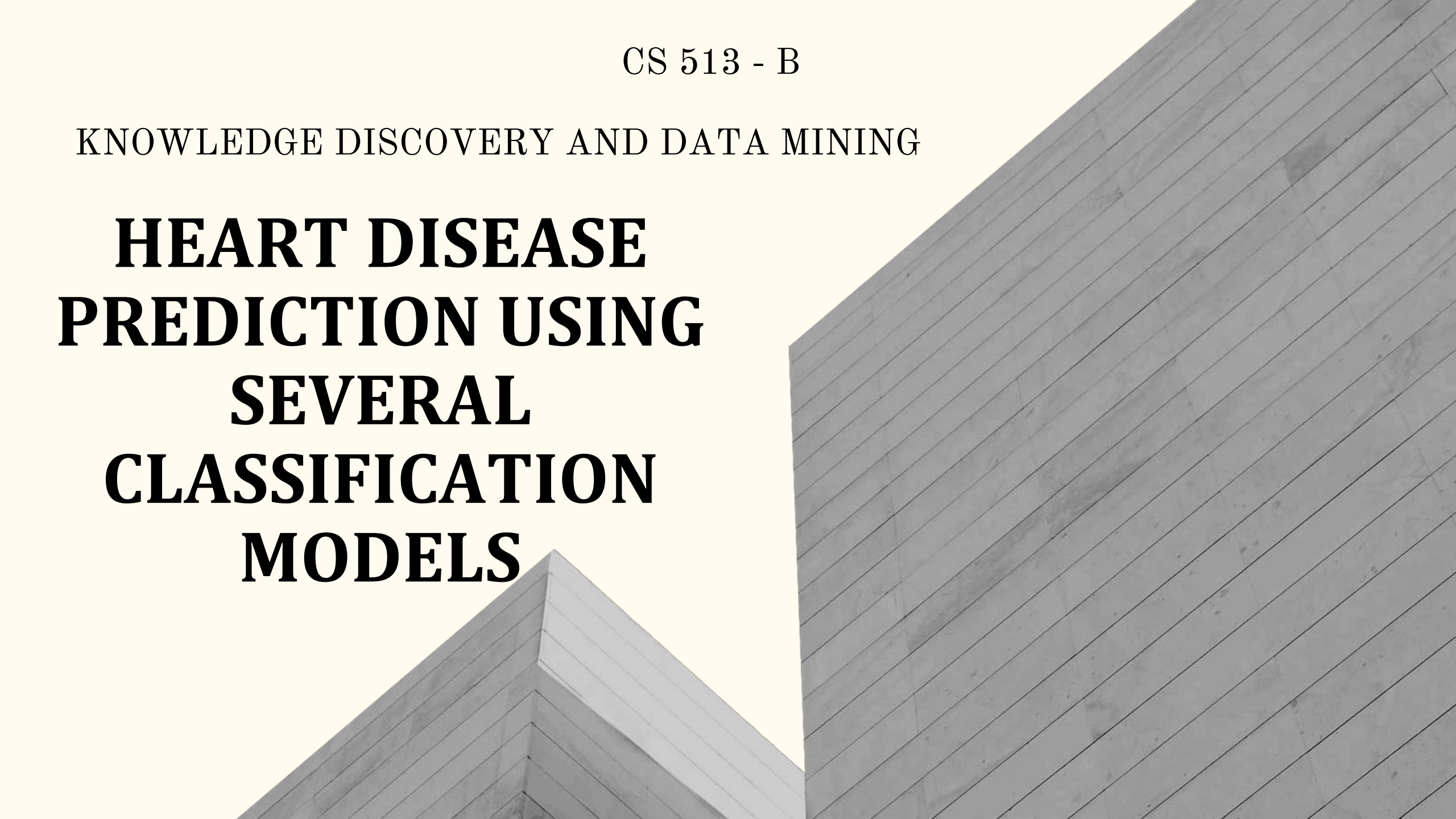
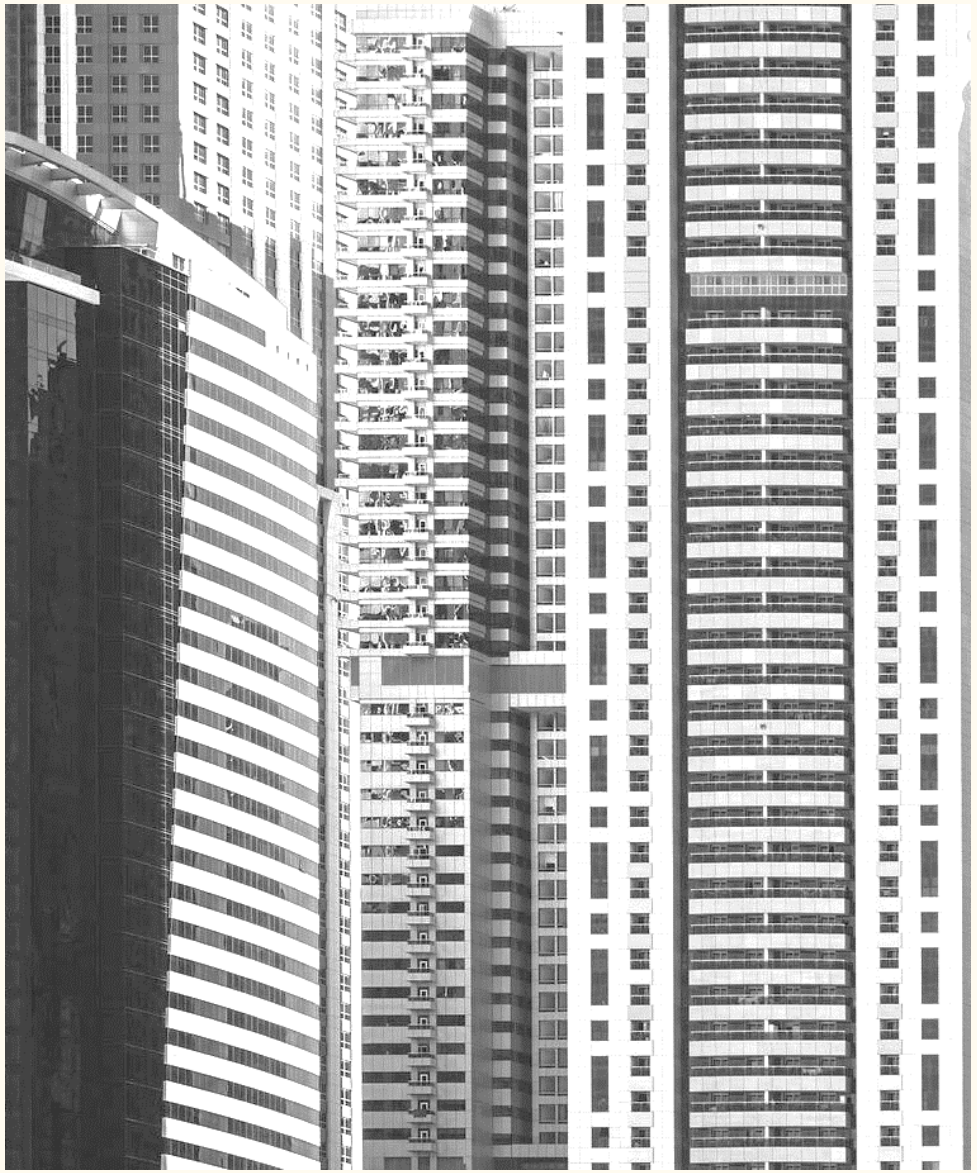
CS 513 - B

