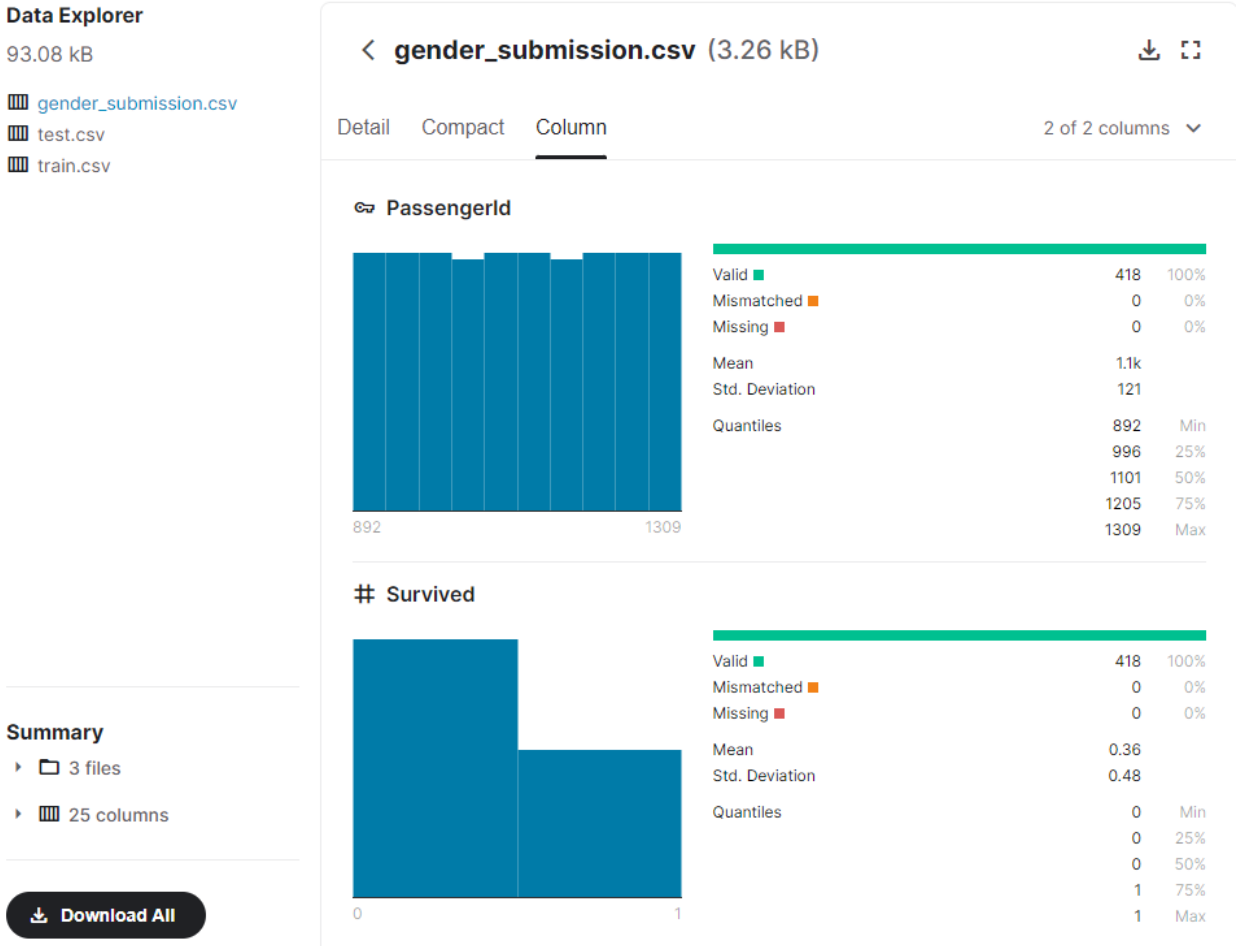
Titanic Report

- CSE445: Machine Learning
- Sec:4
- Semester: Summer ‘21
- Submitted to: Intisar Tahmid Naheen (ITN)
- Submission Date: 23-09-2021

Submitted by: **Udyan Saha Upal_1821359042**

Titanic Dataset is structured data with 12 classes. Our main goal is to work with this bunch of data to predict whether a passenger has survived based on given attributes that they have.



