Spring 2020

PAF-KARACHI INSTITUTE OF ECONOMICS & TECHNOLOGY

ARTIFICIAL INTELLIGENCE LAB

PROJECT REPORT

FAKE NEWS DETECTION

INSTRUCTOR

DR AFFAN ALIM

SIR MINHAL RAZA

GROUP MEMBERS:

JAVERIA HASSAN – 9517

JAWERIA ASIF – 9442

COURSE ID: 102431

FAKE NEWS DETECTION THROUGH MACHINE LEARNING ALGORITHMS

1. PROBLEM DESCRIPTION:

➢ INTRODUCTION:

"Fake News" is a term used to represent fabricated news or propaganda comprising misinformation communicated through traditional media channels like print, and television as well as non-traditional media channels like social media. The general motive to spread such news is to mislead the readers, damage reputation of any entity, or to gain from sensationalism. It is seen as one of the greatest threats to democracy, free debate, and the Western order. Detecting fake news is a challenging task to accomplish as it requires models to summarize the news and compare it to the actual news in order to classify it as fake. Moreover, the task of comparing proposed news with the original news itself is a daunting task as its highly subjective and opinionated.

➢ FAKE NEWS:

Fake news can be come in many forms, including: unintentional errors committed by news aggregators, outright false stories, or the stories which are developed to mislead and influence reader's opinion. While fake news may have multiple forms, the effect that it can have on people, government and organizations may generally be negative since it differs from the facts.

➢ PROBLEM DEFINITION:

Given the social news engagements E among n users for news article a, the task of fake news detection is to predict whether the news article 'a' is a fake news piece or not, i.e., F: E → {0, 1} such that, F (a) = (1, if 'a' is a piece of fake news, 0, otherwise. (1)

*Where '**F**' is the prediction function we want to learn.*

OR

Fake news is a news article that is intentionally and verifiably false.

A made-up story with an intention to deceive

2. DATA SET DESCRIPTION:

➢ COLLECTION OF DATASET:

- ***DETAILS***
  - **Source**: The data is downloaded from data flair.
  - It is a labeled data, so we will use supervised learning algorithms.

➢ INSIDE OF DATASET:

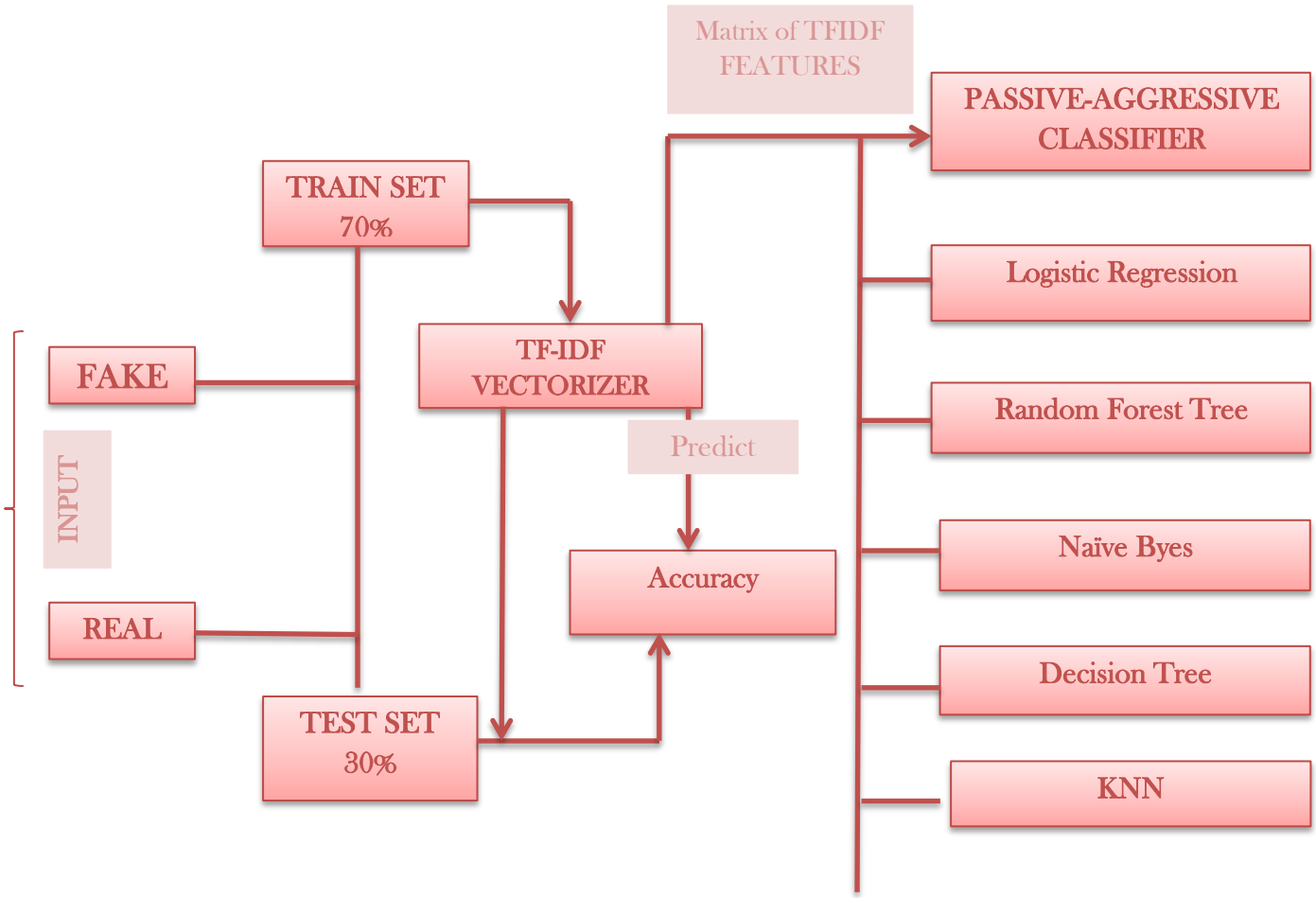
- ***EXAMPLE OF DATA SET:***

A	B	C	D
	title	text	label
8476	You Can Smell Hillaryâ€™s Fear	Daniel Greenfield, a Shillman Journalism Fellow at the Freedom Center, is a New York	FAKE
10294	Watch The Exact Moment Paul Ryan Committed Political Suicide At A Trump Rally (VIDEO)	Google Pinterest Digg Linkedin Reddit Stumbleupon Print Delicious Pocket Tumblr	FAKE
3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this	REAL
10142	Bernie supporters on Twitter erupt in anger against the DNC: 'We tried to warn you!'	â€" Kaydee King (@KaydeeKing) November 9, 2016 The lesson from tonight's Dem	FAKE
875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners Hillary Clinton and Donald Trump are	REAL
6903	Tehran, USA		FAKE
7341	Girl Horrified At What She Watches Boyfriend Do After He Left FaceTime On	Share This Baylee Luciani (left), Screenshot of what Baylee caught on FaceTime (right)	FAKE
95	â€"Britainâ€™s Schindlerâ€™ Dies at 106	A Czech stockbroker who saved more than 650 Jewish children from Nazi Germany has di	REAL
4869	Fact check: Trump and Clinton at the 'commander-in-chief' forum	Hillary Clinton and Donald Trump made some inaccurate claims during an NBC	REAL
2909	Iran reportedly makes new push for uranium concessions in nuclear talks	Iranian negotiators reportedly have made a last-ditch push for more concessions from	REAL
1357	With all three Clintons in Iowa, a glimpse at the fire that has eluded Hillary Clintonâ€™s campaign	CEDAR RAPIDS, Iowa â€" â€œI had one of the most wonderful rallies of my entire career	REAL
988	Donald Trumpâ€™s Shockingly Weak Delegate Game Somehow Got Even Worse	Donald Trumpâ€™s organizational problems have gone from bad to worse to flat-out	REAL
7041	Strong Solar Storm, Tech Risks Today | S0 News Oct.26.2016 [VIDEO]	Click Here To Learn More About Alexandra's Personalized Essences Psychic Protection	FAKE
7623	10 Ways America Is Preparing for World War 3	October 31, 2016 at 4:52 am	FAKE

- ***TRAINING SET***
  - [TITLE, TEXT, LABEL]
  - Pairs of title and text with the appropriate class label for each.
- ***TESTING SET***
  - [TITLE, TEXT, LABEL]
  - Pairs of title and text with the appropriate class label for each.
- ***DESCRIPTION OF DATASET:***

SNO	LABEL	PERC%
01	FAKE	49.99%
02	REAL	50.05%

3. USE CASE:



4. PROBLEMS IN DETECTING FAKE NEWS:

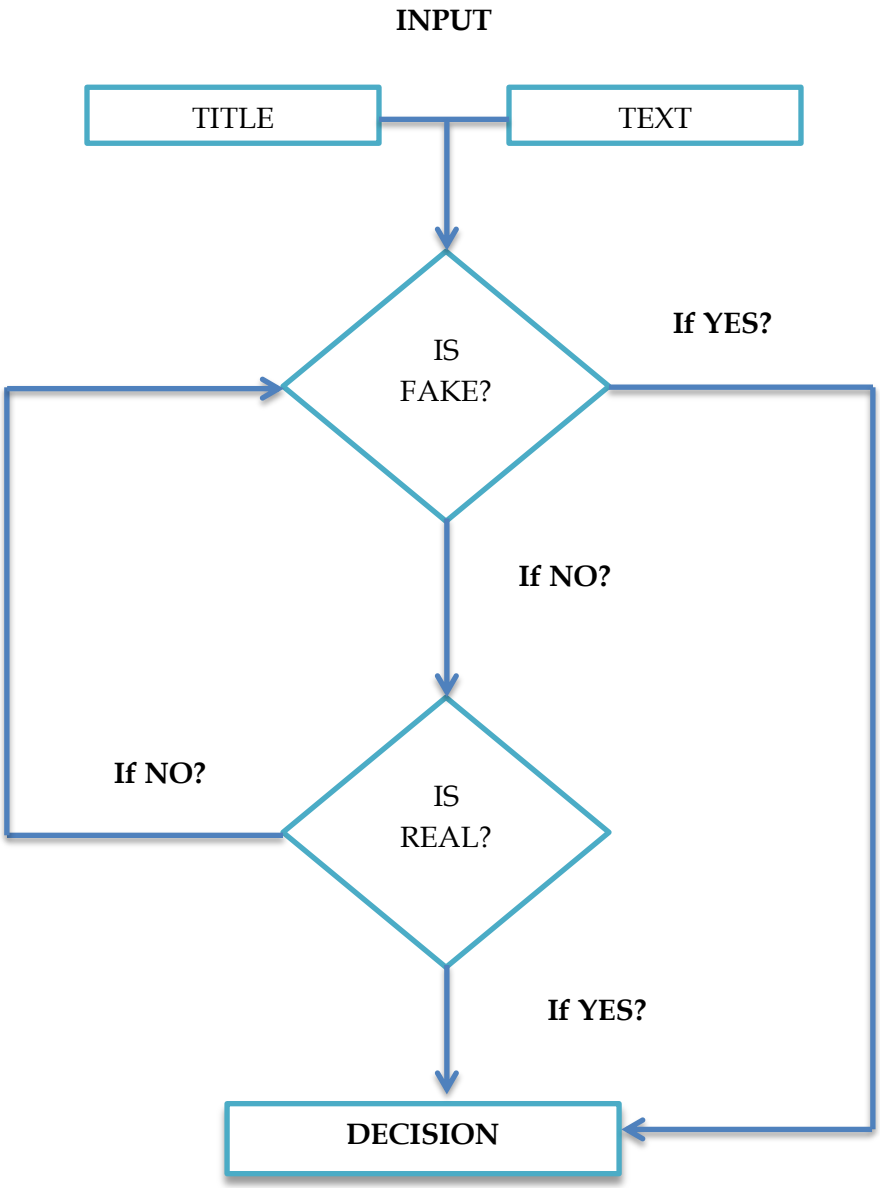
The core task of detecting fake news involves identifying the language (set of words or sentences) which is used to deceive the readers. The idea of classifying fake news by learning word-level meaning is a very challenging task under the skin. Detecting fake news is hard for many reasons. First, manual task of identifying fake news is very subjective. Assessing the veracity of a news story is a complex and cumbersome task, even for trained experts. News is not only spread through traditional media outlets anymore but also through various social media channels. Automated solution requires understanding the natural language processing which is difficult and complex. These complexities make it a daunting task to classify text as fake news.

➢ PROPOSED SOLUTION:

We tried with word vector representation and different machine learning architectures as mentioned in the Methods sub-section. The input features for our model consists of Tf-Idf word vector representations of article-title and text pair. We computed Tf-Idf scores on unigrams and bigrams. The data is labeled (i.e. FAKE or REAL).

