ML Model Development for

Heart Failure Prediction

Prepared by: Crishmi Costa

Date of Submission: 21.11.2021

Introduction

- Cardiovascular diseases (CVDs) are the leading cause of death globally, taking an estimated 17.9 million lives each year.
- The presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established diseases make individuals more prone to CVDs.
- Hence, the machine learning model developed in this project will be of great advantage to people with cardiovascular diseases or people at a high cardiovascular risk as this can be used to provide early detection and management of the disease.

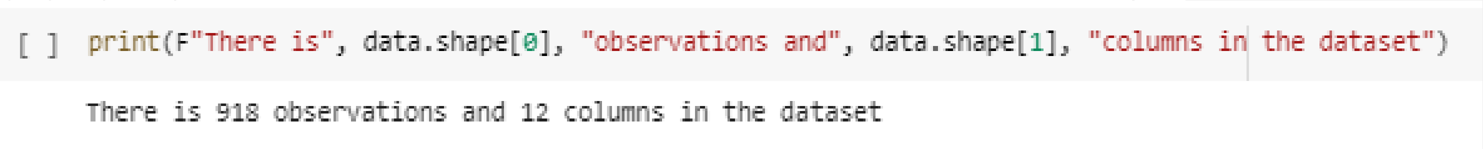
The objective of this project is to build a variety of Classification models and compare the models giving the best prediction on Heart Disease.

Thereby, the best model will be used to make a prediction on the target variable, 'Heart Disease'.

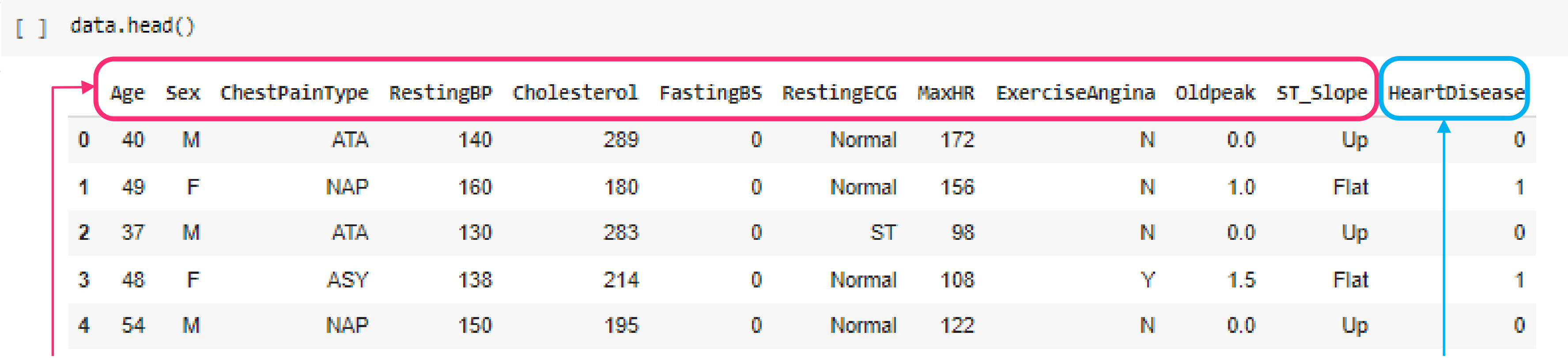
https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

