BRAC
UNIVERSITY

Inspiring Excellence

CSE-422(Artificial Intelligence)
Sec-04
Project Report

Diabetes detection using machine learning models

Done by:
MD FUAD ISLAM
20101060

MD MINHAZUL ISLAM RIMON
20101078

HASNAT JAMIL BHUIYAN
20101411

Introduction:

We are conducting research to predict if someone will get affected by diabetes among both male and females. A stroke prediction dataset is used here which takes the following inputs: gender, age, hypertension, heart-disease, hemoglobin level, blood glucose level, bmi of a person, smoking history of a person to determine whether a person will suffer from diabetes or not.

Motivation:

Diabetes is a common disease among the elderly in the country and also in the sub-continent. It is so prevalent that we know at least one person among us who is affected by diabetes. So, the study here focuses on training machine learning models which will tell us if someone is likely to get affected by diabetes or not based on parameters mentioned above. It will help to predict a person's chances to get affected by the disease and if it is likely that a person will get affected by the disease then he or she will be alerted early and they can change their certain habits to avoid getting affected by it.

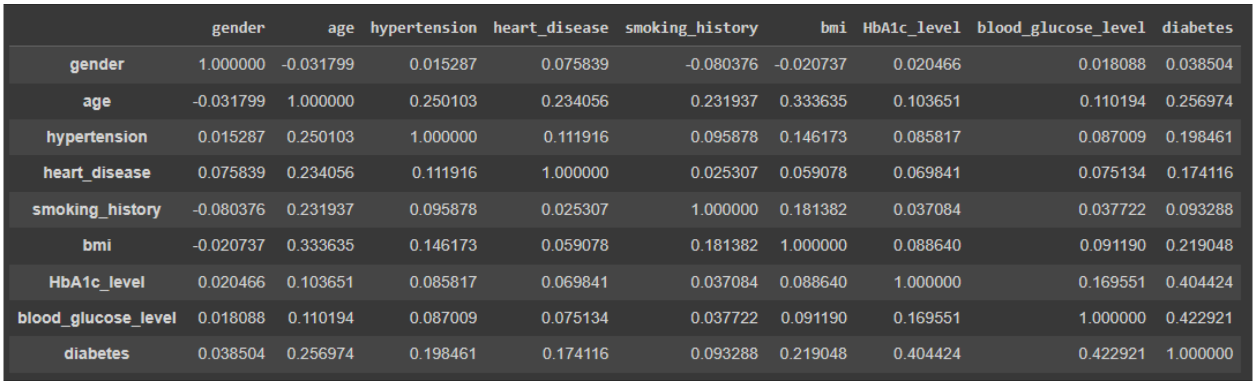
Dataset Description:

Source: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset

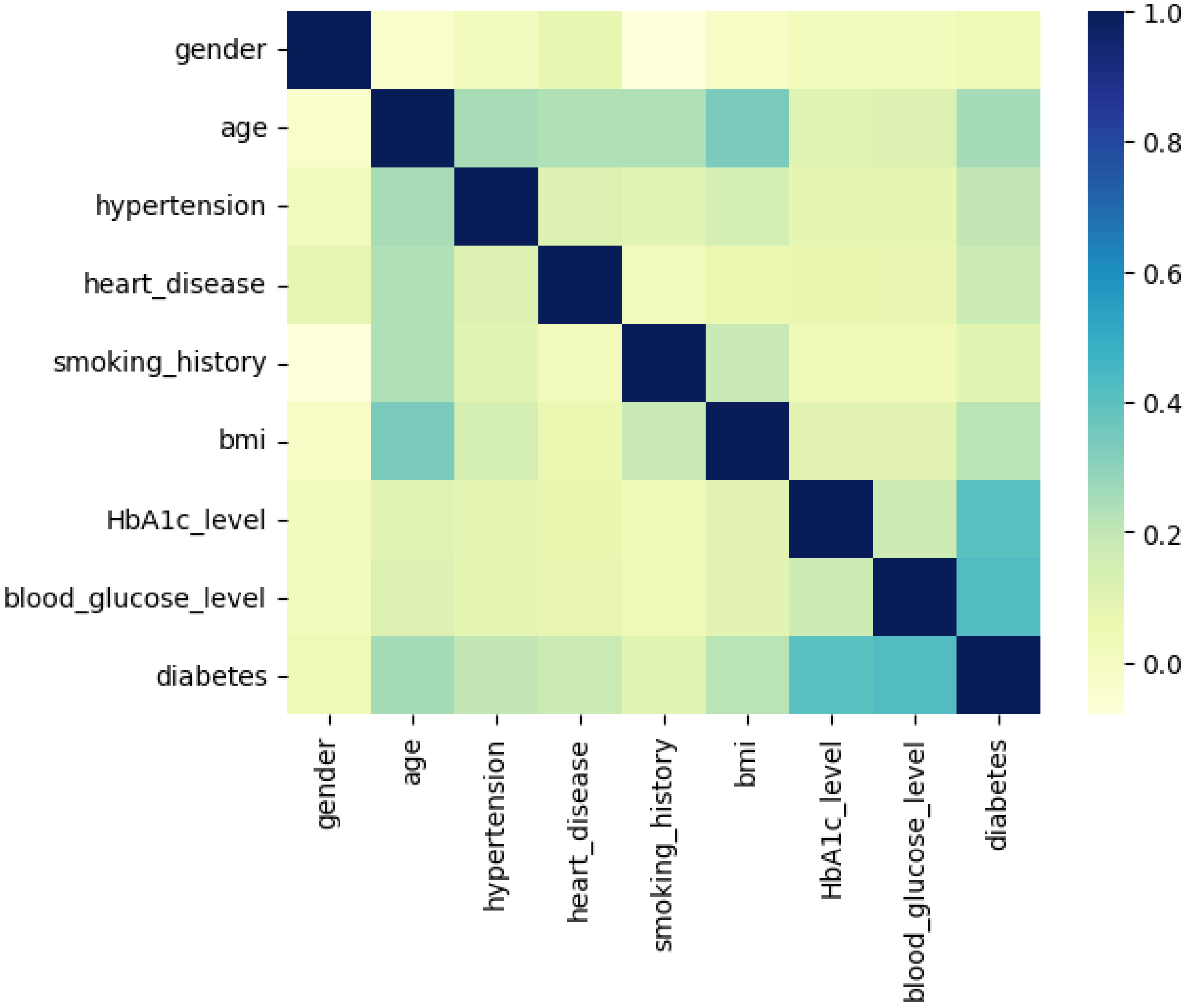
Labels,features,headers:

This dataset contains 9 column headers. Our target is the last column(diabetes). Our dataset contains 8 characteristics. These features are: gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level. A binary class in our target column/label indicates whether or not a diabetes will occur or not.

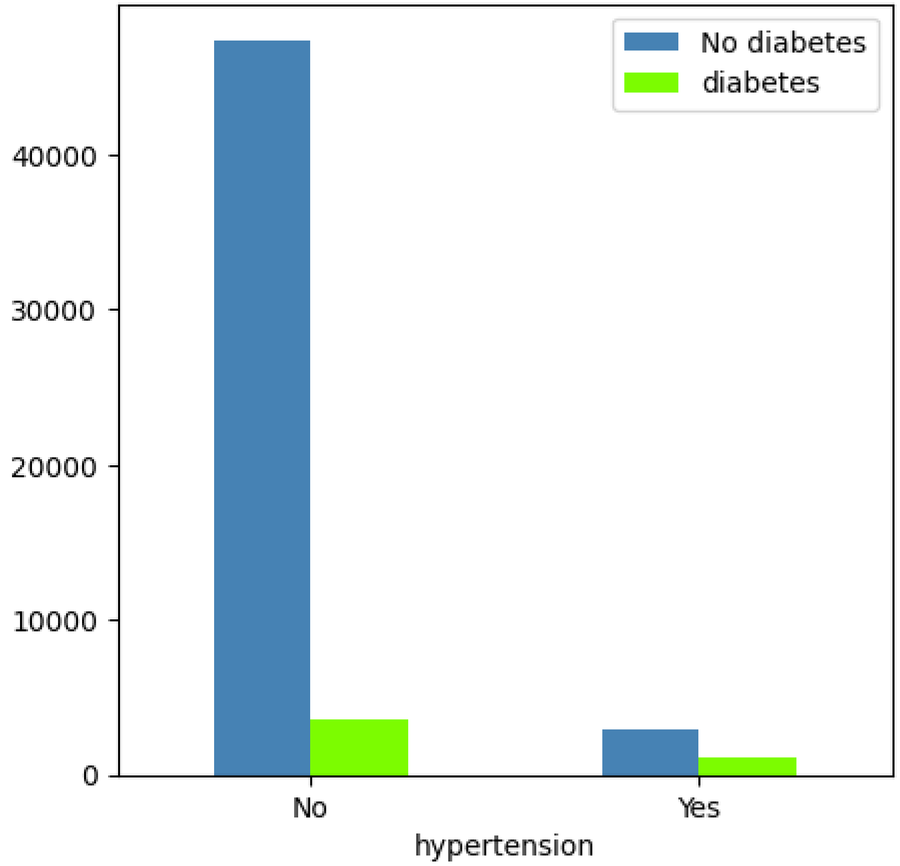
Correlation matrix:

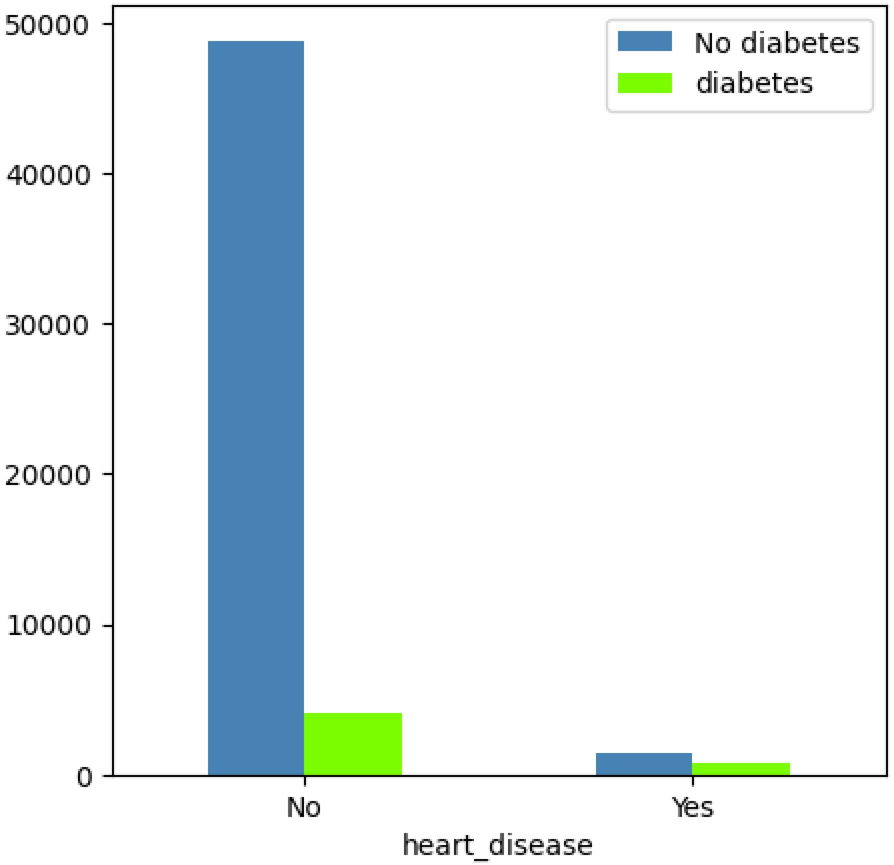


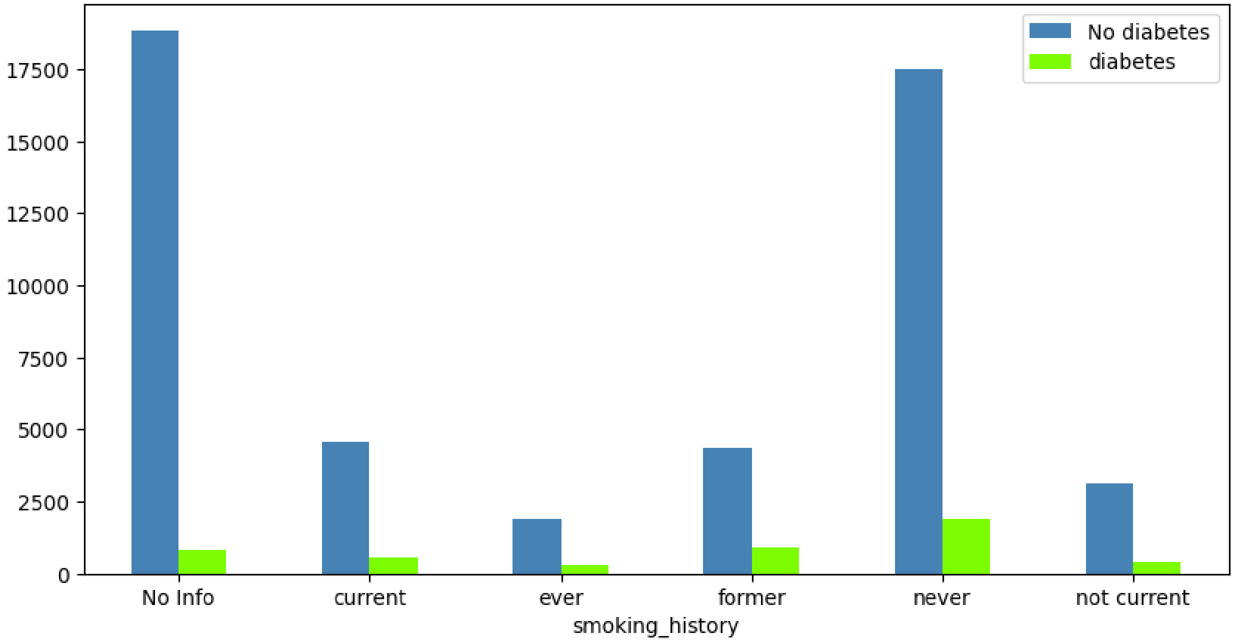
	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
gender	1.000000	-0.031799	0.015287	0.075839	-0.080376	-0.020737	0.020466	0.018088	0.038504
age	-0.031799	1.000000	0.250103	0.234056	0.231937	0.333635	0.103651	0.110194	0.256974
hypertension	0.015287	0.250103	1.000000	0.111916	0.095878	0.146173	0.085817	0.087009	0.198461
heart_disease	0.075839	0.234056	0.111916	1.000000	0.025307	0.059078	0.069841	0.075134	0.174116
smoking_history	-0.080376	0.231937	0.095878	0.025307	1.000000	0.181382	0.037084	0.037722	0.093288
bmi	-0.020737	0.333635	0.146173	0.059078	0.181382	1.000000	0.088640	0.091190	0.219048
HbA1c_level	0.020466	0.103651	0.085817	0.069841	0.037084	0.088640	1.000000	0.169551	0.404424
blood_glucose_level	0.018088	0.110194	0.087009	0.075134	0.037722	0.091190	0.169551	1.000000	0.422921
diabetes	0.038504	0.256974	0.198461	0.174116	0.093288	0.219048	0.404424	0.422921	1.000000


gender
age
hypertension
heart_disease
smoking_history
bmi
HbA1c_level
blood_glucose_level
diabetes
gender
age
hypertension
heart_disease
smoking_history
bmi
HbA1c_level
blood_glucose_level
diabetes
1.0
0.8
0.6
0.4
0.2
0.0

Biasness/Balanced:




50000
40000
30000
20000
10000
0
No diabetes
diabetes
No
Yes
heart_disease


17500
15000
12500
10000
7500
5000
2500
0
No diabetes
diabetes
No Info
current
ever
former
never
not current
smoking_history

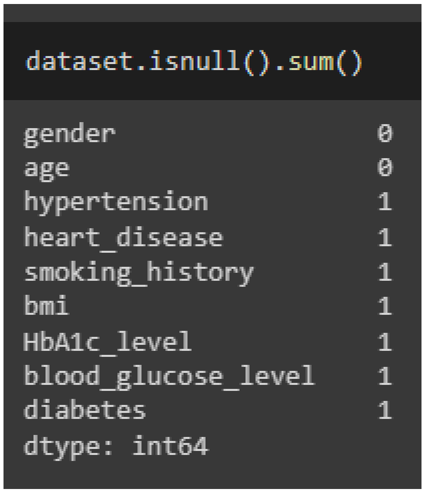
Dataset Pre-processing:

Problem:

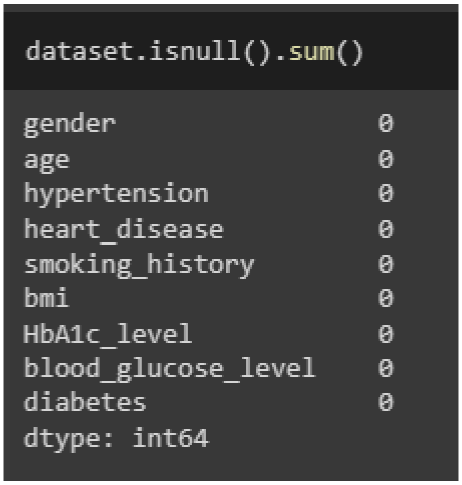
1. Too few entries label ‘others’ in feature ‘gender’.
2. Few entries are missing.

