FAKE NEWS DETECTION USING ML

INDEX

ABSTRACT

Fake news detection refers to the process of identifying and classifying news articles, reports, or other forms of information as either genuine or fabricated with the intent to deceive or mislead readers. As the world is getting digitalize, there are numerous number of platforms to get the information. People are getting confused to know whether the news is real or fake. The proliferation of fake news poses several significant disadvantages and threats to individuals, society, and the integrity of information dissemination. Here are some key disadvantages:

1.Misinformation and Deception: Fake news disseminates false or misleading information, leading to individuals making decisions based on inaccurate or fabricated content. This can result in misunderstandings, misinterpretations, and misguided actions.

2.Damage to Trust and Credibility: The spread of fake news undermines trust in traditional media outlets and reputable sources of information. When individuals are exposed to deceptive content repeatedly, they may become skeptical of all news sources, leading to a general erosion of trust in the media and public institutions.

3.Polarization and Division: Fake news often promotes divisive narratives and amplifies existing social and political tensions. By targeting specific groups or ideologies, fake news can exacerbate polarization, fueling conflicts and hindering constructive dialogue and compromise.

4.Manipulation of Public Opinion: Fake news can be strategically deployed to manipulate public opinion, sway elections, and influence public policy outcomes. By spreading false narratives or propaganda, malicious actors can manipulate perceptions and shape public discourse to serve their own interests.

5.Economic Harm: Fake news can have economic consequences, such as stock market fluctuations, consumer misinformation about products or services, and damage to the reputation of businesses or individuals falsely implicated in misleading stories.

6.Cyber security Risks: Fake news can be used as a vector for spreading malware, phishing attacks, and other forms of cyber threats. Malicious actors may use fake news as bait to lure unsuspecting users into clicking on malicious links or downloading harmful software.

To avoid all these problems, fake news detection is mandatory. Machine learning is one of the technique that is used to solve this problem. By applying different algorithms we can easily get that the news is real or fake.

PROBLEM STATEMENT

The proliferation of fake news in online platforms presents a significant challenge to society, undermining trust in information sources, distorting public discourse, and posing threats to democratic processes. To address this issue, there is a critical need for effective fake news detection systems leveraging machine learning (ML) techniques. To solve this problem, there are certain steps to be involved.

STEPS INVOLVED

1. DATA COLLECTION:

Data collection refers to the process of gathering and accumulating data from various sources for further analysis, research, or decision-making purposes. It is a fundamental step in any data-driven project or study, providing the raw material upon which subsequent analyses, insights, and conclusions are based. Data collection can involve obtaining information from a wide range of sources, including structured databases, unstructured text, sensor readings, surveys, observations, experiments,online sites and more.

2. DATA PRE PROCESSING:

It involves transforming raw data into a format that is suitable for analysis, modeling, and visualization. Data pre processing aims to improve data quality, address missing or erroneous values, reduce noise, and prepare the data for further processing by machine learning algorithms or statistical techniques.In this step, missing values can be dropped or filled and duplicated values are removed, removing unwanted columns and dimensionality reduction all these will be handled.

3. VISUALIZATION:

Data visualization is the graphical representation of data and information using visual elements such as charts, graphs, maps, and dashboards. It is a powerful technique for conveying complex data insights in a clear, intuitive, and actionable manner, enabling users to understand trends, patterns, relationships, and outliers in the data more effectively than through raw numbers or text alone.In this step, we can get an idea about the algorithms that should be used for this project.

4. SPLITTING DATA:

Data splitting, also known as dataset splitting, refers to the process of dividing a dataset into multiple subsets for different purposes, such as model training, validation, and testing. This division is crucial in machine learning and data analysis workflows to assess the performance and generalization capability of models accurately.

TRAINING DATA: The training set is used to train the machine learning model. It contains a large portion of the dataset, and the model learns patterns and relationships within the data by iteratively adjusting its parameters to minimize a chosen loss function.

TESTING DATA: The test set is used to evaluate the final performance of the trained model objectively. It represents an independent dataset that the model has not been exposed to during training or validation. By evaluating the model on the test set, practitioners can assess its ability to generalize to new, unseen data accurately

5. MODELLING:

Data modeling refers to the process of preparing and structuring data in a format suitable for training and evaluating machine learning models. While the principles of data modeling in ML share some similarities with traditional database design, there are specific considerations and techniques tailored to the requirements of ML algorithms. By effectively modeling the data for ML tasks, practitioners can improve the performance, robustness, and interpretability of machine learning models, ultimately leading to more accurate predictions and actionable insights.

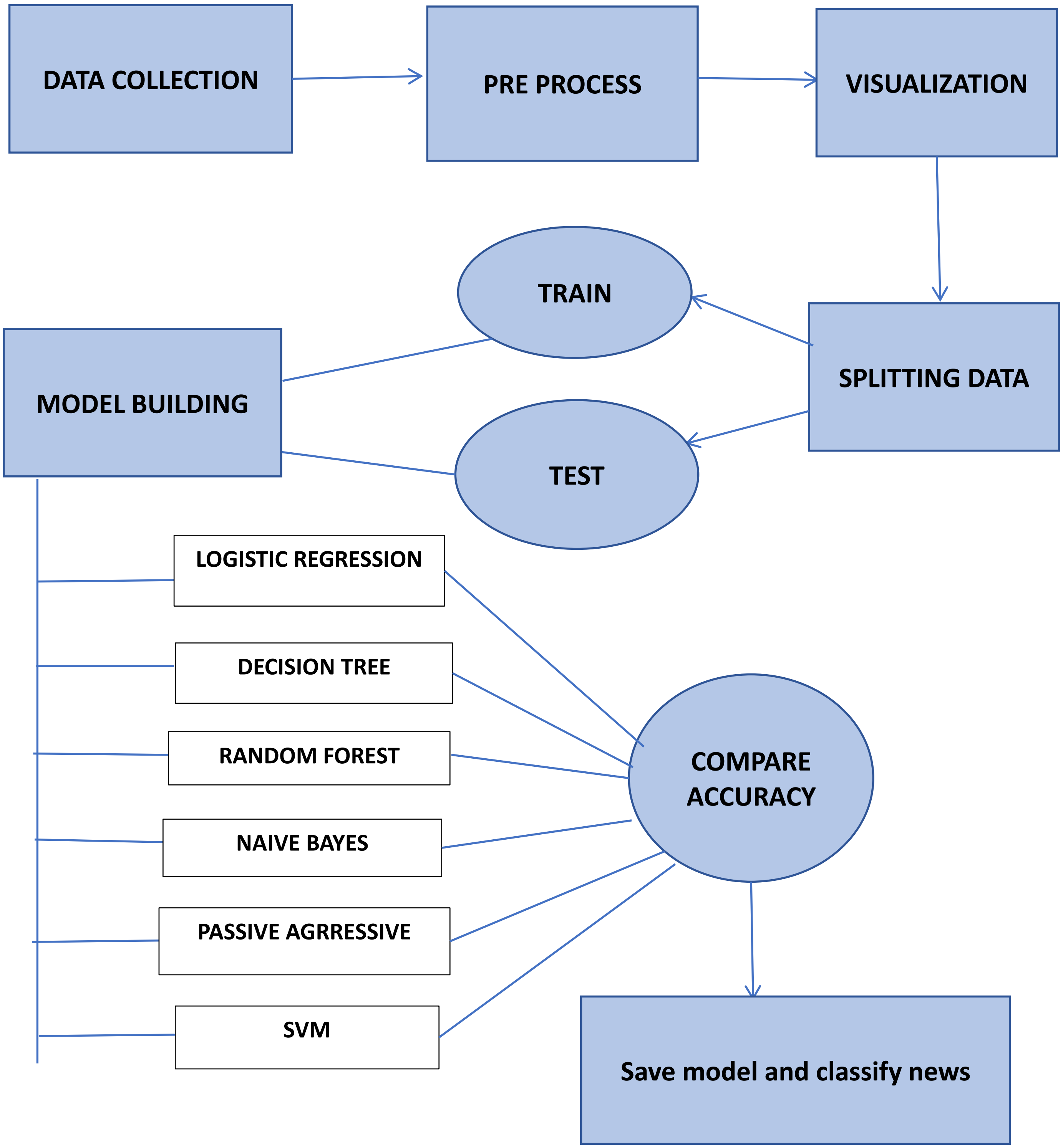
6. COMPARING ACCURACY:

This is the optional step in project. In modelling, if only one algorithm is used then there is no need to compare the accuracy. If multiple algorithms are used then compare the accuracy of every algorithm to choose the best algorithm for particular project.

7. SAVE THE MODEL:

In this step, the model can be saved for future purposes. By saving machine learning model then there is no need to always fit the model again and again. Once a model has been trained on a dataset and has achieved satisfactory performance, it is important to save it so that it can be reused for making predictions on new, unseen data without the need to retrain the model from scratch.

FAKE NEWS DETECTION ARCHITECTURE:



EXPLANATION:

- First, the dataset is collected from Kaagle.The dataset contains fields like index, title, text or news and the label.
- Now, we need to pre process the data. In the dataset, the duplicate values get removed and checking for null or NaN values detected and dropped. Remove unwanted or irrelevant columns to get more performance of model.
- The text or news field should undergo text pre processing such as:
  converting the string into lower case
  Removing hyper links from news
  Removing special characters from news
  Remove stopwords from news
  Split into tokens
- Perform Visualization on the data to know which algorithm to be applied.
- Split the dataset into training data and testing data.
- Now, Choose algorithm for the dataset to get the real or fake label.
- As the problem statement is related to classification like real or fake news, all the classification algorithms are applied in order to check the accuracy.
- The algorithms applied are logistic regression, decision tree, random forest, naive bayes, passive agressor classifier and SVM model.
- Before appliying algorithm, the text field need to be converted to numericals.
- To convert text to numerics, TFIDF vectorizer is used.
- All the algorithms are build and accuracy for each algorithms is verified and confusion matrix is also displayed.
- After performing all algorithms and confusion matrix, accuracy need to be compared.
- SVM model got highest accuracy when compared with all other algorithms.
- To save the SVM model, joblib is used.
- Web application is developed by using streamlit.

