Fake News Identification

Submitted by:

KUMAR GOURABH (kumargourabh94@gmail.com)

ACKNOWLEDGMENT

I would like to thank my mentors at Data Trained, who taught me the concepts of Data Analysis, building a machine learning model, and tuning the parameters for best outcomes.

For this particular task, I referred the following websites and articles when stuck:

- https://towardsdatascience.com/a-common-mistake-to-avoid-when-encoding-ordinal-features-79e402796ab4

- https://stackoverflow.com/questions/43590489/gridsearchcv-random-forest-regressor-tuning-best-params

- https://www.codegrepper.com/code-examples/delphi/scikit+pca+preserve+column+names+pca+pipeline

- https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-pca-with-sklearn

I would also like to thank my mentor in Fliprobo, Manikant Jha, for providing me with the dataset and problem statement for performing this wonderful task.

INTRODUCTION

Business Problem Framing

Need to classify if a news is fake or not, based on the Author's name, Headline and the actual content of the article.

Conceptual Background of the Domain Problem

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed. In this project, you are given a dataset in the fake-news_data.zip folder. The folder contains a CSV files train_news.csv and you have to use the train_news.csv data to build a model to predict whether a news is fake or not fake. You must try out different models on the dataset, evaluate their performance, and finally report the best model you got on the data and its performance.

You must submit the jupyter notebook, in which you have built your best performing model. Your jupyter notebook should be well commented so that it can be easily understood what you are trying to do in the code. Also mention which is your best performing model and the measure of its performance (accuracy score , f1 score etc)

Data- Description:

There are 6 columns in the dataset provided to you. The description of each of the column is given below:

- "id": Unique id of each news article
- "headline": It is the title of the news.
- "news": It contains the full text of the news article
- "Unnamed:0": It is a serial number
- "written_by": It represents the author of the news article
- "label": It tells whether the news is fake (1) or not fake (0).

Analytical Problem Framing

Data Sources and their formats

	Unnamed: 0	id	headline	written_by	news	label
0	0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0
2	2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1
...	...	...	...	...	...	..
20795	20795	5671	NaN	NeverSurrender	No, you'll be a dog licking of the vomit of yo...	1
20796	20796	14831	Albert Pike and the European Migrant Crisis	Rixon Stewart	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	20797	18142	Dakota Access Caught Infiltrating Protests to ...	Eddy Lavine	posted by Eddie You know the Dakota Access Pip...	1
20798	20798	12139	How to Stretch the Summer Solstice - The New Y...	Alison S. Cohn	It's officially summer, and the Society Boutiq...	0
20799	20799	15660	Emory University to Pay for '100 Percent' of U...	Tom Ciccotta	Emory University in Atlanta, Georgia, has anno...	0

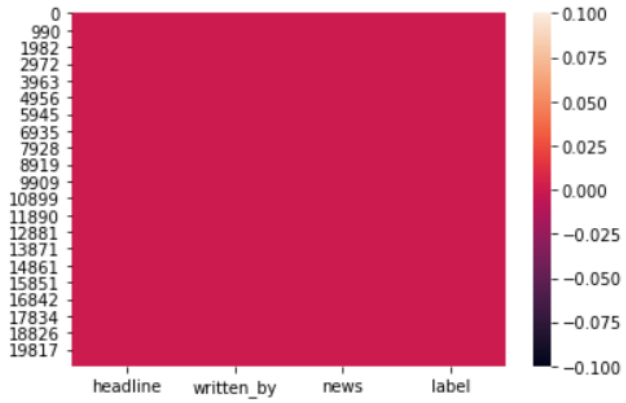
20800 rows × 6 columns

Handled Missing Values:

```
# Treating missing values
# 1. Headline: Replacing with 'No Headline'
df['headline'].fillna('No Headline',inplace=True)

# 2. written_by
df['written_by'].fillna('No Author',inplace=True)

# news: i will drop these rows
df.dropna(subset=['news'],inplace=True)
```

```
# Hetmap for null values
sns.heatmap(df.isnull())
```

```
<AxesSubplot:>
```



