Diabates Classification Problem

Data Overview:

The dataset contains the health and demographic data of 100,000 individuals. This dataset includes information on gender, age, location, race, hypertension, heart disease, smoking history, BMI, HbA1c level, blood glucose level, and diabetes status. The dataset is used to build a classification model to predict the diabetes status of individual. Given below is a screenshot of the sample data.

ian	race:Caucasian	race:Hispanic	race:Other	hypertension	heart_disease	smoking_history	bmi	hbA1c_level	blood_glucose_level	diabetes	clinical_notes
0	0	0	1	0	0	never	27.32	5.0	100	0	Overweight, advised dietary and exercise modif...
1	0	0	0	0	0	never	19.95	5.0	90	0	Healthy BMI range.
0	0	0	1	0	0	never	23.76	4.8	160	0	Young patient, generally lower risk but needs ...
0	1	0	0	0	0	never	27.32	4.0	159	0	Overweight, advised dietary and exercise modif...
0	0	0	0	0	0	never	23.75	6.5	90	0	Healthy BMI range. High HbA1c level, indicativ...

Data Exploration:

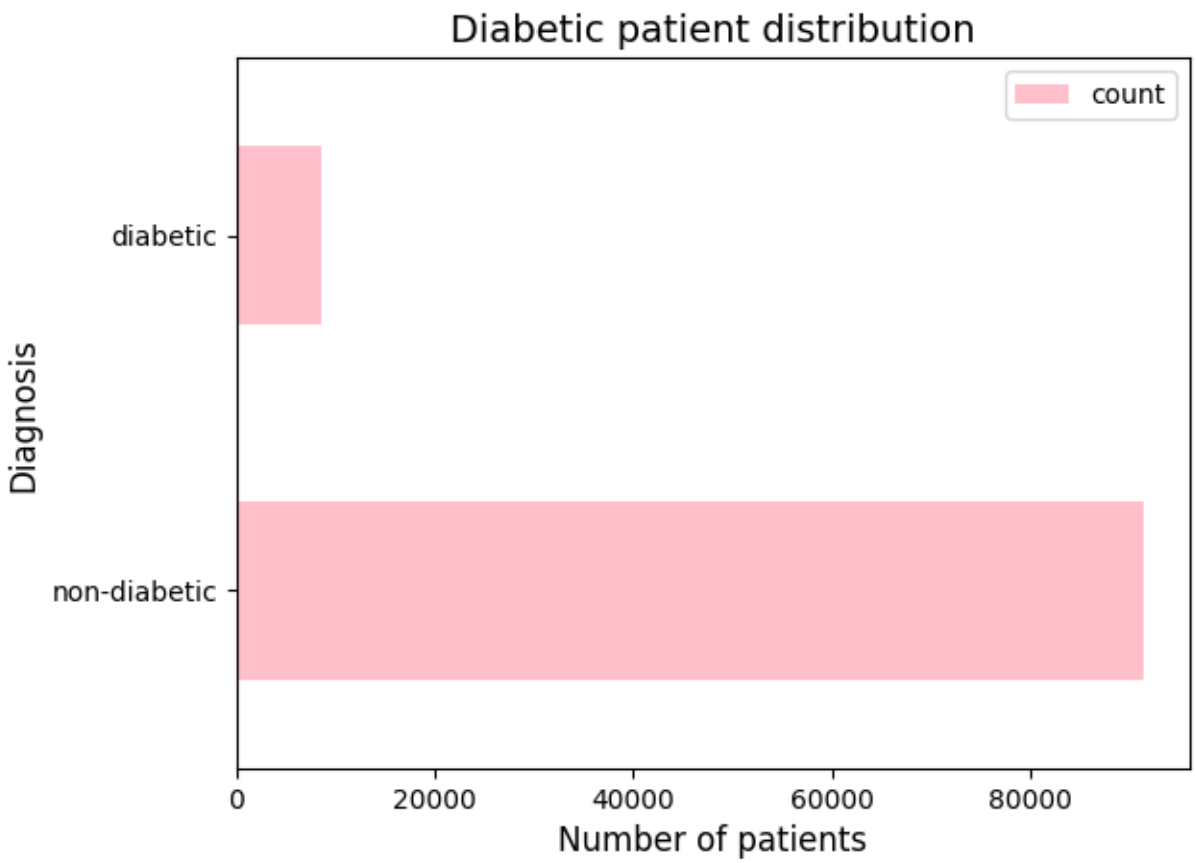
The data contains the following variables as explained below.

