Project Name

University Name

Name: Abed Tabbalat

Professor

Date:

1. Introduction

1.1 Background

I recently found out that a close family member has diabetes. I have never paid close attention to it because it is a very common condition. Once I learned that this could be a genetic passthrough, my curiosity was picqued to learn more about it.

The questions I am aiming to answer are:

- What environmental factors could cause it?
- What variables from the data collected highly impact the outcome of diabetes?

In addition, learning which model best fits certain types of datasets based on what variables it contains is one thing I would want to be excellent with. The ability to look at a dataset and narrowing down which models to use. Of course, this comes with experience and repetition, and this is one of the technical goals that I have.

The main question regarding diabetes would be, what variables are mostly impacting the outcome of having a positive diabetes diagnosis. To add, this topic or healthcare in general has never been my cup of tea, as my interest has always been in the P&C insurance industry. Sometimes life gets in the way, and we forget the importance of health that keeps us alive and well to be able to continue doing what we do. I take that for granted and there is nothing better than learning about it through datasets and predictions. Choosing diabetes in specific, as mentioned above, having a close family member diagnosed with it, while never paid attention to it except for “can’t have sugar” is not enough anymore. This will help me understand finding out what correlates with the outcome what I should monitor for myself and others to avoid

getting it. I believe as common as it is, anyone would be interested in knowing what variables do correlate to be able to monitor themselves and live a better life.

2. Project Design

2.1 Model Types

The plan is to run three different models to see which ones best fit the dataset. The models I am choosing to run are:

3. Random Forest
3. Logistic Regression
3. Decision Tree

The reason I chose the models above is because the dataset shows variables for different people with an outcome variable of ones and zeros that determines if the person is diabetic or not, therefore it is a classification problem which falls under the umbrella of the chosen models. The additional variables that come with it are characteristics of the person that could be triggers on being diabetic. The models will be tested through a confusion matrix to determine the accuracy which will help in deciding which model to go with.

2.2 Results Evaluation

Results will be evaluated through a confusion matrix to determine the accuracy and then an ROC curve to determine how fit the model will be. Accuracy percentage is the key to have a successful model.

2.2 Risks

The main risk I currently see is the dataset is less than 1,000 rows which can impact the model results. The more data and information we have the better the outcome of the model will be.

This risk may not exist depending on how the variables are correlated with each other. In addition, the models I have chosen could result in inaccurate results if overfitting happens which could result in choosing a different approach in predictions.

2.3 Contingency Plan

If the plan does not work out because the analysis shows that there isn't enough data to support the model results, then the plan will be to find a larger dataset within the same topic. That said, having the same topic and possible similar variables should not impact the choice of the models that will be run. If the results are yet to be inaccurate due to the data not having enough features to predict. Finding a different dataset at that time will be a must and starting over. Even though that will be a stretch, a backup plan must be in place.

3. Preliminary Analysis

3.1 Data

The data was downloaded as a CSV file from Kaggle[1], containing 768 rows and 9 columns. Each column is a variable that will be used for the modeling, those variables are the following

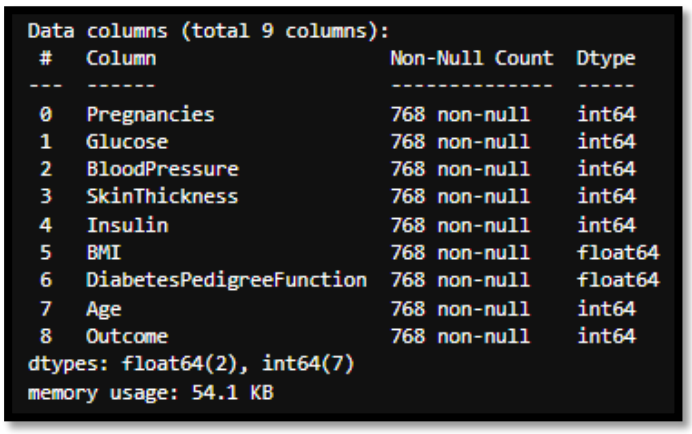
1. **Pregnancies:** Amount of time the patient been pregnant
2. **Glucose:** Patient glucose levels
3. **BloodPressure:** Patient blood pressure
4. **SkinThickness:** Patient skin thickness
5. **Insulin:** Patient insulin level
6. **BMI:** Patient Body Mass Index
7. **DiabetesPedigreeFunction:** Indicates the function which scores the likelihood of diabetes based on family history
8. **Age:** Patient age
9. **Outcome:** 1 for positive diabetes, 0 negative diabetes