5. DIAGRAMS:

➢ FLOW DIAGRAM OF METHODOLOGY:



6. METHODS & PROCEDURES:

➢ MACHINE LEARNING TECHNIQUES:

We are going to implement the following machine learning algorithms in fake news detection:

- ***Word Vector Representation***

The method of representing words as vectors is commonly referred to as word vector representations or word embedding's. In this paper, we experiment with the below mentioned word vector representations:

- ***Bag of Words:***

In a Bag of Words (BoW) model, sentences are represented as multi set of its words. BoW model disregards the order and hence it also disregards the context in which the word occurred. Tf-Idf: Tf-Idf stands for *term frequency-inverse document frequency*. Tf-Idf weight gives us an indication on how important a given word is for a sentence or document in relation to the entire corpus.

- ***TF-IDF Vectorizer***

We have also used the technique "Term Frequency-Inverse Document Frequency" (TF-IDF) for feature extraction. Term Frequency and Inverse Document Frequency are two components of TF-IDF. Term Frequency identifies local importance of a word by its occurrence in a document. Inverse Document Frequency identifies the signature words, which are not appeared more often across the documents Word with a high TF-IDF is a signature word which is important for the document in consideration, has high frequency in the document but is not a common word across other texts.

- ***Sampling Techniques***

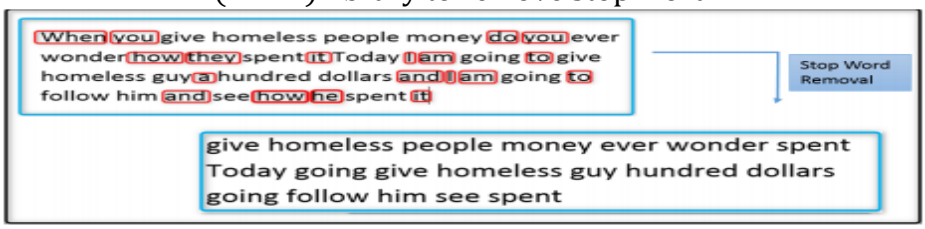
We split our data into Train, Validation and Test data sets. We used stratified proportional allocation and random shuffling to perform our data splits (*i.e use train_split library*). We allocated 70% of our data to the Train set and the remaining 30% to the Test set.

- ***Data Preprocessing***

Text data requires special preprocessing to implement machine learning algorithms on them. There are various techniques widely used to convert text data into a form that is ready for modeling. We use TF_IDF algorithms to convert text into vector form.

- ***Stop Word Removal***

We start with removing stop words from the text data available. Stops Words (most common words in a language which do not provide much context) can be processed and filtered from the text as they are more common and hold less useful information. Stop words acts more like a connecting part of the sentences, for example, conjunctions like "and", "or" and "but", prepositions like "of", "in", "from", "to", etc. and the articles "a", "an", and "the". Such stop words which are of less importance may take up valuable processing time, and hence removing stop words as a part of data preprocessing is a key first step in natural language processing. We used TfidfVectorizer (TF-ID) library to remove stop word.



➢ MACHINE LEARNING ALGORITHMS:

In this section, we discuss several machine learning architectures that we experimented with and we present the model that gave us best results.

• CLASSIFICATION ALGORITHMS:

We trained our model on seven different machine learning models. In this subsection, we provide a high-level summary of algorithms. The details of these models are below:

○ PASSIVE AGRESSIVE CLASSIFIER:

Passive: if correct classification, keep the model; Aggressive: if incorrect classification, update to adjust to this misclassified example.

○ MULTINOMIAL NAIVE BAYES:

Multinomial Naïve Byes is a specialized version of Naive Bayes that is designed more for text documents. Whereas simple naive Bayes would model a document as the presence and absence of particular words, multinomial naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with in.

○ K-NEAREST NEIGHBOR:

KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. Just for reference, this is "where" KNN is positioned in the algorithm list of scikit learn.

○ DECISION TREE:

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

○ LOGISTIC REGRESSION :

The logistic model is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick.

○ RANDOM FOREST ALGORITHM:

Random forest is a supervised learning algorithm which is used for both classification as well as regression. ... Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

○ SUPPORT VECTOR MACHINE (SVM)

SVM is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving SVM model sets of labeled training data for each category, they're able to categorize new text. So you're working on a text classification problem

○ PERFORMANCE MEASURES:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. We can use classification **performance metrics**, Accuracy, AUC (Area under Curve) etc. Another example of metric for evaluation of **machine**

learning algorithms is precision, recall, which can be used for sorting algorithms primarily used by search engines.

- **EVALUATING METRICS**:

To evaluate the performance of algorithms for fake news detection problem, various evaluation metrics have been used. In this subsection, we review the most widely used metrics for fake news detection. Most existing approaches consider the fake news problem as a classification problem that predicts whether a news article is fake or not:

• True Positive (TP): when predicted fake news pieces are actually annotated as fake news
• True Negative (TN): when predicted true news pieces are actually annotated as true news
• False Negative (FN): when predicted true news pieces are actually annotated as fake news
• False Positive (FP): when predicted fake news pieces are actually annotated as true news.

By formulating this as a classification problem, we can define following metrics,

Precision = |T P| / |T P| + |F P|
Recall = |T P| / |T P| + |F N|
F1 = 2 · [Precision · Recall / Precision + Recall]
Accuracy = |T P| + |T N| / |T P| + |T N| + |F P| + |F N|

These metrics are commonly used in the machine learning community and enable us to evaluate the performance of a classifier from different perspectives.

Specifically, *Accuracy measures the similarity between predicted fake news and real fake news*. *Precision measures the fraction of all detected fake news that are annotated as fake news, addressing the important problem of identifying which news is fake*. However, because fake news datasets are often skewed, a high precision can be easily achieved by making fewer positive Predictions. Thus, *Recall is used to measure the sensitivity, or the fraction of annotated fake news articles that are predicted to be fake news*. *F1 is used to combine precision and recall, which can provide an overall prediction performance for fake news detection*. Note that for Precision, Recall, F1, and Accuracy, the higher the value, the better the performance. *The Receiver Operating Characteristics (ROC) curve provides a way of comparing the performance of classifiers by looking at the trade-off in the False Positive Rate (FPR) and the True Positive Rate (TPR)*. To draw the ROC curve, we plot the FPR on the x axis and TPR along the y axis. The ROC curve compares the performance of different classifiers by changing class distributions via a threshold.