KNOWLEDGE DISCOVERY AND DATA MINING

HEART DISEASE PREDICTION USING SEVERAL CLASSIFICATION MODELS





GROUP 4

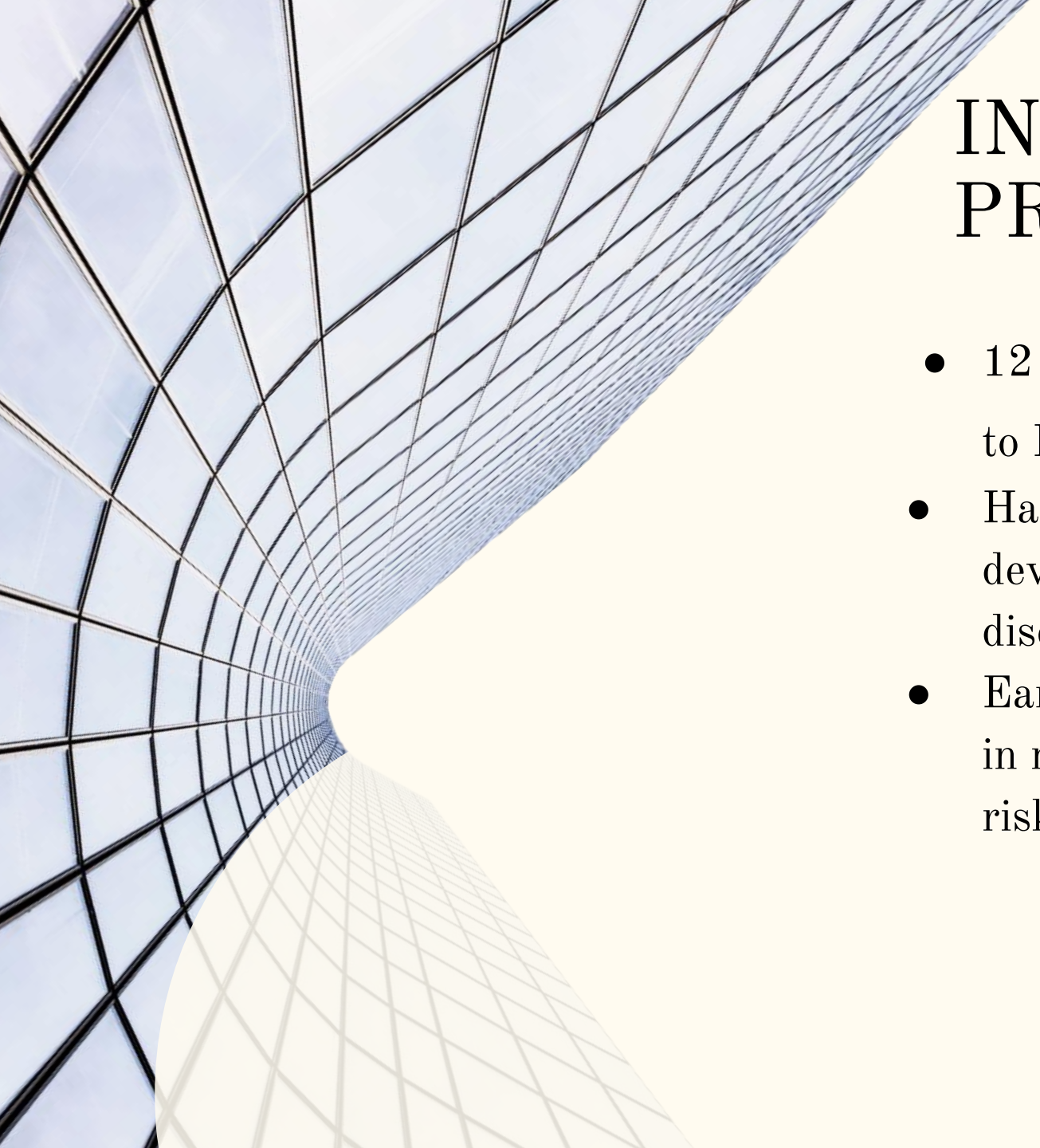
TEAM MEMBERS

ROUSHAN KUMAR | CWID: 20009314

ISHAN ARYENDU | CWID : 10474734

SABAH AHMED | CWID : 10478272

ZHU LI | CWID : 10454296



INTRODUCTION AND PROBLEM OVERVIEW

- 12 million deaths occur worldwide, every year due to Heart diseases.
- Half of the deaths in the United States and other developed countries are due to cardiovascular diseases.
- Early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce complications.

GOAL & OBJECTIVE

- Our classification goal is to determine whether the patient has a 10-year risk of developing/future coronary heart disease (CHD) using several Machine Learning/Data Mining methods from an ongoing cardiovascular study on people of Framingham, Massachusetts.
- Build a predictive model that predicts the patient has a 10-year risk of future coronary heart disease (CHD) or not based on their potential risk factor/ patients' information and compare the ML models
- Predict and Classify what predictors/features might be of importance for the risk of CHD.

ABOUT THE DATASET

Dataset statistics

Number of variables	16
Number of observations	4238
Missing cells	645
Missing cells (%)	1.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	529.9 KiB
Average record size in memory	128.0 B

Variable types

Categorical	8
Numeric	8

Data Source References: https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

DATA FIELDS

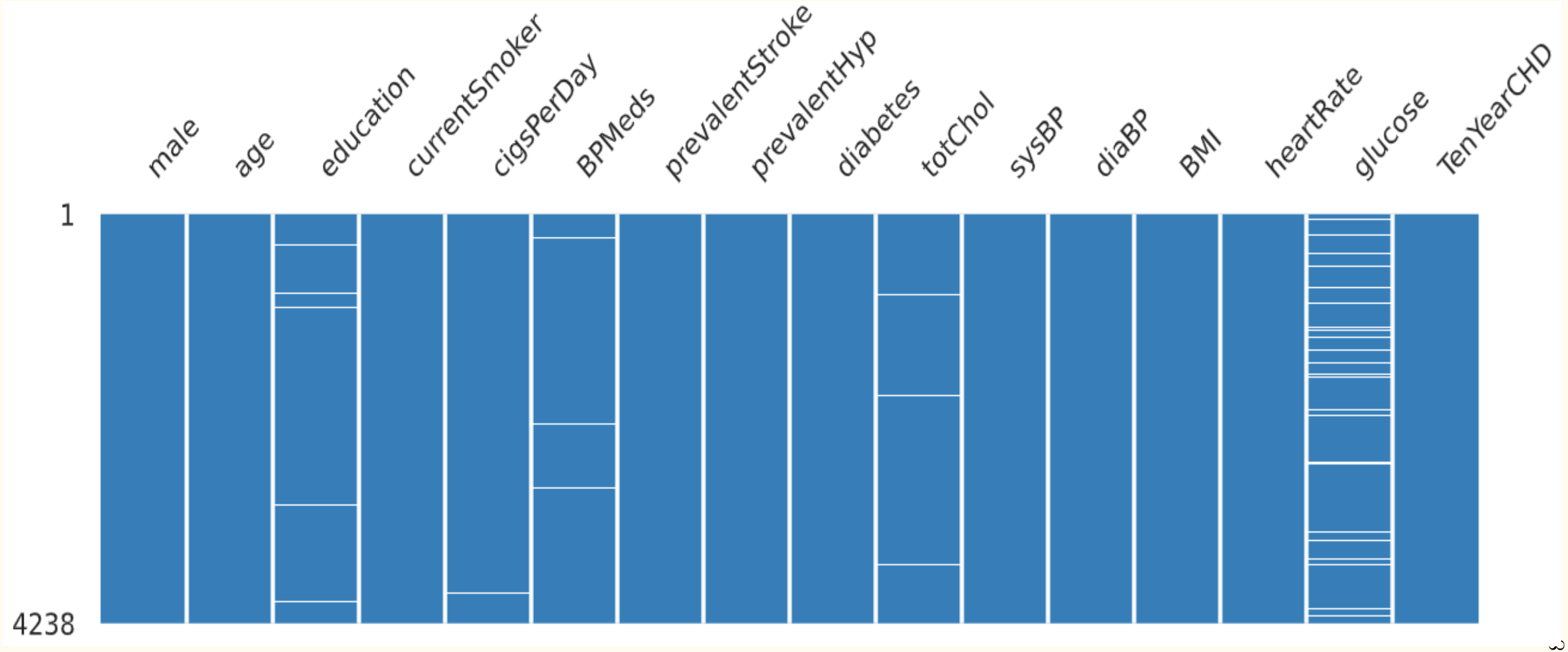
- The dataset provides the patients' information
- Each attribute is a potential risk factor. There are both demographic, behavioral and medical risk factors.