1. Year: The year in which the record has been taken.
2. Gender: Gender of the patient (Male/ Female/Other)
3. Age: Patient's age
4. Location: Patient's Location
5. Race: 5 Variables describing the patient's race
6. Hypertension: Binary variable saying if the patient has hypertension
7. heart_disease: Binary variable saying if the patient has heart disease
8. smoking_history: Categorical variable detailing the smoking history of patient.
9. BMI: Patient's BMI
10. hbA1c_level
11. blood_glucose_level
12. Diabetes: Binary variable detailing if the patient has diabetes
13. Clinical Notes: Diagnosis provided by the doctor

Null value check: The data contains no null values

There are no erroneous values present in any of the categorical variables as well.

The distribution of the target variable is shown in the plot below



As shown in the plot above, target variable has a class imbalance in the predictor variable. About 91% patients are non – diabetic.

This imbalance needs to be accounted for when splitting the data into train and test samples before fitting the model. (stratified samples should be considered)

Feature Engineering:

- Dropped columns like year, location as they don't carry any significance to predict if a patient is diabetic.
- Dropped the column clinical_notes as it is not numerical and can't be directly used in the model.
- Dropped the column race:Other to avoid multi collinearity
- Applied one-hot encoding on Gender and 'smoking_history' as those variables are not ordinal.

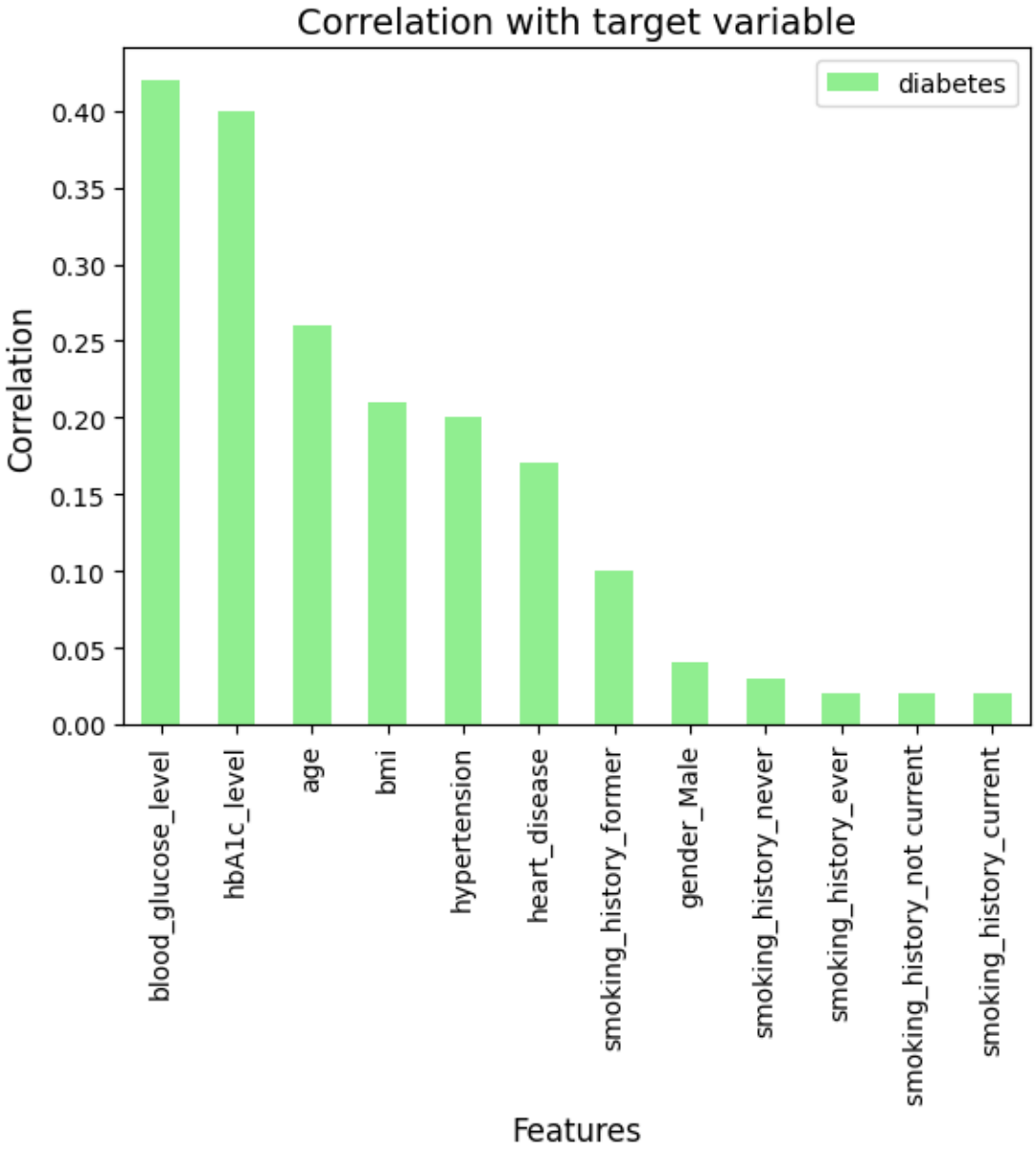
Extracted the correlation between the features and the diabetes variable.

```
age                          0.258008
race:AfricanAmerican         0.004378
race:Asian                   0.003739
race:Caucasian              -0.001749
race:Hispanic               -0.001301
hypertension                 0.197823
heart_disease                0.171727
bmi                          0.214357
hbA1c_level                  0.400660
blood_glucose_level          0.419558
diabetes                     1.000000
gender_Female               -0.037553
gender_Male                  0.037666
smoking_history_current      0.019606
smoking_history_ever         0.024080
smoking_history_former       0.097917
smoking_history_never        0.027267
smoking_history_not current  0.020734
```

```
Name: diabetes, dtype: float64
```