Data

- The data set consists of 12 different columns where 11 of them are considered as the input parameters which are in turn used to predict a single output parameter, which is the target variable.



```
[ ]  print(F"There is", data.shape[0], "observations and", data.shape[1], "columns in the dataset")

     There is 918 observations and 12 columns in the dataset
```



```
[ ]  data.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

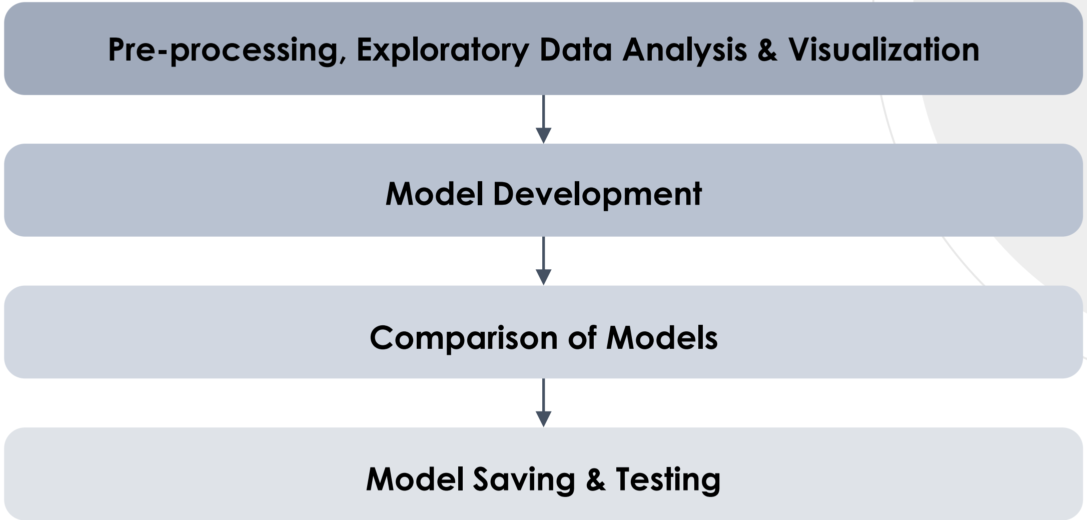
Independent Variables/Input Parameters

Dependent Variable/Output Parameter (Target Variable)

https://www.kaggle.com/andrewmvd/heart-failure-clinical-data

Methodology

- In this study, we analyzed a random dataset of 918 individuals where 508 of them were reportedly positive cases and 410 of them were healthy.
- During the analysis, we applied several machine learning classifiers including **Logistic Regression**, **Decision Tree**, **Random Forest** and **KNN** and examined various aspects to determine and identify the model that best predicts the target variable, ‘Heart Disease’.
- In order to achieve this, a several steps were followed.



Pre-processing, Exploratory Data Analysis & Visualization

Model Development

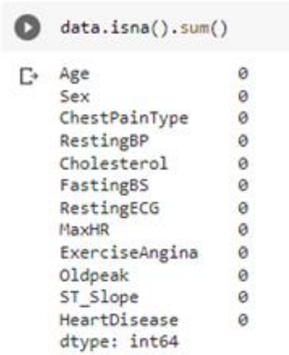
Comparison of Models

Model Saving & Testing

Methodology

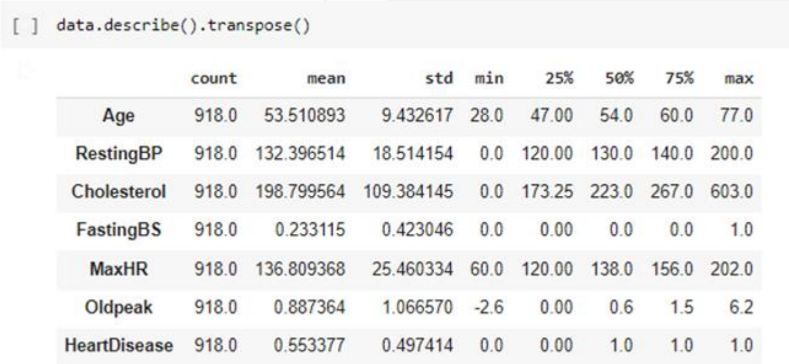
Pre-processing, Exploratory Data Analysis & Visualization

- We have taken our data set through a several pre-processing stages to cleanse the data before training and modeling stages, so as to avoid any errors in the final results.
- For example,
  - **Null & missing value** identification
  - **Descriptive statistics** to check & identify the distribution of the data set.
  - The data set was also split into **numerical and categorical** variables and were further analyzed based upon that.
  - Different methods of **data visualization** to analyze various features
  - **Correlation** between target variable and independent variables



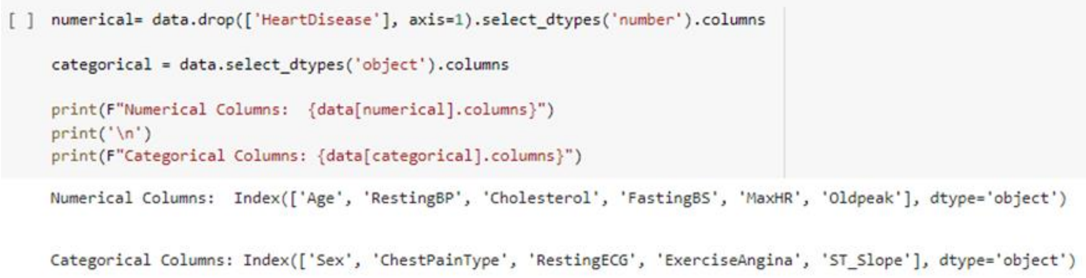
```
data.isna().sum()
```