Calculated Lengths of Features:

	headline	written_by	news	label	length_headline	length_written_by	length_news
0	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0	84	26	7936
1	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0	72	15	6112
2	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0	100	7	425
3	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0	100	12	6516
4	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1	28	17	9164

Using NLP to Solve Problem:

Here the idea is to vectorize the contents so that the meaning of the article can be understood by the machine. The following pre-processing steps were done:

```
df['headline'] = df.headline.str.lower()
df['written_by'] = df.written_by.str.lower()
df['news'] = df.news.str.lower()
```

```
cols=['headline','written_by','news']
for j in cols:
    # Replace email addresses with 'email'
    df[j] = df[j].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$',
                                      'emailaddress')

    # Replace URLs with 'webaddress'
    df[j] = df[j].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$',
                                      'webaddress')

    # Replace money symbols with 'moneysymb' (£ can by typed with ALT key + 156)
    df[j] = df[j].str.replace(r'£|\$', 'dollers')

    # Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
    df[j] = df[j].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$',
                                      'phonenumber')


    # Replace numbers with 'numbr'
    df[j] = df[j].str.replace(r'\d+(\.\d+)?', 'numbr')

    # Remove punctuation
    df[j] = df[j].str.replace(r'[^\w\d\s]', ' ')

    # Replace whitespace between terms with a single space
    df[j] = df[j].str.replace(r'\s+', ' ')

    # Remove leading and trailing whitespace
    df[j] = df[j].str.replace(r'^\s+|\s+?$', '')
```

Stop Words Removal: We need to remove words like for, of, the etc. which donot actually add to the context so that our processing speed is reduced.

```
# Remove stopwords
import string
import nltk
from nltk.corpus import  stopwords

stop_words = set(stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure'])

# 'headline','written_by','news'

df['headline'] = df['headline'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))

df['written_by'] = df['written_by'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))

df['news'] = df['news'].apply(lambda x: ' '.join(
    term for term in x.split() if term not in stop_words))
```

```
from nltk.tokenize import RegexpTokenizer
tokenizer=RegexpTokenizer(r'\w+')
df['headline'] = df['headline'].apply(lambda x: tokenizer.tokenize(x.lower()))
df['written_by'] = df['written_by'].apply(lambda x: tokenizer.tokenize(x.lower()))
df['news'] = df['news'].apply(lambda x: tokenizer.tokenize(x.lower()))
df.head()
```

	headline	written_by	news	label	length_headline	length_written_by	length_news
0	[ethics, questions, dogged, agriculture, nomin...	[eric, lipton, steve, eder]	[washington, sonny, perdue, telling, georgians...	0	84	26	7936
1	[must, dig, deep, stop, argentina, lionel, mes...	[david, waldstein]	[houston, venezuela, plan, tactical, approach,...	0	72	15	6112
2	[cotton, house, walk, plank, vote, bill, canno...	[pam, key]	[sunday, abc, week, discussing, republican, pl...	0	100	7	425
3	[paul, lepage, besieged, maine, governor, send...	[jess, bidgood]	[augusta, beleaguered, republican, governor, m...	0	100	12	6516