First I downloaded the dataset from kaggle with the kaggle API. I used google colab as my platform for Jupyter Notebook.
Then I imported the necessary libraries.

1.
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_predict
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report, confusion_matrix, roc_curve
```

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

EDA:

- As the data has been loaded, I want to find out the size of this data frame using df.shape command, which the result indicates that our train.csv contains
    - 891 rows (each representing a passenger) and
    - 12 columns (the attributes of each passenger)
- Checking all the data types:We can see here that there are int64, float64 and object.

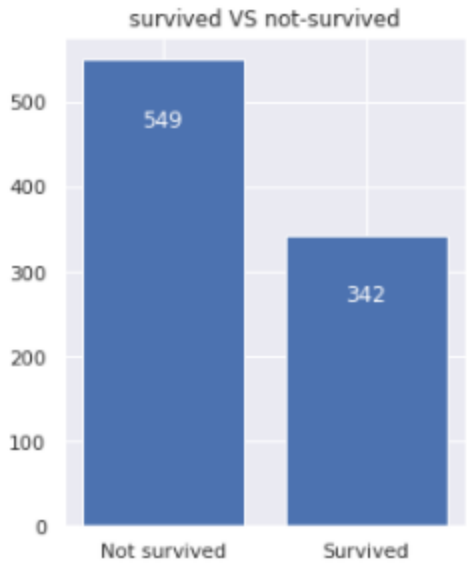
```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

-
- Checking the NaN(Not a Number) values.

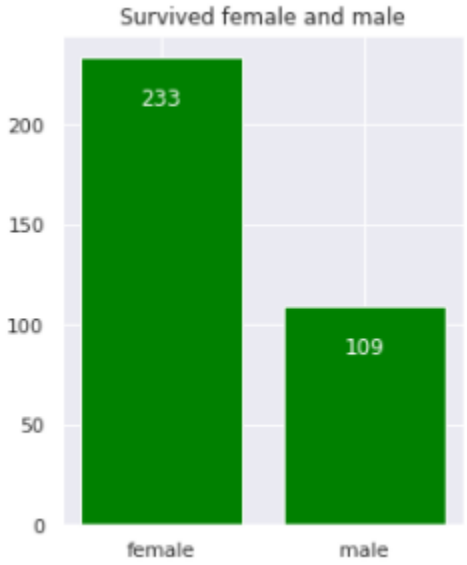
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

