A project Report

on

FAKE NEWS DETECTION

submitted for partial fulfillment of the requirements

of the course

Machine learning Lab

By

A ANTO NIGIN (142202001)

ADITYA DANDRIYAL (142202003)

ABHINAV RAJ (142202026)

under the guidance of

Dr. C.K Narayanan

IIT PALAKKAD

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD
PALAKKAD - 678557, KERALA

1. INTRODUCTION

Fake news is false or misleading information presented as news. Fake news often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue. The prevalence of fake news has increased with the recent rise of social media and this misinformation is gradually seeping its way into the mainstream media. Fake news can reduce the impact of real news by competing with and also carry the potential to undermine trust in serious media coverage. Thus a need arises to identify whether a news is fake or not. With this project we aim to do the same.

2. METHODOLOGY

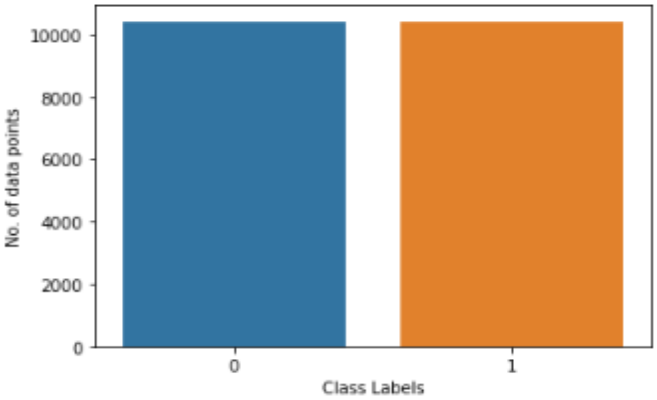
The objective of this project is to classify whether a news is fake news or not , thus a binary classification problem with labels 0 for true news and 1 for fake news.

1. Dataset: The dataset used was downloaded from kaggle.It has 20k data points with the following attributes: id: unique id for a news article , title: the title of a news article, author: author of the news article, text: the text of the article; could be incomplete, label: a label that marks the article as potentially unreliable (1: fake , 0: real)
2. Data-Preprocessing: Firstly we dealt with the null values in the dataset by replacing them with an empty string. Then we merged the columns ‘title’ and ‘author’ into a single column and used this column only as we proceeded further .Since the data is textual in nature so stemming , lemmatization and stopwords removal techniques were used to clean the data.
3. Vectorization Techniques: Converts data from its raw textual format into vectors of real numbers so we can feed it to a machine learning model .The vectorization techniques used by us are:
    - Bag of Words: in this method a sparse matrix is created for the input, out of the frequency of vocabulary words..In this sparse matrix each row is a document and each column represents word in a corpus
    - TF-IDF vectorizer: we in this method a sparse matrix is created for the input, out of the tf-idf values of vocabulary words..In this sparse matrix each row is a document and each column represents words in a corpus
    - Word2Vec:This approach uses the power of a simple neural network to generate word embeddings .Each word is represented as a 100 dimensional vector and words having similar meaning exist in close proximity to each other in the 100 dimensional hyperspace. Each document is represented as a vector by taking the mean of all the words in the document in vector format.
4. Feature Reduction: In order to drop some columns and reduce the matrix dimensionality of the sparse matix in TF-IDF and Bag of Words we carry out some feature selection using Chi-Squared Test to determine whether a feature and the target are independent and to keep only the features with a certain p-value from the Chi-Square test.
5. Model Training: We trained various models namely Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Multinomial Naïve Bayes Classifier, K-Nearest -Neighbour and Support Vector Classifier.
   For dataset obtained through word2vec vectorization, instead of Multinomial we make use of Gaussian Naïve Bayes Classifier as Multinomial NB Classifier doesn’t take negative values.

6. Model Selection: We used K-fold cross validation with scoring parameter as accuracy to test the models and find the accuracy of all the models. Further we are using Hypothesis testing (using 5x2 CV Paired T Test) on the top two performers of the K-fold cross validation to verify the claim.