SOFTWARE REQUIREMENTS

The required modules for the project are Pandas,re, nltk, matplotlib, seaborn, sklearn, joblib and streamlit.

PANDAS:

The dataset is stored in a structured format such as CSV, Excel, or a database, pandas can be used to load this data into a DataFrame, which can then be manipulated and used for further processing.Pandas is one of the important module to build machine learning projects. It provides powerful tools for data preprocessing, including handling missing values, encoding categorical variables, and scaling numerical features.

RE:

RE is a built-in module that provides support for regular expressions (regex). Regular expressions are sequences of characters that define a search pattern, mainly for string manipulation. The re module allows you to work with regular expressions in Python, enabling you to search, match, and manipulate strings based on specific patterns.Regular expressions are powerful tools for text processing and pattern matching in Python. They are commonly used in tasks such as text parsing, data validation, and string manipulation.

NLTK:

NLTK (Natural Language Toolkit) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces and lexical resources, such as WordNet. Additionally, NLTK includes a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.In this project, nltk is used to remove stopwords from the news.

MATPLOTLIB:

Matplotlib is a powerful plotting library for Python that enables you to create a wide variety of plots, charts, and visualizations. It is extensively used for data visualization tasks in fields such as data science, machine learning, finance, and scientific computing.Overall, Matplotlib is a versatile and flexible library that provides extensive functionality for creating high-quality plots and visualizations in Python. Whether you need to explore data, analyze trends, or communicate results, Matplotlib offers the tools you need to create informative and visually appealing plots.

SEABORN:

Seaborn is a Python data visualization library based on Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics. Seaborn is designed to work well with Pandas DataFrames and NumPy arrays, making it an excellent tool for exploring and visualizing structured data. It provides a convenient and intuitive interface for exploring data and communicating insights effectively through visualizations. Whether you're analyzing data for exploratory purposes or preparing visualizations for presentations or reports, Seaborn offers the tools you need to create informative and visually appealing plots.

SKLEARN:

Scikit-learn, also known as sklearn, is a popular machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It is built on top of other scientific computing libraries such as NumPy, SciPy, and Matplotlib, and it is designed to be user-friendly, modular, and easily extensible.In this project, sklearn is used to get all the algorithms and even the accuracy metrics, classification reports. It is used for modelling purpose in this project.scikit-learn is a powerful and versatile library for machine learning in Python, suitable for both beginners and experienced practitioners. It provides a wide range of algorithms and tools for building, evaluating, and deploying machine learning models across various domains and applications.

JOBLIB:

Joblib is a Python library that provides utilities for saving and loading Python objects, especially large NumPy arrays, efficiently to and from disk. It is particularly useful for saving and loading trained machine learning models, which often involve large parameter sets or complex data structures.In this project, after comparing accuracy the SVM got highest accuracy so that model is saved using joblib.

STREAMLIT:

Streamlit is an open-source Python library that allows you to create interactive web applications for data science and machine learning projects with ease. It is designed to simplify the process of building and sharing data-centric web apps, enabling data scientists and machine learning engineers to quickly prototype and deploy their projects without requiring web development expertise. In this project, streamlit is used to design the web application.

MACHINE LEARNING:

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans.Theability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. We have three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Supervised Machine Learning:

Supervised learning is the types of machine learning in which machines are trained using well "labelled"training data, and on basis of that data, machines predict the output. The labelled data means some inputdata is already tagged with the correct output. In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Unsupervised Machine Learning:

In the previous we got to know about supervised machine learning in which models are trained using labeled data under the supervision of training data. But there may be many cases in which we do not havelabeled data and need to find the hidden patterns from the given dataset. So, to solve such types of cases in machine learning, we need unsupervised learning techniques. Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

ALGORITHMS USED:

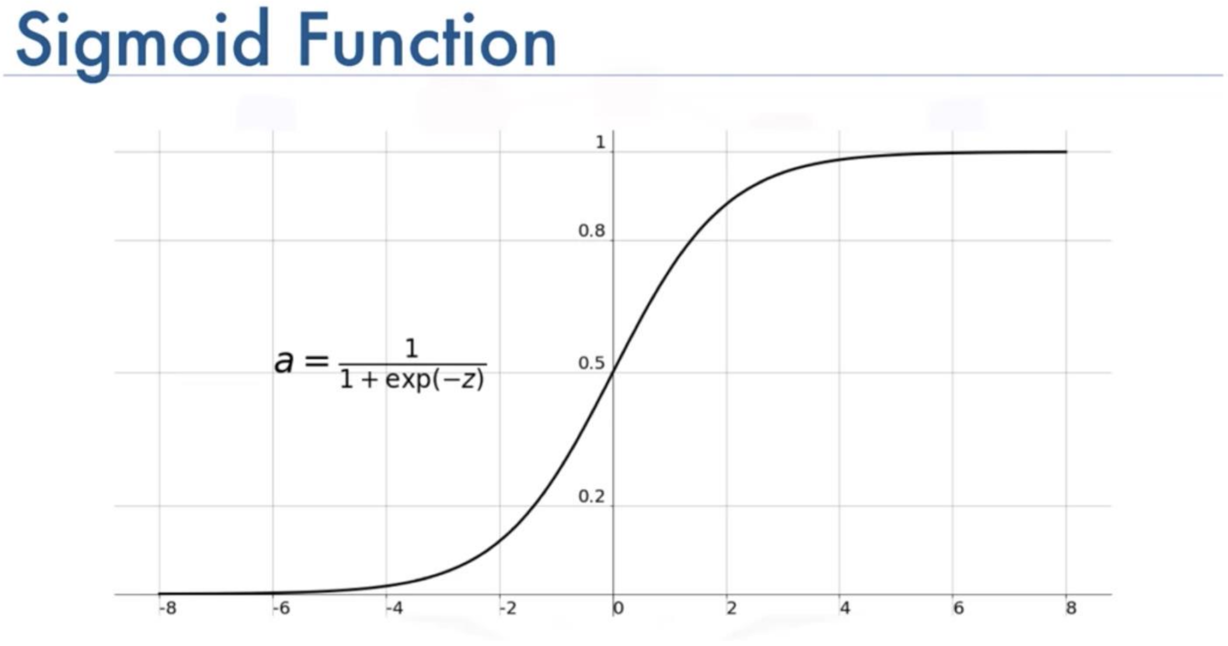
The algorithms used in project are:

1. Logistic regression
2. Decision tree
3. Random forest
4. Naive bayes
5. Passive aggressive
6. SVM

LOGISTIC REGRESSION:

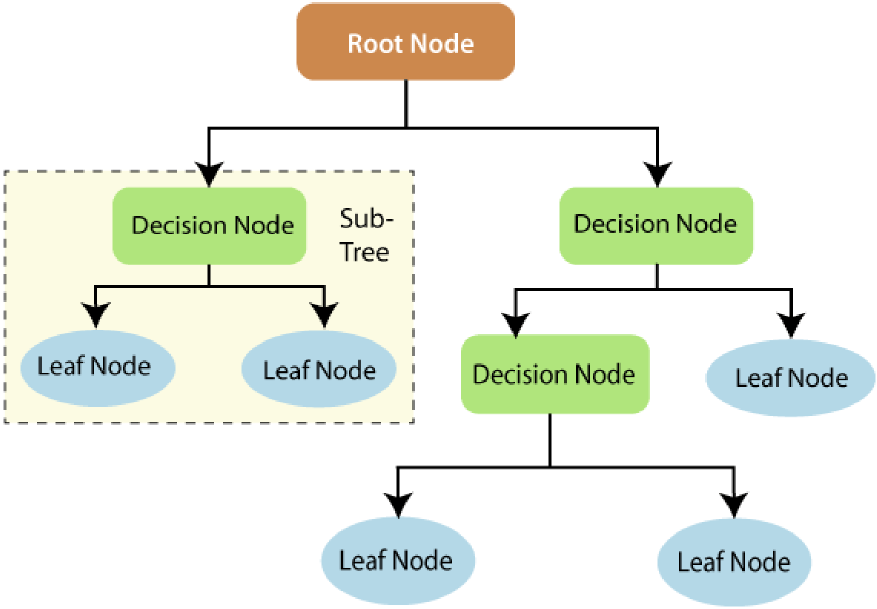
- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.



DECISION TREE:

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- In decision tree algorithm, it is mandatory to set random state otherwise it produce different values each time you run.



RANDOM FOREST:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

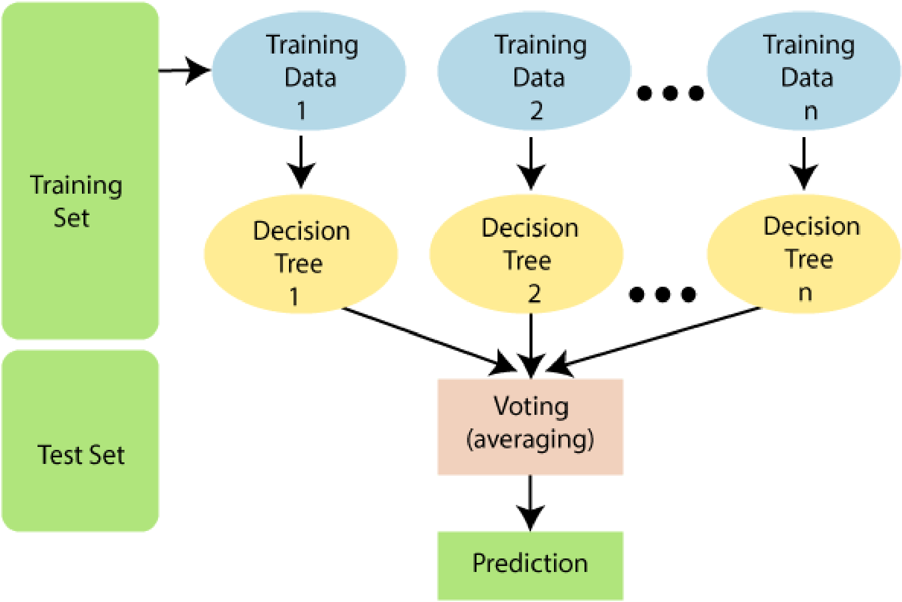
The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output.

