# ELG 5142:
# Ubiquitous Sensing / Smart Cities
# Assignment 2

Dr. Burak Kantarci

GROUP 14

# Table of Contents

# Table of Figures

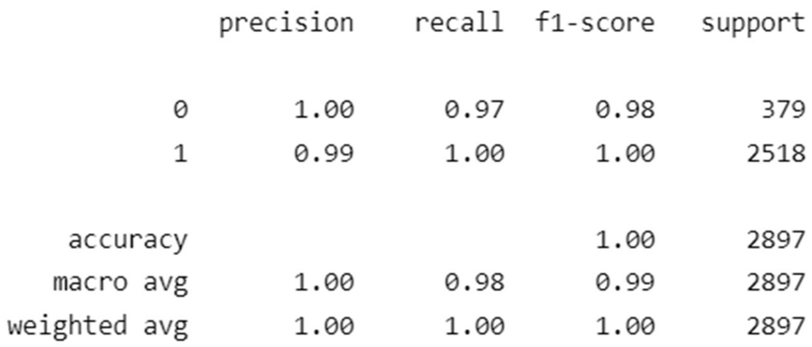
## 1. Code:

```
# Import Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, accuracy_score
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
import seaborn as sns # for data visualization
sns.reset_defaults()

# Loading the data
df = pd.read_csv('MCSDatasetNEXTCONLab.csv')
df.head()

# Splitting
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
print(X_train.shape)
print(X_test.shape)

# Random Forest model training
rf = RandomForestClassifier().fit(X_train, y_train)

# Adaboost model training
adaboost = AdaBoostClassifier(n_estimators=100, random_state=0).fit(X_train, y_train)

# Gaussian Naive Bayes
gnb = GaussianNB().fit(X_train, y_train)

# Function to show classification report and plot the confusion matrix
def report_and_confusion(y_test, y_pred):
    labels1 = ['Fake', 'Legitimate']
    print(classification_report(y_test, y_pred))
    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels1)
    disp.plot()

    plt.show()
# Random Forest model testing
```

```
y_pred1 = rf.predict(X_test)
report_and_confusion(y_test, y_pred1)

# Adaboost model testing
y_pred2 = adaboost.predict(X_test)
report_and_confusion(y_test, y_pred2)

# Gaussian Naive testing
y_pred3 = gnb.predict(X_test)
report_and_confusion(y_test,y_pred3)

# Agrregator based on majority voting rule
y_pred4 = (y_pred1 + y_pred2 + y_pred3) >= 2
report_and_confusion(y_test, y_pred4)

# Agrregator based on weighted sum of classifiers decisions
pred1 = rf.predict(X_train)
acc1 = accuracy_score(y_train, pred1)
pred2 = adaboost.predict(X_train)
acc2 = accuracy_score(y_train, pred2)
pred3 = gnb.predict(X_train)
acc3 = accuracy_score(y_train, pred3)
wrf = acc1/(acc1 + acc2 + acc3)
wada = acc2/(acc1 + acc2 + acc3)
wgnb = acc3/(acc1 + acc2 + acc3)
y_pred5 = (y_pred1 * wrf + y_pred2 * wada + y_pred3 * wgnb) > 0.5
print("Train Accuracies: ", acc1, acc2, acc3)
print("Weights: ", wrf, wada, wgnb)
report_and_confusion(y_test, y_pred5)

# Calculating accuracies and plotting the bar plot
test_acc1 = accuracy_score(y_test, y_pred1)
test_acc2 = accuracy_score(y_test, y_pred2)
test_acc3 = accuracy_score(y_test, y_pred3)
test_acc4 = accuracy_score(y_test, y_pred4)
test_acc5 = accuracy_score(y_test, y_pred5)
test_acc_ls=[test_acc1,test_acc2,test_acc3,test_acc4,test_acc5]
print(test_acc_ls)
sns.set()
plt.figure(figsize=(10,5))
g=sns.barplot(x = ['RF', 'Adaboost', 'NB', 'Ensemble-vote', 'Ensemble-weighted'], y =
test_acc_ls)
g.set(ylim=(0.82, 1));
counter=0
for va in test_acc_ls:
  v=str(round(va,4)*100)+' %'
  g.text(counter,va,v, color='black', ha="center")
  counter+=1
plt.show()
```