Solution:

1. Delete all the entries with ‘others’ label in ‘gender’ feature.
2. Remove the missing entries to optimize the dataset.

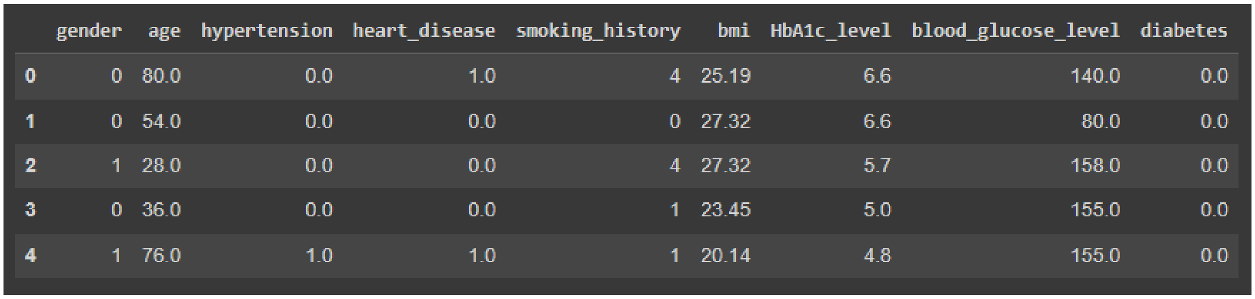


```
dataset.isnull().sum()

gender                 0
age                    0
hypertension           1
heart_disease          1
smoking_history        1
bmi                    1
HbA1c_level            1
blood_glucose_level    1
diabetes               1
dtype: int64
```



```
dataset.isnull().sum()

gender                 0
age                    0
hypertension           0
heart_disease          0
smoking_history        0
bmi                    0
HbA1c_level            0
blood_glucose_level    0
diabetes               0
dtype: int64
```

Encoding: There are few features with string labels. Using label encoder, they were converted to optimize the dataset.

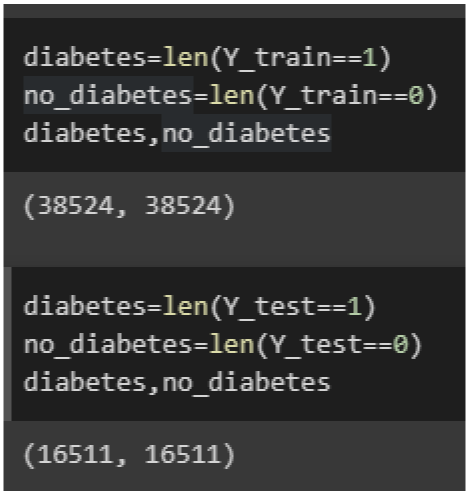


	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	0	80.0	0.0	1.0	4	25.19	6.6	140.0	0.0
1	0	54.0	0.0	0.0	0	27.32	6.6	80.0	0.0
2	1	28.0	0.0	0.0	4	27.32	5.7	158.0	0.0
3	0	36.0	0.0	0.0	1	23.45	5.0	155.0	0.0
4	1	76.0	1.0	1.0	1	20.14	4.8	155.0	0.0

Feature Scaling: We decided to use Decision tree, random forest, xgboost machine learning algorithm for our training our model. The benefit of feature scaling on the computation of these said algorithms is negligible. Hence, feature scaling was avoided.

Data Splitting:

The entire data was divided where 70% of it was used for training the model and the other 30% was reserved for testing. The data was evenly distributed using stratify.



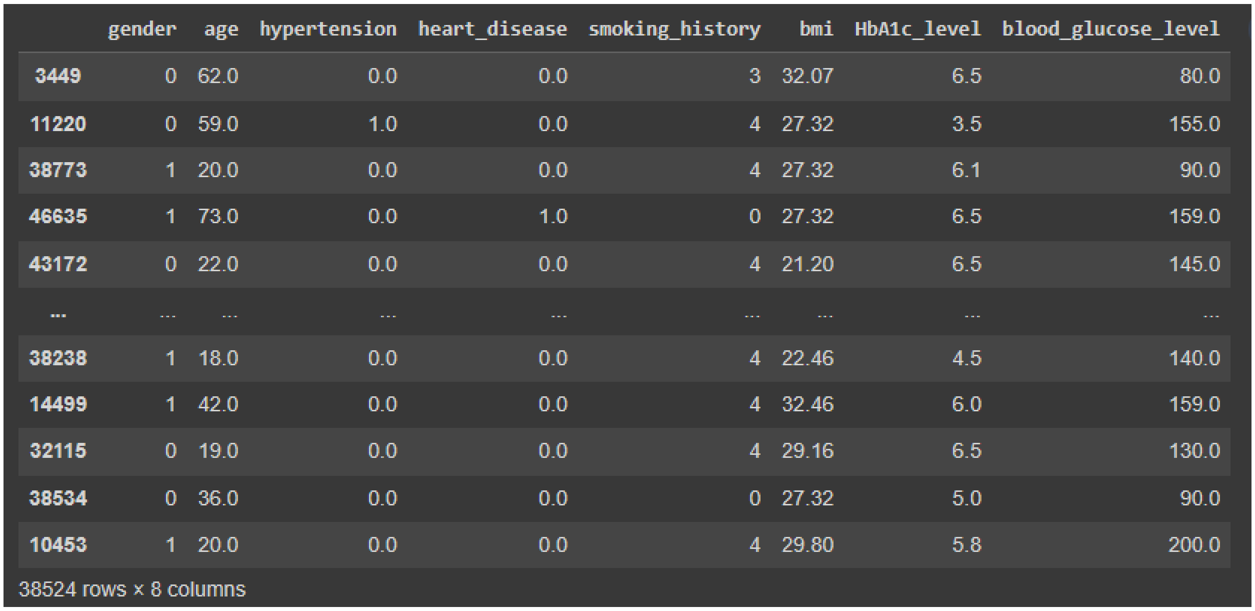
```
diabetes=len(Y_train==1)
no_diabetes=len(Y_train==0)
diabetes,no_diabetes
```

```
(38524, 38524)
```

```
diabetes=len(Y_test==1)
no_diabetes=len(Y_test==0)
diabetes,no_diabetes
```

```
(16511, 16511)
```