It takes less training time as compared to other algorithms.

It predicts output with high accuracy, even for the large dataset it runs efficiently.

It can also maintain accuracy when a large proportion of data is missing.



NAIVE BAYES:

- Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

- Naive: It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

- Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$\mathrm{P(A|B)} = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

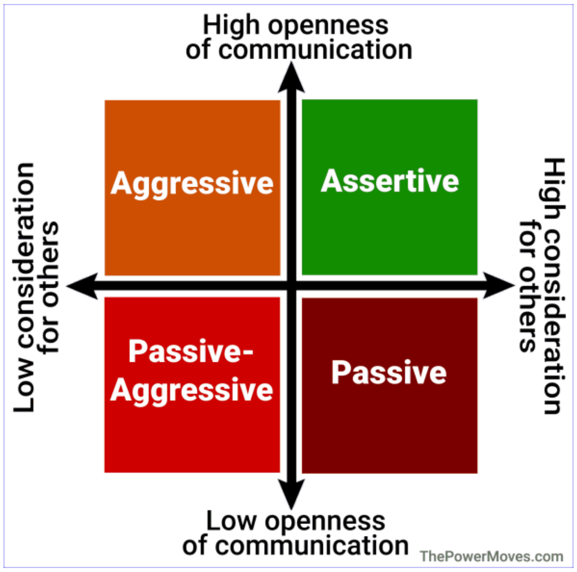
PASSIVE AGGRESSIVE CLASSIFIER:

Passive Aggressive Classifier belongs to the category of online learning algorithms in machine learning. It works by responding as passive for correct classifications and responding as aggressive for any miscalculation.Passive-Aggressive algorithms are generally used for large-scale learning. It is one of the few ‘online-learning algorithms‘. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data.

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

The key idea behind the Passive-Aggressive algorithm is to make aggressive updates to the model parameters when the current prediction is incorrect, while making passive updates when the prediction is correct. This approach allows the algorithm to quickly adapt to changes in the data distribution while still maintaining stability and convergence properties.Passive-Aggressive classifiers are often used in scenarios where memory and processing time are limited, such as online advertising, spam detection, and text classification tasks. They are efficient and easy to implement, making them a popular choice for real-time and streaming data applications. However, their performance may vary depending on the characteristics of the data and the specific variant of the algorithm used.



SVM(SUPPORT VECTOR MACHINE):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

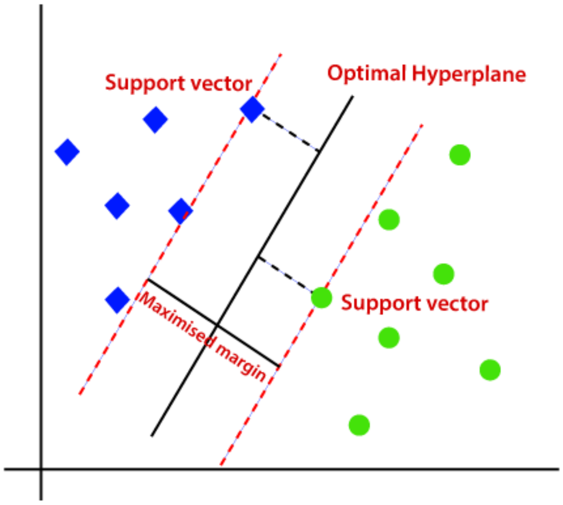
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.



SOURCE CODE

Loading the dataset:

```
df = pd.read_csv("news.csv")
df.head()
df.info()
df.describe()
df.tail()
df.nunique()
df.columns
df['label'].value_counts
df.shape
df['text'].value_counts
number_msgs=len(df['text'].unique())print("Number of news samples in dataset:",number_msgs)
```

PRE PROCESSING DATA

```
df.isnull().sum()
df.columns=['Index','Title','Text','Label']
df.columns
df.dropna()
df.drop('Index',axis=1,inplace=True)
df.columns
df.drop('Title',axis=1,inplace=True)
df.columns

df['Text']=df['Text'].str.lower()
df.head()
df['Text']=df['Text'].str.replace(r'https\S+',",regex=True)
df.head()
df['Text']=df['Text'].str.replace('[^A-Za-z\s]+', ",regex=True)
df.head()
import nltkfrom nltk.corpus import stopwords
nltk.download('stopwords')
sw=stopwords.words('english')
df['Text']=df['Text'].apply(lambda words:' '.join(word.lower() for word in words.split() if word
not in sw))
t=nltk.tokenize.WhitespaceTokenizer()l=nltk.stem.WordNetLemmatizer()def lt(text):
    return [l.lemmatize(w) for w in t.tokenize(text)]
df['Tokens']=df['Text'].apply(lt)
df.head()
df['word_length']=df['Text'].str.split().str.len()
df.head()
df.drop_duplicates(subset='Text',inplace=True)
df.shape
df.head()
```

VISUALIZATION

```
df['Label'].value_counts()
df['Label'].value_counts().plot.bar()
import matplotlib.pyplot as pltimport seaborn as sns
sns.set(color_codes=True)
plt.figure(figsize=(15,7))
cmap=['red','green']
labels=['REAL','FAKE']
for label,clr in zip(labels,cmap):
    sns.kdeplot(df.loc[(df['Label']==label),'word_length'],color=clr,shade=True,label=label)
    plt.title("Density of True and False values",color="Blue")
    plt.show()
sns.set(color_codes=True)
plt.figure(figsize=(15,7))
cmap=['red','green']
labels=['REAL','FAKE']
for label,clr in zip(labels,cmap):
    sns.kdeplot(df.loc[(df['Label']==label),'word_length'],color=clr,shade=True,label=label)
    plt.title("Real and Fake Densities",color="Blue")

import itertoolsimport collections
import pandas as pd
lt=list(df['Tokens'])
t=list(itertools.chain(*lt))
c=collections.Counter(t)
d=pd.DataFrame(c.most_common(30),columns=['words','count'])fig,ax=plt.subplots(figsize=(15,
7))
d.sort_values(by='count').plot.barh(x='words',y='count',ax=ax,color='blue')plt.title("Most
Frequent Words in dataset",color="Red")
plt.show()
```

SPLITTING DATA

```
x=df.iloc[:,0].values
y=df.iloc[:,1].values
print(x)
print(y)
```

TEXT TO NUMERICS

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=23)
from sklearn.feature_extraction.text import TfidfVectorizer
  vectorization = TfidfVectorizer() x_train = vectorization.fit_transform(x_train)
x_test = vectorization.transform(x_test)
```

LOGISTIC REGRESSION

```
reg=LogisticRegression()
model1=reg.fit(x_train, y_train)
y_pred=model1.predict(x_test)
import sklearn.metrics
accuracy1=sklearn.metrics.accuracy_score(y_test,y_pred)
print("Accuracy of Logistic Regression model:",accuracy1)
print("Accuracy of trained data:")
print(accuracy_score(y_train, model1.predict(x_train)))
print("Accuracy of test data:")
print(accuracy_score(y_test, model1.predict(x_test)))
from sklearn.metrics import accuracy_score, classification_report
classification_rep1 = classification_report(y_test, y_pred)
print("Classification report of Logistic Regression:")
print(classification_rep1)
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test,y_pred)  cm
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show()
```

DECISION TREE

```
from sklearn.tree import DecisionTreeClassifier
  dt = DecisionTreeClassifier(random_state=0)
model2=dt.fit(x_train, y_train)
y_pred=model2.predict(x_test)
import sklearn.metrics
accuracy2=sklearn.metrics.accuracy_score(y_test,y_pred)
```

```
print("Accuracy of Decision tree model:",accuracy2)
print("Accuracy of trained data:")
print(accuracy_score(y_train, model2.predict(x_train)))
 print("Accuracy of test data:")
print(accuracy_score(y_test, model2.predict(x_test)))
from sklearn.metrics import accuracy_score, classification_report
classification_rep2 = classification_report(y_test, y_pred)
print("Classification report of Decision Tree:")
print(classification_rep2)
from sklearn.metrics import confusion_matrix
 cm= confusion_matrix(y_test,y_pred)  cm
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show
```

RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=50,random_state=42)
model3=rf.fit(x_train,y_train)
y_pred=model3.predict(x_test)
accuracy3=sklearn.metrics.accuracy_score(y_test,y_pred)
print("Accuracy of Random Forest Classifier is:",accuracy3)
print("Accuracy of trained data:")
print(accuracy_score(y_train, model3.predict(x_train)))
print("Accuracy of test data:")
print(accuracy_score(y_test, model3.predict(x_test)))
from sklearn.metrics import accuracy_score, classification_report
classification_rep3 = classification_report(y_test, y_pred)
 print("Classification Report of Random Forest: ",)print(classification_rep3)
from sklearn.metrics import confusion_matrix
 cm= confusion_matrix(y_test,y_pred)
cm
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show
```

PASSIVE AGGRESSIVE CLASSIFIER

```
from sklearn.linear_model import PassiveAggressiveClassifier
pac=PassiveAggressiveClassifier(max_iter=40,random_state=0)
model4=pac.fit(x_train,y_train)
y_pred=model4.predict(x_test)
accuracy4=accuracy_score(y_test,y_pred)
print("Accuracy of PassiveAggressiveClassifier model:",accuracy4)
from sklearn.metrics import accuracy_score, classification_report
classification_rep4 = classification_report(y_test, y_pred)
print("Classification report of Passive Aggresive Classifier:")
print(classification_rep4)
cm=confusion_matrix(y_test,y_pred)
print(cm)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show
```

NAIVE BAYES

```
from sklearn.naive_bayes import MultinomialNB
nv= MultinomialNB()
model5=nv.fit(x_train,y_train)
y_pred=model5.predict(x_test)
accuracy5=accuracy_score(y_test,y_pred)
print("Accuracy of Naive Bayes model:",accuracy5)
from sklearn.metrics import accuracy_score, classification_report
classification_rep5 = classification_report(y_test, y_pred)
print("Classification report of Naive Bayes:")print(classification_rep5)
cm=confusion_matrix(y_test,y_pred)
print(cm)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show
```