-
- Number of survived vs not survived passengers

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

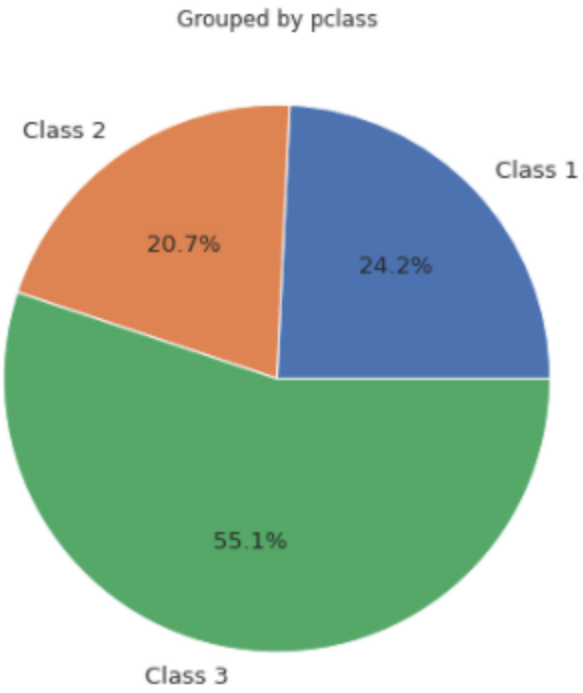


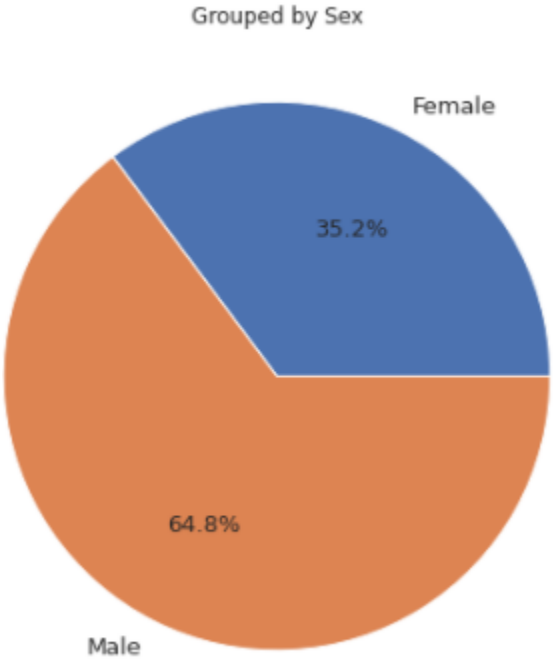
- I did the similar thing in order to find out the number of survived persons based on their gender.



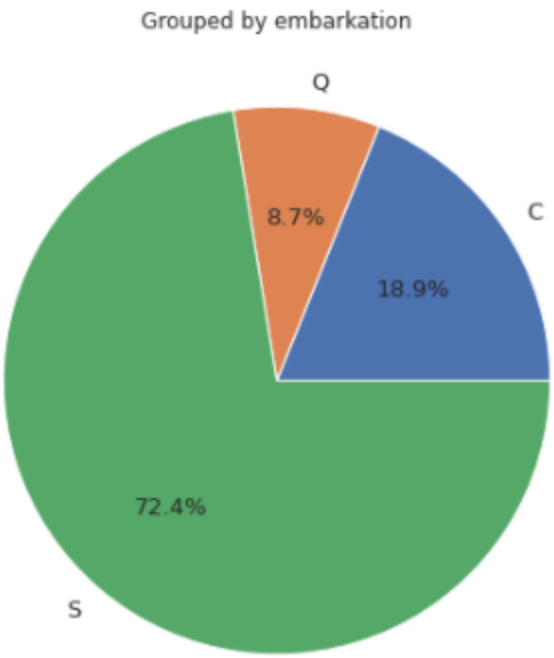
- Passenger class, gender and embarkation distro:

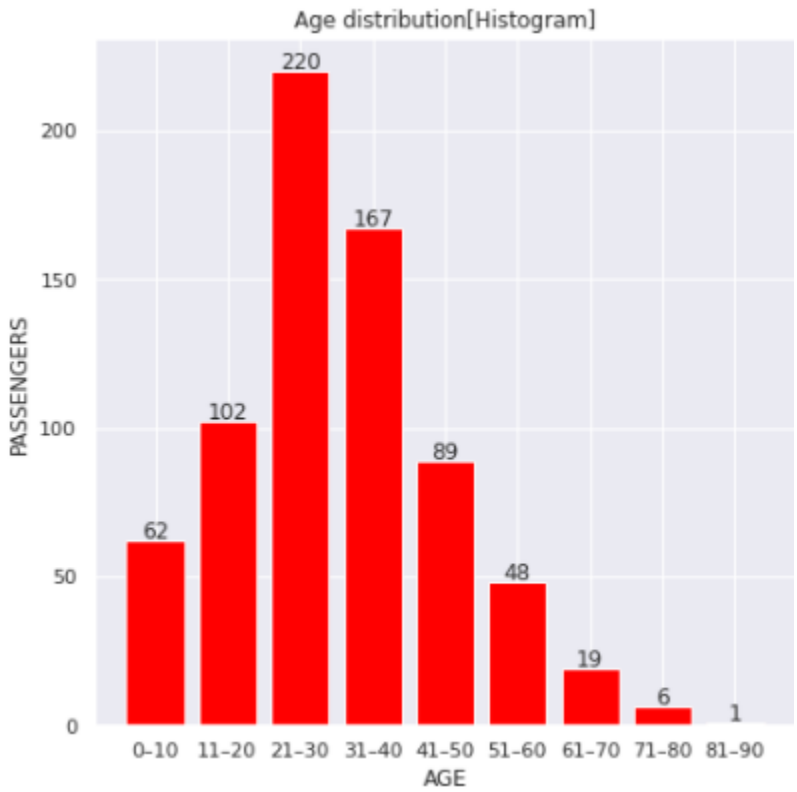
Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