Demographic	**Behavioral**	**Medical(history)**	**Medical(current)**
Male: Whether the patient is male or female	**Current Smoker:** whether or not the patient is a current smoker	**BP Meds:** whether or not the patient was on blood pressure medication	**Tot Chol:** total cholesterol level
Age: Age of the patient	**Cigs Per Day:** the number of cigarettes that the person smoked on average in one day	**Prevalent Stroke:** whether or not the patient had previously had a stroke	**Sys BP:** systolic blood pressure
Education: **1-** Primary education **2-** Secondary education **3-** Postsecondary education **4-** Graduate and above		**Prevalent Hyp:** whether or not the patient was hypertensive	**Dia BP:** diastolic blood pressure
		Diabetes: whether or not the patient had diabetes	**BMI:** Body Mass Index
			Heart Rate: heart rate
			Glucose: glucose level
Predict variable (desired target) : 10 year risk of coronary heart disease CHD (binary: "1", means "Yes", "0" means "No")			

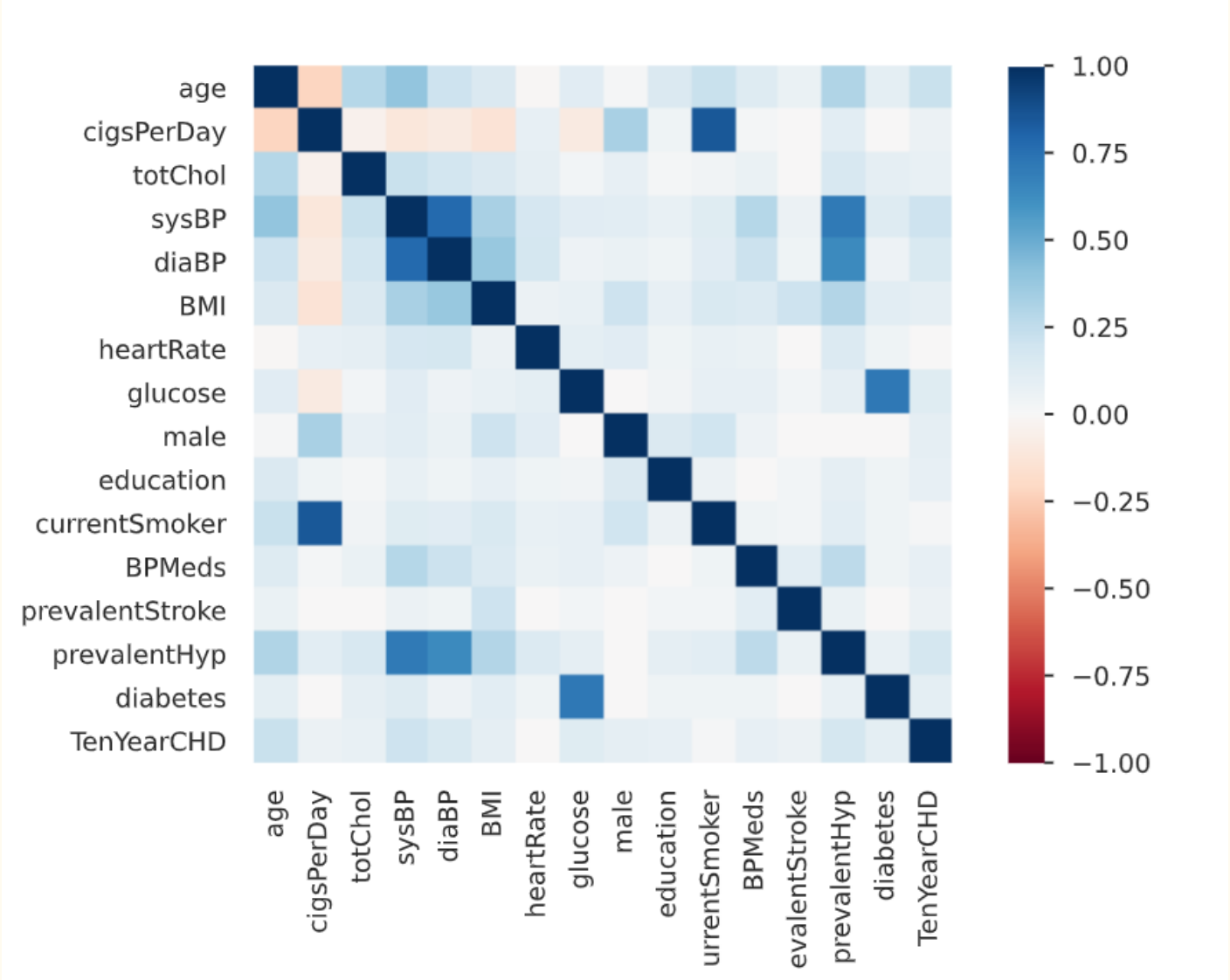
ORIGINAL DATASET

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
2	1	39	4	0	0	0	0	0	0	195	106	70	26.97	80	77	0
3	0	46	2	0	0	0	0	0	0	250	121	81	28.73	95	76	0
4	1	48	1	1	20	0	0	0	0	245	127.5	80	25.34	75	70	0
5	0	61	3	1	30	0	0	1	0	225	150	95	28.58	65	103	1
6	0	46	3	1	23	0	0	0	0	285	130	84	23.1	85	85	0
7	0	43	2	0	0	0	0	1	0	228	180	110	30.3	77	99	0
8	0	63	1	0	0	0	0	0	0	205	138	71	33.11	60	85	1
9	0	45	2	1	20	0	0	0	0	313	100	71	21.68	79	78	0
10	1	52	1	0	0	0	0	1	0	260	141.5	89	26.36	76	79	0
11	1	43	1	1	30	0	0	1	0	225	162	107	23.61	93	88	0
12	0	50	1	0	0	0	0	0	0	254	133	76	22.91	75	76	0
13	0	43	2	0	0	0	0	0	0	247	131	88	27.64	72	61	0
14	1	46	1	1	15	0	0	1	0	294	142	94	26.31	98	64	0
15	0	41	3	0	0	1	0	1	0	332	124	88	31.31	65	84	0
16	0	39	2	1	9	0	0	0	0	226	114	64	22.35	85	NA	0
17	0	38	2	1	20	0	0	1	0	221	140	90	21.35	95	70	1
18	1	48	3	1	10	0	0	1	0	232	138	90	22.37	64	72	0

MISSING VALUES



CORRELATIONS HEATMAP



ALERTS ON FEATURES

`cigsPerDay` is highly overall correlated with `currentSmoker`	High correlation
`sysBP` is highly overall correlated with `prevalentHyp` and 1 other fields	High correlation
`diaBP` is highly overall correlated with `prevalentHyp` and 1 other fields	High correlation
`glucose` is highly overall correlated with `diabetes`	High correlation
`currentSmoker` is highly overall correlated with `cigsPerDay`	High correlation
`prevalentHyp` is highly overall correlated with `sysBP` and 1 other fields	High correlation
`diabetes` is highly overall correlated with `glucose`	High correlation
`education` has 105 (2.5%) missing values	Missing
`BPMeds` has 53 (1.3%) missing values	Missing
`totChol` has 50 (1.2%) missing values	Missing
`glucose` has 388 (9.2%) missing values	Missing
`cigsPerDay` has 2144 (50.6%) zeros	Zeros

PREPROCESSING

- The feature prevalentHyp was removed from the training and testing datasets because it had a high correlation with the sysBP and diaBP feature
- Because currentSmoker had a strong correlation with the cigsPerDay feature, they were taken out of the training and testing datasets.
- The missing values were replaced with the mode of the feature set and min-max scaler was used to normalize the rest of the features in the dataset.
- The dataset contained bias. So, to balance the training dataset we used SMOTETomek.

MODEL BUILDING

- We used the features [**'male', 'age', 'education', 'cigsPerDay', 'BPMeds', 'prevalentStroke', 'diabetes', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'**] for training the model.
- Our target feature is **'TenYearCHD'**.
- The training and testing datasets were divided **80:20** from the normalized and imputed data.
- We used **k-fold cross validation** to validate the model against the validation dataset.