SVM

```
from sklearn import model_selection,svm
svm=svm.SVC(C=1.9,kernel="linear")
model6=svm.fit(x_train,y_train)
y_pred=model6.predict(x_test)
accuracy6=accuracy_score(y_test,y_pred)
print("Accuracy of SVM Model:",accuracy6)
from sklearn.metrics import accuracy_score, classification_report
Classification_rep6 = classification_report(y_test, y_pred)
print("Classification report of SVM:")print(classification_rep6)

cm=confusion_matrix(y_test,y_pred)
print(cm)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['REAL','FAKE'],
            yticklabels=['REAL','FAKE'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show
```

COMPARING ACCURACY

```
accuarcy_list=pd.DataFrame({'Models':['Logistic Regression','Decision Tree','Random Forest',
                                  'PassiveAggressiveClassifier','Naive Bayes','SVM'],
                            'Accuracy':[accuracy1*100,accuracy2*100,
                                        accuracy3*100,accuracy4*100,
                                        accuracy5*100,accuracy6*100]})
print(accuarcy_list)
plt.barh(accuarcy_list['Models'],accuarcy_list['Accuracy'])
```

SAVING MODEL

```
import joblib
final_model=joblib.load('FakeNews_detector.pkl')
final_y=final_model.predict(x_test)
final_acc=accuracy_score(y_test,final_y)
print("Accuracy:",final_acc)
vector_form = joblib.load(open('vector.pkl', 'rb'))
```

PREDICTION SYSTEM

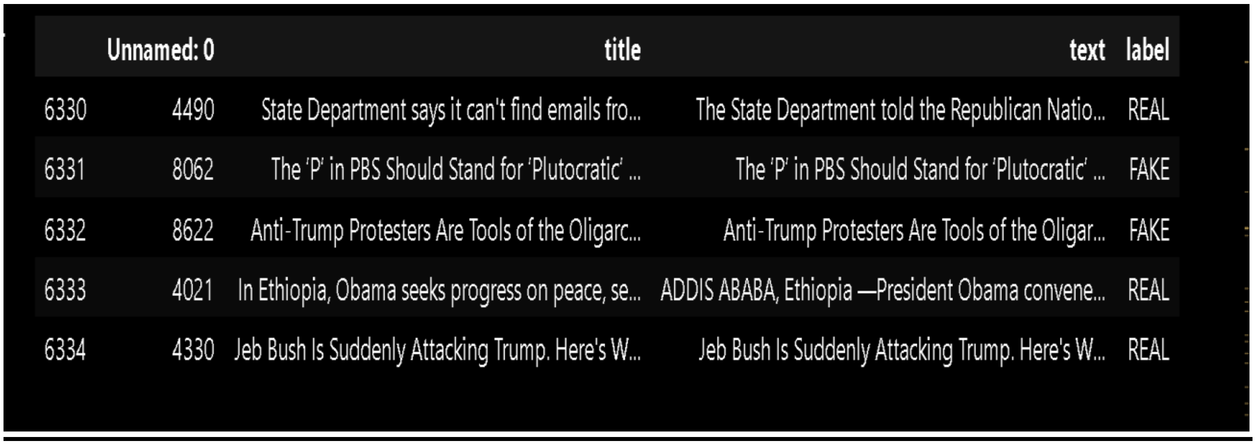
```
import redef wordopt(text):
    text=text.lower()
    text=re.sub('\[,*?\]',",text)
    text=re.sub('\\W',",text)
    text=re.sub('https?://\S+|www\.\S+',",text)
    text=re.sub('<,*?>+',",text)
    text=re.sub('\n',",text)
    text=re.sub('\w*\d\w*',",text)
    return text
def testing(news):
    news=wordopt(news)
    input_data=[news]
    vector_form1=vector_form.transform(input_data)
    prediction = final_model.predict(vector_form1)
    return prediction[0]
```

WEB APPLICATION

```
    import streamlit as st
    import pandas as pd
    import joblib
    import re
    from nltk.corpus import stopwords
    from sklearn.feature_extraction.text import TfidfVectorizer
    vector_form = joblib.load(open('vector.pkl', 'rb'))
    load_model = joblib.load('FakeNews_detector.pkl')
    def wordopt(text):
        text=text.lower()
        text=re.sub('\[,*?\]',",text)
        text=re.sub('\\W',",text)
        text=re.sub('https?://\S+|www\.\S+',",text)
        text=re.sub('<,*?>+',",text)
        text=re.sub('\n',",text)
        text=re.sub('\w*\d\w*',",text)
        return text
    def testing(news):
        news=wordopt(news)
        input_data=[news]
        vector_form1=vector_form.transform(input_data)
        prediction = load_model.predict(vector_form1)
        return prediction[0]



    if __name__ == '__main__':
        st.title('Fake News Classification app ')
        st.subheader("Input the News content below")
        sentence = st.text_area("Enter your news content here", "",height=200)
        predict_btt = st.button("predict")
        if predict_btt:
            prediction_class=testing(sentence)
            if prediction_class == 'REAL':
                st.success('The news is REAL!!')
            if prediction_class == 'FAKE':
                st.warning('The news is FAKE!!')
```

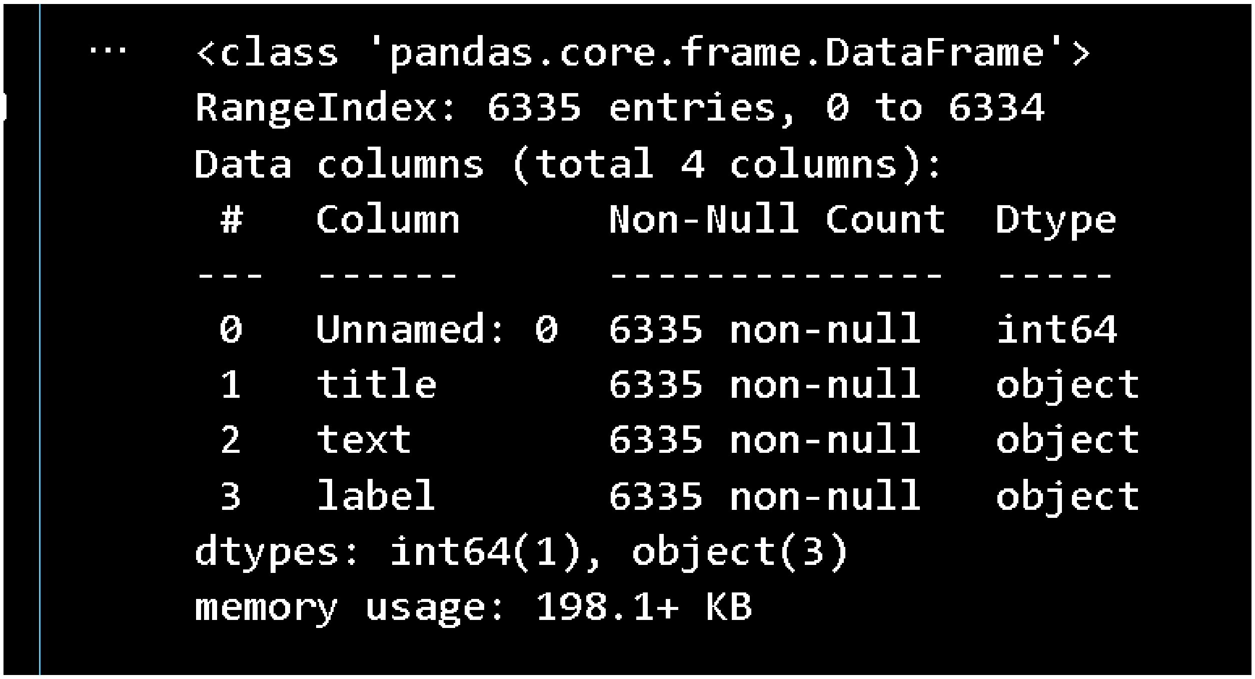
OUTPUT SCREENSHOTS

Tail of dataset: Last five records



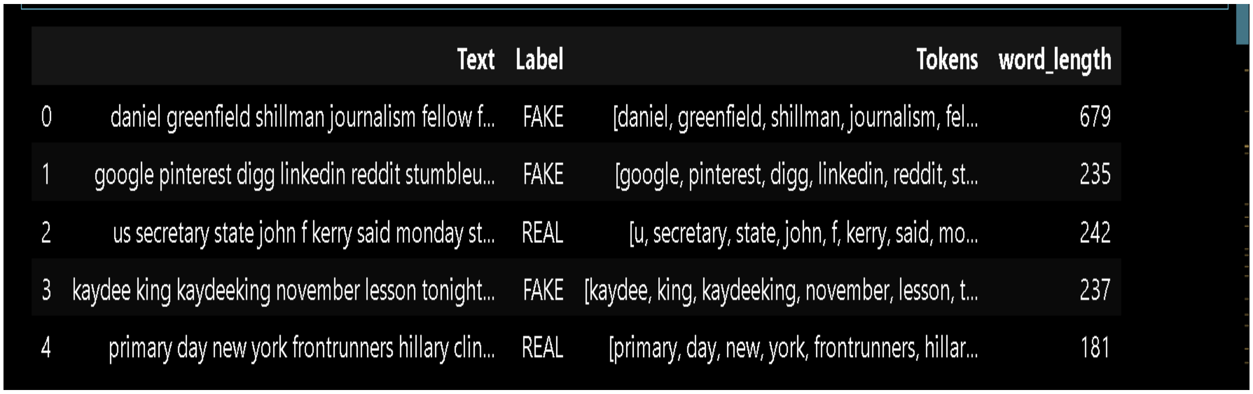
	Unnamed: 0	title	text	label
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