3.2 Data Exploration

The main question towards the dataset and the project is whether it is possible to predict the outcome based on the variables available. After some data exploration with the data, I believe predicting the outcome with the variables provided is possible with a chance of error. Finding out how accurate the model would be in predicting the outcome is key.

[1] https://www.kaggle.com/datasets/mathchi/diabetes-data-set

Fig (1) shows the data types of the dataset. The data is all numeric which means applying a regression model would be best as a quick observation.



```
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Fig (1): Data Columns

Going further with data exploration, fig (2) shows exploratory analysis on each variable. Blanks are also checked, as can be depicted in fig (3).

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

Fig (2): Explanatory analysis on variables

```
------- Blanks count -------
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Fig (3): Blanks check

3.3 Data Visualization

Many visualizations came in handy to understand the data better. The target variable is the Outcome column which determines if the patient is positive with diabetes or not. Therefore, the first exploratory visualization, seen in Fig(4), is to compare the split between positive and negative outcomes to diabetes on the dataset. 35% (268 out of 768) of the patients have diabetes and 65% (500 out of 768) as seen in Fig (4).

Fig (4): Exploratory Visualization

If we focus on the pool with positive outcome, we can check the weighted percentages of pregnancies. The bar chart represented in Fig (5) shows the outcome for diabetes for each patient's pregnancy count and we can determine that for positive outcome, 14% (38 out of 268) is the highest count of positive outcome with no pregnancies.