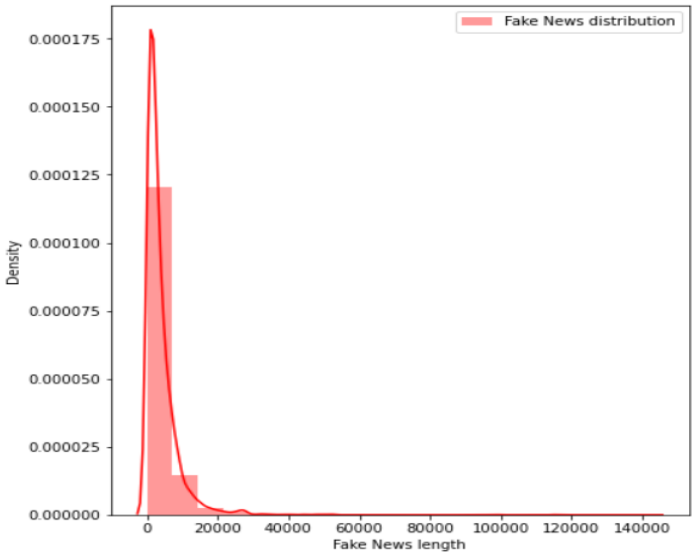
I have performed lemmatization and stemming on the features – Headline and news. This is done to find the root words.

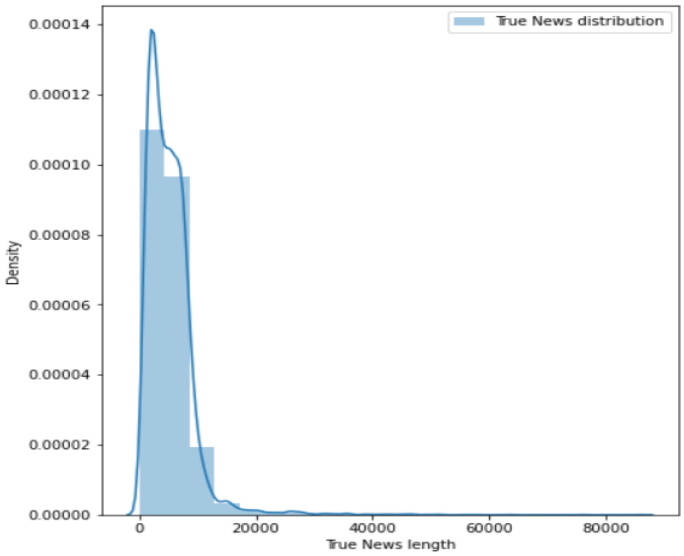
```
# writing function for the entire dataset
# Lemmatizing and then Stemming with Snowball to get root words and further reducing characters

from nltk.stem import SnowballStemmer, WordNetLemmatizer
stemmer = SnowballStemmer("english")
import gensim
def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text,pos='v'))

#Tokenize and Lemmatize
def preprocess(text):
    result=[]
    for token in text:
        if len(token)>=3:
            result.append(lemmatize_stemming(token))

    return result
```

Finally, we can see the counts of actual data and data after pre-processing

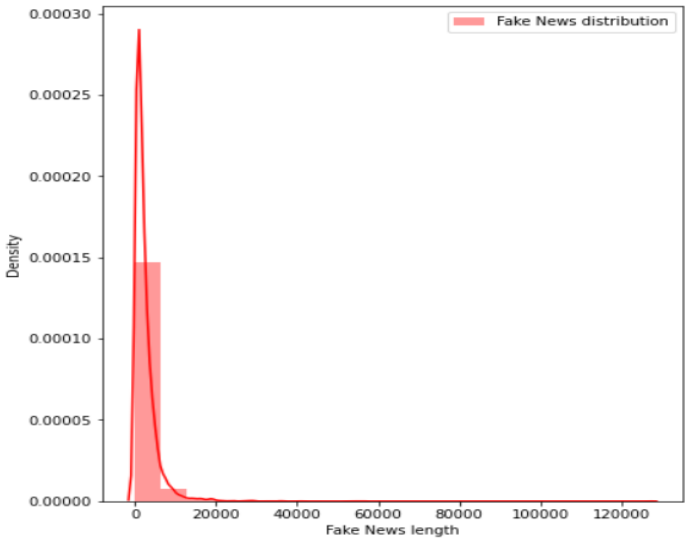
	headline	written_by	news	label	length_headline	length_written_by	length_news	clean_headline	clean_news	clean_length_headline	clean_length_
0	ethic question dog agricultur nomine georgia g...	eric lipton steve eder	washington sonni perdu tell georgian grow wear...	0	84	26	7936	[ethic, question, dog, agricultur, nomine, geo...	[washington, sonni, perdu, tell, georgian, gro...	67	
1	must dig deep stop argentina lionel messi new ...	david waldstein	houston venezuela plan tactic approach design ...	0	72	15	6112	[must, dig, deep, stop, argentina, lionel, mes...	[houston, venezuela, plan, tactic, approach, d...	55	
2	cotton hous walk plank vote bill cannot pass s...	pam key	sunday abc week discuss republican plan repeal...	0	100	7	425	[cotton, hous, walk, plank, vote, bill, cannot...	[sunday, abc, week, discuss, republican, plan,...	60	