CLASSIFICATION ALGORITHMS

- Logistic Regression
- Naive Bayes
- Random Forest
- Extreme Gradient Boost

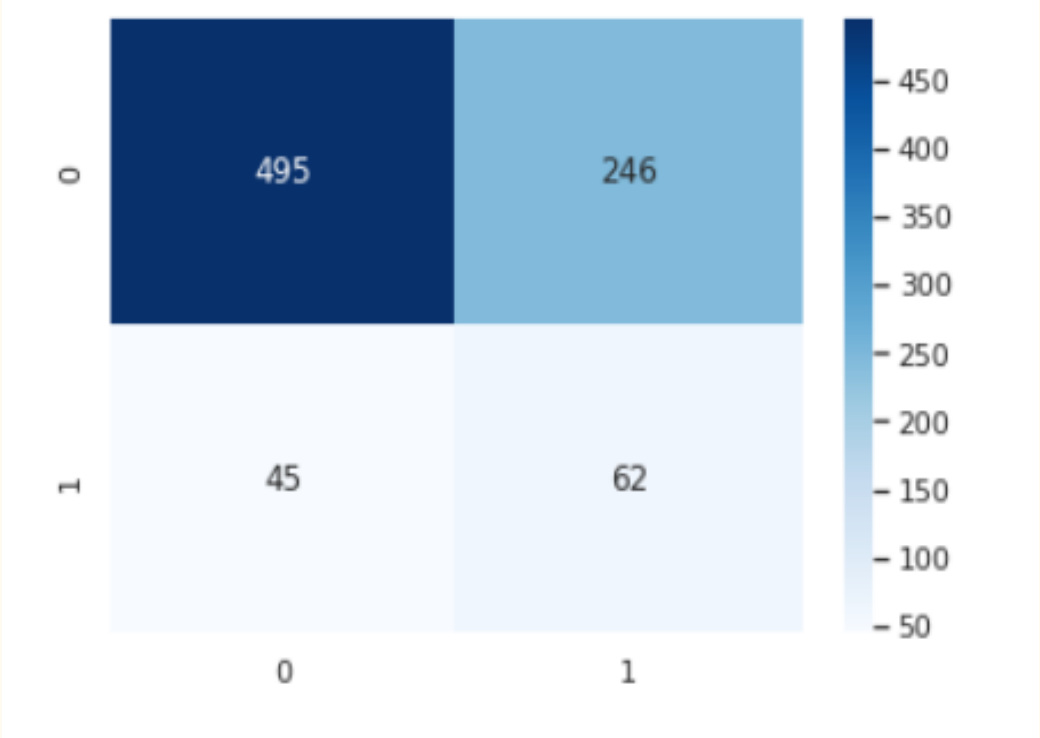
- KNeighbors
- Decision Tree
- Support Vector Classifier

LOGISTIC REGRESSION

```
Accuracy of Logistic Regression: 65.68396226415094

              precision    recall  f1-score   support

         0.0       0.92      0.67      0.77       741
         1.0       0.20      0.58      0.30       107

    accuracy                           0.66       848
   macro avg       0.56      0.62      0.54       848
weighted avg       0.83      0.66      0.71       848
```

```
[ ] score1=cross_val_score(lr,X_valid, y_valid,cv=10)
    print(f"After k-fold cross validation score is {score1.mean()}")
```

```
After k-fold cross validation score is 0.8738375350140057
```



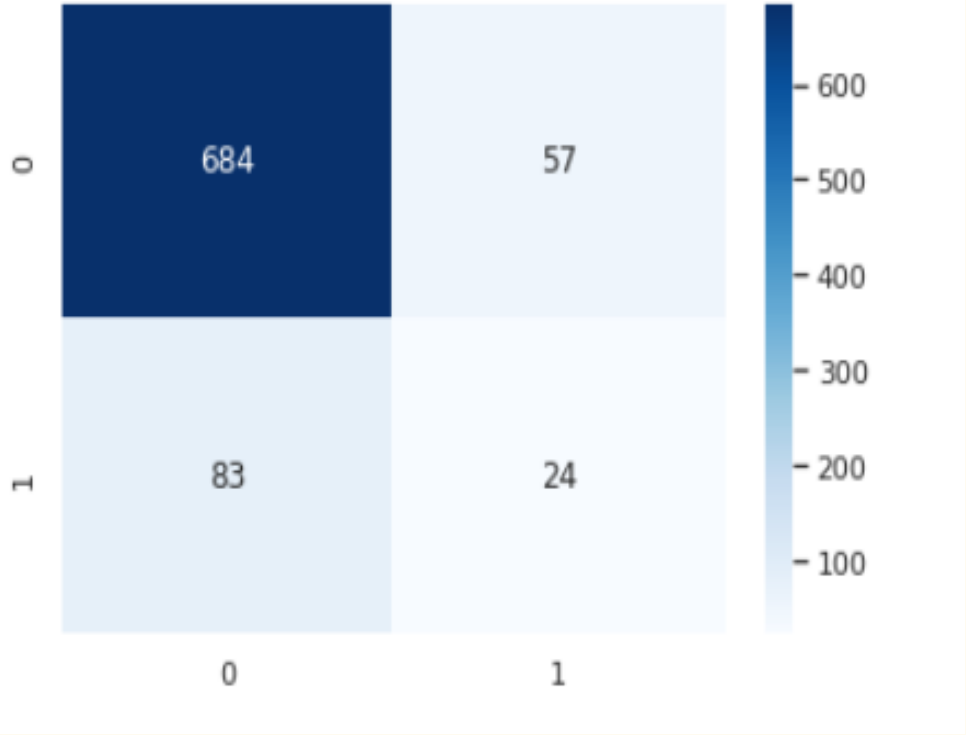
Confusion Matrix

NAIVE BAYES

```
Accuracy of Naive Bayes model: 83.49056603773585

              precision    recall  f1-score   support

         0.0       0.89      0.92      0.91       741
         1.0       0.30      0.22      0.26       107

    accuracy                           0.83       848
   macro avg       0.59      0.57      0.58       848
weighted avg       0.82      0.83      0.82       848
```

```
[ ] score2=cross_val_score(nb,X_valid,y_valid,cv=10)
    print(f"After k-fold cross validation score is {score2.mean()}")
```

```
After k-fold cross validation score is 0.7643697478991596
```



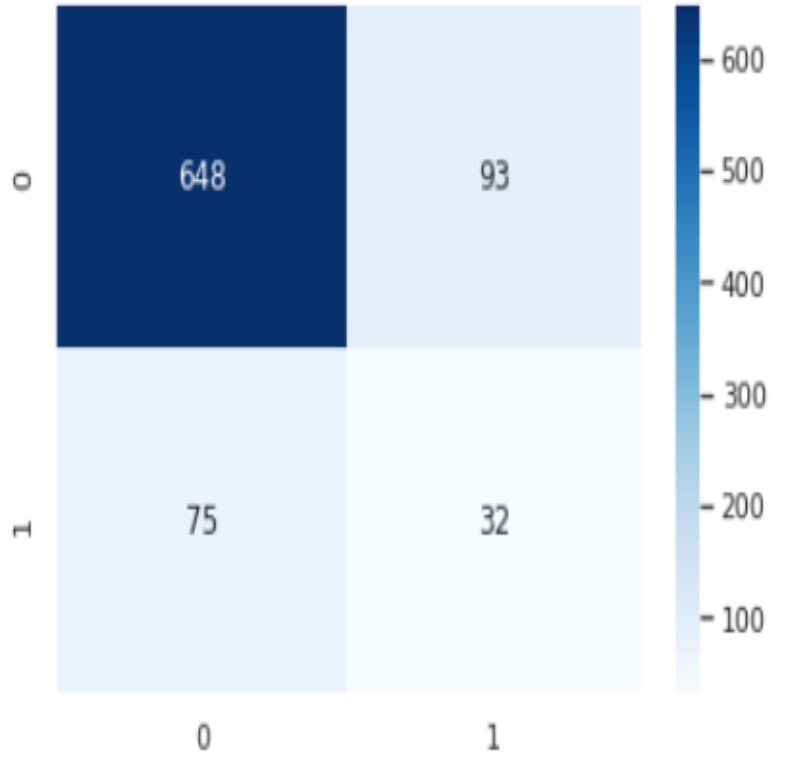
Confusion Matrix

RANDOM FOREST

```
Accuracy of Random Forest: 80.18867924528303

              precision    recall  f1-score   support

         0.0       0.90      0.87      0.89       741
         1.0       0.26      0.30      0.28       107

    accuracy                           0.80       848
   macro avg       0.58      0.59      0.58       848
weighted avg       0.82      0.80      0.81       848
```

```
[ ] score3=cross_val_score(rf,X_valid,y_valid,cv=10)
    print(f"After k-fold cross validation score is {score3.mean()}")
```

```
After k-fold cross validation score is 0.8679411764705882
```