Information about dataset



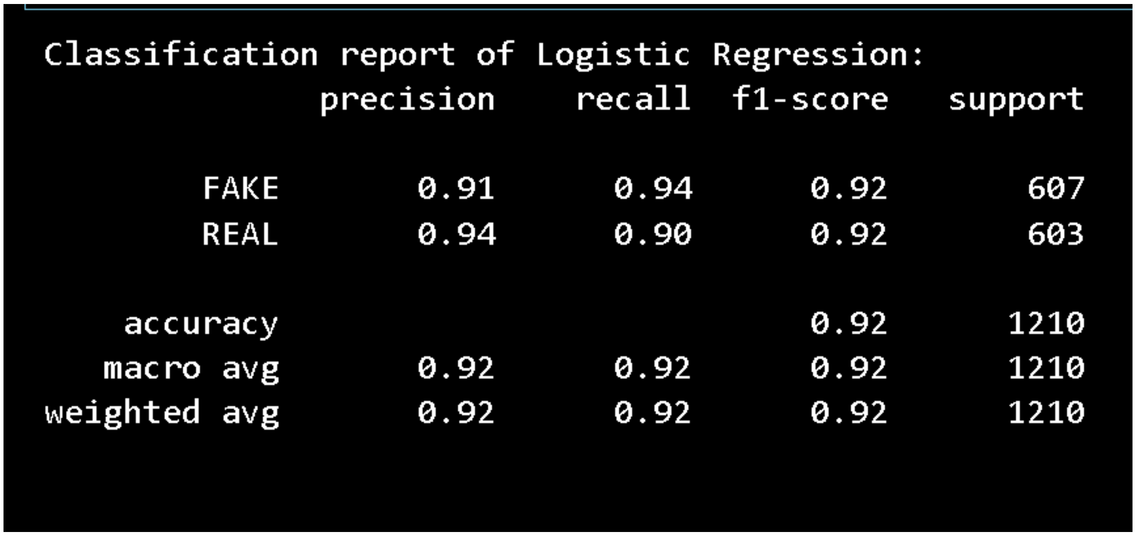
```
...  <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 6335 entries, 0 to 6334
     Data columns (total 4 columns):
      #   Column      Non-Null Count  Dtype 
     ---  ------      --------------  ----- 
      0   Unnamed: 0  6335 non-null   int64 
      1   title       6335 non-null   object
      2   text        6335 non-null   object
      3   label       6335 non-null   object
     dtypes: int64(1), object(3)
     memory usage: 198.1+ KB
```

After pre processing



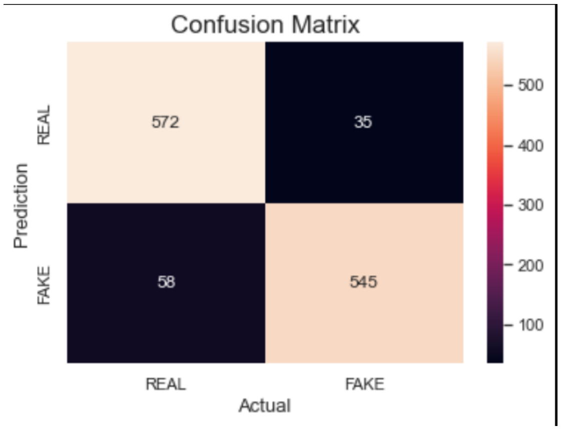
	Text	Label	Tokens	word_length
0	daniel greenfield shillman journalism fellow f...	FAKE	[daniel, greenfield, shillman, journalism, fel...	679
1	google pinterest digg linkedin reddit stumbleu...	FAKE	[google, pinterest, digg, linkedin, reddit, st...	235
2	us secretary state john f kerry said monday st...	REAL	[u, secretary, state, john, f, kerry, said, mo...	242
3	kaydee king kaydeeking november lesson tonight...	FAKE	[kaydee, king, kaydeeking, november, lesson, t...	237
4	primary day new york frontrunners hillary clin...	REAL	[primary, day, new, york, frontrunners, hillar...	181

Logistic regression

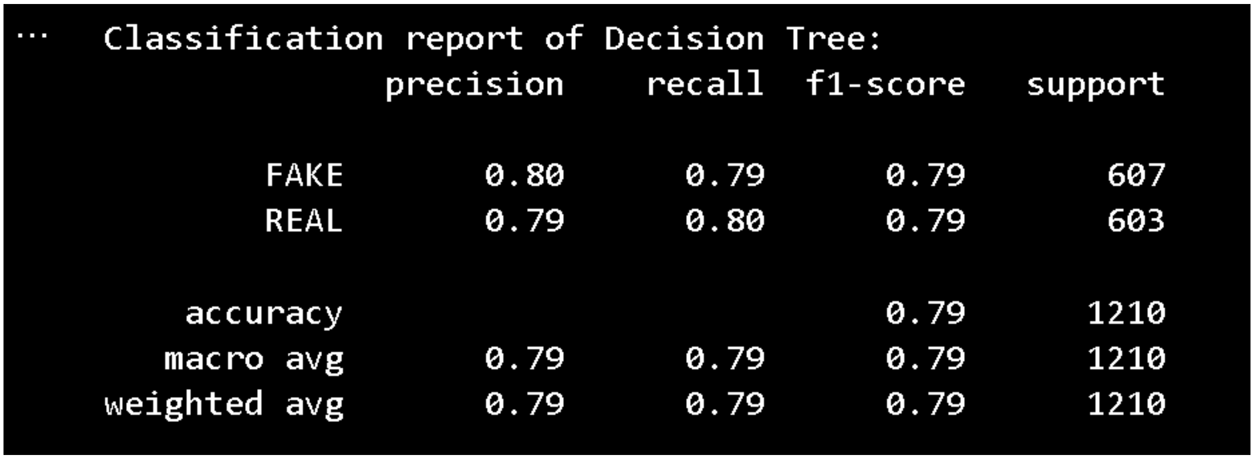


```
Classification report of Logistic Regression:
              precision    recall  f1-score   support

        FAKE       0.91      0.94      0.92       607
        REAL       0.94      0.90      0.92       603

    accuracy                           0.92      1210
   macro avg       0.92      0.92      0.92      1210
weighted avg       0.92      0.92      0.92      1210
```

Confusion matrix for logistic regression

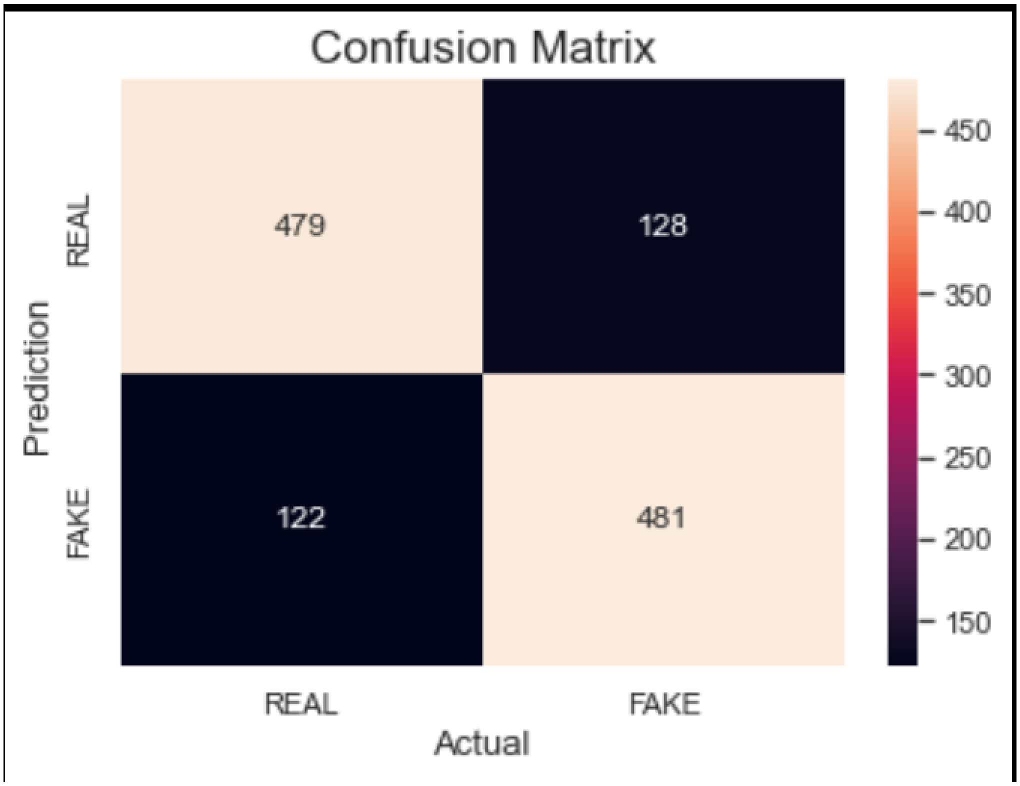


Decision tree

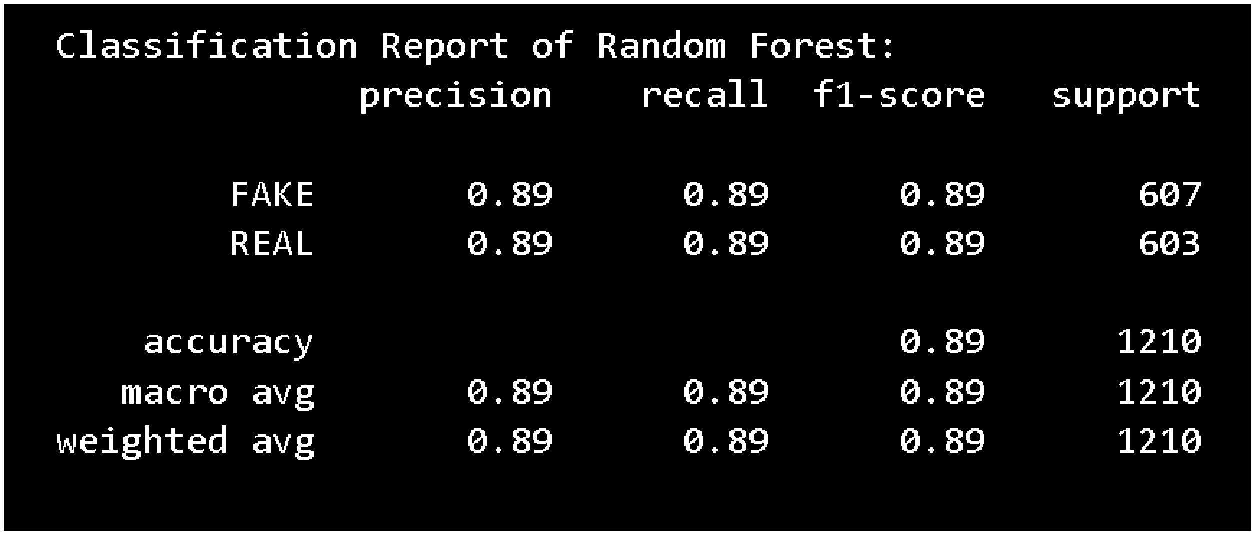


```
...   Classification report of Decision Tree:
                     precision    recall  f1-score   support

              FAKE        0.80      0.79      0.79       607
              REAL        0.79      0.80      0.79       603

          accuracy                            0.79      1210
         macro avg        0.79      0.79      0.79      1210
      weighted avg        0.79      0.79      0.79      1210
```