```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```



```
[ ]  data.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Age	918.0	53.510893	9.432617	28.0	47.00	54.0	60.0	77.0
RestingBP	918.0	132.396514	18.514154	0.0	120.00	130.0	140.0	200.0
Cholesterol	918.0	198.799564	109.384145	0.0	173.25	223.0	267.0	603.0
FastingBS	918.0	0.233115	0.423046	0.0	0.00	0.0	0.0	1.0
MaxHR	918.0	136.809368	25.460334	60.0	120.00	138.0	156.0	202.0
Oldpeak	918.0	0.887364	1.066570	-2.6	0.00	0.6	1.5	6.2
HeartDisease	918.0	0.553377	0.497414	0.0	0.00	1.0	1.0	1.0

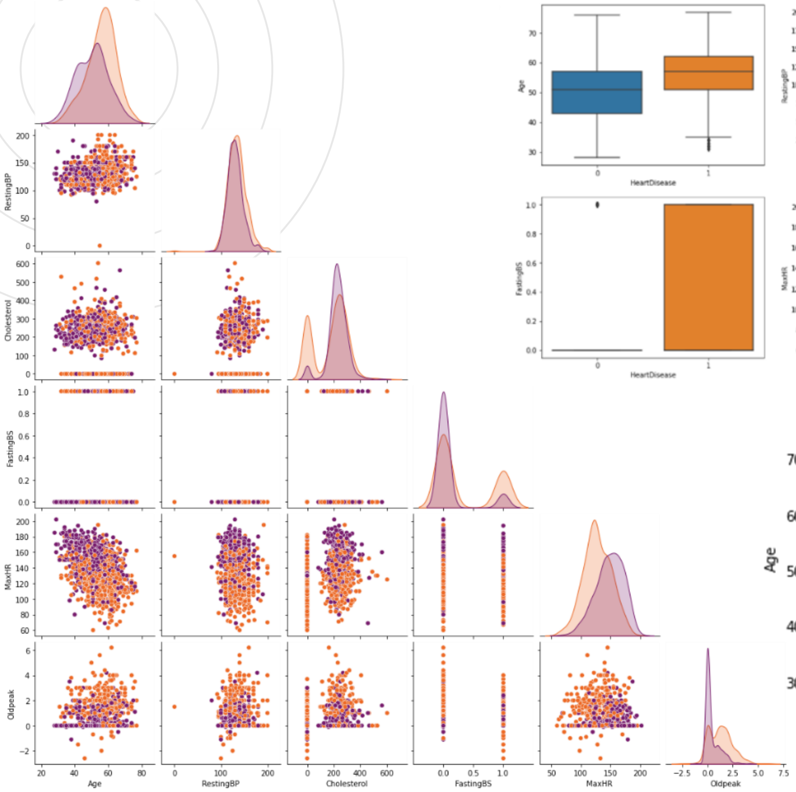


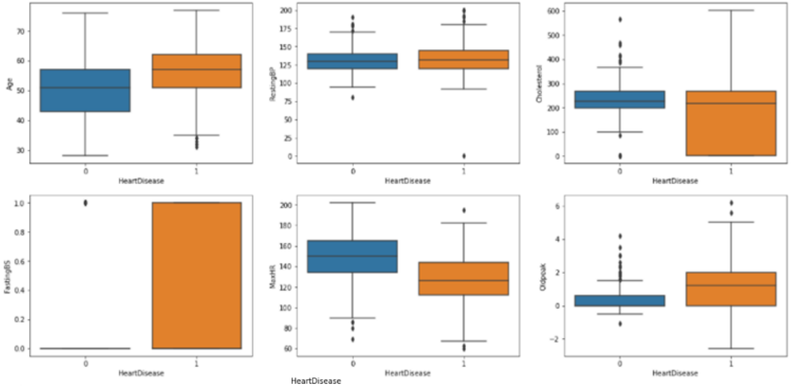
```
[ ]  numerical= data.drop(['HeartDisease'], axis=1).select_dtypes('number').columns

