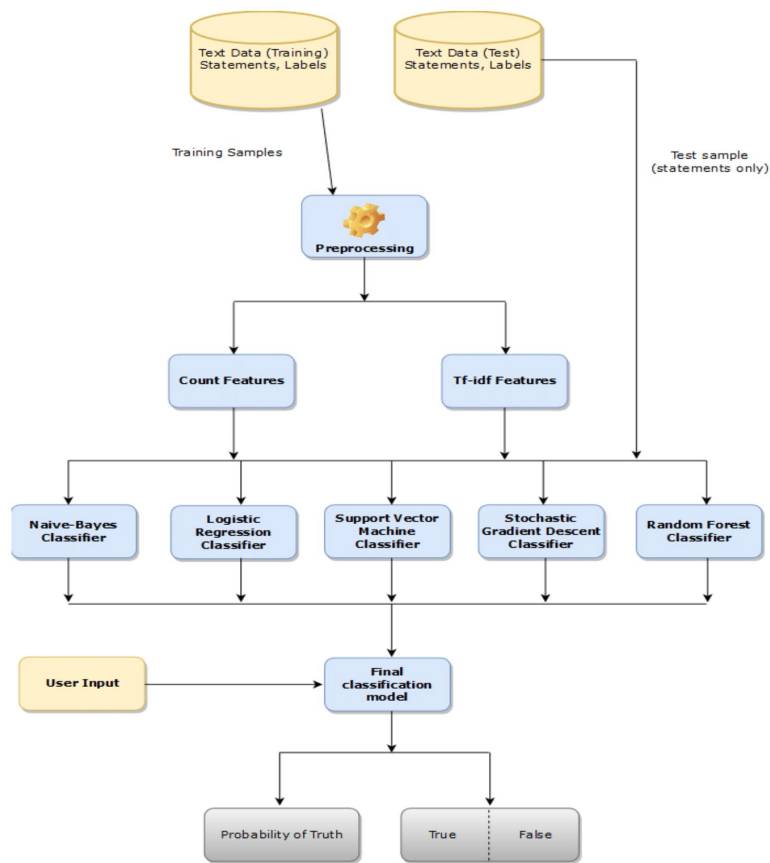
Fake news has also been used in order to influence the referendum in the United Kingdom for the "Brexit".

Execution Plan.

For the execution of the project, we have used Python as it has various modules and packages to implement machine learning algorithms. The dataset is taken from Kaggle. The data includes both fake and truthful news articles from multiple domains. The truthful news articles published contains a true description of real-world events, while fake news websites contain claims that are not aligned with facts. The input features are then used to train the different machine-learning models. Each dataset is divided into training and testing set with a 70/30 split, respectively. The articles are shuffled to ensure a fair allocation of fake and true articles in training and test instances. The machine learning algorithms are trained with different hyperparameters to achieve maximum accuracy for a given dataset, with an optimal balance between variance and bias. Each model is trained multiple times with a set of different parameters using a grid search to optimize the model for the best outcome. In this way, we applied machine learning techniques to the dataset and obtained good results.

Plan of Action:





Related Work.

There are two main categories of state of the art that are interesting for this work: previous work on fake news detection and on general text classification. Works on fake news detection is almost in existent and mainly focus in 2016 US presidential elections or does not use the same features. That is, when this work focus on automatic features extraction using machine learning and deep learning, other works make use of hand-crafted features, such as psycholinguistic features which are not the goal here. Current research focus mostly on using social features and speaker information in order to improve the quality of classifications.

In addition to texts and social features, Yang used visual features such as images with a convolutional neural network. Wang also used visual features for classifying fake news but uses adversarial neural networks to do so.

Data Preprocessing

The code reads in three CSV files - test, train, and valid. The data is then observed using the `data_obs()` function, which displays the size and head of each dataset.

`create_distribution()` function is used to visualize the distribution of the classes in the datasets using a countplot.

Additionally, the `data_qualityCheck()` function is used to check for any missing values in the datasets and display information on the datasets

The `process_data()` function is included to preprocess the text data by converting all words to lowercase, stemming the words, and removing stop words. However, it is currently commented out and not utilized in the code.

TF-IDF

- TF-IDF (Term Frequency Inverse Document Frequency) is a numerical statistic that is commonly used in natural language processing (NLP) and information retrieval to represent the importance of a term in a document or a corpus. The term frequency is the number of times a term appears in a document, while the inverse document frequency is a measure of how rare a term is in the corpus.
- A term that appears frequently in a document but rarely in the corpus as a whole is likely to be more important to that document than a term that appears frequently in both the document and the corpus.
- To calculate the TF-IDF score for a term in a document, we multiply the term frequency by the inverse document frequency.
- The inverse document frequency is typically calculated as the logarithm of the total number of documents in the corpus divided by the number of documents that contain the term.
- TF-IDF scores are used to represent the importance of a term in a document or a corpus. They are often used as features in machine learning models for NLP tasks such as text classification, clustering, and information retrieval.

N-grams Creation

The n-gram feature extraction is a way of breaking down text into smaller chunks of contiguous words to capture more information about the context. For example, a bigram ($n=2$) for the sentence "The cat sat on the mat" would be "The cat", "cat sat", "sat on", "on the", "the mat".