Confusion matrix for decision tree

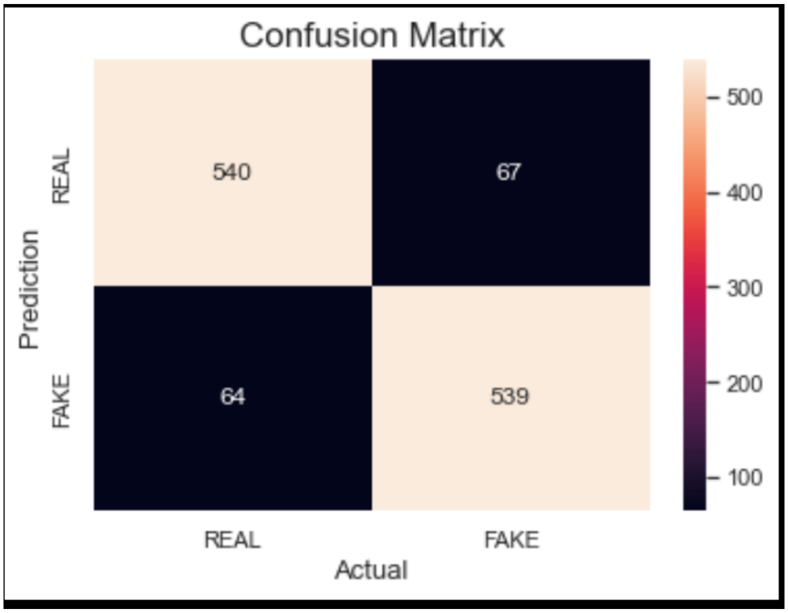


Accuracy for random forest

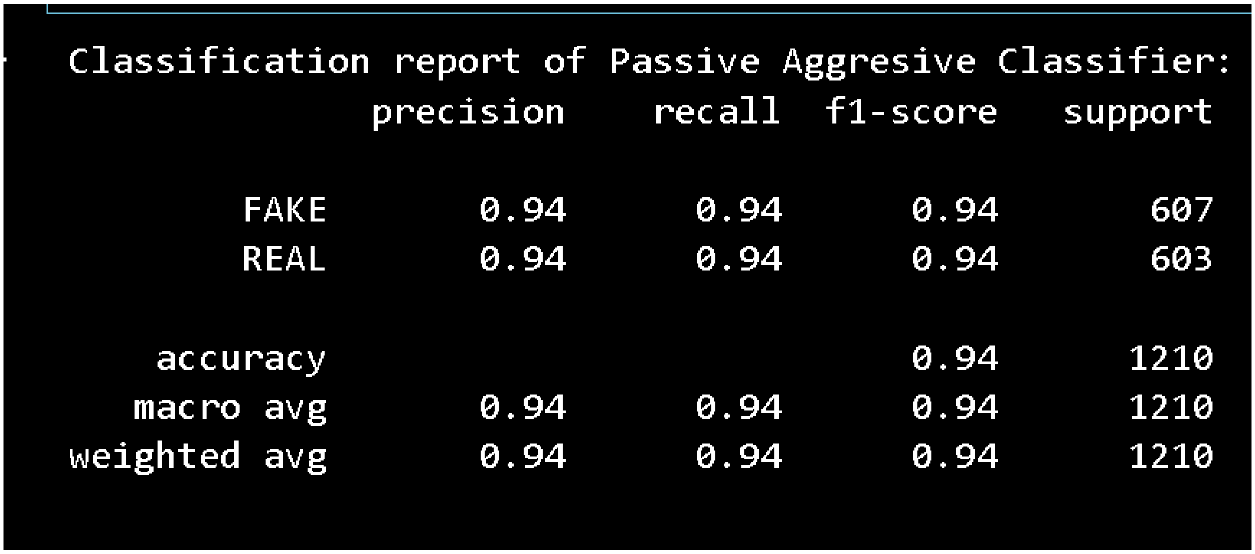


```
Classification Report of Random Forest:
             precision    recall  f1-score   support

        FAKE      0.89      0.89      0.89       607
        REAL      0.89      0.89      0.89       603

    accuracy                          0.89      1210
   macro avg      0.89      0.89      0.89      1210
weighted avg      0.89      0.89      0.89      1210
```

Confusion matrix for random forest

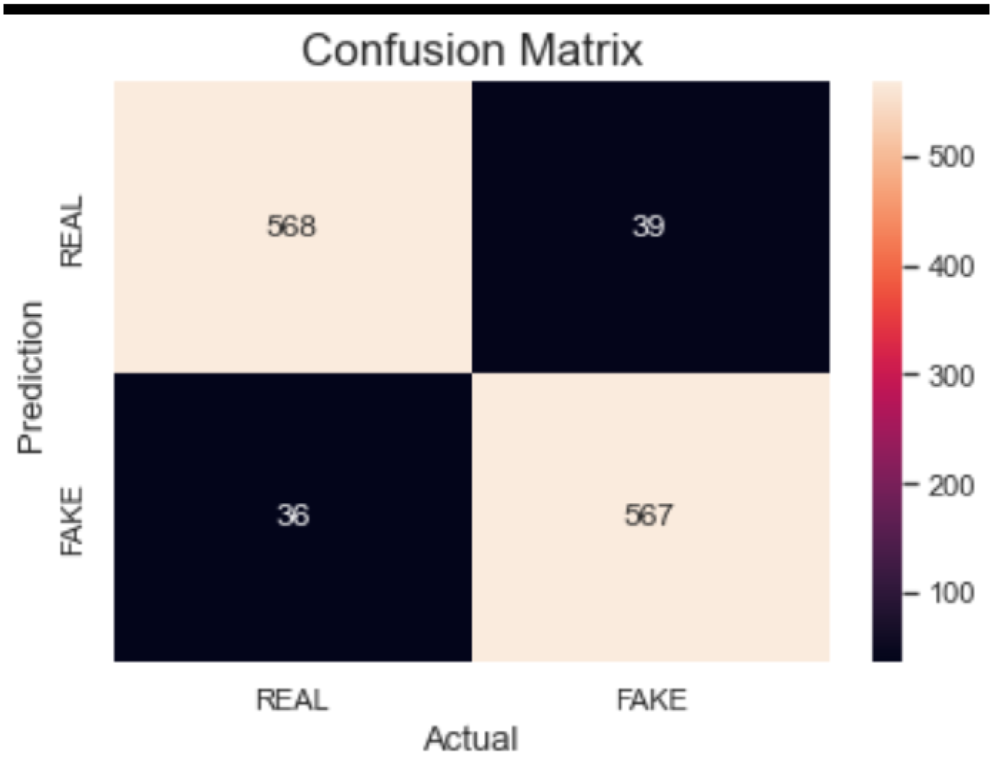


Accuracy for passive aggressive classifier

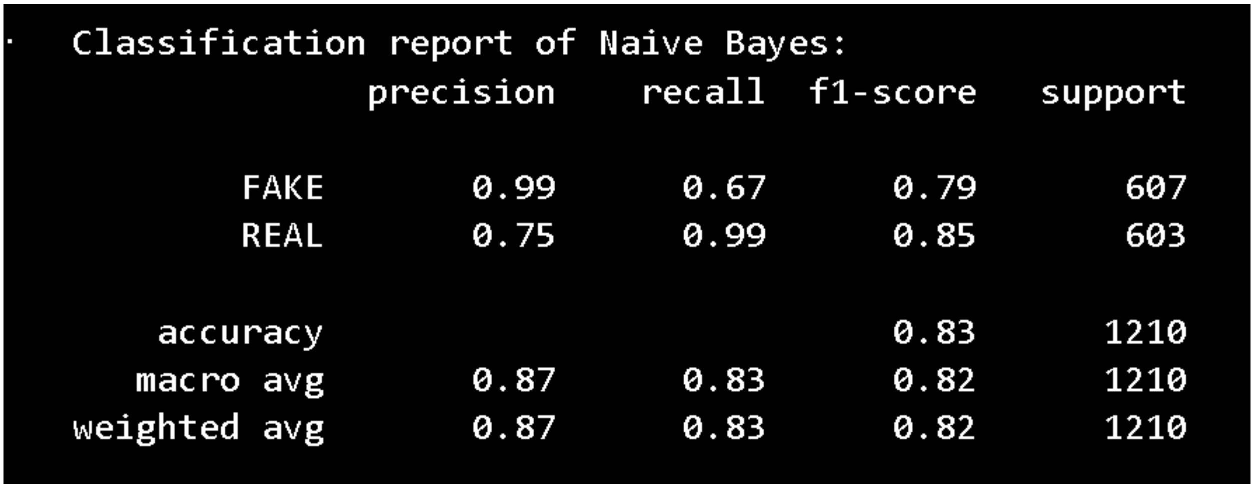


```
Classification report of Passive Aggresive Classifier:
              precision    recall  f1-score   support

        FAKE       0.94      0.94      0.94       607
        REAL       0.94      0.94      0.94       603

    accuracy                           0.94      1210
   macro avg       0.94      0.94      0.94      1210
weighted avg       0.94      0.94      0.94      1210
```