The gender_Female variable has been dropped to avoid multi-collinearity.
All the race variables have been dropped because they carry very less correlation with predictor.

The plot below describes the correlation of the final variables selected for modelling with the predictor variable



Objective:

The main objective of this project is to come up with a predictor which best classifies the data. Accuracy of the prediction takes precedence over the model's explainability.

We shall initially begin by fitting basic models like Logistic Regression and KNN as these models are simple and have very good interpretability.

If the results are not promising enough, we will proceed to fit ensemble models using boosting and bagging techniques.

In each of these models, F1 Score is given higher importance over accuracy of the prediction as there is already very high class imbalance. Hence the aim will be to reduce the False Positives and False Negatives to maximum extent.

Model fitting and results:

Splitting the data:

The data has been split into train and test in a stratified split to take care of the class imbalance.

The train data set contains 70,000 records and the test dataset contains 30,000 records.

Both the datasets are now scaled using MinMaxScaler, to fit models like KNN Classifier and Support Vector Classifier.

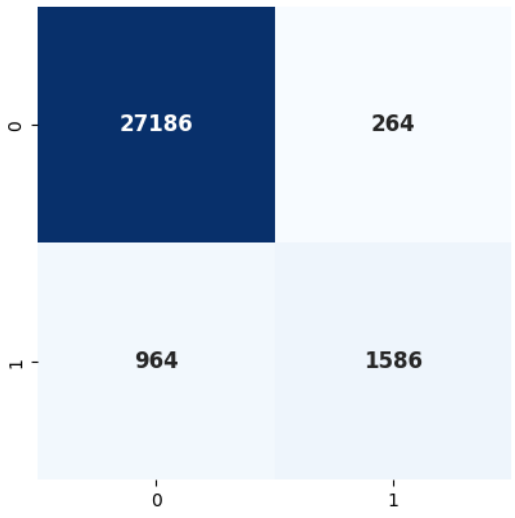
Model 1: Logistic Regression:

3 logistic regression models were fit. One a regular model and two others with l1 and l2 regularizations respectively.

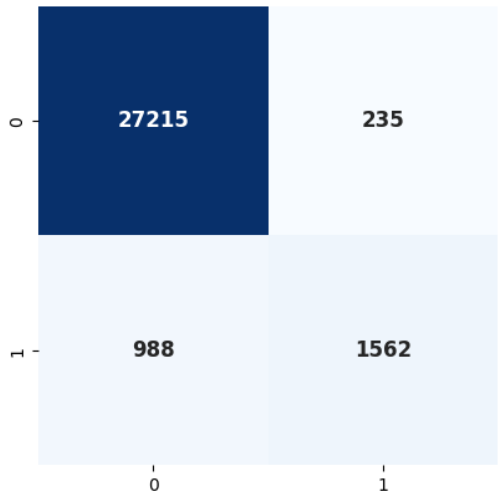
All the 3 models gave similar outputs as shown below.