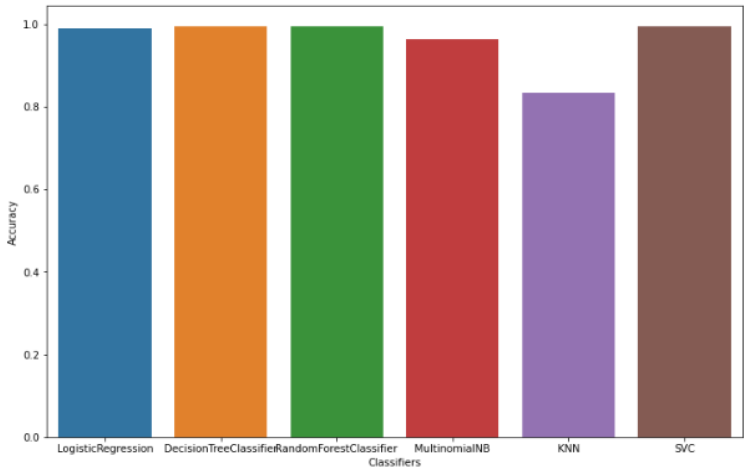
3. EXPERIMENTAL RESULTS

1. **No class imbalance was found in our dataset.**



2. **Results from the Bag of words :**

The input is a sparse matrix of size 20800x17128(17128 words in corpus and 20800 documents).



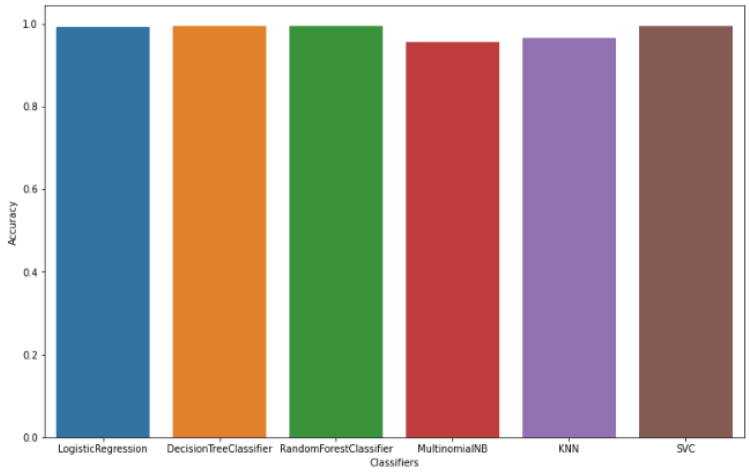
CLASSIFIERS	ACCURACY
Logistic Regression	0.990577
Decision Tree Classifier	0.995385
Random Forest Classifier	0.993894
Multinomial Naïve Bayes	0.963558
KNN (n=3)	0.834615
SVC (linear kernel)	0.994904

The top 2 models in terms of accuracy are Decision Tree and SVC.

Hypothesis testing result: p-value = 0.63749
Since p>0.05, we cannot reject the null hypothesis and may conclude that the performance of the two models are probably same.
Thus **Decision Tree Classifier** and **SVC** are the most accurate classifiers when using this approach.

3. **Results from Bag of Words + Chi Square Test for feature reduction:**

In chi square test the only features with p value less than 0.05 taken. Thus the number of features were reduced from 17128 to 3036.Thus the final input sparse matrix is of size 20800x3036.