TPR and FPR are defined as follows (note that TPR is the same as recall defined above):

T P R = |T P| / |T P| + |F N|
F P R = |F P| / |F P| + |T N|

Based on the ROC curve, we can compute *the Area under the Curve (AUC) value, which measures the overall performance of how likely the classifier is to rank the fake news higher than any true news*. AUC is defined as below:

AUC = P(n0 + n1 + 1 − ri) − n0(n0 + 1)/2 n0n1

where ri is the rank of ith fake news piece and n0 (n1) is the number of fake (true) news pieces. It is worth mentioning that AUC is more statistically consistent and more discriminating than accuracy [47], and it is usually applied in an imbalanced classification problem, such as fake news classification, where the number of ground truth fake news articles and and true news articles have a very imbalanced distribution.

7. CONSOLE BASED / GUI BASED:

- Jupyter Console with graph plotting.

8. CODE:

➢ READING A DATASET

```
df=pd.read_csv('news.csv')
```

➢ SPLITTING A DATASET

```
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.3, random_state=7)
```

➢ APPLYING TF-IDF

```
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)
tfidf_train=tfidf_vectorizer.fit_transform(x_train)
tfidf_test=tfidf_vectorizer.transform(x_test)
```

➢ APPLYING PASSIVE-AGGRESSIVE

```
PA_Model=PassiveAggressiveClassifier()
PA_Model.fit(tfidf_train,y_train)
pred_PA=PA_Model.predict(tfidf_test)
```

➢ APPLYING MULTINOMIAL NAÏVE BYES

```
NaveByes = MultinomialNB()
NaveByes.fit(tfidf_train, y_train)
pred_NB = NaveByes.predict(tfidf_test)
```

➢ APPLYING LOGISTIC REGRESSION

```
model_Logistic = LogisticRegression()
model_Logistic.fit(tfidf_train, y_train)
pred_Log = model_Logistic.predict(tfidf_test)
```

➢ APPLYING DECISION TREE

```
model_Tree=tree.DecisionTreeClassifier()
model_Tree.fit(tfidf_train,y_train)
DT_predict=model_Tree.predict(tfidf_test)
```

➢ APPLYING K_NEAREST NEIGHBOUR

```
KNN_model = KNeighborsClassifier()
KNN_model.fit(tfidf_train,y_train)
KNN_pred = KNN_model.predict(tfidf_test)
```

➢ APPLYING RANDOM FOREST

```
model_RF=RandomForestClassifier()
model_RF.fit(tfidf_train,y_train)
RF_predict=model_RF.predict(tfidf_test)
```

➢ APPLYING SUPPORT VECTOR MACHINE

```
svm = SVC()
svm.fit(tfidf_train, y_train)
svm_pred=svm.predict(tfidf_test)
```

➢ CALCULATING ACCURACY

```
------general code for every model-----
accuracy=accuracy_score(y_test,y_predict)
accuracy =round(accuracy *100,2)
print(f'Accuracy  : { accuracy }%')
```

➢ CALCULATING PERFORMANCE MEASURES

```
Tp= True Positive
Tn= True Negative
Fn= False Negative
Fp=False Positive
tn,fp,fn,tp=confusion_matrix(y_test,y_predict).ravel()
print('True Negative - TN = ',tn,"\nFalse Positive - FP = ",fp,"\n False Negative - FN = ",fn,"\nTrue
Positive - TP = ",
    tp)
sensitivity=tp/(tp+fn)
print("Sensitivity = ",sensitivity)
specificity=tn/(tn+fp)
print("Specifity = ",specificity)
recall=tp/(tp+fn)
print("Recall = ",recall)
Precision=tp/(tp+fp)
print("Precision = ",Precision)
conf_matrix=confusion_matrix(y_test,y_predict)
print('Confusion Matrix  :\n',conf_matrix)
print("Classification Report \n",classification_report(y_test,y_predict))
```

➢ PLOTTING GRAPH BETWEEN PREDICTED & TARGET DATA

```
plt.scatter(y_test,y_predict, color = "purple")
plt.plot(y_test,y_predict, color = "pink")
plt.title("Algorithm")
plt.xlabel("Test DATA")
plt.ylabel('PREDICTED DATA')
```

➢ PLOTTING CONFUSION MATRIX

```
df_cm = pd.DataFrame(conf_matrix, columns=np.unique(y_test), index = np.unique(y_test))
df_cm.index.name = 'Actual'
df_cm.columns.name = 'Predicted'
plt.figure(figsize = (10,7))
sns.set(font_scale=1.4)
```

```
sns.heatmap(df_cm, cmap="rocket", annot=True,annot_kws={"size": 16})
```

➢ COMAPRING ACCURACY

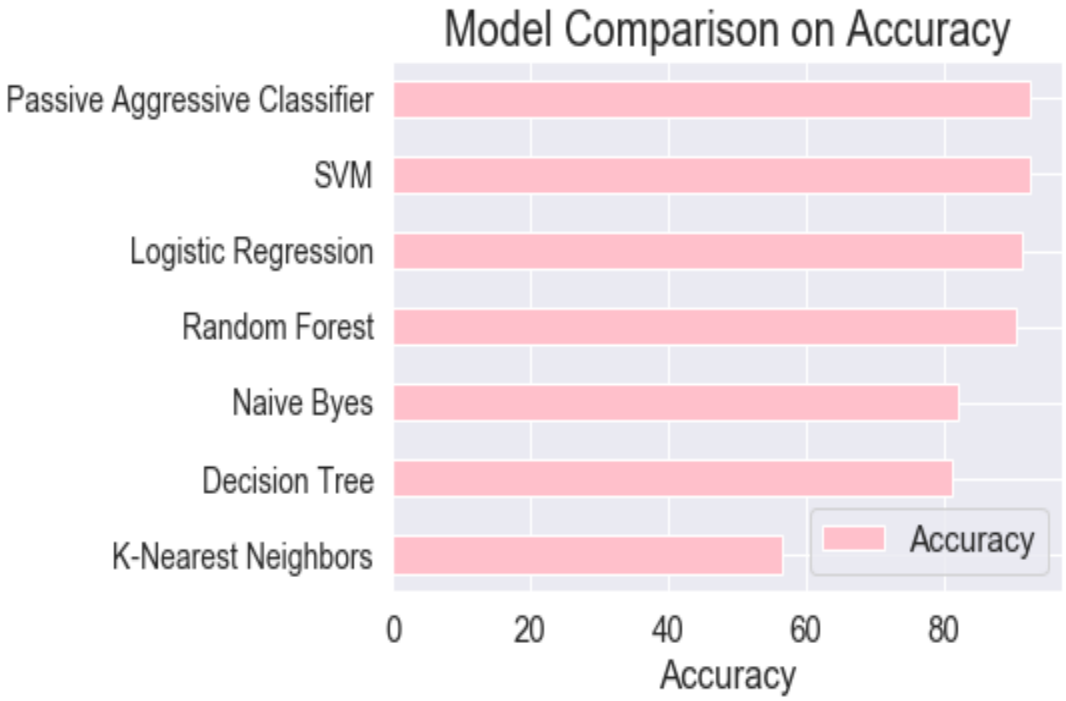
```
model_comparison = pd.DataFrame({'ML_Models': ['Passive Aggressive Classifier', 'Naive Byes',
                                               'Logistic Regression', 'Decision Tree',
                                               'K-Nearest Neighbors', 'Random Forest','SVM'],
                                 'Accuracy': [accuracy_PA, accuracy_NB, accuracy_Log,
                                              acc_DT, accKNN,accRF,accSVM]})

model_comparison.sort_values('Accuracy', ascending = True).plot(x = 'ML_Models', y =
'Accuracy', kind = 'barh',
                                                                color = 'pink', edgecolor = 'white')

plt.ylabel('');
plt.yticks(size = 14);
plt.xlabel('Accuracy');
plt.xticks(size = 15)
plt.title('Model Comparison on Accuracy', size = 20);
```