## 2. Test Results:

### 1. RandomForest model results:



```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       379
           1       0.99      1.00      1.00      2518

    accuracy                           1.00      2897
   macro avg       1.00      0.98      0.99      2897
weighted avg       1.00      1.00      1.00      2897
```

*Figure 1 Classification report of Random Forest model*

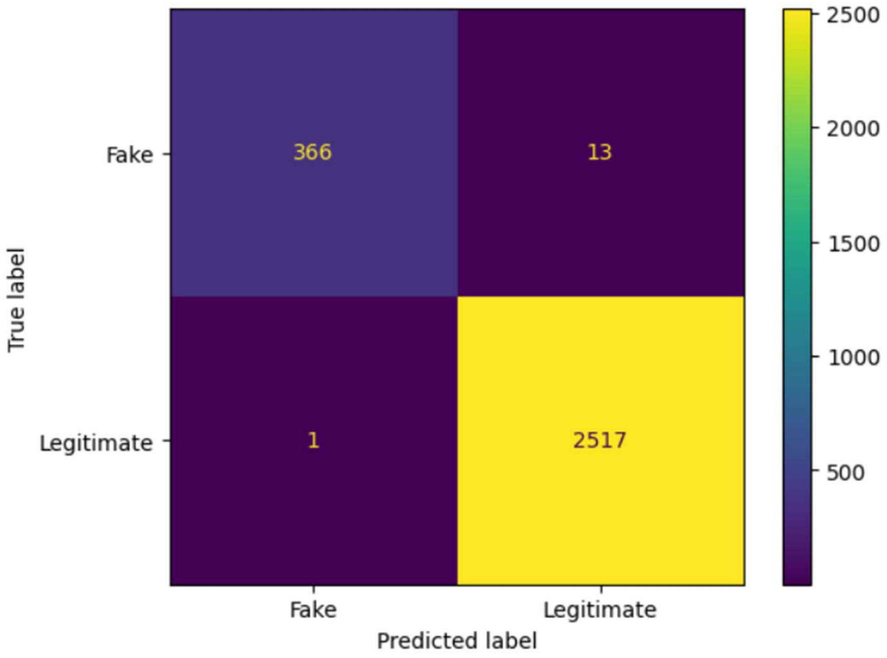
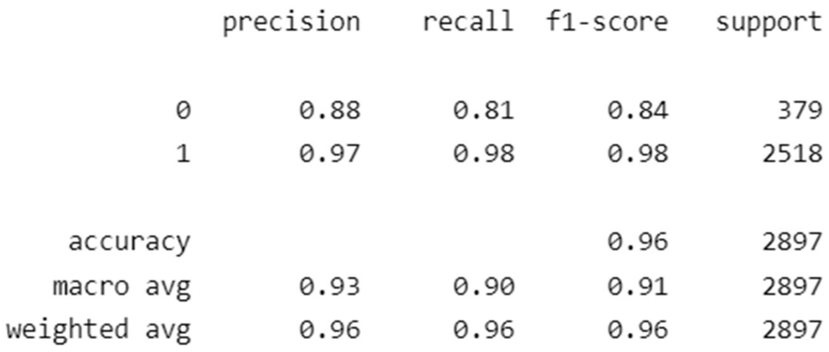


*Figure 2 Confusion matrix of Random Forest model*

## 2. Adaboost model results:



```
              precision    recall  f1-score   support

           0       0.88      0.81      0.84       379
           1       0.97      0.98      0.98      2518

    accuracy                           0.96      2897
   macro avg       0.93      0.90      0.91      2897
weighted avg       0.96      0.96      0.96      2897
```