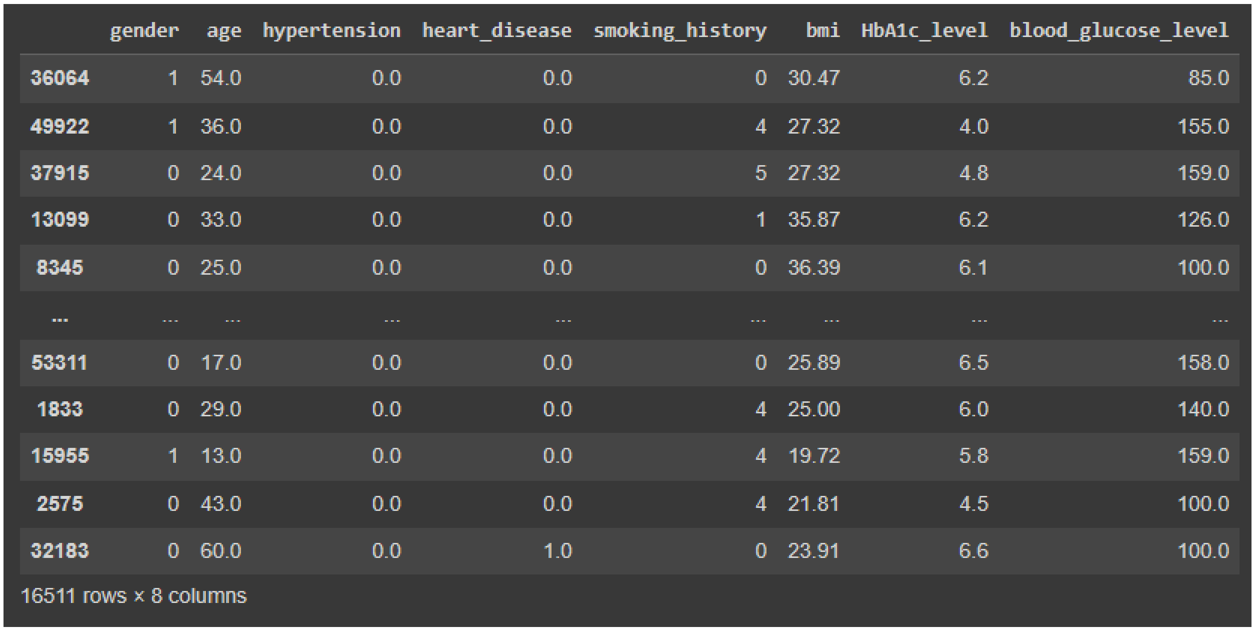
Train set:



	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level
3449	0	62.0	0.0	0.0	3	32.07	6.5	80.0
11220	0	59.0	1.0	0.0	4	27.32	3.5	155.0
38773	1	20.0	0.0	0.0	4	27.32	6.1	90.0
46635	1	73.0	0.0	1.0	0	27.32	6.5	159.0
43172	0	22.0	0.0	0.0	4	21.20	6.5	145.0
...	...	...	...	...	...	...	...	...
38238	1	18.0	0.0	0.0	4	22.46	4.5	140.0
14499	1	42.0	0.0	0.0	4	32.46	6.0	159.0
32115	0	19.0	0.0	0.0	4	29.16	6.5	130.0
38534	0	36.0	0.0	0.0	0	27.32	5.0	90.0
10453	1	20.0	0.0	0.0	4	29.80	5.8	200.0

38524 rows × 8 columns

Test set:



	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level
36064	1	54.0	0.0	0.0	0	30.47	6.2	85.0
49922	1	36.0	0.0	0.0	4	27.32	4.0	155.0
37915	0	24.0	0.0	0.0	5	27.32	4.8	159.0
13099	0	33.0	0.0	0.0	1	35.87	6.2	126.0
8345	0	25.0	0.0	0.0	0	36.39	6.1	100.0
...	...	...	...	...	...	...	...	...
53311	0	17.0	0.0	0.0	0	25.89	6.5	158.0
1833	0	29.0	0.0	0.0	4	25.00	6.0	140.0
15955	1	13.0	0.0	0.0	4	19.72	5.8	159.0
2575	0	43.0	0.0	0.0	4	21.81	4.5	100.0
32183	0	60.0	0.0	1.0	0	23.91	6.6	100.0

16511 rows × 8 columns

Model training:

Decision Tree:

Although it is frequently chosen for doing so, this supervised learning approach can be used to solve classification and regression problems. It is a tree-structured classifier in which the dataset's features are represented by the inside nodes and each leaf node represents the classification result. For larger datasets, it is an excellent approach. The unique value from the characteristics is selected using this method's application of entropy.