Confusion matrix for passive aggressive classifier

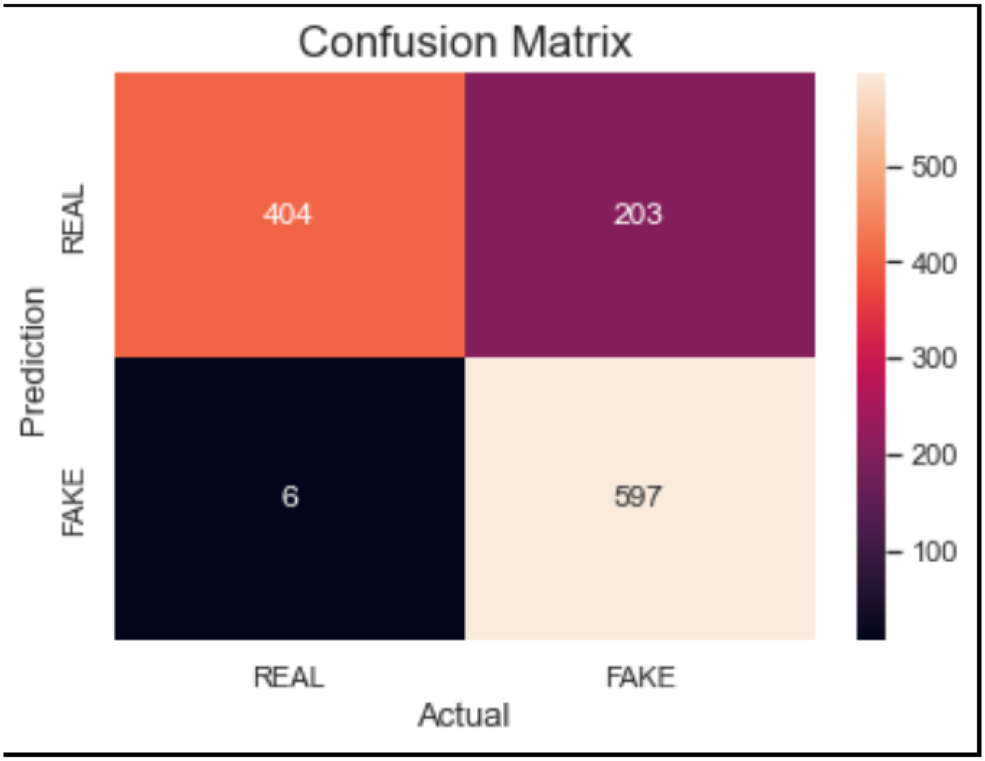


Accuracy for Naive bayes

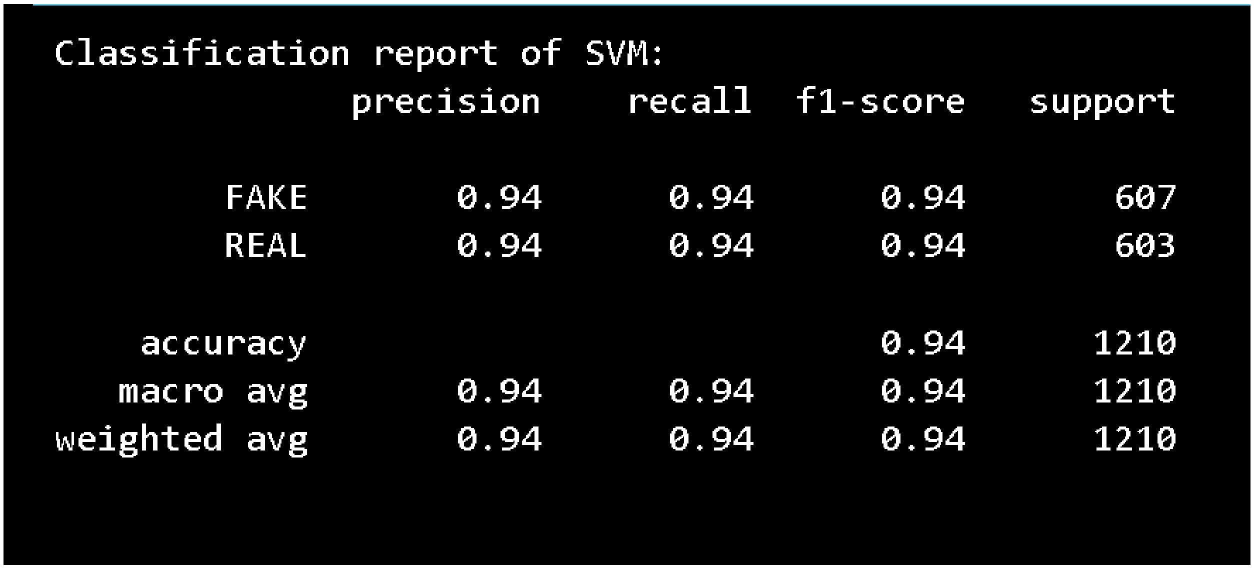


```
   Classification report of Naive Bayes:
                 precision    recall  f1-score   support

         FAKE       0.99      0.67      0.79       607
         REAL       0.75      0.99      0.85       603

     accuracy                           0.83      1210
    macro avg       0.87      0.83      0.82      1210
 weighted avg       0.87      0.83      0.82      1210
```

Confusion matrix for Naive bayes

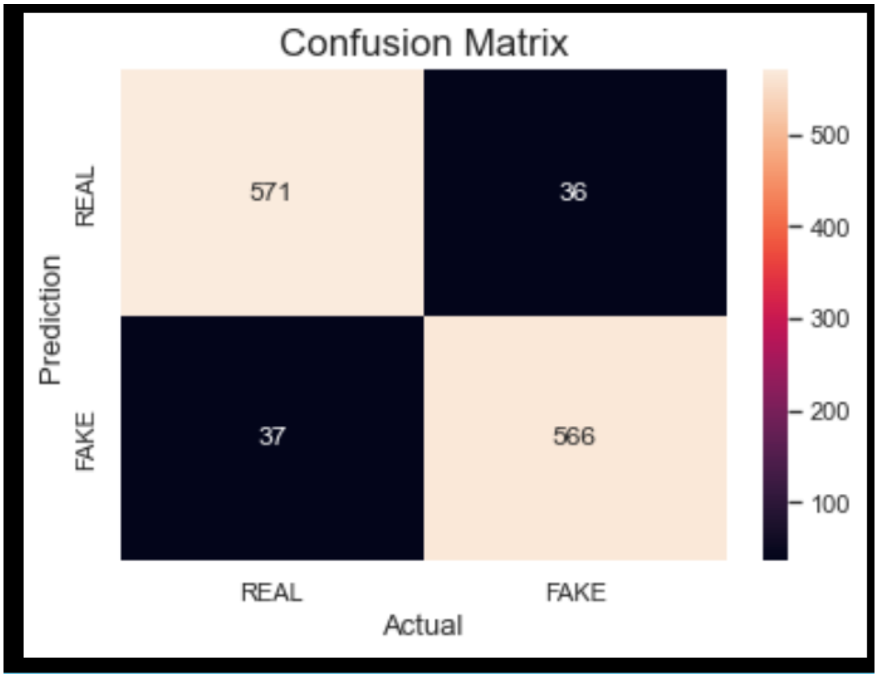


Accuracy for SVM model

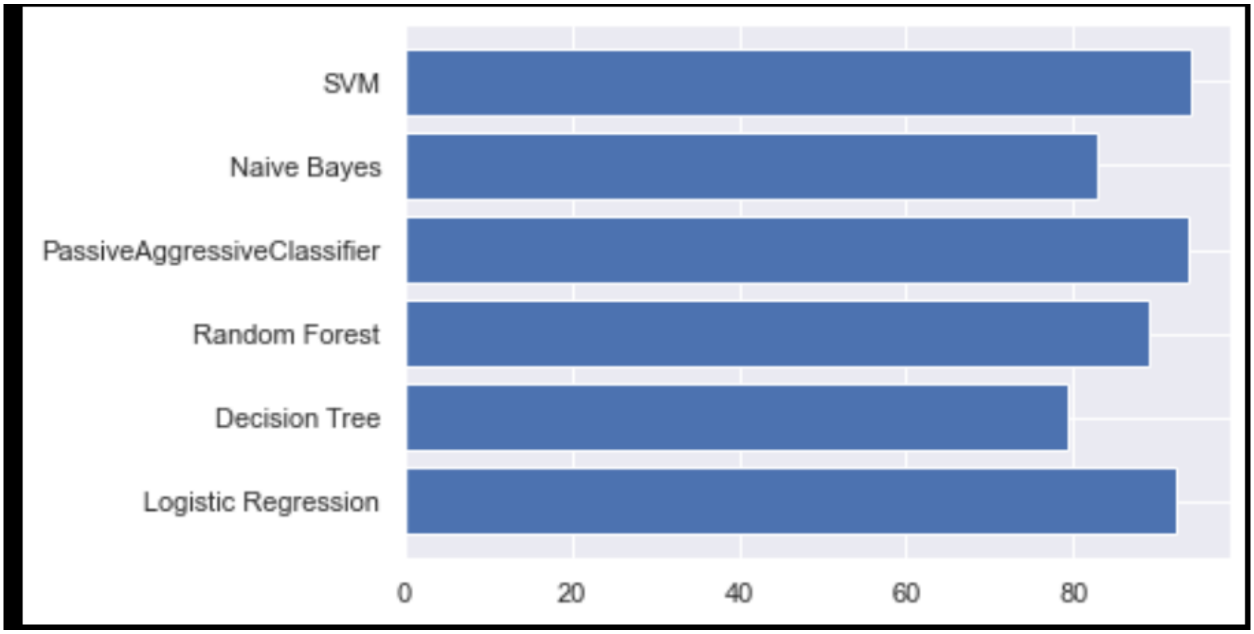


```
Classification report of SVM:
              precision    recall  f1-score   support

        FAKE       0.94      0.94      0.94       607
        REAL       0.94      0.94      0.94       603

    accuracy                           0.94      1210
   macro avg       0.94      0.94      0.94      1210
weighted avg       0.94      0.94      0.94      1210
```