*Figure 3 Classification report of Adaboost model*

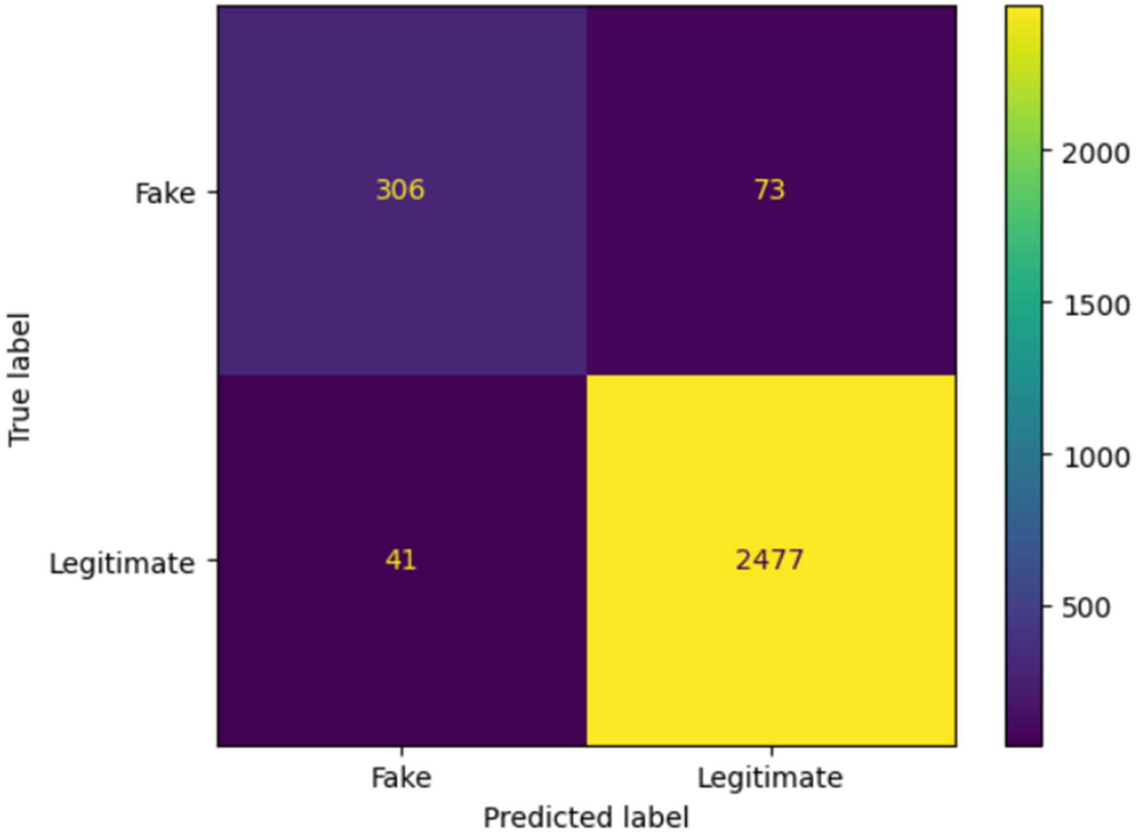
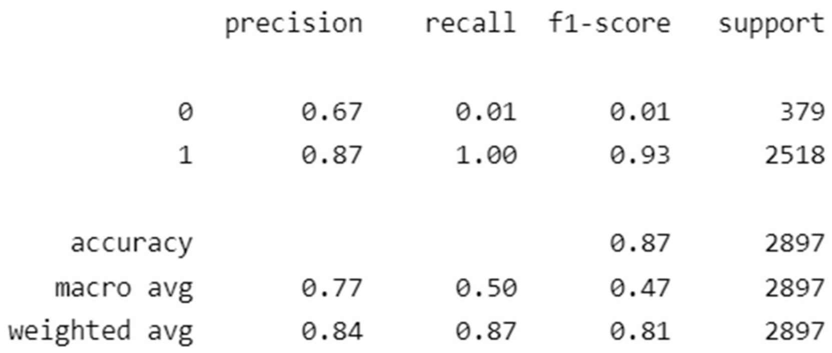