- 

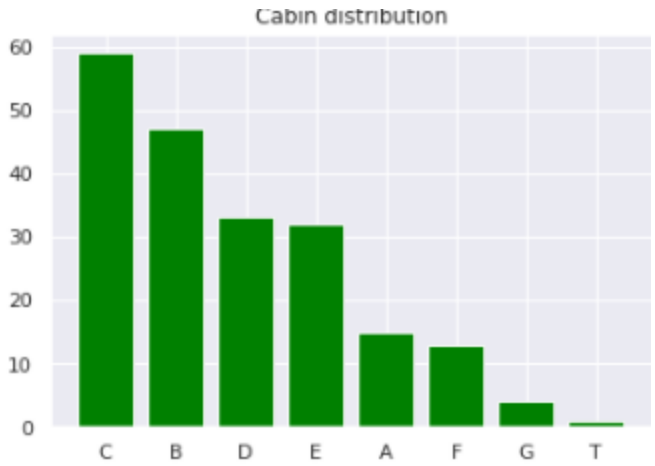
- 

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu





Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

- 

Data Preparation:

- Feature Engineering[SibSp & Parch]: I did some feature engineering stuff from the SibSp and Parch columns. According to the dataset details (which you can access from this link), the two columns represent the number of siblings/spouses and the number of parents/children aboard the Titanic respectively. The main idea here is to create a new column called **FamilySize** in which the value is taken from the two columns I mentioned earlier. This action is taken based on the assumption that larger family sizes may have greater opportunity to survive as they can stay intact with each other better than those who travel alone.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FamilySize
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1

- There were 2 missing values in the ‘Embarked’ column. Since it’s not a significant number, I removed it.Next, if I take the unique values of this column, we will find that there are 3 possible values, namely C, Q and S (which stands for Cherbourg, Queenstown and Southampton). Here I decided to convert this column values into something like **one-hot encoding** since any machine learning algos will never work with non-numerical data.

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Embarked_C	Embarked_Q	Embarked_S
0	0	1
1	0	0
0	0	1

-
- I found that the values of the Cabin column contain plenty of missing values. Thus, I decided to fill that out with 'Unknown'. It can simply be achieved using the fillna() method. Then One hot encoded the 'Cabin'. It creates,

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked', 'FamilySize',
       'Embarked_C', 'Embarked_Q', 'Embarked_S', 'Cabin_A', 'Cabin_B',
       'Cabin_C', 'Cabin_D', 'Cabin_E', 'Cabin_F', 'Cabin_G', 'Cabin_T',
       'Cabin_u'],
      dtype='object')
```

-
- Now dropping all categorical values and from the dataframe and Normalize the data
- Here I select feature and target variable and select 20% as a test data