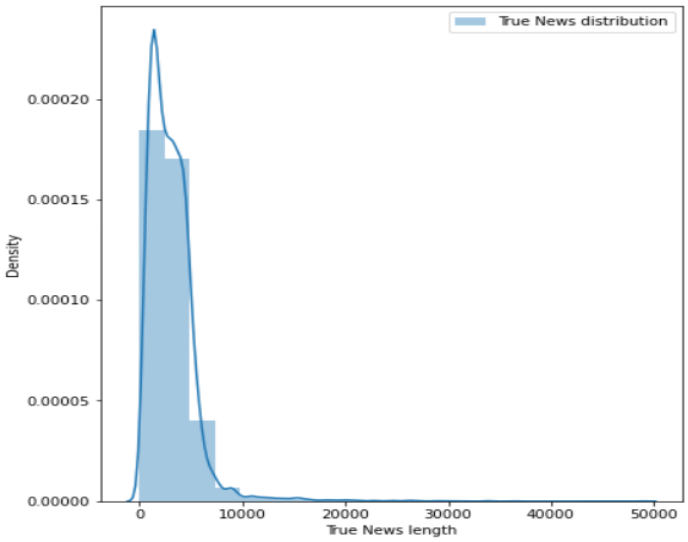
Distribution of News before Cleaning



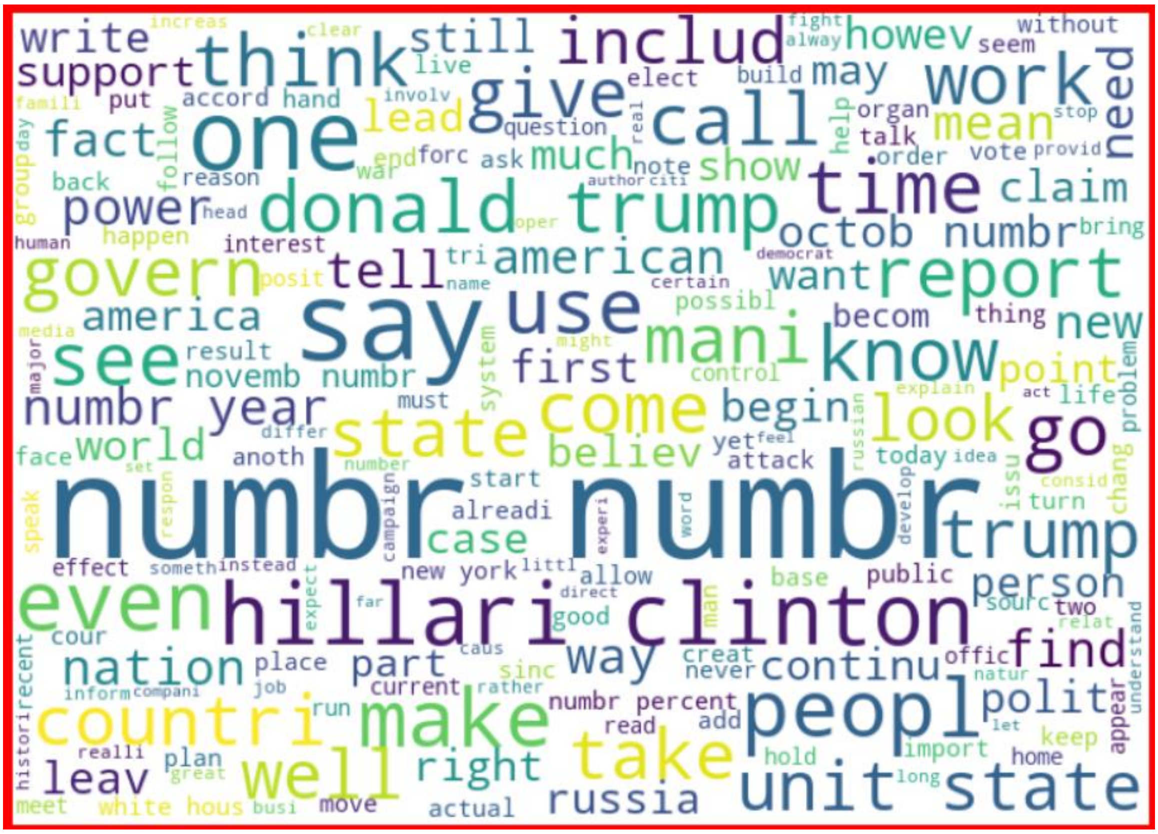


Distribution of News after Cleaning

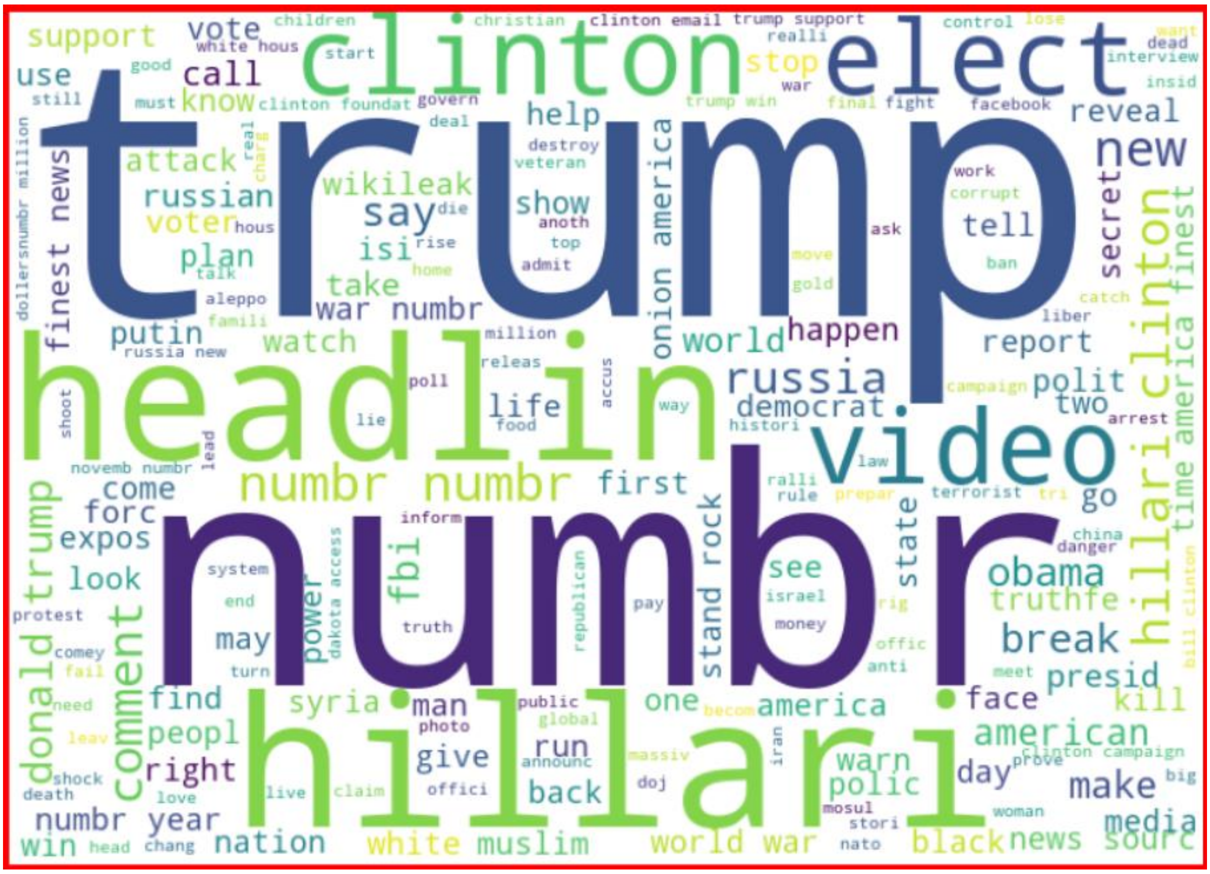




Word Cloud for Fake News for News Column



Word Cloud for Fake News for Headline Column



Next, I have vectorized using tf-idf vectorizer so that words are arranged in a 2-d area based on similarity or difference in their meaning.