*Figure 4 Confusion matrix of Adaboost model*

## 3. Naïve Bayes model results:



```
              precision    recall  f1-score   support

           0       0.67      0.01      0.01       379
           1       0.87      1.00      0.93      2518

    accuracy                           0.87      2897
   macro avg       0.77      0.50      0.47      2897
weighted avg       0.84      0.87      0.81      2897
```

*Figure 5 Classification report of Naive Bayes model*

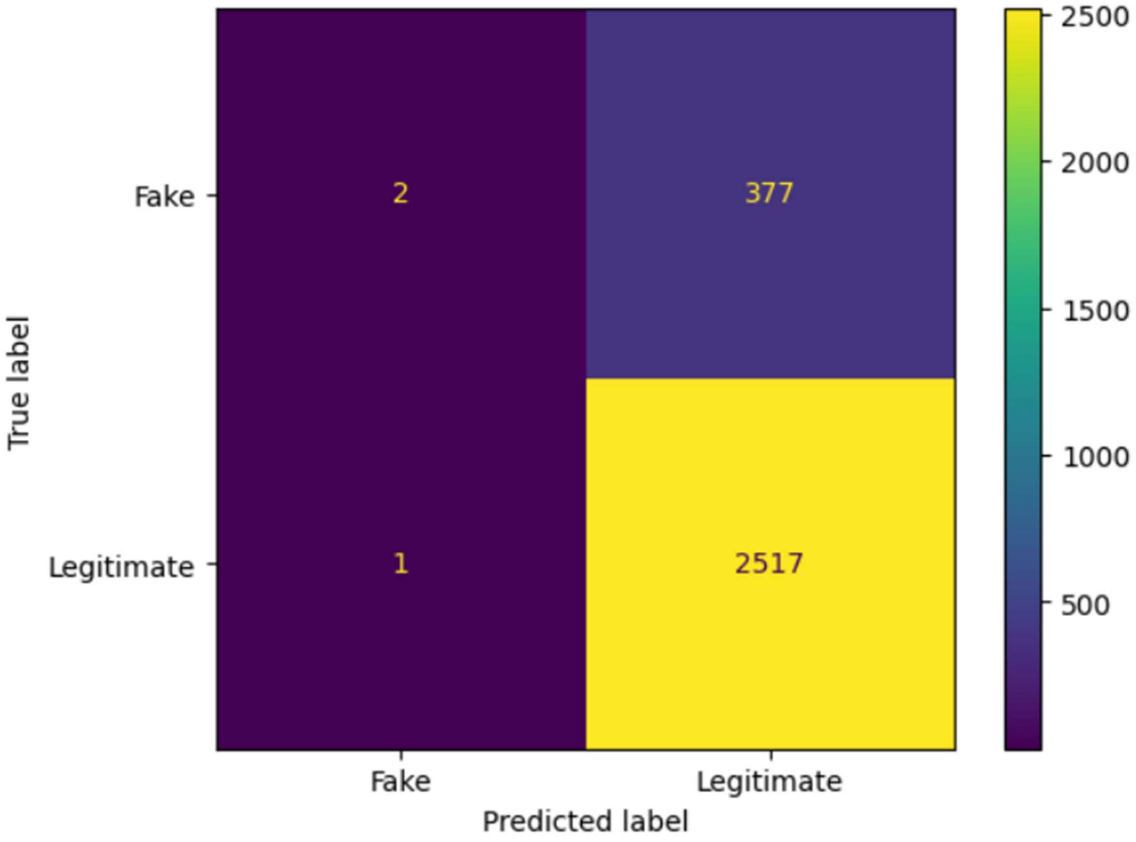
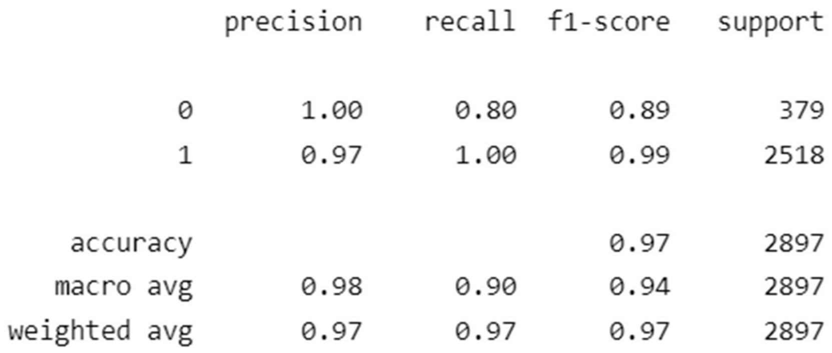