     categorical = data.select_dtypes('object').columns

     print(F"Numerical Columns:  {data[numerical].columns}")
     print('\n')
     print(F"Categorical Columns: {data[categorical].columns}")
```

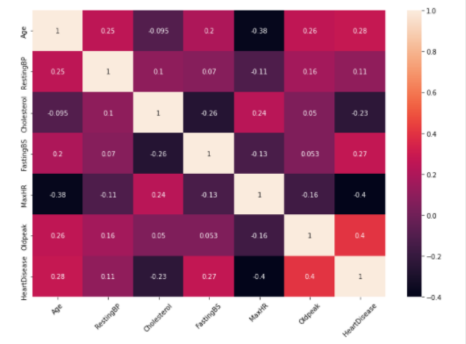
```
Numerical Columns:  Index(['Age', 'RestingBP', 'Cholesterol', 'FastingBS', 'MaxHR', 'Oldpeak'], dtype='object')


Categorical Columns: Index(['Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', 'ST_Slope'], dtype='object')
```

Methodology

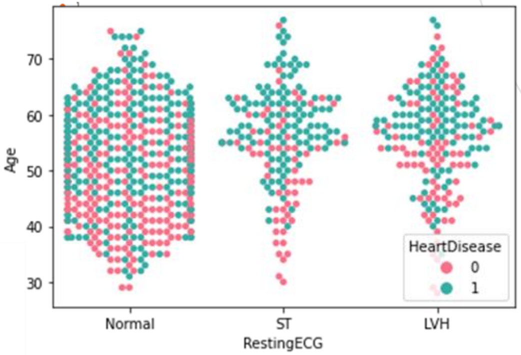
Pre-processing, Exploratory Data Analysis & Visualization

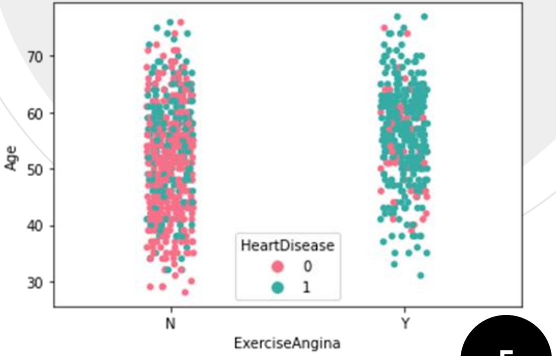






	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1	0.25	-0.095	0.2	-0.38	0.26	0.28
RestingBP	0.25	1	0.1	0.07	-0.11	0.16	0.11
Cholesterol	-0.095	0.1	1	-0.26	0.24	0.05	-0.23
FastingBS	0.2	0.07	-0.26	1	-0.13	0.053	0.27
MaxHR	-0.38	-0.11	0.24	-0.13	1	-0.16	-0.4
Oldpeak	0.26	0.16	0.05	0.053	-0.16	1	0.4
HeartDisease	0.28	0.11	-0.23	0.27	-0.4	0.4	1

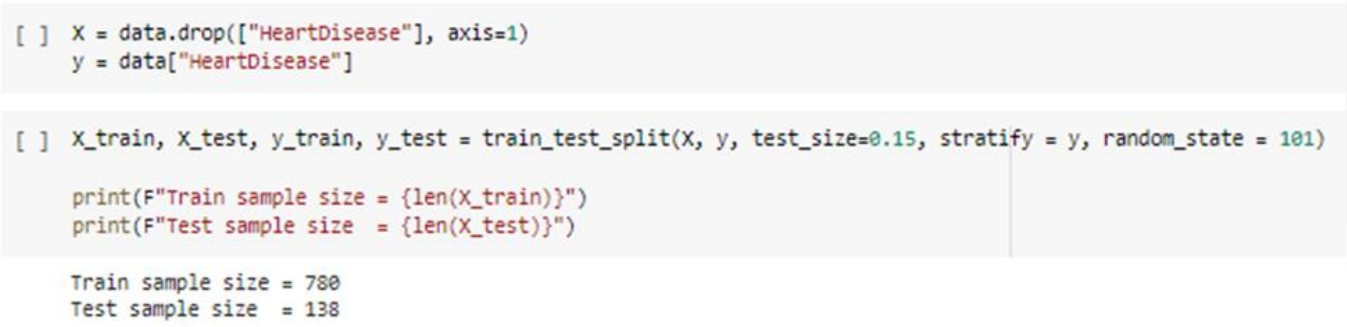




Methodology

Model Development

Types of Models used	Logistic Regression, Decision Tree, Random Forest, KNN
Independent variables	Age, Sex, Chest Pain Type, Resting BP, Cholesterol, Fasting BS, Resting ECG, Max HR, Exercise Angina, Old peak, ST_Slope
Dependent variables	Heart Disease



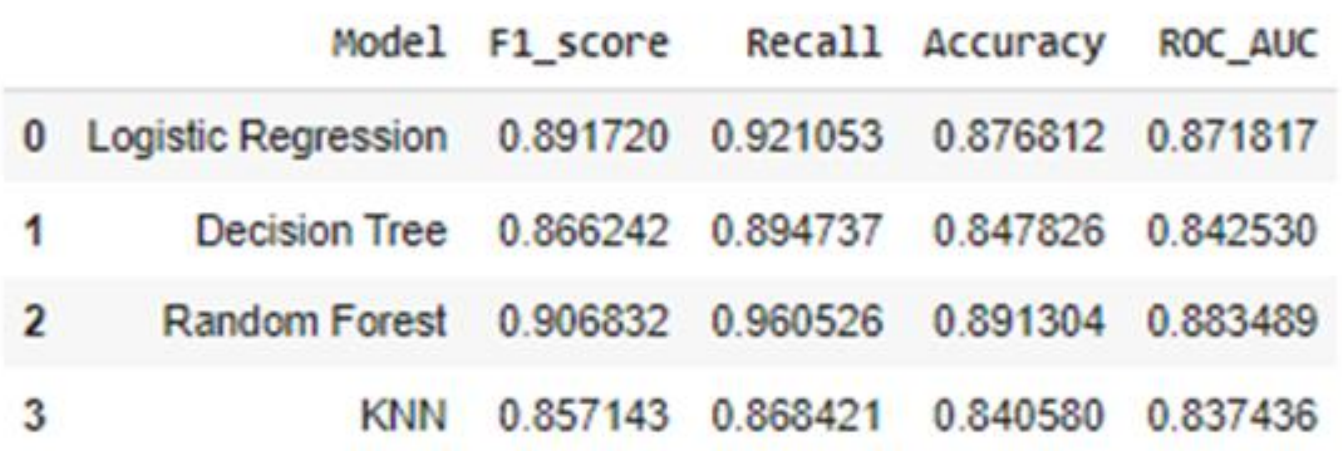
```
[ ] X = data.drop(["HeartDisease"], axis=1)
    y = data["HeartDisease"]
```

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, stratify = y, random_state = 101)

    print(F"Train sample size = {len(X_train)}")
    print(F"Test sample size  = {len(X_test)}")

    Train sample size = 780
    Test sample size  = 138
```

Methodology

Comparison of Models

- For the four model types that were used to analyze the data, the **F1 score**, **accuracy**, **recall** and the **ROC** values were used to evaluate and identify the model that provides the best prediction of the target variable, ‘Heart Disease’.