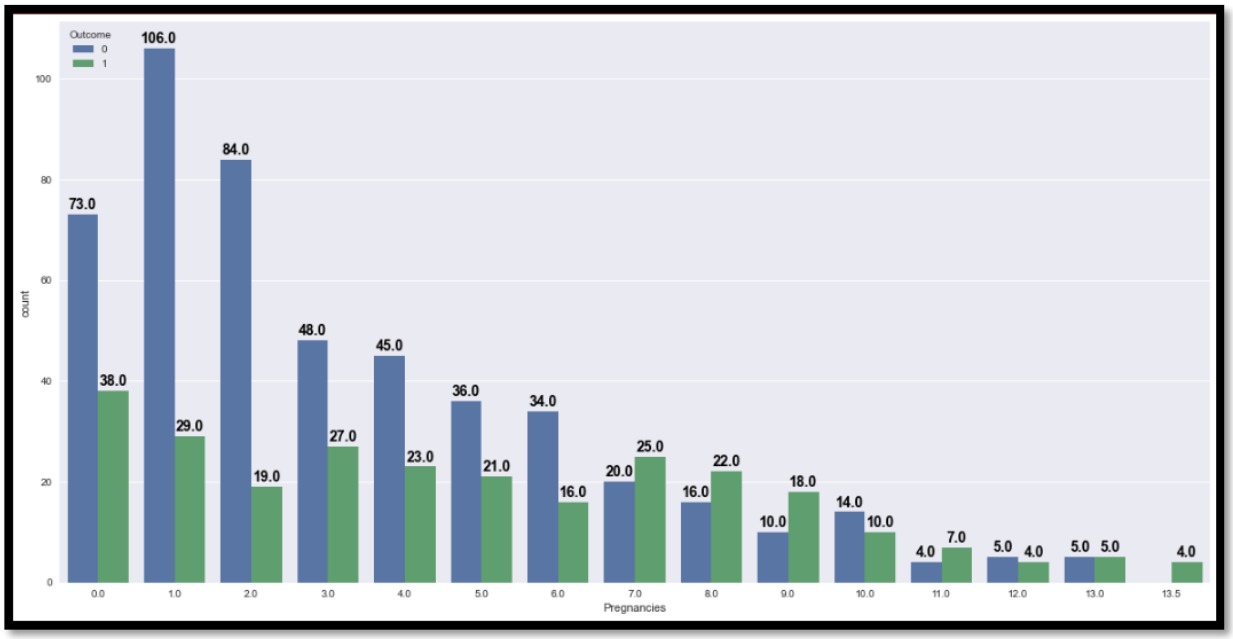


Fig (5): Diabetes outcome for each patient

	Pregnancies	Outcome	Outcome_Pct
0	0.0	38	14.179104
1	1.0	29	10.820896
2	2.0	19	7.089552
3	3.0	27	10.074627
4	4.0	23	8.582090
5	5.0	21	7.835821
6	6.0	16	5.970149
7	7.0	25	9.328358
8	8.0	22	8.208955
9	9.0	18	6.716418
10	10.0	10	3.731343
11	11.0	7	2.611940
12	12.0	4	1.492537
13	13.0	5	1.865672
14	13.5	4	1.492537

Fig (6): Percentage of outcome for each number of pregnancies

Next step was a spearman correlation matrix represented in fig (7), to determine which variables are highly correlated. This could help with the modeling process and the accuracy of the model.