```
# X = feature; y = target variable
X = titanic_df.iloc[:,1:].values
y = titanic_df['Survived'].values
```

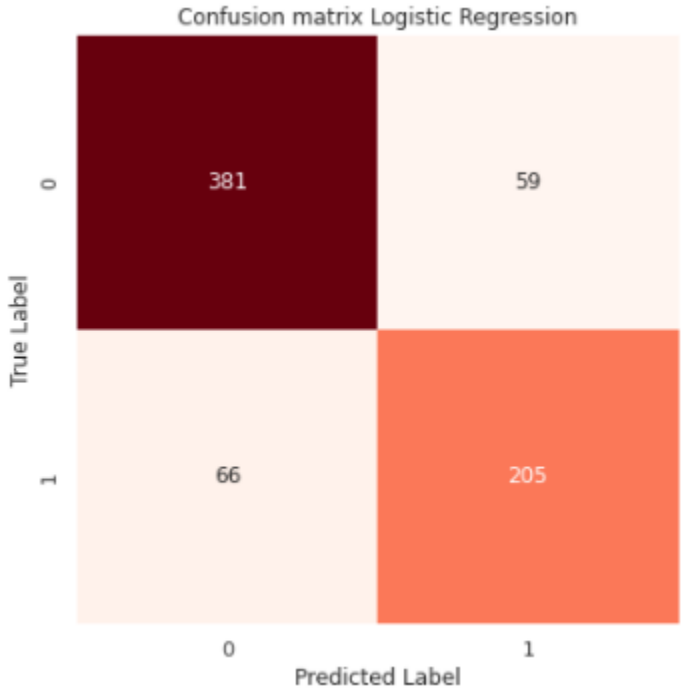
```
# train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 42)
```

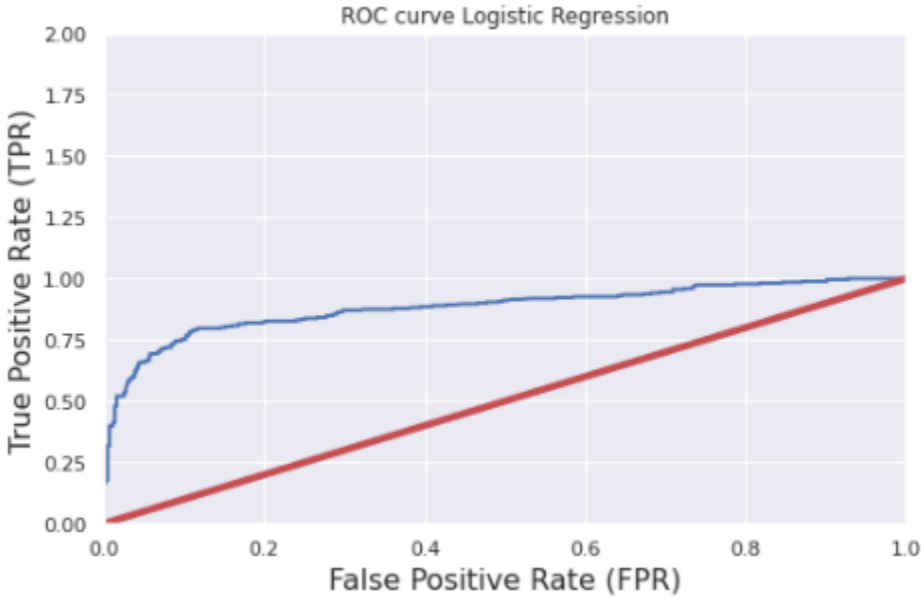
-

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Then I fit the model with Logistic Regression, Gaussian Naive Bayes, Support Vector Machine, Decision Tree & Random Forest as the ques asked.