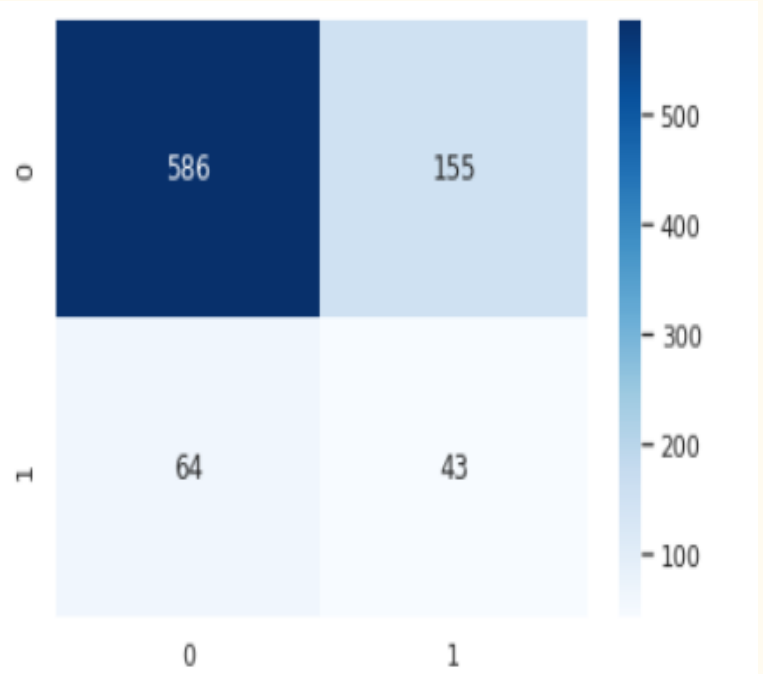
Confusion Matrix

EXTREME GRADIENT BOOST

```
Accuracy of Extreme Gradient Boost: 74.1745283018868

              precision    recall  f1-score   support

         0.0       0.90      0.79      0.84       741
         1.0       0.22      0.40      0.28       107

    accuracy                           0.74       848
   macro avg       0.56      0.60      0.56       848
weighted avg       0.82      0.74      0.77       848
```

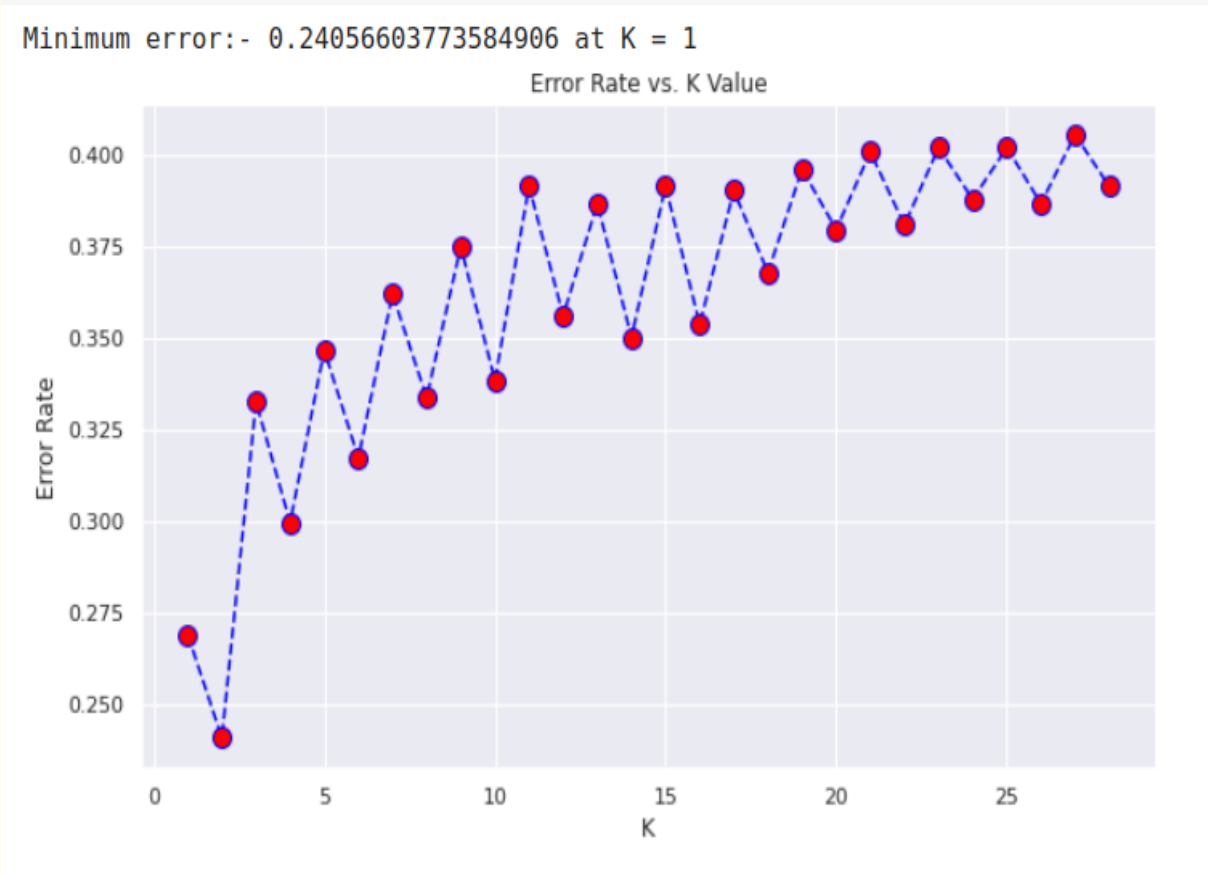
```
[ ]  score4=cross_val_score(xgb,X_valid,y_valid,cv=10)
     print(f"After k-fold cross validation score is {score4.mean()}")
```

```
After k-fold cross validation score is 0.8773669467787114
```