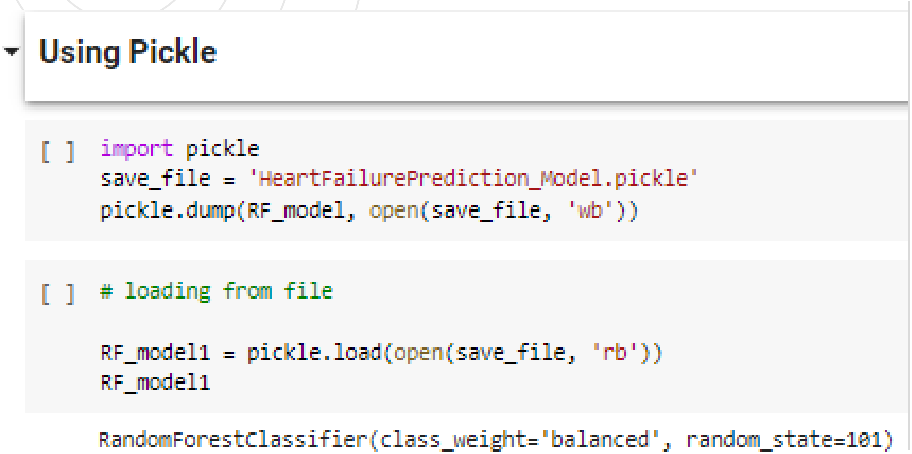


	Model	F1_score	Recall	Accuracy	ROC_AUC
0	Logistic Regression	0.891720	0.921053	0.876812	0.871817
1	Decision Tree	0.866242	0.894737	0.847826	0.842530
2	Random Forest	0.906832	0.960526	0.891304	0.883489
3	KNN	0.857143	0.868421	0.840580	0.837436

Methodology

Model Testing & Saving

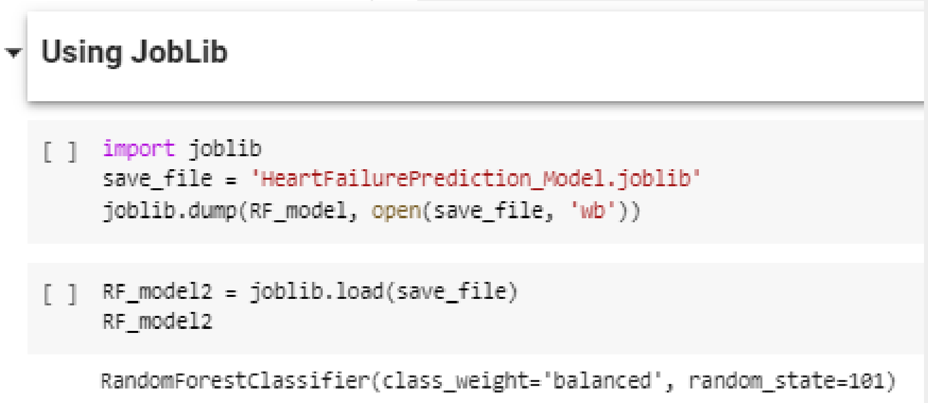
- Once the model was finalized, two methods of model saving using ‘Pickle’ and ‘Joblib’ were carried out.



Using Pickle

```
[ ] import pickle
    save_file = 'HeartFailurePrediction_Model.pickle'
    pickle.dump(RF_model, open(save_file, 'wb'))
```

```
[ ] # loading from file

    RF_model1 = pickle.load(open(save_file, 'rb'))
    RF_model1

    RandomForestClassifier(class_weight='balanced', random_state=101)
```