I used sk-learn libraries to fit different models and evaluate those models with accuracy score, confusion matrix and ROC curve. Here are those.

Logistic Regression:

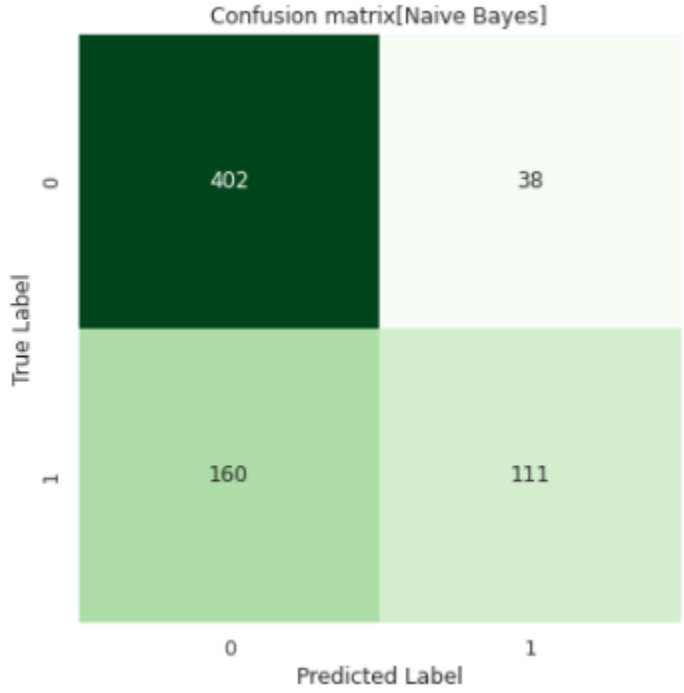
- Accuracy: 84.95%
- 

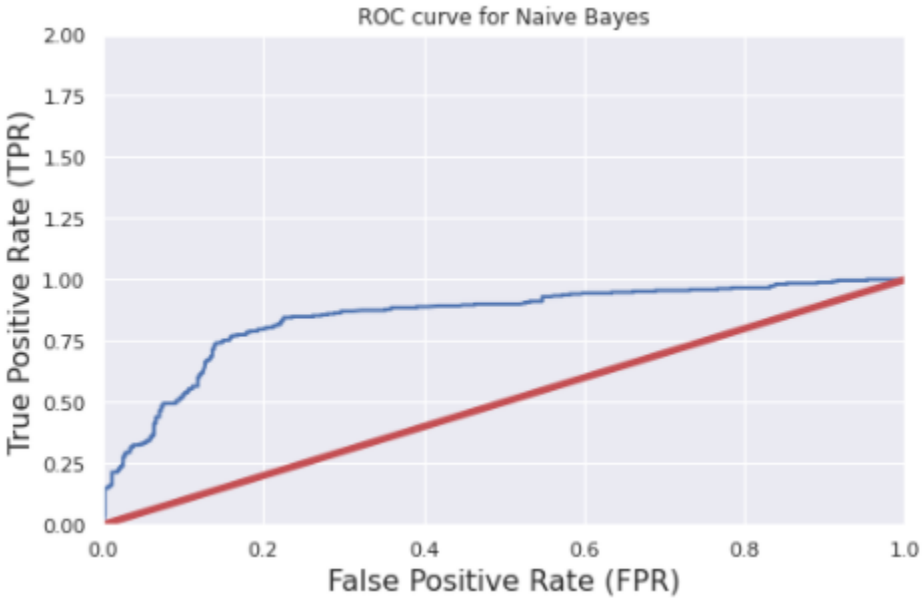
- 

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Gaussian Naive Bayes:

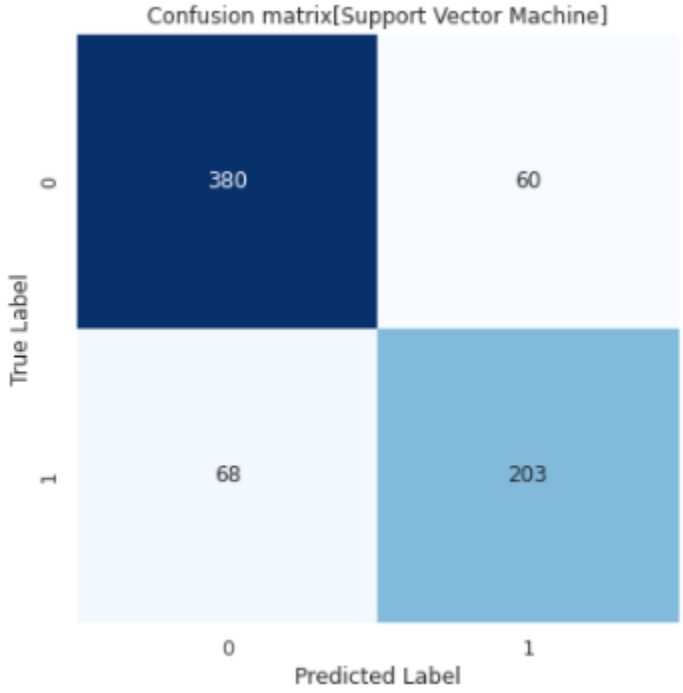
- Accuracy: 69.34%





Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Support Vector Machine

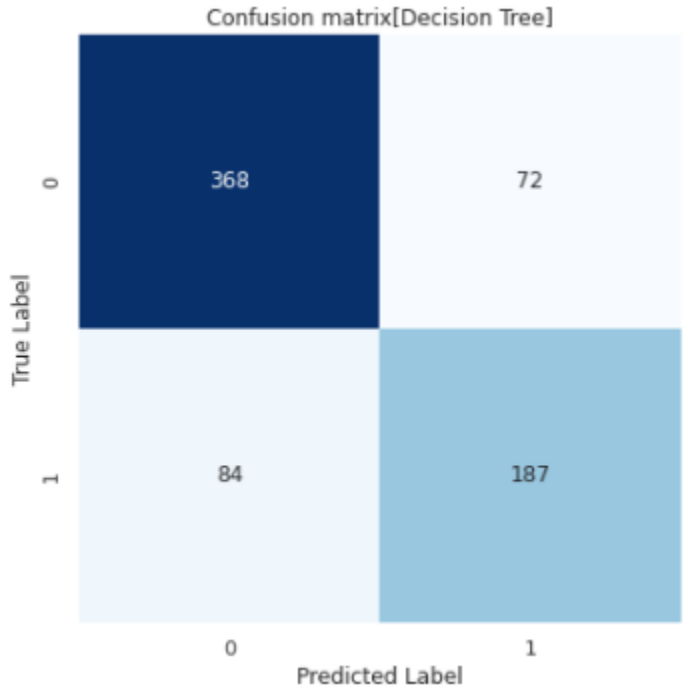
- Accuracy:83.97%
- 

- 

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Decision Tree:

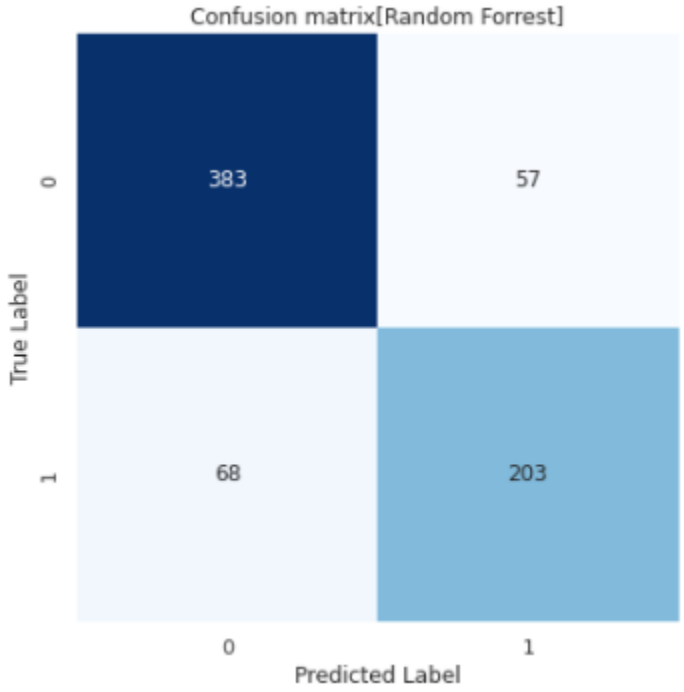
- Accuracy:98.87%



Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu

Random Forest:

- Accuracy: 98.87%



.

Finished

Udyan Saha Upal_1821359
Sec_4, CSE 445.4, Summer21
Udyan.upal@northsouth.edu