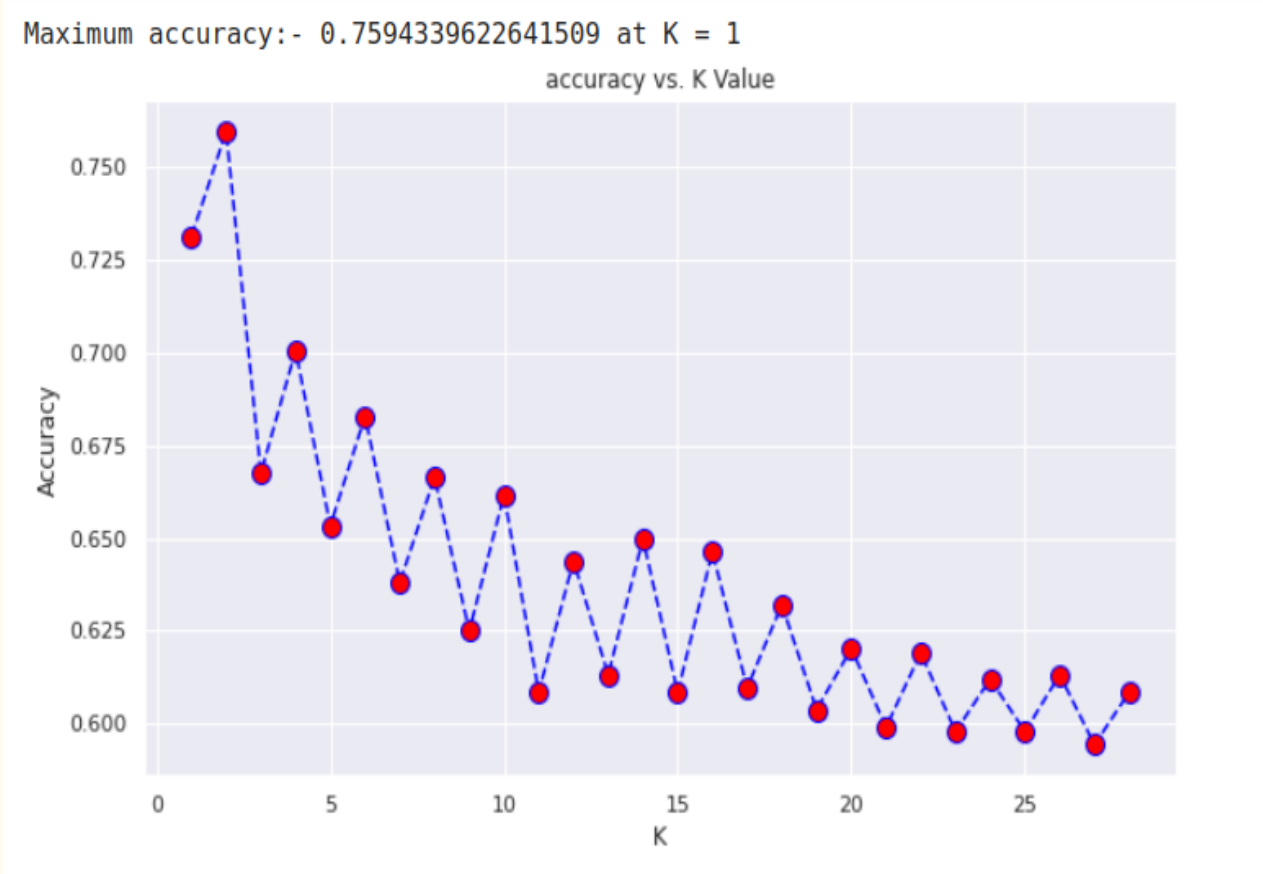


Confusion Matrix

KNEIGHBORS



Error Rate



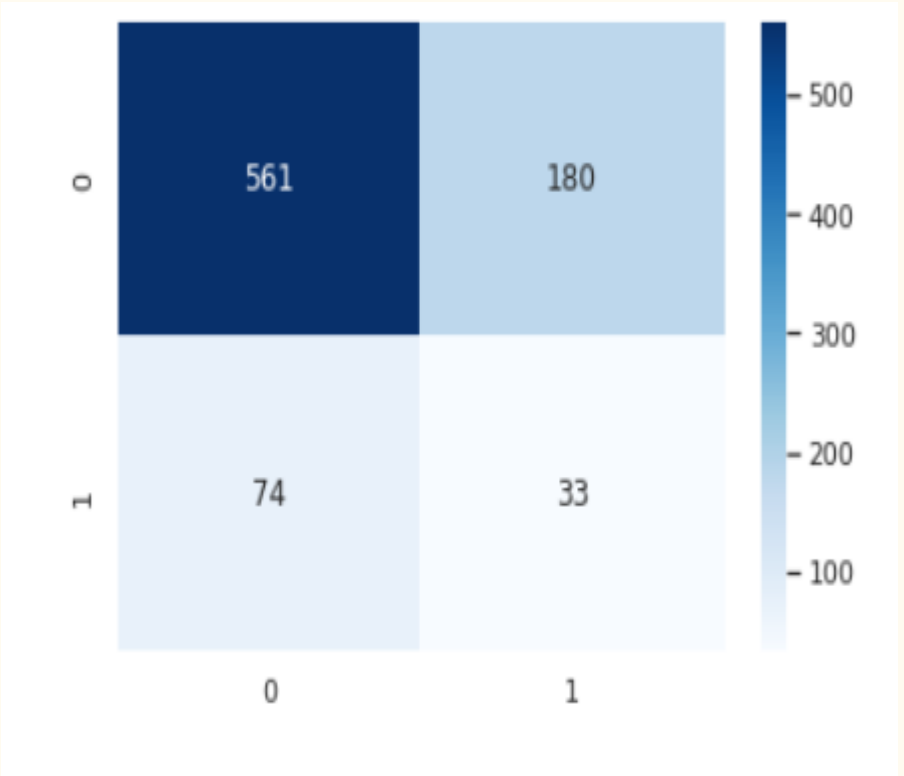
Accuracy Rate

KNEIGHBORS

```
Accuracy of K-NeighborsClassifier: 70.04716981132076

              precision    recall  f1-score   support

         0.0       0.88      0.76      0.82       741
         1.0       0.15      0.31      0.21       107

    accuracy                           0.70       848
   macro avg       0.52      0.53      0.51       848
weighted avg       0.79      0.70      0.74       848
```

```
[ ] score5=cross_val_score(knn,X_valid,y_valid,cv=10)
    print(f"After k-fold cross validation score is {score5.mean()}")
```

```
After k-fold cross validation score is 0.8714705882352941
```



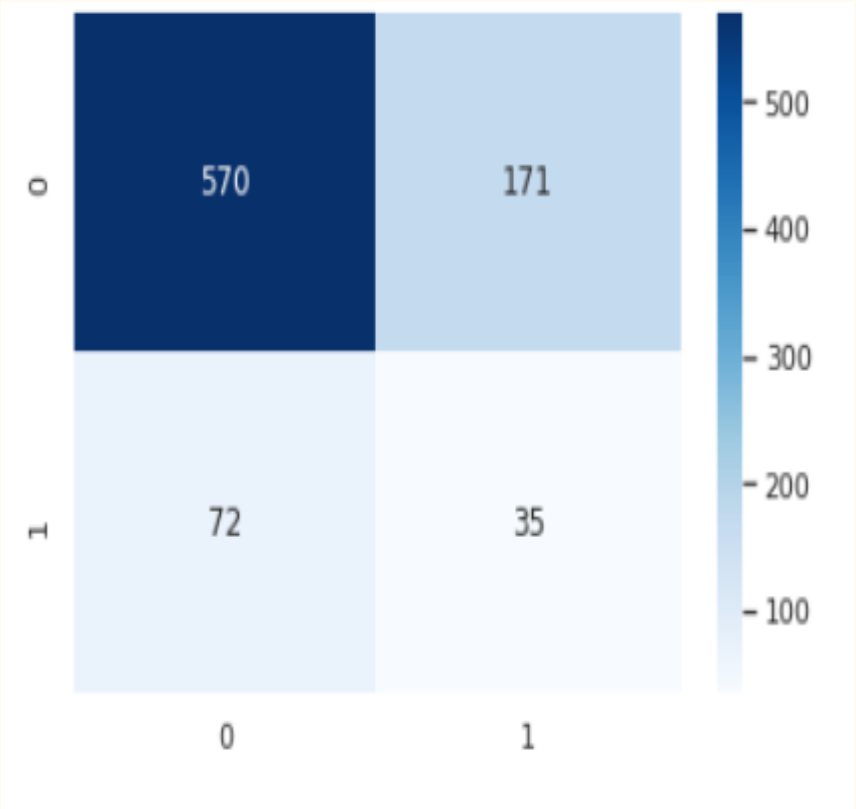
Confusion Matrix

DECISION TREE

```
Accuracy of DecisionTreeClassifier: 71.34433962264151

              precision    recall  f1-score   support

         0.0       0.89      0.77      0.82       741
         1.0       0.17      0.33      0.22       107

    accuracy                           0.71       848
   macro avg       0.53      0.55      0.52       848
weighted avg       0.80      0.71      0.75       848
```

```
[ ] score6=cross_val_score(dt,X_valid,y_valid,cv=10)
    print(f"After k-fold cross validation score is {score6.mean()}")
```

```
After k-fold cross validation score is 0.7936554621848739
```



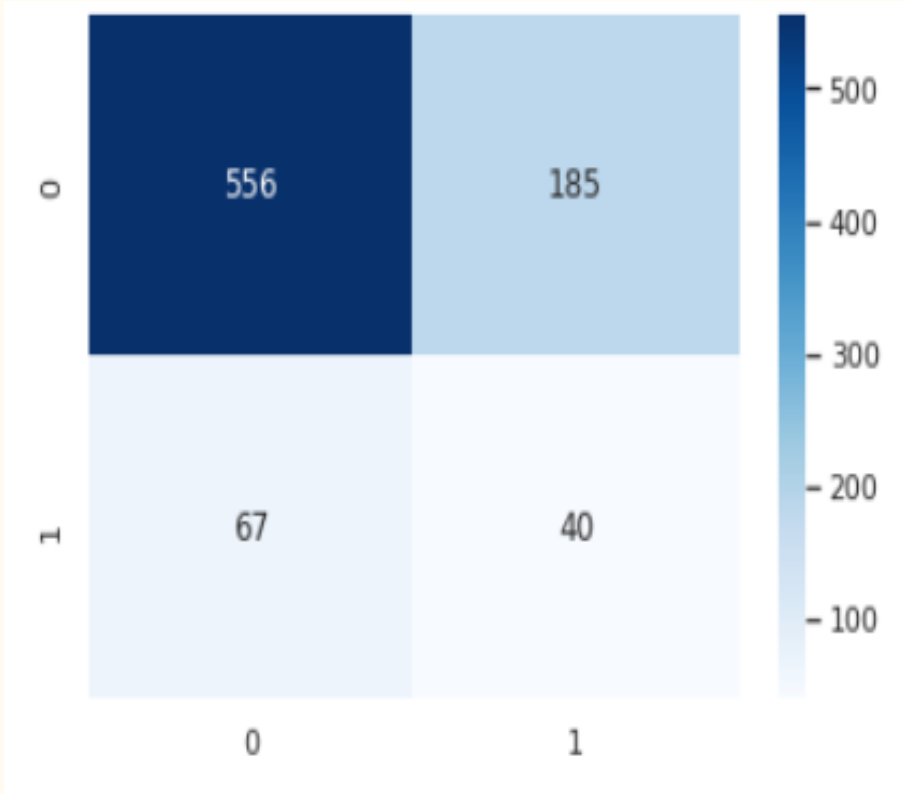
Confusion Matrix

SUPPORT VECTOR CLASSIFIER

```
Accuracy of Support Vector Classifier: 70.28301886792453

              precision    recall  f1-score   support

         0.0       0.89      0.75      0.82       741
         1.0       0.18      0.37      0.24       107

    accuracy                           0.70       848
   macro avg       0.54      0.56      0.53       848
weighted avg       0.80      0.70      0.74       848
```