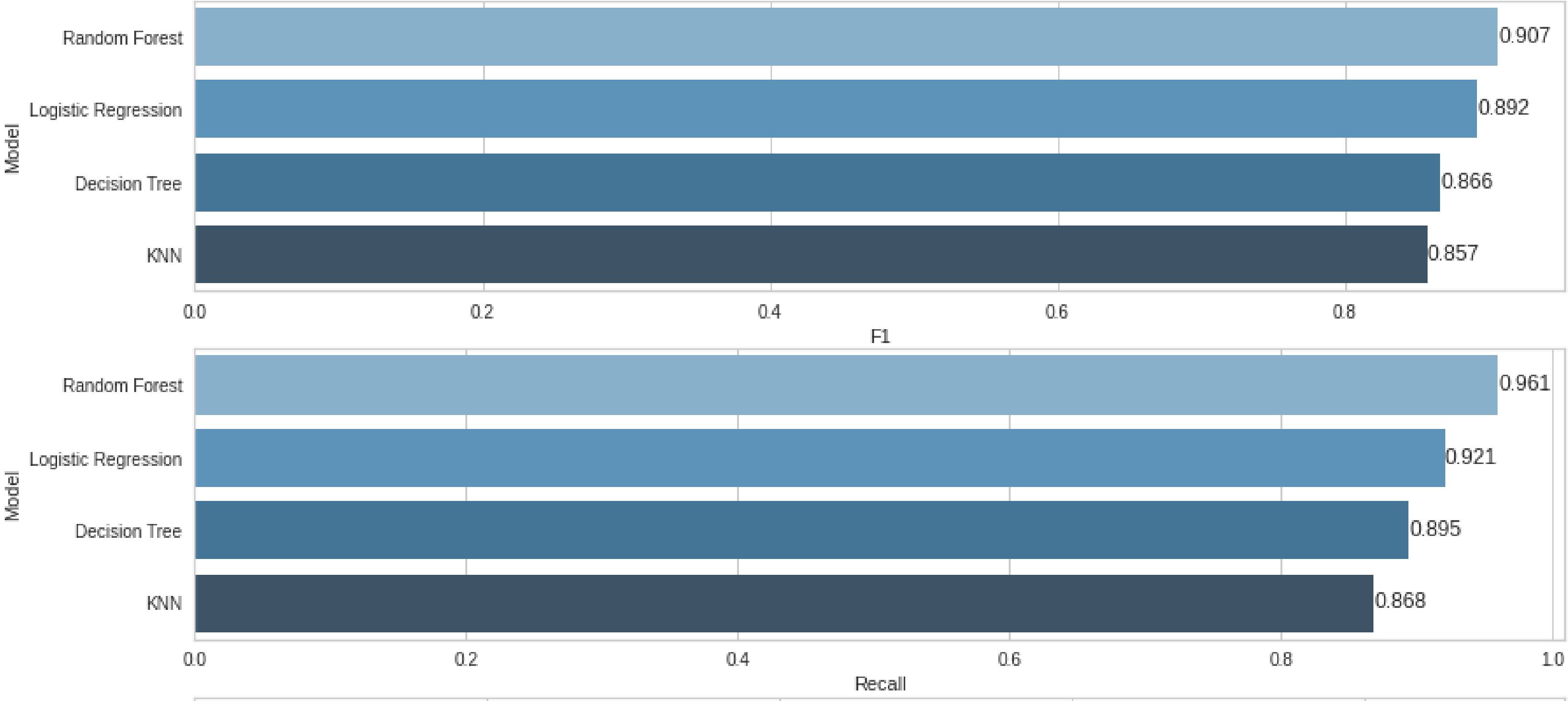
Using JobLib

```
[ ] import joblib
    save_file = 'HeartFailurePrediction_Model.joblib'
    joblib.dump(RF_model, open(save_file, 'wb'))