9. OUTPUT SCREEN SHOTS:

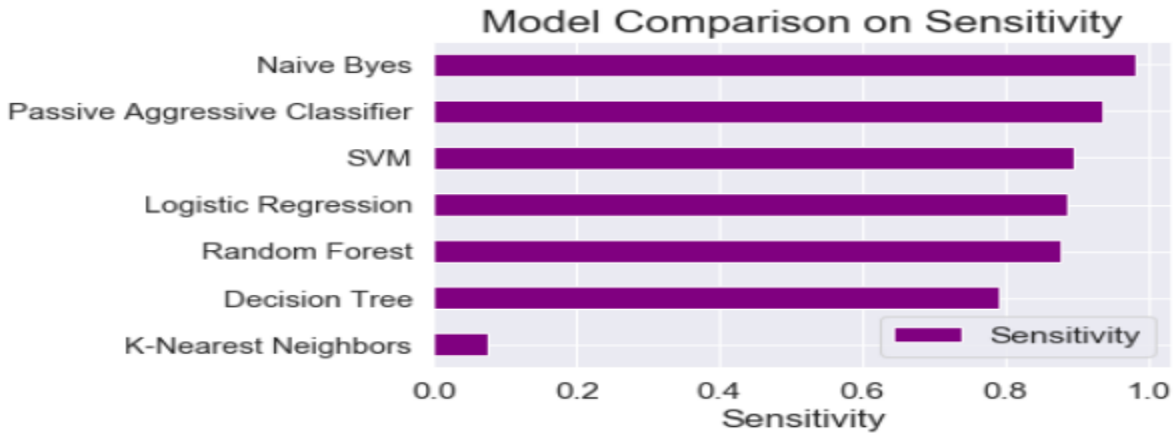
➢ RESULTS

After a thorough hyper parameter tuning on our best performing model, we evaluated the model on test data. Since our intention is to accurately measure the closeness of the predicted stance to the original, we choose '*Classification Accuracy*', '*Sensitivity*' and '*Specifity*' as our evaluation metric. The predicted accuracy for the models is presented in Table below:

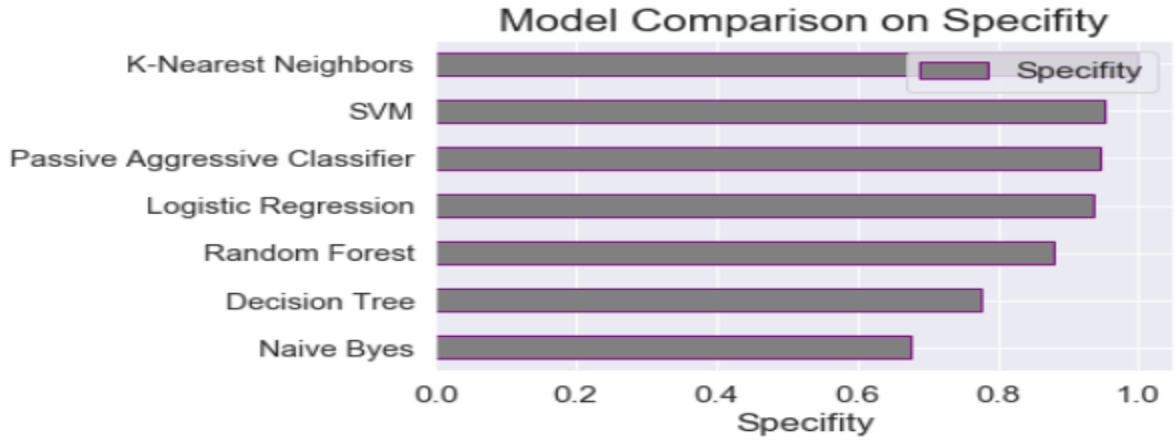
- **MODELS COMPARISON ON ACCURACY:**



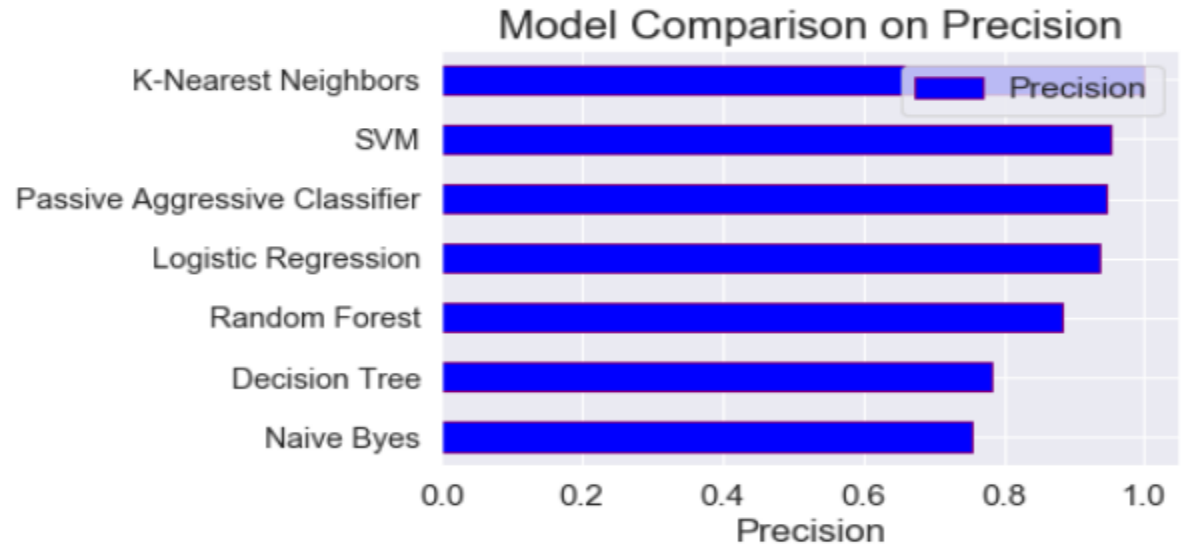
- **MODELS COMPARISON ON SENSITIVITY:**



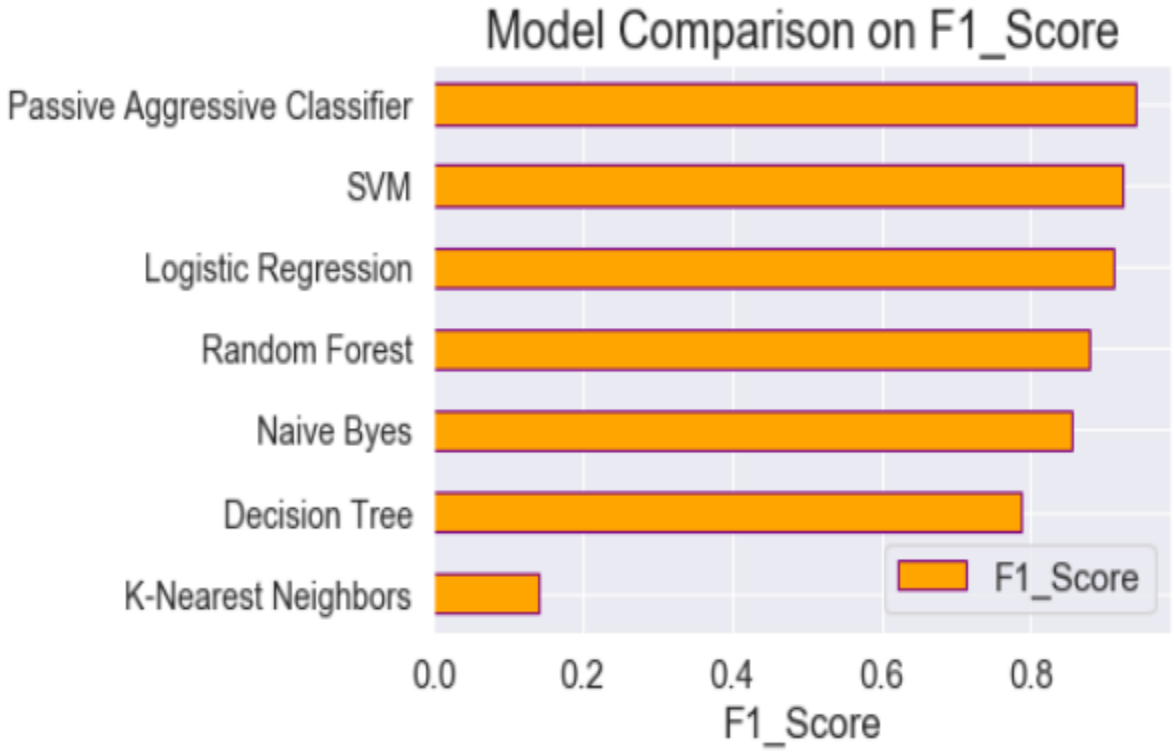
- **MODELS COMPARISON ON SPECIFITY:**



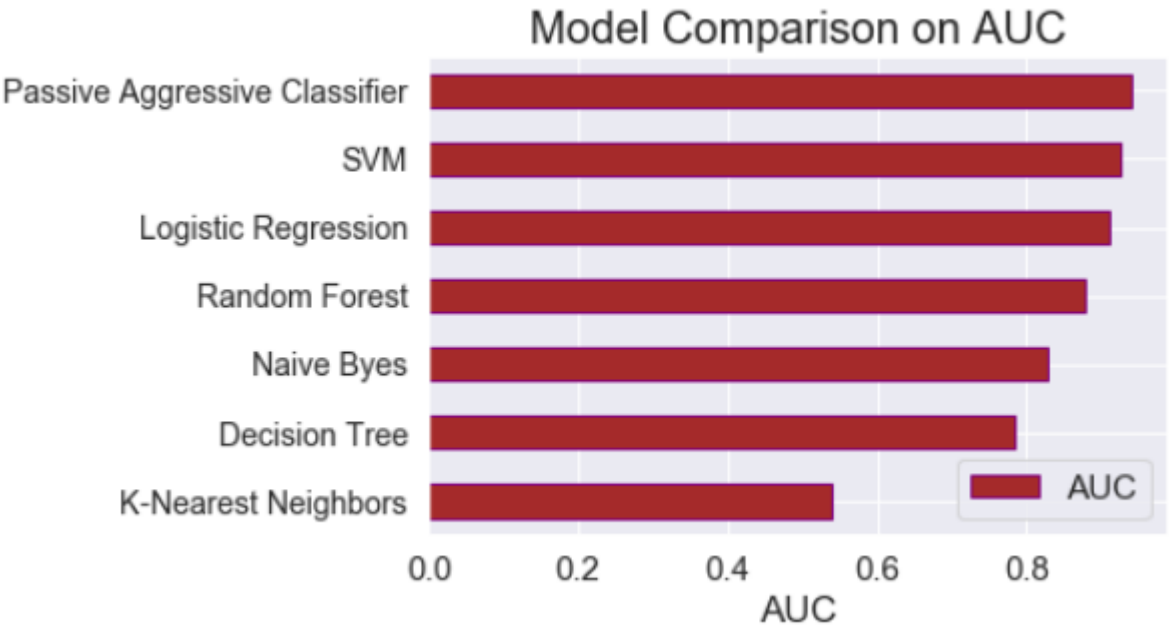