Model	LR	LR_L1	LR_L2
precision	0.857297	0.848055	0.848341
recall	0.621961	0.632549	0.631765
fscore	0.720909	0.724618	0.724208
accuracy	0.959067	0.959133	0.959100
auc	0.960644	0.960614	0.960653

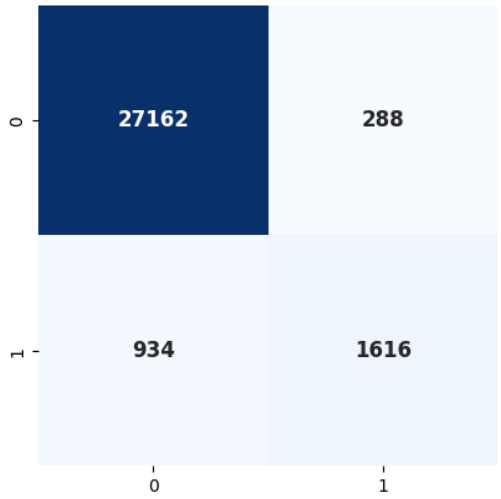
LR confusion matrix:



LR_L1 confusion matrix:



LR_L2 confusion matrix:



- All the 3 models gave similar results.
- While the accuracy of the predictions is high, this can't be taken as the only measure as there is a class imbalance in the model.
- The F1 score currently stands at 0.72 which needs to be further improved by taking care of the misclassification.

Model 2: KNN

A KNN model was fit and the hyperparameters were experimented with using a GridSearch.

Considered the following set of neighbour counts - [5,15,25,50,100,200,300] along with the weights parameter being – ['uniform','distance']

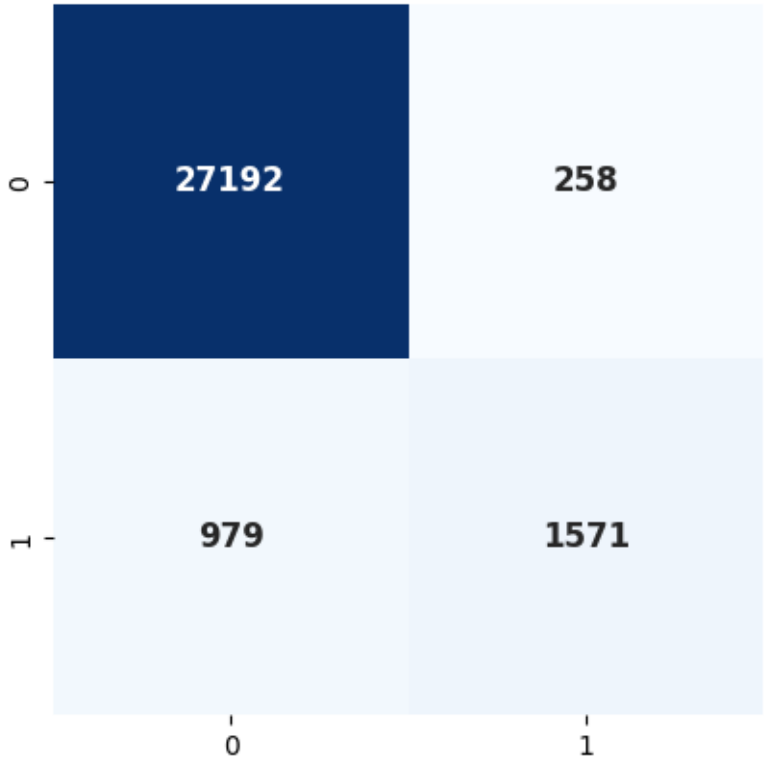
The best model was achieved with 5 neighbours and the weights parameter as ‘distance’

GridsearchCV was used with 5 folds to avoid overfitting.

Given below are the results on the test data

	KNN
precision	0.858939
recall	0.616078
fscore	0.717515
accuracy	0.958767
auc	0.803340

Confusion Matrix:



- This model performed slightly worse than logistic regression with an F1 score of 0.717
- Hence considering more complex models, to get a better fit and model a better solution.