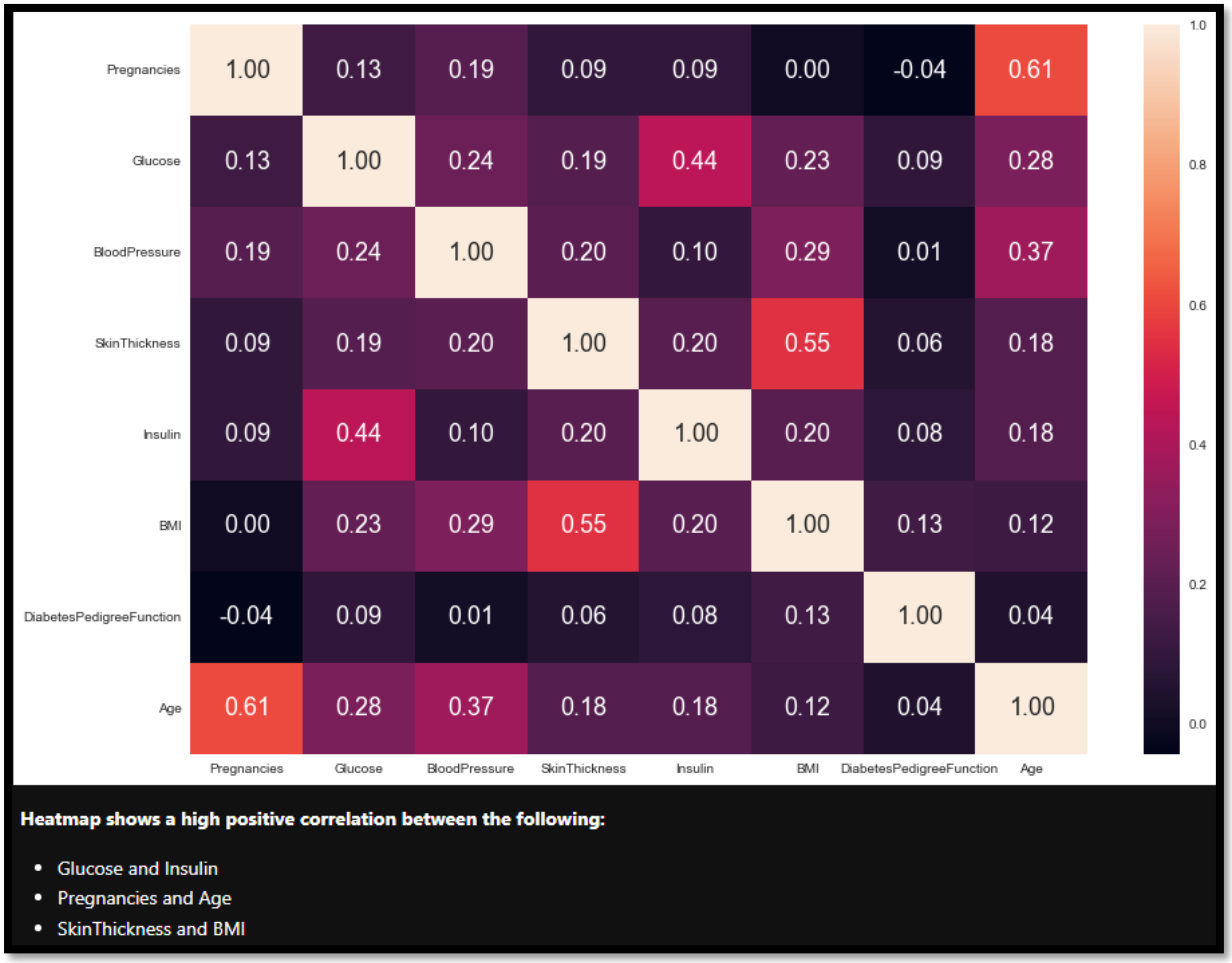


Fig (7): Spearmen Correlation Matrix

Based on general knowledge the results make sense. The higher the glucose levels are the higher the insulin is. The older the patient is the more likely the patient has been through pregnancies. The higher the body mass index, skin thickness would more parallel to it.

Final area to visualize would be running bar charts of the target variable against all other variables which is fig (11).

