Introduction

Fake News Detection: Build a system to identify unreliable news articles.

The advent of the World Wide Web and the rapid adoption of social media platforms paved the way for information dissemination. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading. In addition, since the automated classification of news articles is a challenging task, we need ways to predict the genuineness of the news articles. So in this project, we propose to build an automated machine learning classifier for the classification of news articles.

Data Transformation

Initially, we removed rows from the data that contained blank or missing entries. We then transformed the text to lowercase in order to map words with different cases to lowercase form since python is case sensitive. Subsequently, we used a tokenizer to split news text into a set of words with the help of python's nltk (Natural Language Toolkit) library to do further analysis on words.

After that, we iterate over the above set of words, check whether it is a stop word or a non-alphabetical word, and remove them. By doing so, we remove the low-level information text from our dataset and draw more focus on important information. Removing these stop words also helped in reducing the dataset and thus reducing the training time.

Further, to normalize the words, we used the Stemming technique to reduce a given word to its root word by chopping a part of the word. However, we observed that stemming resulted in words that are not actual words, changed the meaning of the word, and also could not convert words with different forms based on grammatical constructs. To overcome these limitations, we used a technique called Lemmatization that considers the context of the words and converts them into their meaningful base form. Then based on the user context, we identified the part-of-speech i.e. POS tag for the word using Wordnet to establish structured semantic relationships between the words.

The final news text generated is then passed to sklearn's TfidfVectorizer which converts raw text to a matrix of TF-IDF features. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This helps to adjust for the fact that some words appear more frequently in general. Sklearn's TfidfVectorizer incorporates the importance of words not just the frequency of words unlike the bag of words.

Machine Learning Techniques

1. Naive Bayes

Since our goal is to classify given news as fake or genuine, we use Naive Bayes as our first model as it is simple and effective. Naive Bayes is a supervised learning algorithm which applies Bayes' theorem with the 'naive' assumption of conditional independence between every pair of features given the value of the class variable. We used **MultinomialNB,** a classic naive Bayes variant used in text classification which is known to work well with tf-idf vectors. The model we used correctly classified the genuineness of news with an accuracy of **87.11.**

2. Support Vector Machines (SVMs)

A Support Vector Machine (SVM) is a supervised machine learning algorithm where we plot each data item as a point in n-dimensional space (where n is a number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes. We implemented this using the Scikit Learn inbuilt SVM classifier which can be tuned using different parameters. For our model, we have considered parameters like Kernel with values {linear, poly, rbf, sigmoid}, Gamma which takes values {scale, auto} and C values ranging from 1 to 5. The F1 score is highest for the below parameters which is **94.12**, while lowest F1 score is **66.66.**

	Kernel	**C (Regularisation)**	**Gamma**
Max	RBF	3.0	Scale
Min	Poly (Deg = 2)	1.0	Auto

3. Logistic Regression

Logistic regression is a statistical method for predicting binary classes. It works by measuring the relationship between the dependent variable (what we want to predict) and one or more independent variables (the features). It does this by estimating the probabilities with the help of its underlying logistic function. For our model, we have considered parameters like Solver with values {newton-cg, lbfgs, liblinear, sag, saga}, Penalty which takes {L1,L2}, C ranging from 1 to 5 and Maximum Iterations with values {250, 500, 1000, 1500, 2500}. The maximum F1 score is **93.60** and minimum is **91.75** for the below parameters

	Solver	**Penalty**	**C (Regularisation)**	**Maximum Iterations**
Max	Liblinear	L1	2.0	250
Min	newton-cg	none	1.0	250

Tuning

At the time of pre-processing data, we initially used the Stemming technique to change the words into their base form and observed that the classification accuracy was low. To improve the accuracy, we used a Lemmatizing technique which preserves the context of the words. This technique resulted in a significant increase in the overall accuracy. For example, Naive Bayes accuracy increased from 82.16% to 87.11%.

Also, we performed hyperparameter tuning to determine the right combination of hyperparameters that allows the model to maximize model performance. We first defined a list of relevant parameters to iterate over and then fit the models to get their corresponding F1 scores. We then selected the parameters with maximum F1 scores. This approach helped us find the optimal combination of parameters.

Role of each Team Member

Preprocessing was done by all the team members.

- Priyanka Savla - Naive Bayes
- Humaid Desai - Logistic Regression
- Anubhav Nanda - Support Vector Machine

Evaluation Details

- Confusion Matrix - In the binary news classification problem, the confusion matrix has 2 rows and 2 columns where the rows represent actual values and the columns represent predicted values.
- Precision - It is the number of actual reliable news divided by the total number of reliable news predicted. Precision can be thought of as a measure of a classifier's exactness.
- Recall - It is the number of actual reliable news divided by the total number of reliable news in the test data. Recall can be thought of as a measure of a classifier's completeness.
- F1 Score - The F1 score can be used to convey the balance between the precision and the recall. The F1 Score is 2*((precision*recall)/(precision+recall)).

	Naive Bayes	**Logistic Regression**	**Support Vector Machine (SVM)**
Accuracy	87.66	93.19	94.32
Precision	83.71	93.29	94.10
Recall	90.99	93.16	94.56
F1 Score	87.20	93.22	94.33