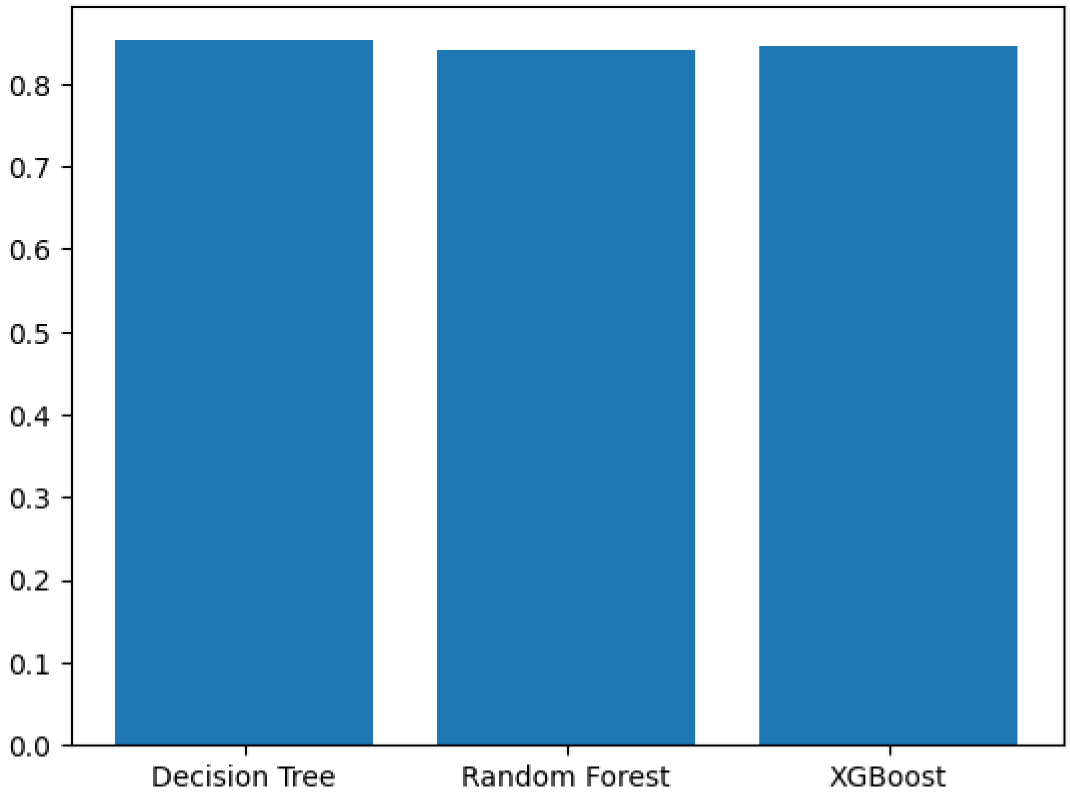
Random Forest:

Random Forest is used for classification and regression tasks due to its high accuracy, robustness, feature importance, versatility, and scalability. It reduces the overfitting problem by averaging multiple decision trees and is less sensitive to noise and outliers in the data. It provides a measure of feature importance, which can be useful for feature selection and data interpretation.

XGBoost:

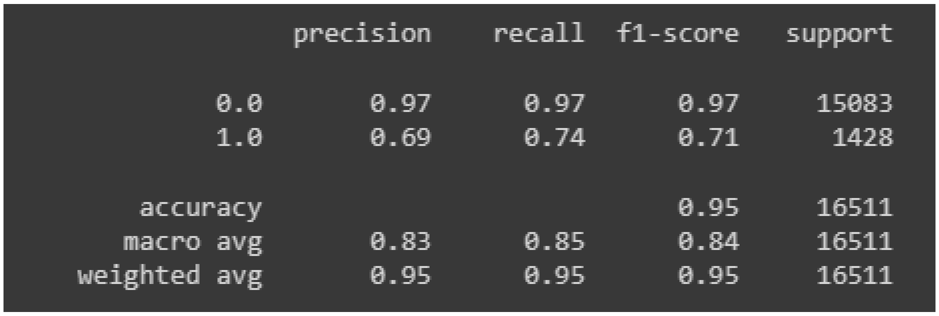
XGBoost is used for these two reasons: execution speed and model performance.It is designed for speed, ease of use, and performance on large datasets. It does not require optimization of the parameters or tuning, which means that it can be used immediately after installation without any further configuration. XGBoost offers regularization, which allows us to control overfitting by introducing L1/L2 penalties on the weights and biases of each tree.Another feature of XGBoost is its ability to handle sparse data sets using the weighted quantile sketch algorithm. This algorithm allows us to deal with non-zero entries in the feature matrix while retaining the same computational complexity as other algorithms like stochastic gradient descent.XGBoost also has a block structure for parallel learning. It makes it easy to scale up on multicore machines or clusters. It also uses cache awareness, which helps reduce memory usage when training models with large datasets.Finally, XGBoost offers out-of-core computing capabilities using disk-based data structures instead of in-memory ones during the computation phase.