Model 3 - Random Forest:

Fit a random forest model and used Grid Search CV to tune the hyperparameters.

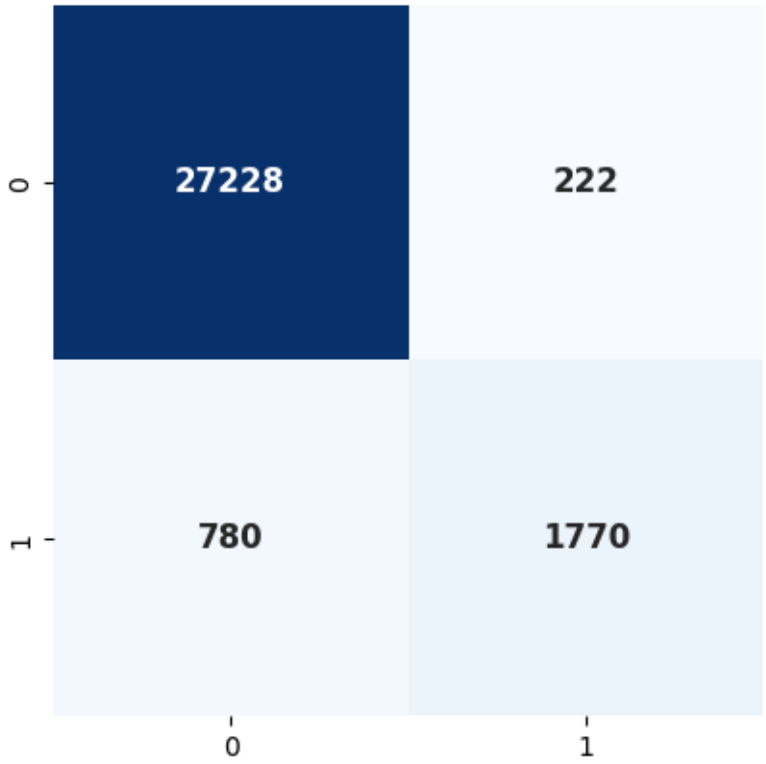
The following combinations were considered.

```python
rf = RandomForestClassifier(random_state=24,max_features = 'sqrt',class_weight = 'balanced')

param_grid = {'n_estimators':[50,100,200], 'max_depth':[10,20],'min_samples_split':[2,5,10]}
```

The best estimator gave the following results

precision	0.888554
recall	0.694118
fscore	0.779392
accuracy	0.966600
auc	0.843015

Confusion Matrix:



- This is slightly better fit than the previous models and produced an F1 score of 0.77.
- It reduced both FPs and FNs
- Boosting models have also been explored for even better fits

Model 4 – Gradient Boosting Classifier:

Fit a gradient boosting model and used Grid Search CV to tune the hyperparameters.

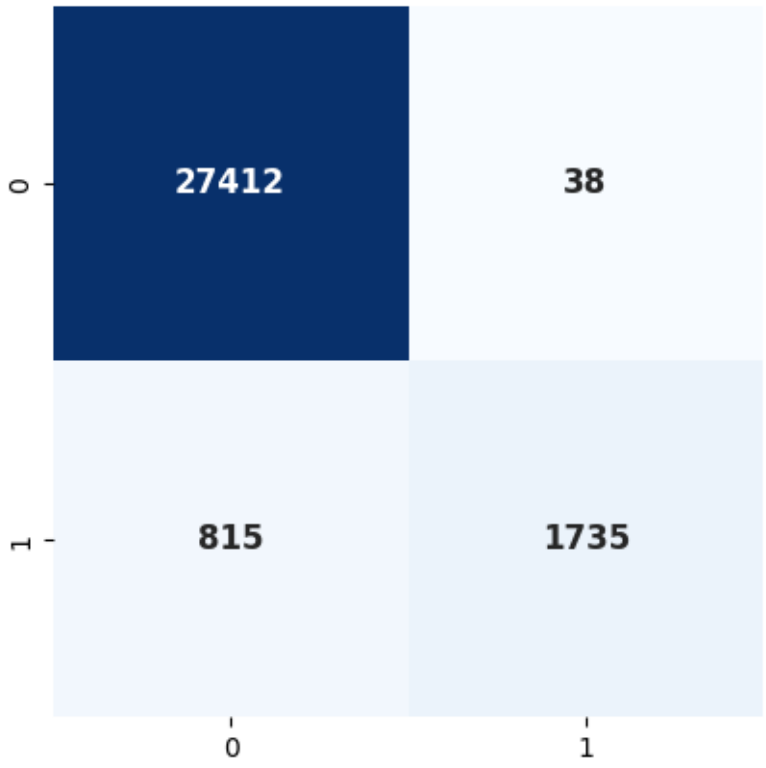
The following combinations were considered.

```python

gbc = GradientBoostingClassifier(random_state=24,max_features = 'sqrt')

param_grid = {'n_estimators':[50,100,200], 'max_depth':[3,5],'min_samples_split':[2,5,10], 'learning_rate': [0.01, 0.1]}