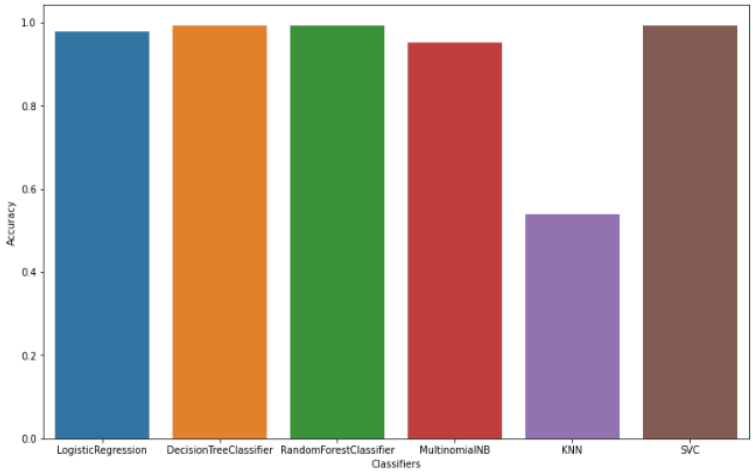
CLASSIFIERS	ACCURACY
Logistic Regression	0.99153
Decision Tree Classifier	0.993702
Random Forest Classifier	0.994087
Multinomial Naïve Bayes	0.955529
KNN (n=3)	0.965385
SVC (linear kernel)	0.995000

The top 2 models in terms of accuracy are Random Forest and SVC.

Hypothesis testing result: p-value = 0.00563
Since p<0.05, We can reject the null-hypothesis that both models perform equally well on this dataset.
We may conclude that the two models are significantly different.
Thus **SVC** is the most accurate classifier when using this approach.

4. **Results from TF-IDF Vectorizer:**

The input is a sparse matrix of size 20800x17128.



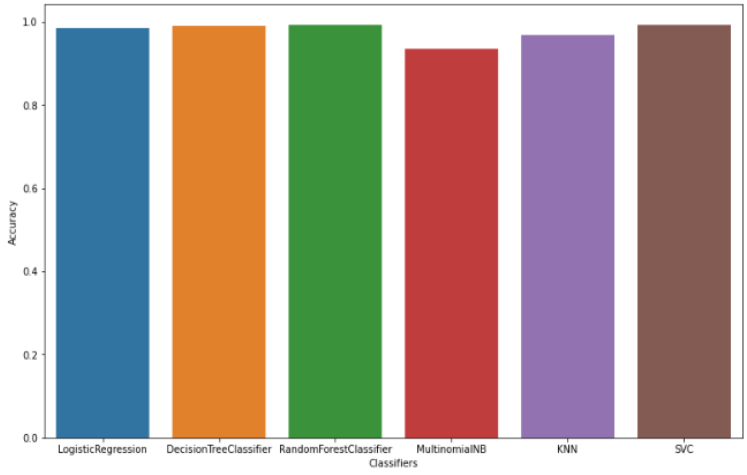
CLASSIFIERS	ACCURACY
Logistic Regression	0.978269
Decision Tree Classifier	0.993654
Random Forest Classifier	0.993750
Multinomial Naïve Bayes	0.952163
KNN (n=3)	0.539712
SVC (linear kernel)	0.992308

The top 2 models in terms of accuracy are Random Forest and Decision Tree Classifier.

Hypothesis testing result: p-value = 0.07280
Since p>0.05, we cannot reject the null hypothesis and may conclude that the performance of the two models are probably same.
Thus **Random Forest Classifier** and **Decision Tree Classifier** are the most accurate classifier when using this approach.

5. **Results from TF-IDF Vectorizer + Chi Square Test for feature reduction:**

   In chi square test the only features with p value less than 0.05 is taken .Thus the number of features were reduced from 17128 to 1063.Thus the final input sparse matrix is of size 20800x1063.



CLASSIFIERS	ACCURACY
Logistic Regression	0.985913
Decision Tree Classifier	0.990865
Random Forest Classifier	0.993365
Multinomial Naïve Bayes	0.934375
KNN (n=3)	0.968798
SVC (rbf kernel)	0.992885

The top 2 models in terms of accuracy are Random Forest and SVC.