*Figure 6 Confusion matrix of Naive Bayes model*

## 4. Ensemble voting model results:



```
              precision    recall  f1-score   support

           0       1.00      0.80      0.89       379
           1       0.97      1.00      0.99      2518

    accuracy                           0.97      2897
   macro avg       0.98      0.90      0.94      2897
weighted avg       0.97      0.97      0.97      2897
```

*Figure 7 Classification report of Ensemble voting model*

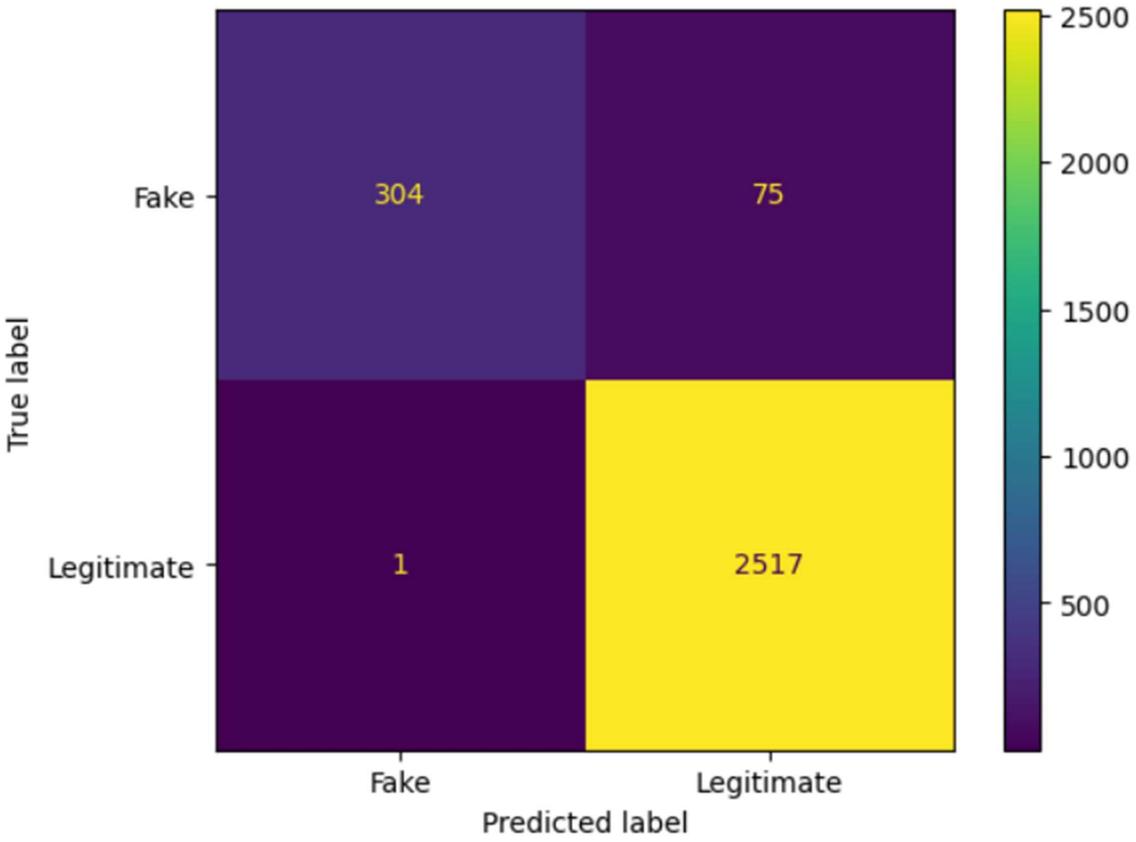
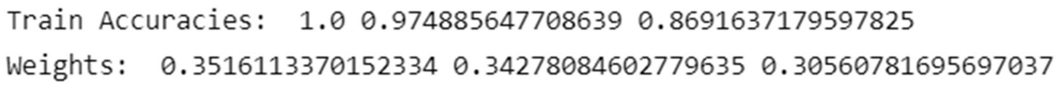