Also, prepared X and y variables for model building.

```
tf_vec = TfidfVectorizer()
features = tf_vec.fit_transform(df['written_by'] + df['headline'] + df['news'])

X = features
y = df['label']
```

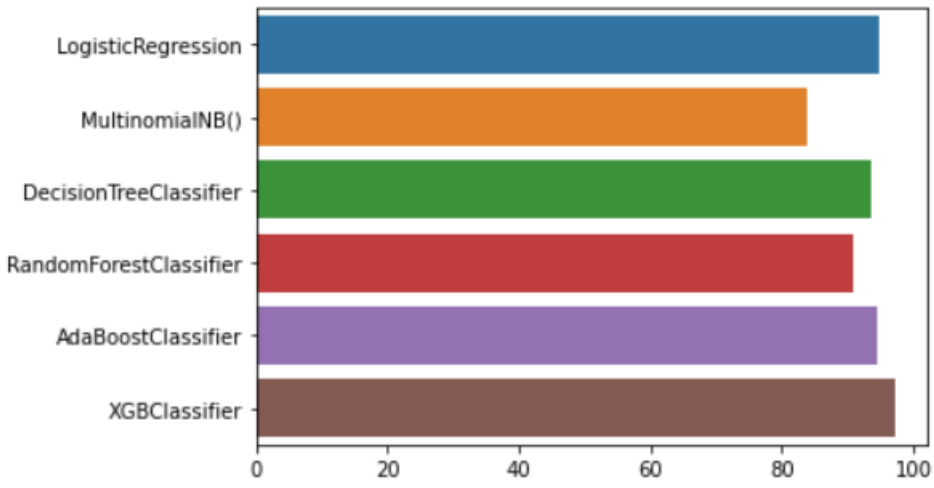
Let's have a look at the Model Performances

	Model	Learning Score	Accuracy Score	Cross Val Score	Roc_Auc_curve
0	LogisticRegression	96.786402	94.894847	98.765770	94.894784
1	MultinomialNB()	87.737407	83.833681	97.444174	83.826232
2	DecisionTreeClassifier	100.000000	93.417884	93.218256	93.417712
3	RandomForestClassifier	100.000000	90.961631	97.631336	90.959982
4	AdaBoostClassifier	94.068263	94.349013	98.317529	94.349197
5	XGBClassifier	99.944949	97.126345	99.536547	97.126259

Accuracy Scores of Various:

```
sns.barplot(y=Model,x=Acc_score)
```

```
<AxesSubplot:>
```