```

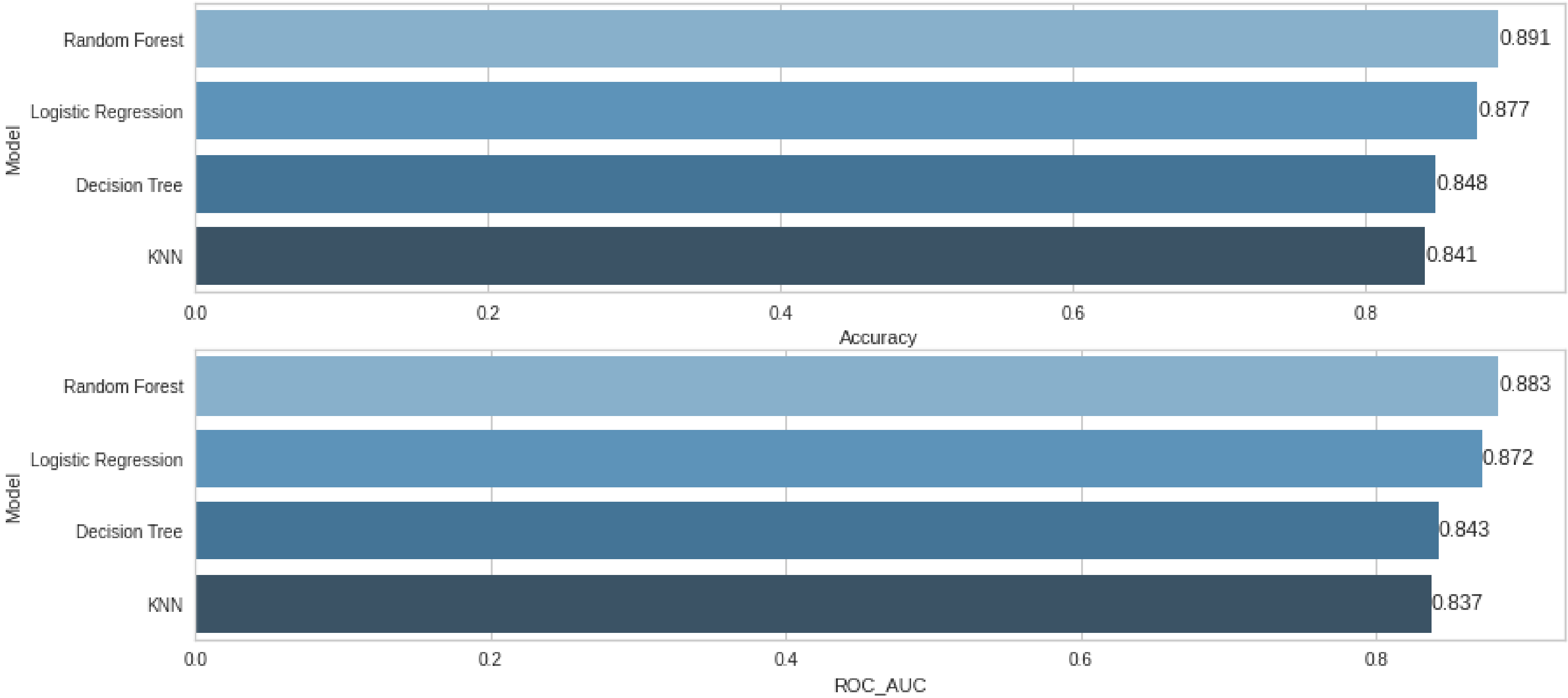
```
[ ] RF_model2 = joblib.load(save_file)
    RF_model2

    RandomForestClassifier(class_weight='balanced', random_state=101)
```