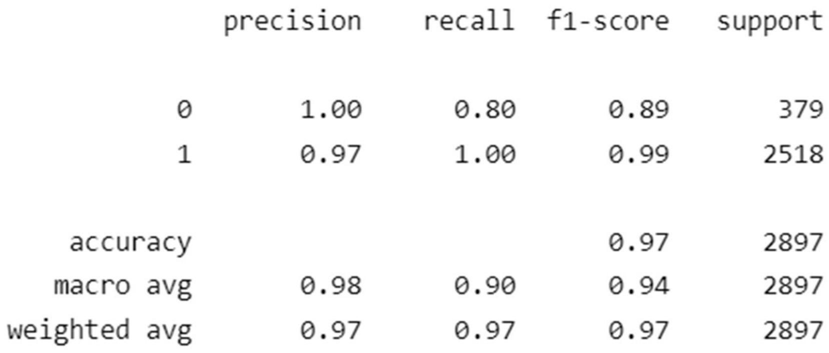
*Figure 8 Confusion matrix of Ensemble voting model*

## 5. Ensemble weighted model results:



```
Train Accuracies:  1.0 0.974885647708639 0.8691637179597825
Weights:  0.3516113370152334 0.34278084602779635 0.30560781695697037
```

*Figure 9 Train Accuracies and weights that are used to make Ensemble weighted model*



```
              precision    recall  f1-score   support

           0       1.00      0.80      0.89       379
           1       0.97      1.00      0.99      2518

    accuracy                           0.97      2897
   macro avg       0.98      0.90      0.94      2897
weighted avg       0.97      0.97      0.97      2897
```