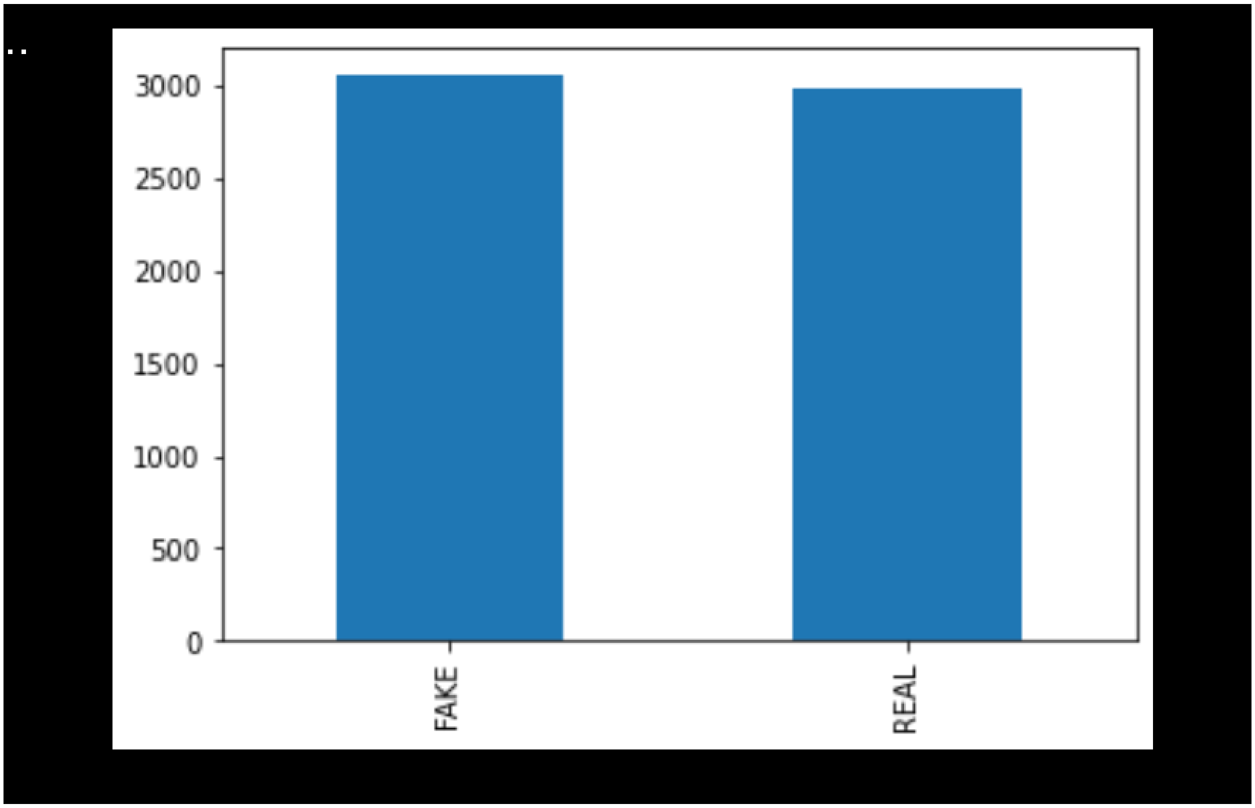
Confusion matrix for SVM



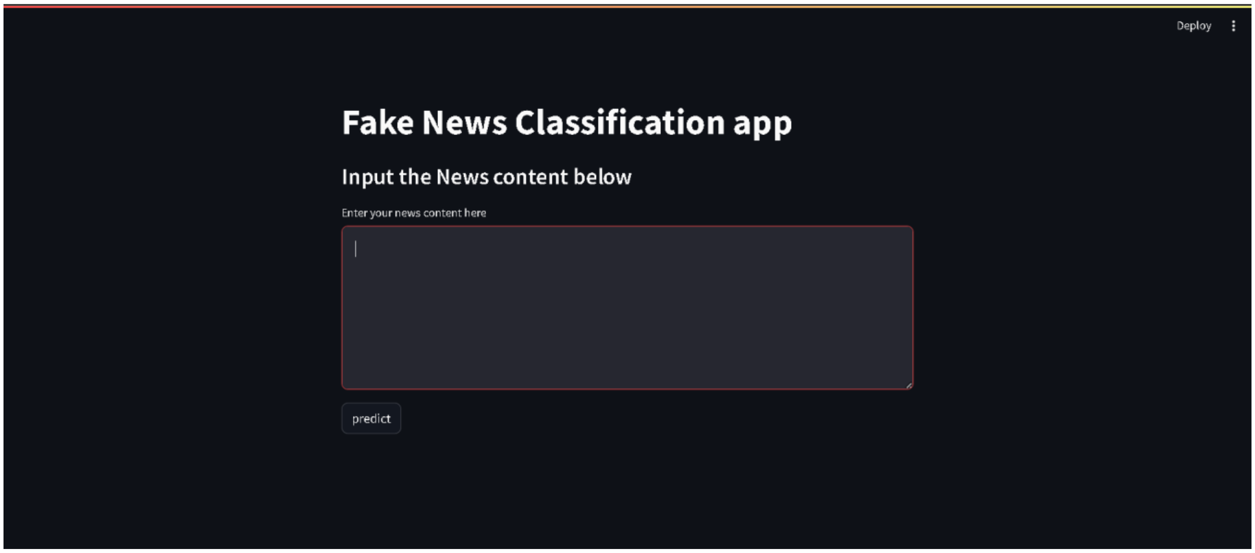
Comparing accuracy



Visualization



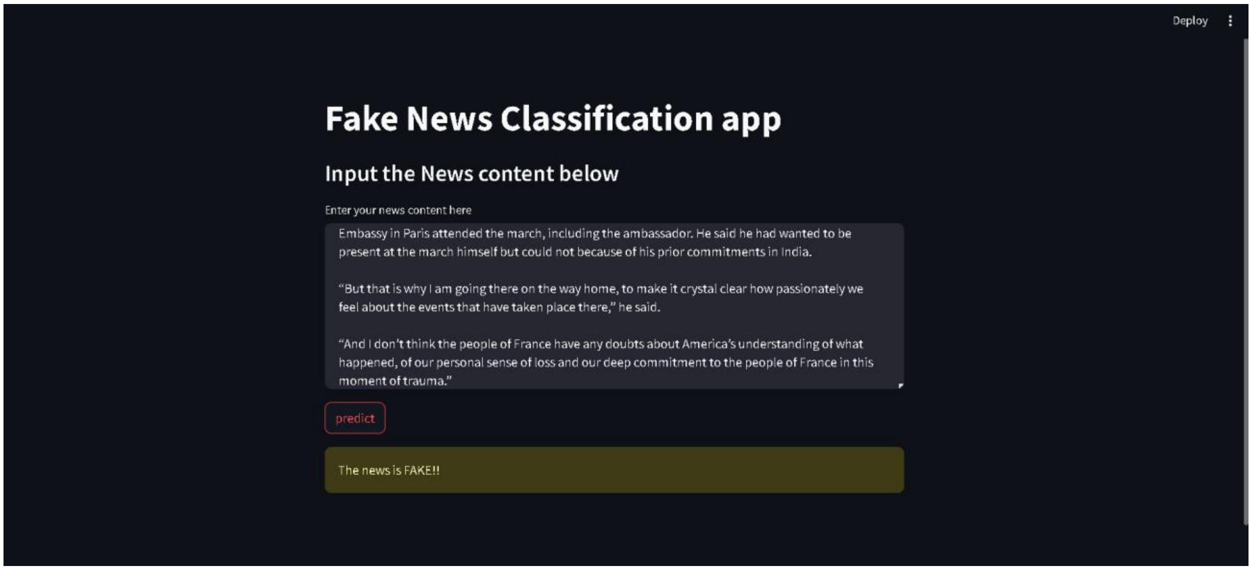
Web application



Fake News Classification app

Input the News content below

Enter your news content here



Fake News Classification app

Input the News content below

Enter your news content here

Embassy in Paris attended the march, including the ambassador. He said he had wanted to be present at the march himself but could not because of his prior commitments in India.

“But that is why I am going there on the way home, to make it crystal clear how passionately we feel about the events that have taken place there,” he said.

“And I don’t think the people of France have any doubts about America’s understanding of what happened, of our personal sense of loss and our deep commitment to the people of France in this moment of trauma.”

The news is FAKE!!

CONCLUSION

Finally we can conclude that by using different machine learning algorithms we have saved the one model which gives the better accuracy than other models. The saved model is the SVM which is good when compared to other models. And finally after saving the model the we designed the streamlit model to predict the given news is fake or real.

The different of accuracies of the models are as follows:

NAME OF THE MODEL	ACCURACY
LOGISTIC REGRESSION	92.479339
DECISION TREE	79.33884
RANDOM FOREST	89.173554
NAÏVE BAYES	82.727273
PASSIVE AGGRESSIVE	93.801653
SUPPORT VECTOR MACHINE	93.966942

These are the accuracies of the different machine learning models.

ML models can analyze large volumes of textual data, extracting patterns and features that distinguish between reliable and unreliable information sources. By leveraging labeled datasets for training, these models can learn to recognize subtle linguistic cues, contextual inconsistencies, and other indicators of falsehoods or misinformation.

PROJECT LINKS

B. Durga : https://github.com/BDurga26/Fake-News-Detection

B.Lavanya : https://github.com/lavanya120/FAKENEWS

G. Bandhavi : https://github.com/bandhavi2913/Fake-news-detection

A. Sai lakshmi: https://github.com/Sailakshmi35/Fake_News_Detection

SUBMITTED BY

ANDE SAILAKSHMI

BOYINA LAVANYA

DURGA BOMMAREDDY

GARIKIPATI BANDHAVI