Results

- The following observations were made with regards to the F1 score, accuracy, recall and ROC values of the four model types that were used.

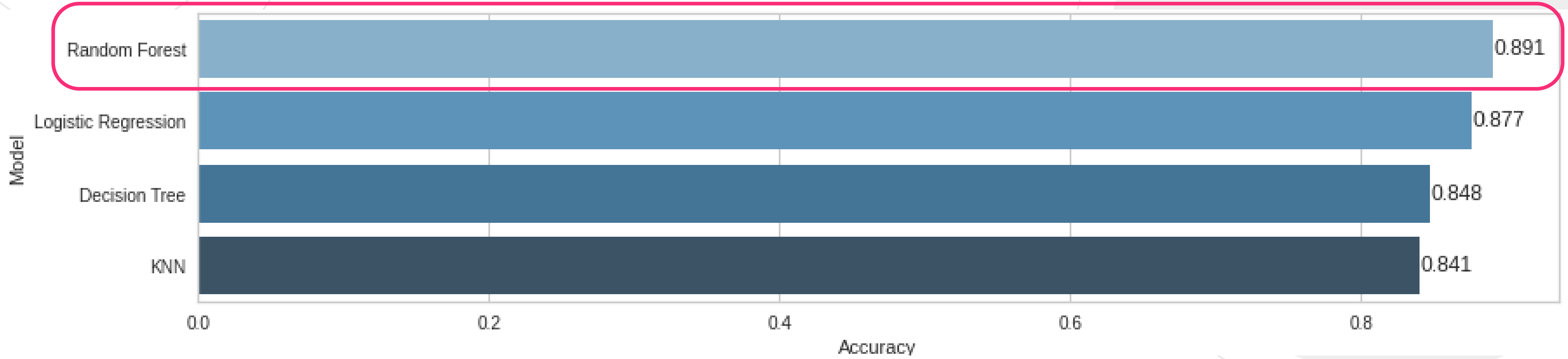


Results


Model
Random Forest
Logistic Regression
Decision Tree
KNN
0.891
0.877
0.848
0.841
0.0
0.2
0.4
0.6
0.8
Accuracy
Model
Random Forest
Logistic Regression
Decision Tree
KNN
0.883
0.872
0.843
0.837
0.0
0.2
0.4
0.6
0.8
ROC_AUC

Conclusion

- As observed, it can be concluded that the best model to predict the risk of 'Heart Disease' of an individual would be the Random Forest Classifier.
- The model shows an approximate accuracy of about **89%.**



Thank You!