*Figure 10 Classification report of Ensemble weighted model*

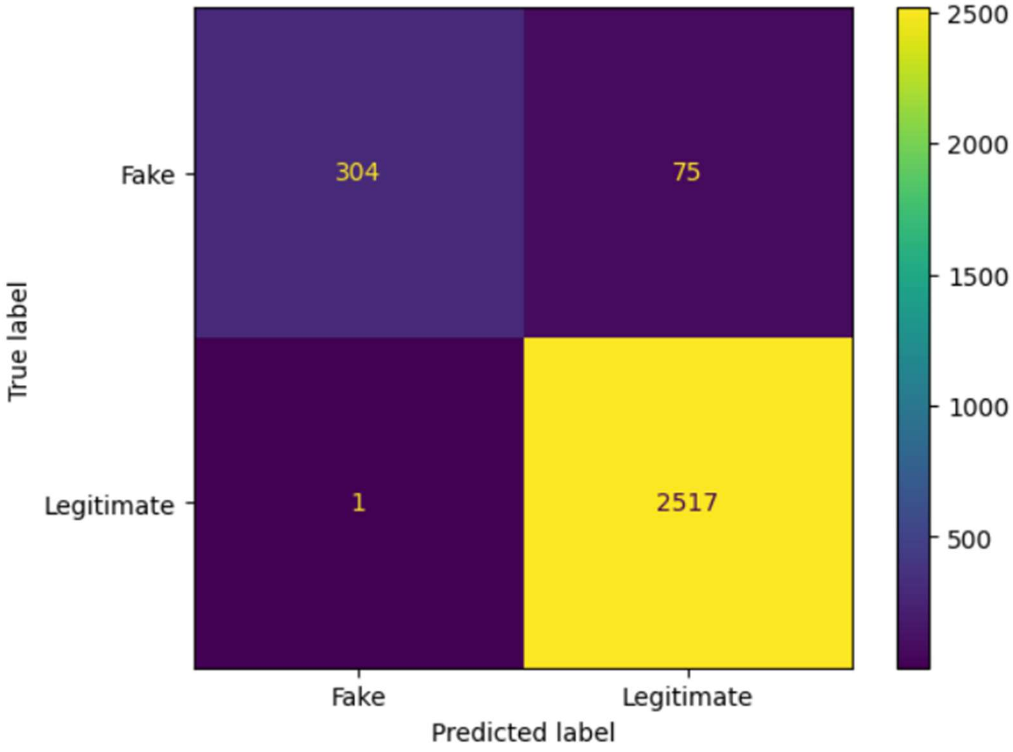


*Figure 11 Confusion matrix of Ensemble weighted model*

## 3. Accuracy Comparison:

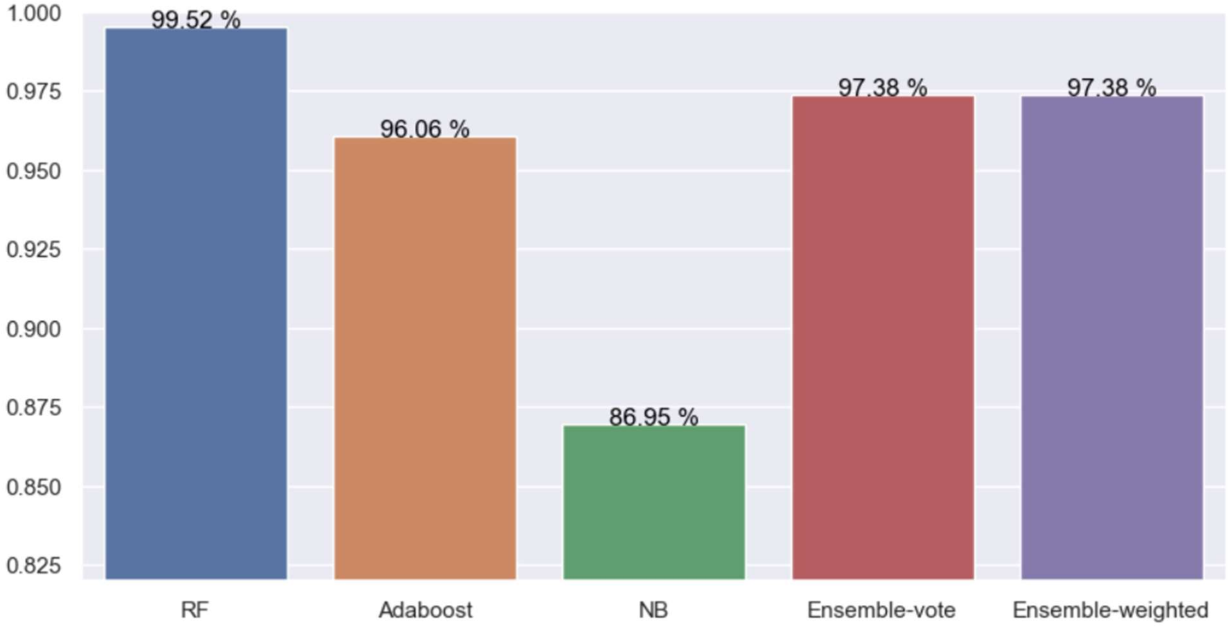


*Figure 12 Bar plot to compare accuracies of different models*

It’s concluded that Random Forest model has the highest accuracy. Ensemble voting and Ensemble weighted models have the same accuracy as the accuracies of used base estimators are high. The accuracy of Ensemble models is near to the accuracy of Adaboost model as its accuracy is higher than Naïve Bayes accuracy and lower than Random Forest accuracy (median accuracy).