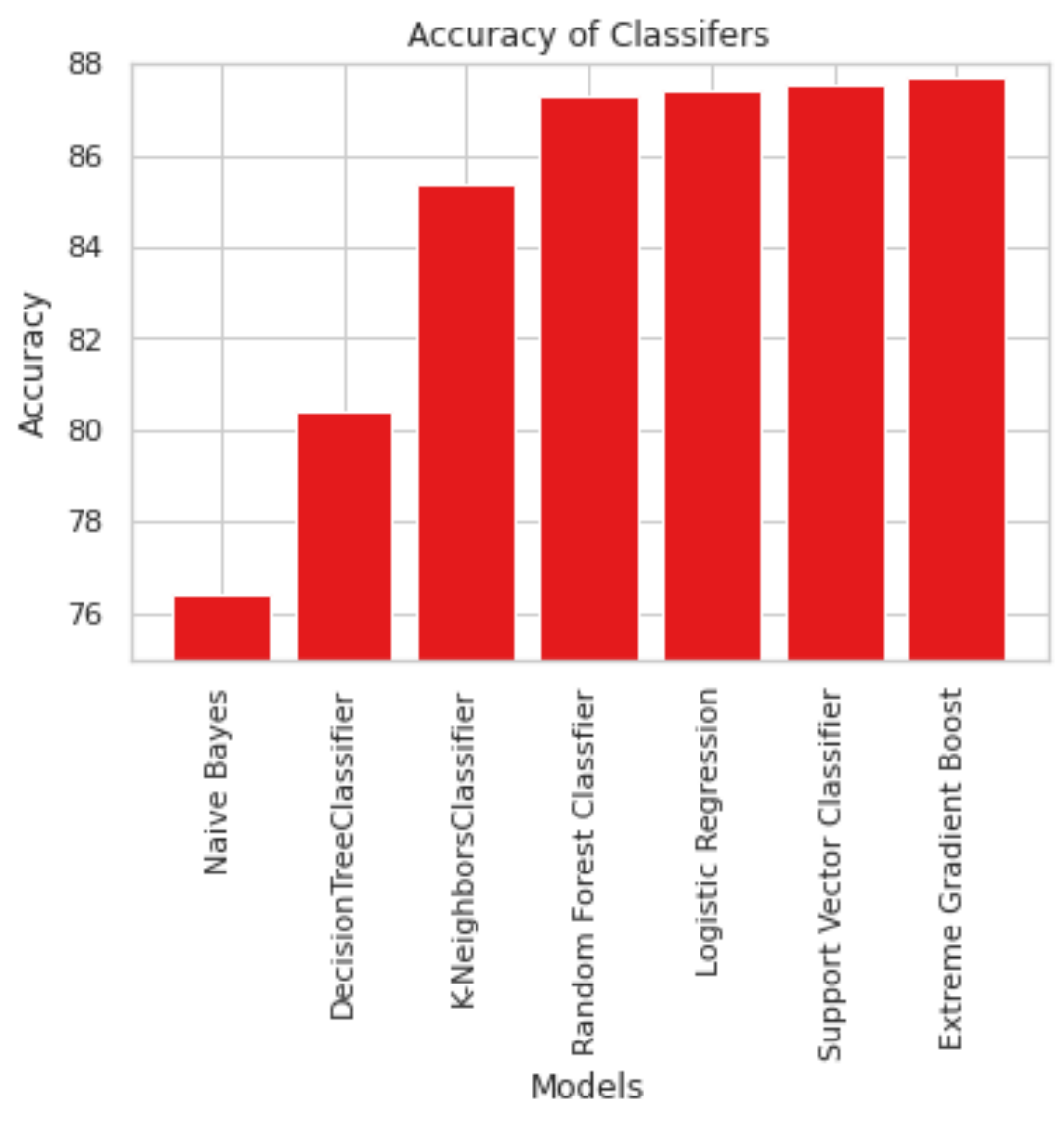
```
[ ] score7=cross_val_score(svc,X_valid,y_valid,cv=10)
    print(f"After k-fold cross validation score is {score7.mean()}")
```

```
After k-fold cross validation score is 0.875014005602241
```



Confusion Matrix

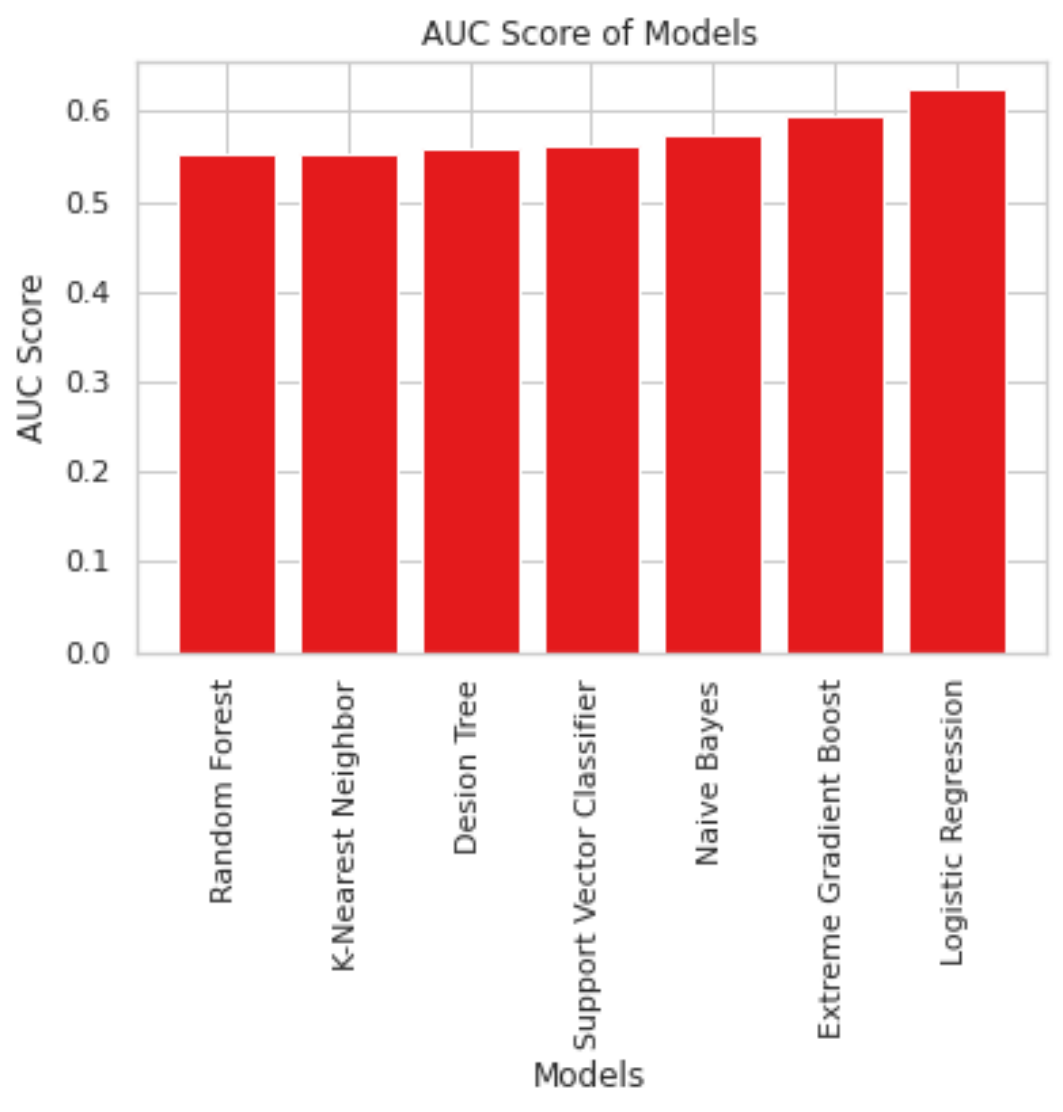
ACCURACY COMPARISON



As we can see,

Extreme Gradient Boost algorithm is most accurate for this dataset.