- **MODELS COMPARISON ON PRECISION:**



- **MODELS COMPARISON ON F1-SCORE:**



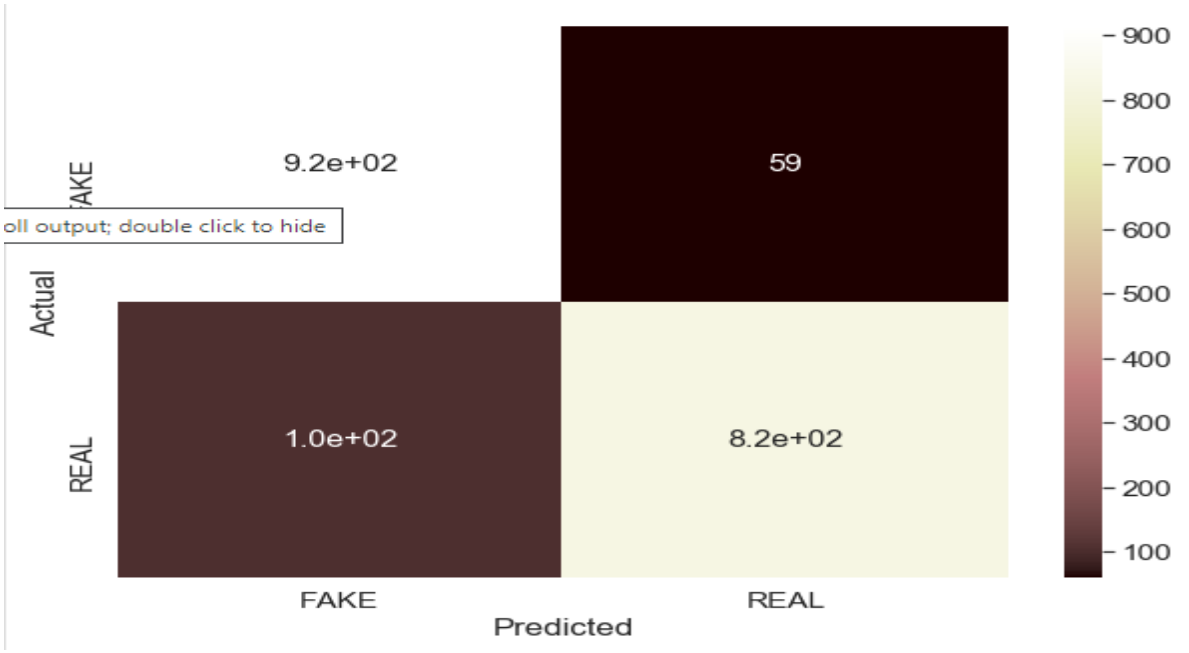
- **MODELS COMPARISON ON AUC:**



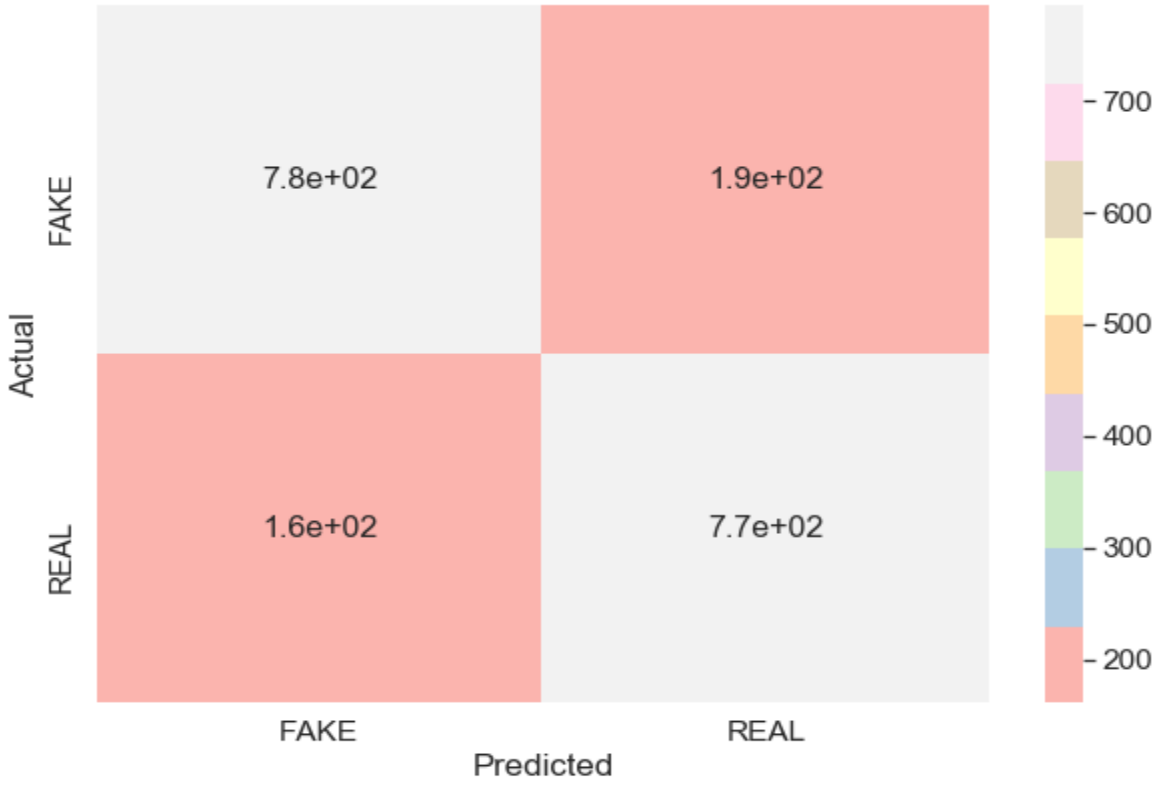
CONFUSION MATRIX OF PASSIVE AGGRESSIVE CLASSIFIER:

- ## CONFUSION MATRIX OF MULTINOMIAL NAÏVE BYES:

- **CONFUSION MATRIX OF LOGISTIC REGRESSION:**

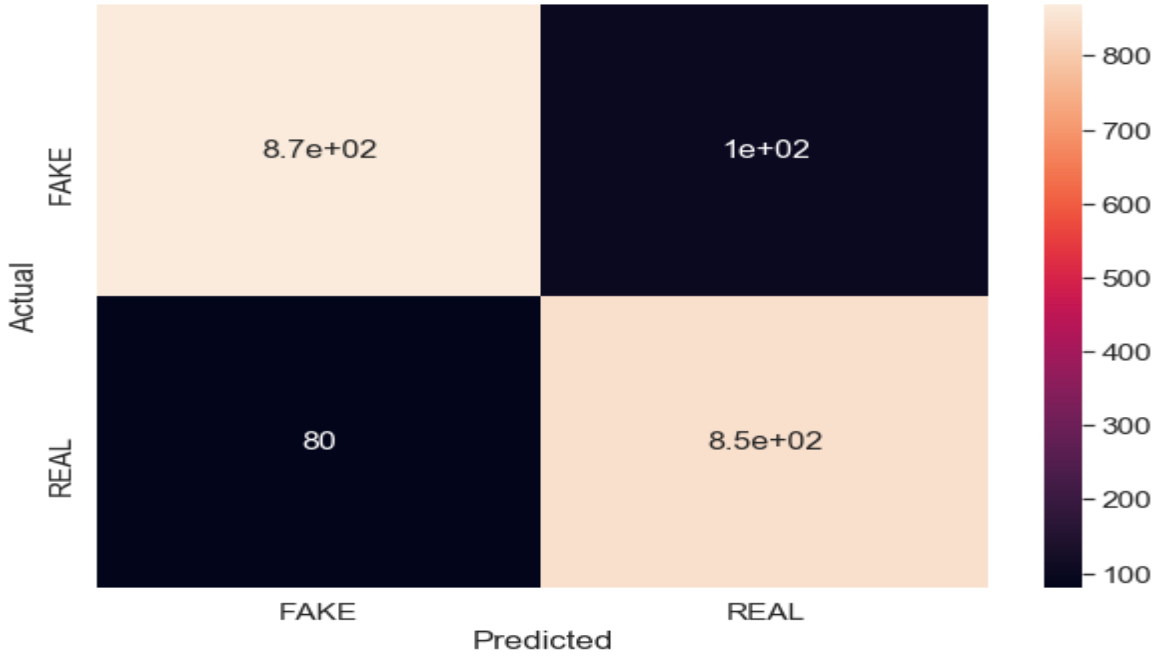


- **CONFUSION MATRIX OF DECISION TREE:**

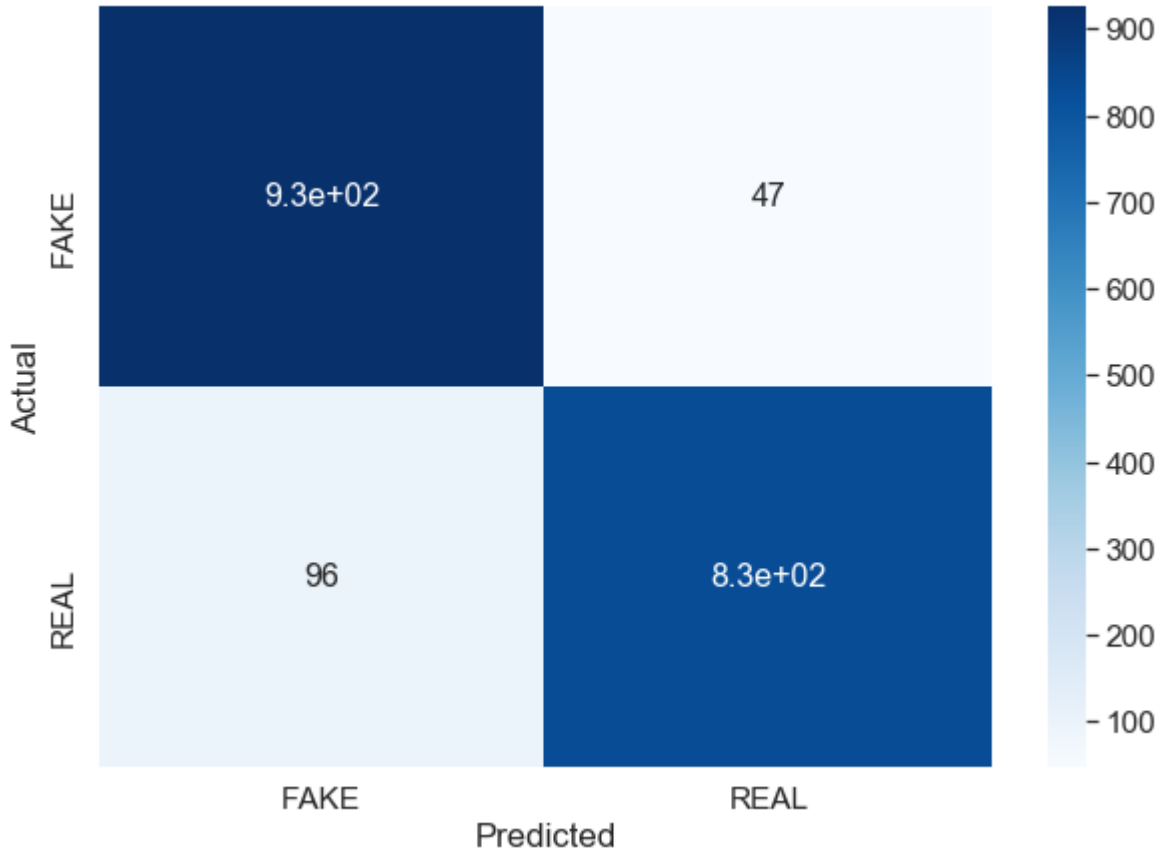


- **CONFUSION MATRIX OF K-NEAREST NEIGHBOUR:**

- **CONFUSION MATRIX OF RANDOM FOREST:**



- **CONFUSION MATRIX OF SUPPORT VECTOR MACHINE (SVM):**



10. SUPPORTING TOOLS:

➢ LANGUAGE

- PYTHON

➢ SOFTWARE

- Anaconda (JUPYTER NOTEBOOK)

➢ LIBRARIES

```
import numpy as np
import pandas as pd
```

- **MACHINE LEARNING MODELS**

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.svm import SVC
from sklearn import tree
```

- **GRAPH PLOTTING**

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, confusion_matrix
```

11. CONCLUSION:

We use TF-IDF for text transformation into metrix. We use 6 different machine learning model for detecting whether a news article is Fake or Real. We got the following accuracy:

SNO	MODELS	ACCURACY	SENSITIVITY	SPECIFITY	PRECISION	RECALL	F1_SCORE	AUC
01	Passive Aggressive Classifier	92.48%	93%	94%	94%	93%	94%	0.9437
02	Multinomial Naïve Byes	83.11%	98%	67%	75%	98%	85%	0.8301
03	Logistic Regression	91.37%	88%	93%	93%	88%	90%	0.9161
04	Decision Tree	78.30%	79%	77%	78%	79%	78%	0.8023
05	Random Forest	87.85%	87%	87%	87%	87%	88%	0.9047
06	K-Nearest Neighbor	56.60%	0.01%	100%	100%	0%	13%	0.5585
07	Support Vector Machine	92.48%	89%	95%	95%	89%	92%	0.9350

This study provides a baseline for the future tests and broadens scope of the solutions dealing with fake news detection. In future, we will work on Fake images and videos detection.

12. REFERENCE:

- https://scholar.smu.edu/cgi/viewcontent.cgi?article=1036&context=datasciencereview