```

The best estimator gave the following results

precision	**0.978567**
recall	**0.680392**
fscore	**0.802683**
accuracy	**0.971567**
auc	**0.839504**

Confusion Matrix:



- This is better than all the above models and produced an F1 score of 0.8
- It trimmed down almost all FPs, with just a slight increase in FNs

Model 5 – AdaBoost:

Fit an adaboost model and used Grid Search CV to tune the hyperparameters.

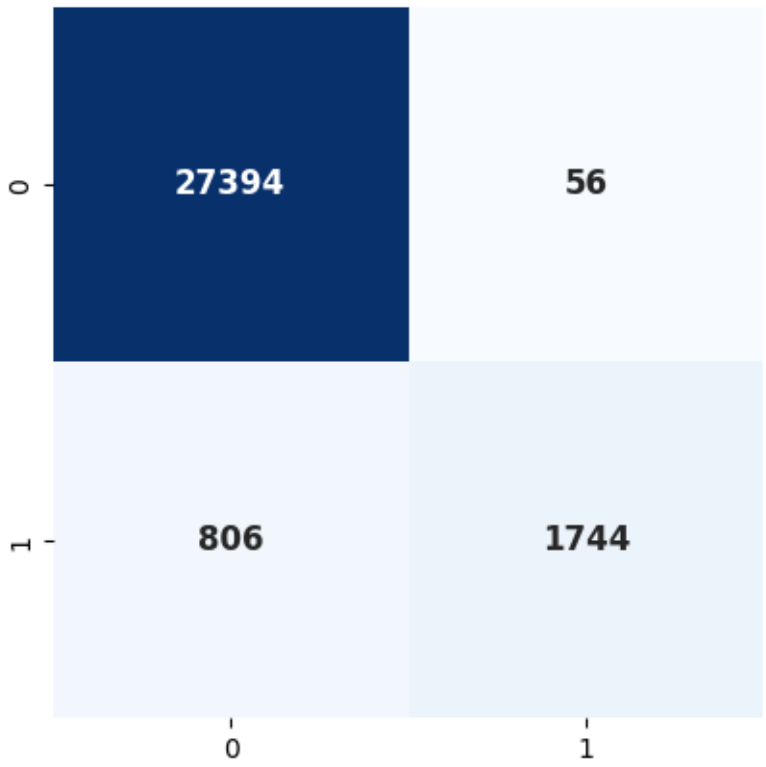
The following combinations were considered.

```python
abc = AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=1), random_state=24)
param_grid = {'n_estimators':[50,100,200], 'algorithm':['SAMME.R','SAMME'], 'learning_rate': [0.1,0.5,1]}
```

The best estimator gave the following results

precision	0.968889
recall	0.683922
fscore	0.801839
accuracy	0.971267
auc	0.840941

Confusion Matrix:



These results are almost identical to the GBM.

Conclusion:

1. The data provided is clean with no missing values or erroneous entries.
2. In the feature engineering step, some of the unimportant features have been removed using domain knowledge and correlation values. Some of the categorical values have been one hot encoded to help in model fitting. All the variables have been scaled to values between 0 and 1 using MinMaxScaler
3. Multiple models have been fit starting from basic models like logistic regression, KNN to ensemble models like random forests and gradient boosting.
4. The feature coefficients from logistic regression revealed the most important determinant of diabetes is `HBA1C - Level`.
5. The best model that's finally selected is the gradient boosting model(model 4), which provided an F1 score of 0.8.

Next Steps:

1. Features can be extracted from clinical notes provided for each patient, which could improve the model results.
2. Other Machine Learning techniques like neural networks can be used to come up with a better predictor