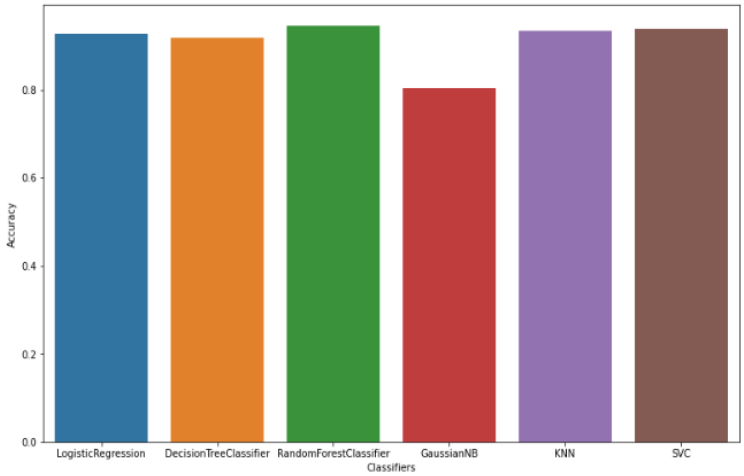
Hypothesis testing result: p-value = 0.028636

Since p<0.05, We can reject the null-hypothesis that both models perform equally well on this dataset. We may conclude that the two models are significantly different.

Thus **Random forest classifier** is the most accurate classifier when using this approach.

6. **Results from Word2Vec:**

   All 20800 documents is represented as a vector of size 100.



CLASSIFIERS	ACCURACY
Logistic Regression	0.926875
Decision Tree Classifier	0.917452
Random Forest Classifier	0.946010
Gaussian Naïve Bayes	0.804375
KNN (n=7)	0.933221
SVC (linear kernel)	0.938558

The top 2 models in terms of accuracy are Random Forest and SVC.

Hypothesis testing result: p-value = 0.1710219

Since p>0.05, we cannot reject the null hypothesis and may conclude that the performance of the two models are probably same.

Thus **Random Forest Classifier** and **SVC** are the most accurate classifiers when using this approach.

7. **Overall comparizon among the best models:**

S.No.	VECTORIZATION TECHNIQUE	CLASSIFICATION ALGORITHM	ACCURACY
1	Bag of Words	Decision Tree & SVC (Linear kernel)	99.538 & 99.49 %
2	Bag of words + Chi Square Test	SVC (Linear kernel)	99.50 %
3	TF-IDF	Random Forest & Decision Tree	99.375 & 99.365 %
4	TF-IDF + Chi Square Test	Random Forest	99.336 %
5	Word2Vec	Random Forest & SVC (Linear kernel)	94.60 & 93.85 %

4. CONCLUSION

Analyzing the text data is critical. This project is focused on applying vectorization techniques such as bag of words, TF-IDF and word2vec to preprocess and vectorize text and evaluate its effectiveness by running them through Logistic regression, Decision Tree, Random Forest , Multinomial Naïve Bayes , K-Nearest Neighbors and SVM classifiers.
The result of the above approaches showed that bag of words vectorization technique with Decision Tree or SVC (Linear kernel) showed the highest accuracy, but since **Bag of words vectorization with features reduction using chi-square test with SVC (Linear kernel) classifier** also has almost the same accuracy ,hence considering the benefits of feature reduction this approach comes out to be the better one for text classification.

REFERENCES:

1. Fake news dataset link: https://www.kaggle.com/competitions/fake-news/data
2. Salma El Shahawy – Evaluate ML Classifier Performance using Statistical Hypothesis Testing in Python (https://towardsdatascience.com/evaluate-ml-classifier-performance-using-statistical-hypothesis-testing-in-python-e4b90eb27dce)
3. Sampath kumar gajawada - Chi-Square Test for Feature Selection in Machine learning (https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223)
4. Jason Brownlee - Statistical Significance Tests for Comparing Machine Learning Algorithms (https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/)
5. Dilip Valeti – Classification using Word2Vec (https://medium.com/@dilip.voleti/classification-using-word2vec-b1d79d375381)