XG Boost seems to be the best performing model. But Let's try to tune the hyperparameters of Logistic Regression to see if we get a better model score

```
from sklearn.model_selection import GridSearchCV
def grid_cv(mod,parameters,scoring):
    clf = GridSearchCV(mod,parameters,scoring, cv=10)
    clf.fit(X,y)
    print(clf.best_params_)
```

```
# Using Grid Search CV
lr=LogisticRegression()
parameters={'penalty': ['l1', 'l2'],'C':[0.001,.009,0.01,.09,1,5,10,25]}
grid_cv(lr,parameters,'accuracy')
```

```
{'C': 25, 'penalty': 'l2'}
```

```
clf_lr = LogisticRegression(C=25,penalty='l2')
max_acc_score(clf_lr,X,y)
```

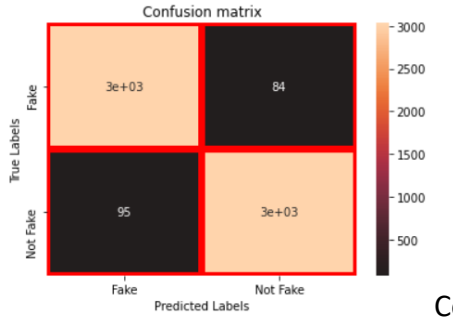
```
Max Accuracy Score corresponding to Random State  58 is: 0.9616310804302456


58
```

```
# Accuracy Score after tuning Logistic Regression is 0.96 and XG Boost is 0.97. I'm selecting XGBoost as the final model
```

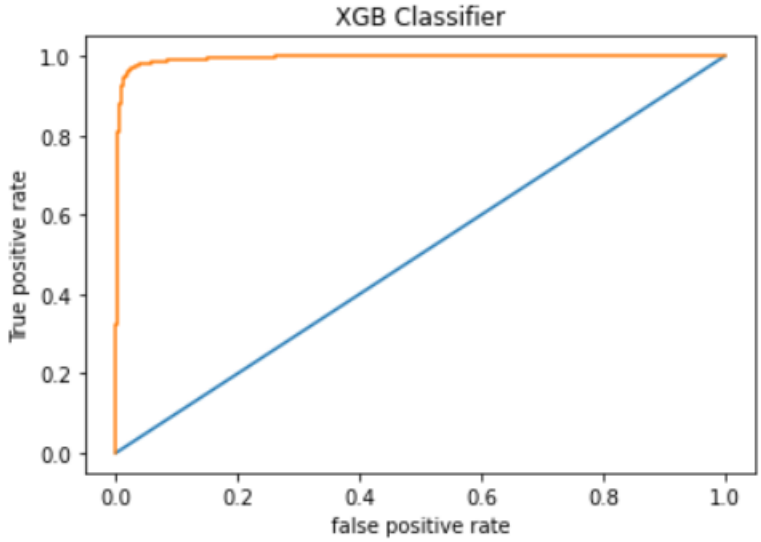
Final Model

```
x_train,x_test,y_train,y_test=train_test_split(X,y,random_state=98,test_size=.30,stratify=y)
XG=XGBClassifier()
XG.fit(x_train,y_train)
XG.score(x_train,y_train)
XGpred=XG.predict(x_test)
print('Accuracy Score:',accuracy_score(y_test,XGpred))
print('Confusion Matrix:\n',confusion_matrix(y_test,XGpred))
print('Classification Report:','\n',classification_report(y_test,XGpred))
```

```
[00:43:00] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.
0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly se
t eval_metric if you'd like to restore the old behavior.
Accuracy Score: 0.9712634451757907
Confusion Matrix:
 [[3032   84]
 [  95 3018]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      3116
           1       0.97      0.97      0.97      3113

    accuracy                           0.97      6229
   macro avg       0.97      0.97      0.97      6229
weighted avg       0.97      0.97      0.97      6229
```



Confusion Matrix



roc_auc_score = 0.9952208779531115

	label	Predicted values
7342	1	1
2997	0	0
7882	1	1
13115	1	1
6538	1	1
...	...	...
7832	1	1
250	1	1
15308	0	0
15397	0	0
510	1	1

6229 rows × 2 columns