Results:

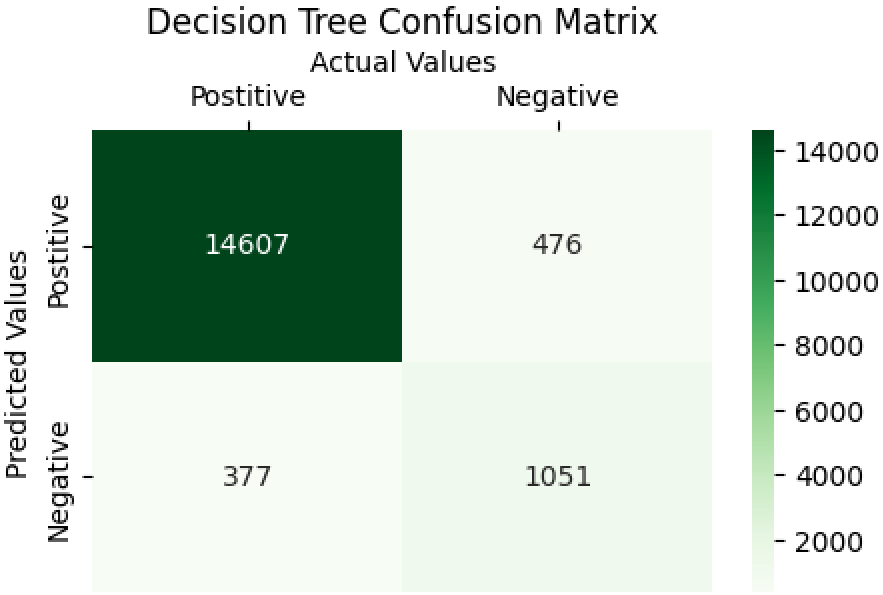


Accuracy, Precision, Recall, F1 score, Confusion matrix:

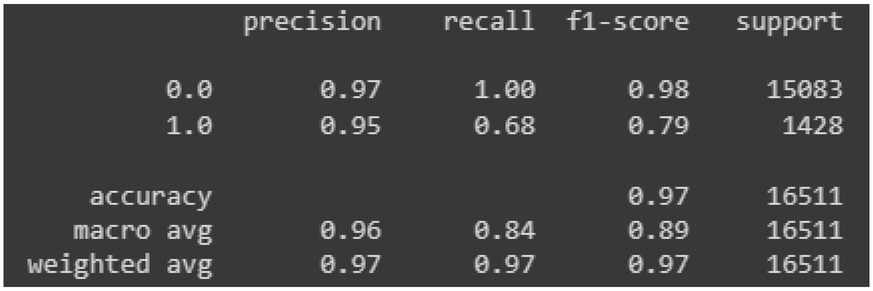
1. Decision Tree:



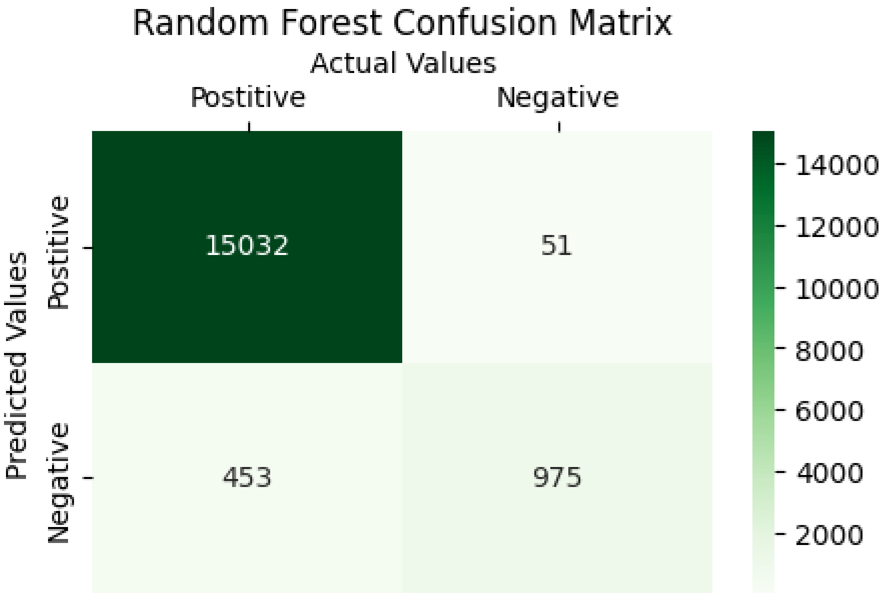
```
              precision    recall  f1-score   support

         0.0       0.97      0.97      0.97     15083
         1.0       0.69      0.74      0.71      1428

    accuracy                           0.95     16511
   macro avg       0.83      0.85      0.84     16511
weighted avg       0.95      0.95      0.95     16511
```



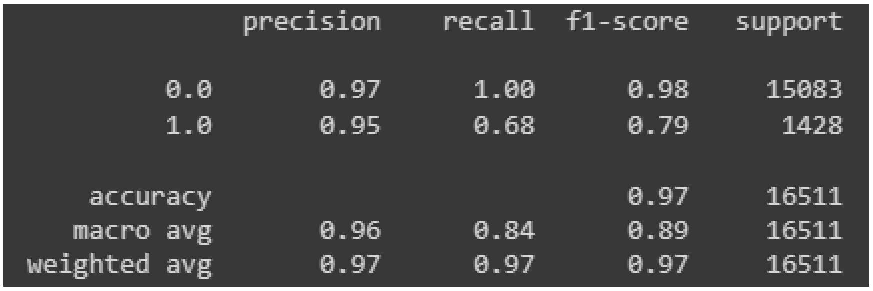
2. Random Forest:



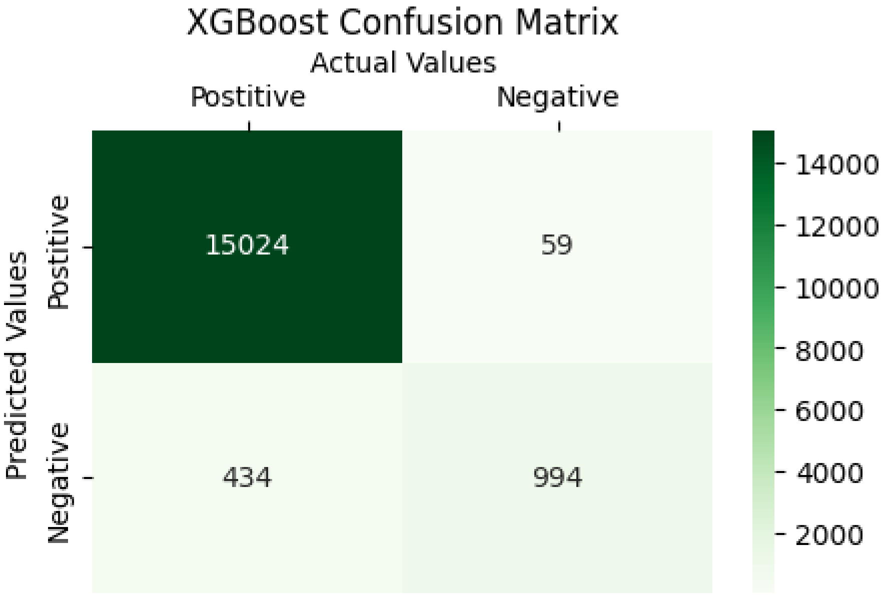
```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     15083
         1.0       0.95      0.68      0.79      1428

    accuracy                           0.97     16511
   macro avg       0.96      0.84      0.89     16511
weighted avg       0.97      0.97      0.97     16511
```



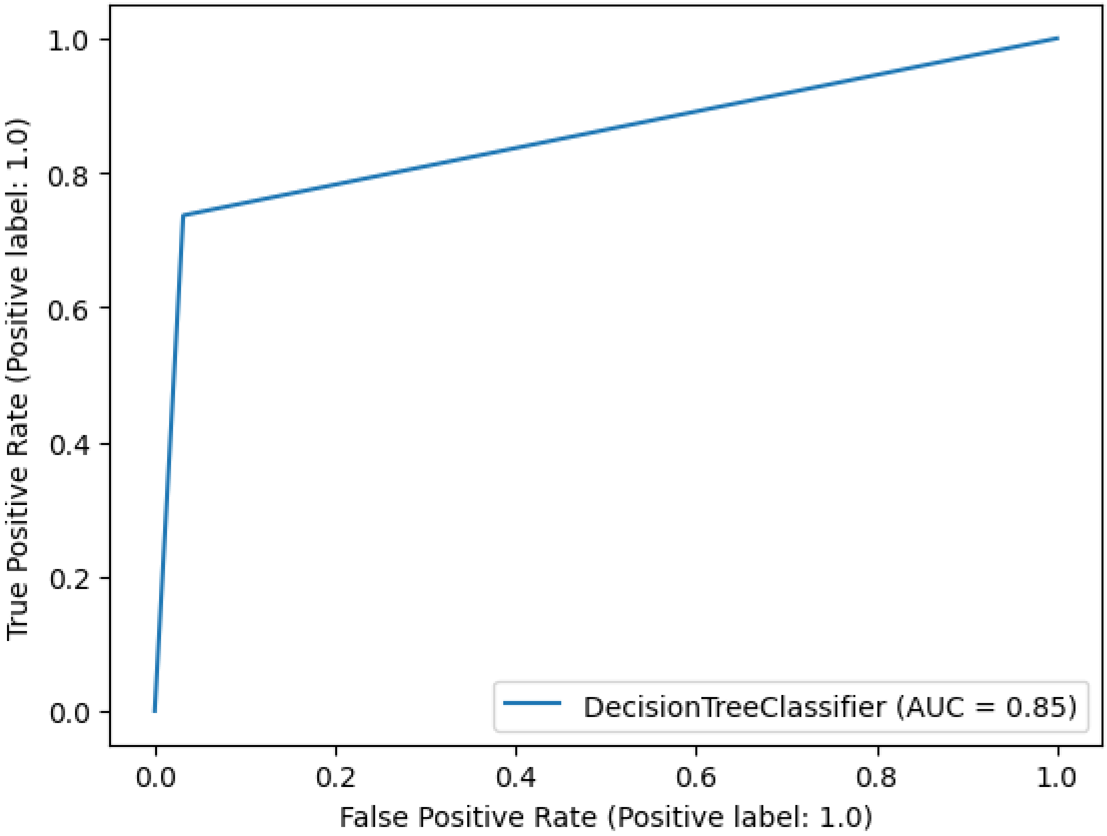
3. <u>xgBoost:</u>



```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     15083
         1.0       0.95      0.68      0.79      1428

    accuracy                           0.97     16511
   macro avg       0.96      0.84      0.89     16511
weighted avg       0.97      0.97      0.97     16511
```



ROC, AUC curve:



Conclusion:

Our trained models have passed with overall high accuracy scores for all three models. Among the three models, the model trained with the decision tree seems to have the highest accuracy of all. So, we can say that the decision tree works best for this dataset.

Future work:

Though we have managed to get high accuracy scores in all our models, none of them are 100% accurate yet. We can utilize advanced machine learning algorithms to ensure the model works better. We can also fine tune the models to our specific need to achieve a better result in future projects.

References:

1. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=A.%20Random%20Forest%20is%20a,and%20outliers%20in%20the%20data.
2. https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article#:~:text=XGBoost%20is%20a%20robust%20algorithm,any%20decision%20tree%2Dbased%20model.