AUC COMPARISON

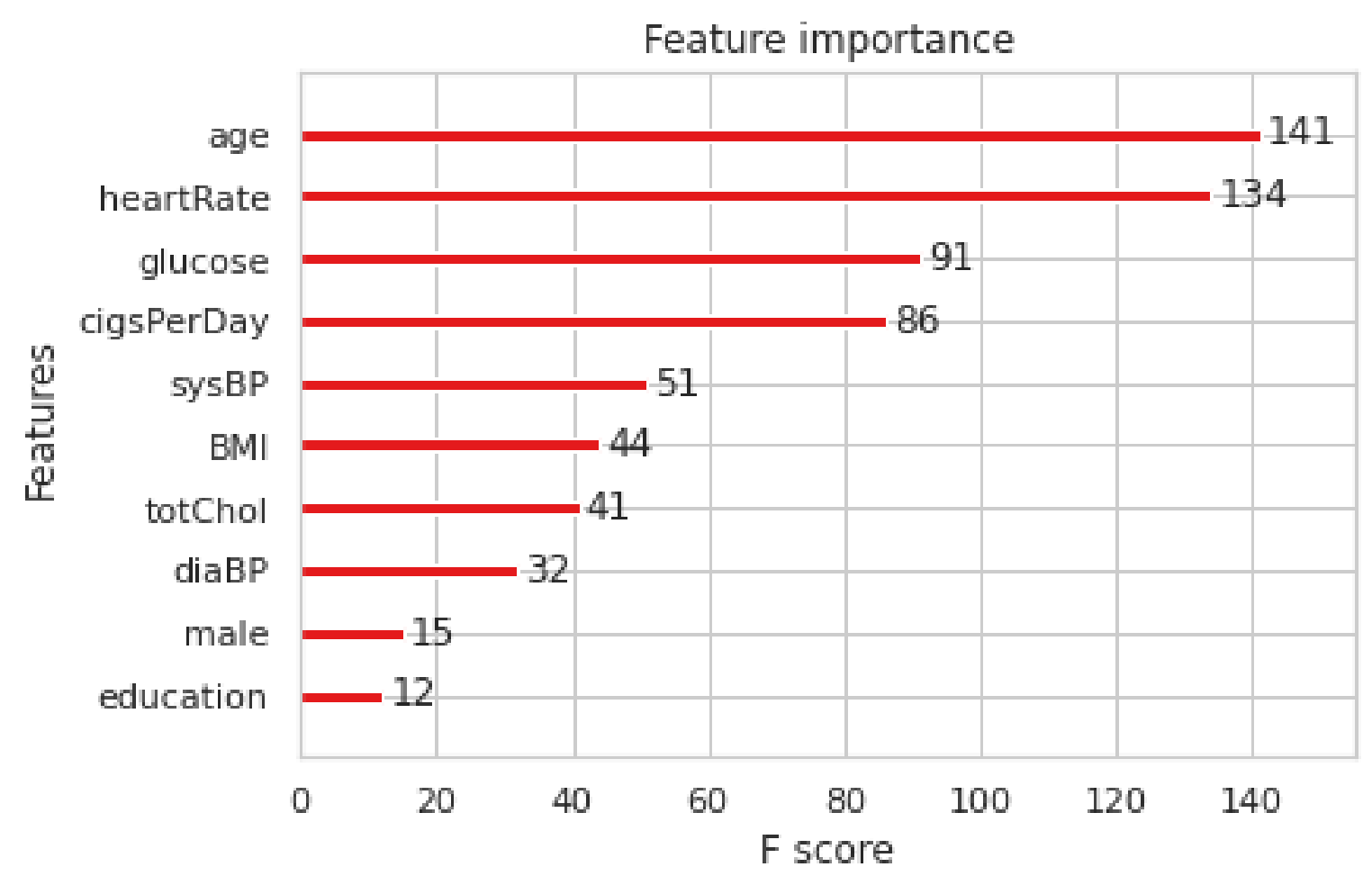


As we can see,

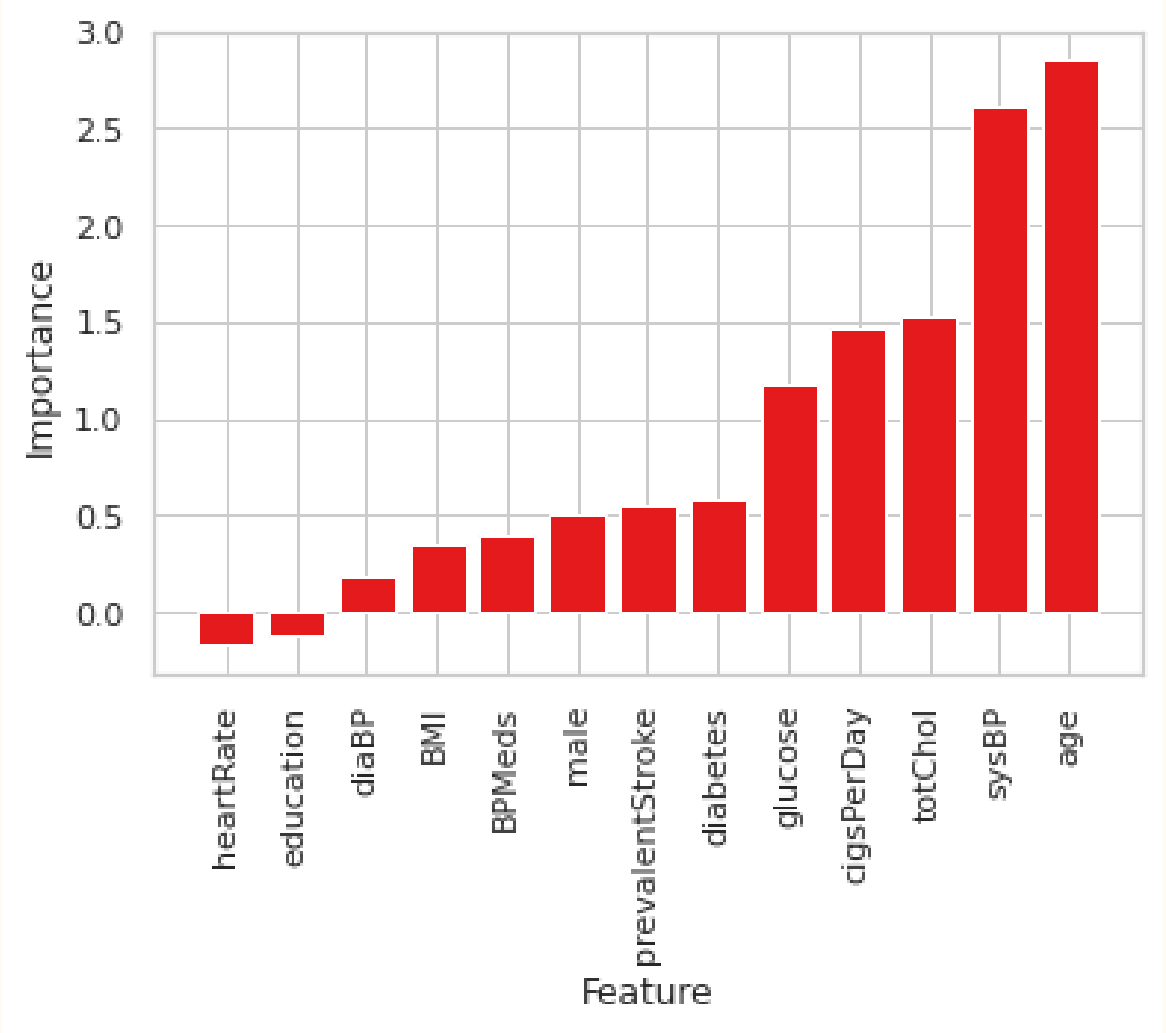
Logistic Regression

algorithm has most AUC for this dataset.

IMPORTANT FEATURES OF EXTREME GRADIENT BOOST CLASSIFIER



IMPORTANT FEATURES OF EXTREME LOGISTIC REGRESSION CLASSIFIER



Prof. Khashayar Dehnad

THANK YOU