- In the program, a TF-IDF vectorizer is used to create n-gram features, where stop words (commonly used words like "the" and "and") are removed, and n-grams of size 1 to 4 are created to capture the context and relationships between different words in the text.
- This technique helps in identifying important words and phrases in the text, which can then be used to detect fake news by comparing with a pre-labeled dataset.

POS TAGGING

Part-of-speech (POS) tagging is the process of labeling words in a sentence with their corresponding part of speech, such as noun, verb, adjective, etc.

- POS tagging can be used to identify the usage of specific parts of speech that are commonly associated with fake news. For example, fake news articles may contain more adjectives, adverbs, or exaggerated language than real news articles. By analyzing the POS tags of the words in a given text, we can identify these patterns and use them to distinguish between real and fake news.
- To perform POS tagging, a machine learning model is trained on a corpus of labeled data, where each word is annotated with its corresponding POS tag. The trained model can then be used to automatically assign POS tags to new, unlabeled text data.
- In the program, a POS tagging model is trained on the Treebank corpus, and the resulting model is used to extract features from the news articles for fake news detection.

Naive-Bayes

- **Naive bayes** is a simple algorithm that assumes that the features are independent and uses the Bayes theorem to calculate the probabilities of each class.
- I developed a pipeline that converts news statements into word counts and uses the multinomial distribution in order to predict the validity of news statements. The classifier exceeded random guessing with a mean accuracy score of 0.61 on the test data.
- In the Naive classifier we use the two functions `NBCV` and `nb_clf`
- The classification analysis can be used to predict the probability of news article being fake or real.
- Overall, the Naive Bayes algorithm is a popular and effective choice for text classification tasks, especially when dealing with large datasets.

Logistic Regression

- **Logistic regression** is a popular and powerful algorithm that uses a logistic function to model the probability of each class.
- I used logistic regression to classify news statements as true or false, based on their word counts. I built a pipeline that transforms the statements into word counts and applies the logistic function. The classifier achieved a mean accuracy score of 0.62 on the test data, which is slightly better than naive bayes.
- I used this algorithm because it is popular and powerful, and it can model the probability of each class. It can also handle binary or multiclass problems, and it can be easily tuned with different parameters. It is also fast and simple to implement with sklearn.

SVM Classifier

SVM (Support Vector Machine) is a type of machine learning algorithm that can be used for classification tasks, such as detecting fake news. In the program, SVM classifier is used to classify news articles as either real or fake based on their features.

- The SVM classifier is trained on a set of labeled news articles, where each article is represented by a set of features, such as the frequency of certain words or the presence of certain patterns of words.
- In the program, a pipeline is used to combine the TF-IDF vectorizer (which creates numerical features from the text data) and the SVM classifier.
- To optimize the performance of the classifier, a grid search is performed to find the best combination of hyperparameters (such as the range of n-grams to consider and whether to use IDF weighting or not). The accuracy of the classifier is recorded for each combination of hyperparameters.
- Once all the combinations have been evaluated, the grid search returns the set of hyperparameters that resulted in the highest accuracy score. By tuning the hyperparameters using the grid search method, the performance of the SVM classifier can be optimized to achieve the highest possible accuracy on the test data.

SDG CLASSIFIER

- SDG Classifier is a variant of the linear Support Vector Machine (SVM) that uses stochastic gradient descent optimization for the loss function. The algorithm works by iteratively updating the model parameters using small random batches of training data, which makes it efficient for large datasets.
- The loss function used in the SGDClassifier is hinge loss. In addition to the hinge loss, the SGDClassifier can use other loss functions as well, such as logistic loss (for binary classification) or softmax loss (for multiclass classification). It can also apply regularization to prevent overfitting.
- The SGD Classifier is a versatile algorithm that can handle various types of data, including text, image, and numerical data. It has been widely used in natural language processing (NLP) tasks, such as text classification and sentiment analysis.

Random Forest

The feature selection and random forest classification. The feature selection step is done using the count vectorizer method, which converts the text data into a matrix of token counts. The random forest classifier is then trained using the transformed data with 200 estimators and 3 jobs for parallel processing.

The model is trained on the training dataset 'Statement' column and the 'Label' column is used as the target variable. The trained model is then used to predict the labels of the test dataset's 'Statement' column.

Regression analysis can be used to predict the probability of a news article being fake or not.

Evaluating the results...

We evaluated the performance of these classifiers using the F1 score and found that the random forest classifier performed the best with an F1 score of 0.70.

However, we wanted to enhance the features by using term frequency weights with various n-grams, and we applied this technique to the same classifiers. We found that the random forest classifier still performed the best with an F1 score of 0.76.

We also performed K-fold cross-validation on all the classifiers, and the results showed that the random forest classifier with n-grams had the highest accuracy of 93%.

REFERENCES:

- Jeffrey Gottfried and Elisa Shearer. News Use Across Social Media Platforms 2016. Pew Research Center, 2016.
- Craig Silverman and Lawrence Alexander. How teens in the Balkans are duping Trump supporters with fake news. BuzzFeed News, 3, 2016.
- Craig Silverman and Lawrence Alexander. How teens in the Balkans are dumping Trump supporters with fake news. BuzzFeed News 3.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. J. Mach. Learn. Res. , 9:1871 – 1874..
- Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for idf.
- Harry Zhang. The Optimality of Naive Bayes. page 6.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9:1735–1780.
- Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1):22 – 36.

спасибо 谢谢
GRACIAS

THANK YOU

ありがとうございました MERCI

DANKE धन्यवाद

شُكراً OBRIGADO