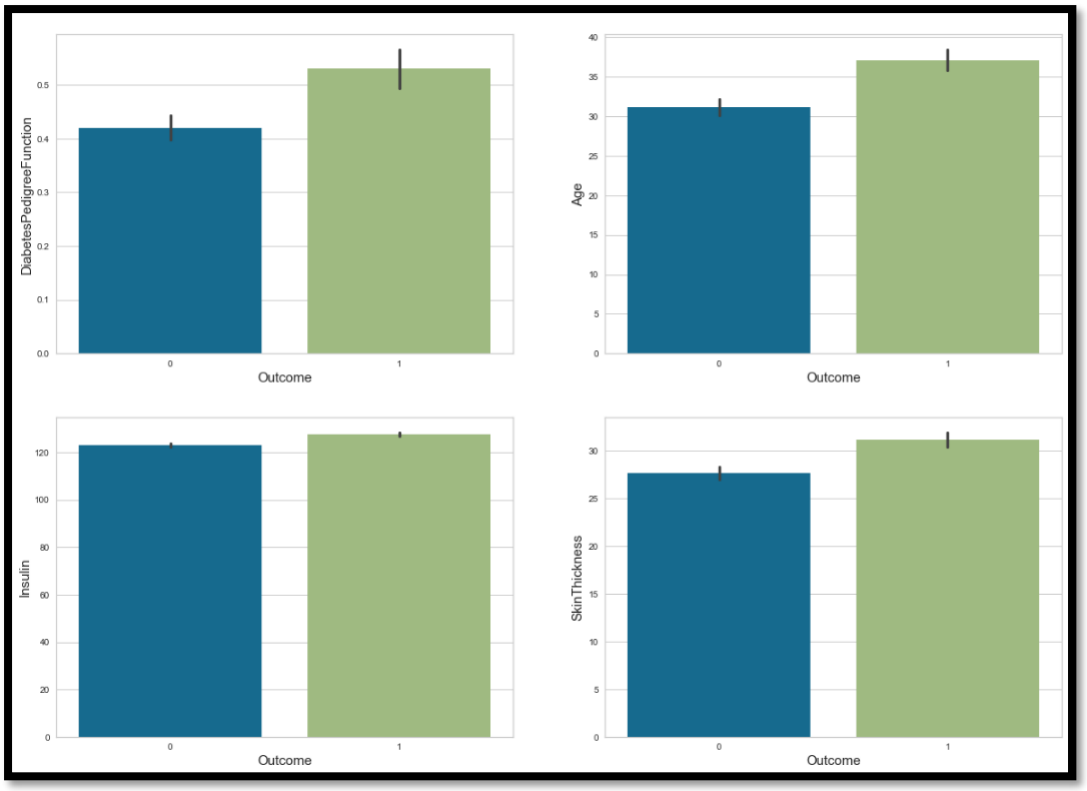


Fig (11): Bar chart of target variable against variables

The bar charts clearly state that patients with positive outcome have higher readings in all the variables than patients with negative outcome.

3. Results Analysis

4.1 Model/ Evaluation Choice Adjustments.

Until this point, there is still no model of choice, as the first step in the modeling process is to determine which model is the best fit. However, after exploring the data, realizing all the variables are numeric, the best choice would be applying a logistic regression to the model. Instead of running each model separately to determine the accuracy, applying a GridSearch could help determine the model of choice quicker.

4.2 Original Expectations

After analyzing the data, and understanding what each variable contributes, my original expectations remain reasonable. The variables that are highly correlated make sense and would be vital in the modeling process. The original expectations are the variables are positively correlated, and therefore, the higher the readings the more likely for the patient to be positive in diabetes. The EDA performed compliments the original expectations.

5. Finalizing Results

5.1 Modeling

Once the data is in shape to go through the modeling process, the steps taken to apply the models would be to split the data, referenced in fig (12). The first step is to break out `Outcome` from the dataset since this is the target variable.

```
# Splitting Outcome (target variable) out of the dataframe
X = df.drop(columns=['Outcome'])
y = df['Outcome']

print(X)
print(y)
```

Fig (12): Splitting the data

As mentioned earlier, the data contains 9 columns and 768 rows, which are not considered to be large, therefore I split the data into 80% for training and 20% for testing, as shown below in fig (13).

```
#Splitting the data into training data and test data
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=60)

print(f'\n----- Shape of the splits -----')
print(f'X_train: {X_train.shape[0]} rows and {X_train.shape[1]} columns')
print(f'X_test: {X_test.shape[0]} rows and {X_test.shape[1]} columns\n')
```

```
----- Shape of the splits -----
X_train: 614 rows and 8 columns
X_test: 154 rows and 8 columns
```

Fig (13): Splitting the data into training and testing.

The following steps have been taken to show how the modeling process has been handled:

- Fitting the model to the training set
- Predicting the model on the test set
- Applying a confusion matrix for accuracy

- Creating a ROC curve to determine the AUC (area under the curve)
- Creating a classification report to determine the F1 score
- Appending results to the final table

Fig (14-17) show the code example doing the process above:

Logistic Regression (All Data)

```
# Fitting and predicting using Logistic Regression
lr = LogisticRegression(max_iter=500)
lr.fit(X_train, y_train)
lr_test = lr.predict(X_test)
```

```
plt.figure(figsize=(15,10))

# Computing results of Linear Regression
cm_lr = ConfusionMatrix(lr, encoder={0 : 'Negative Outcome', 1: 'Positive Outcome'})
cm_lr.fit(X_test, y_test)

# Computing accuracy of the model
lr_accuracy = round(cm_lr.score(X_test, y_test) * 100, 2)
print(f'\nAccuracy: {lr_accuracy}%\n')

# Confusion matrix vizualization
for l in cm_lr.ax.texts:
    l.set_size(30)
cm_lr.show()
```

Fig (14): Logistic Regression (All Data)

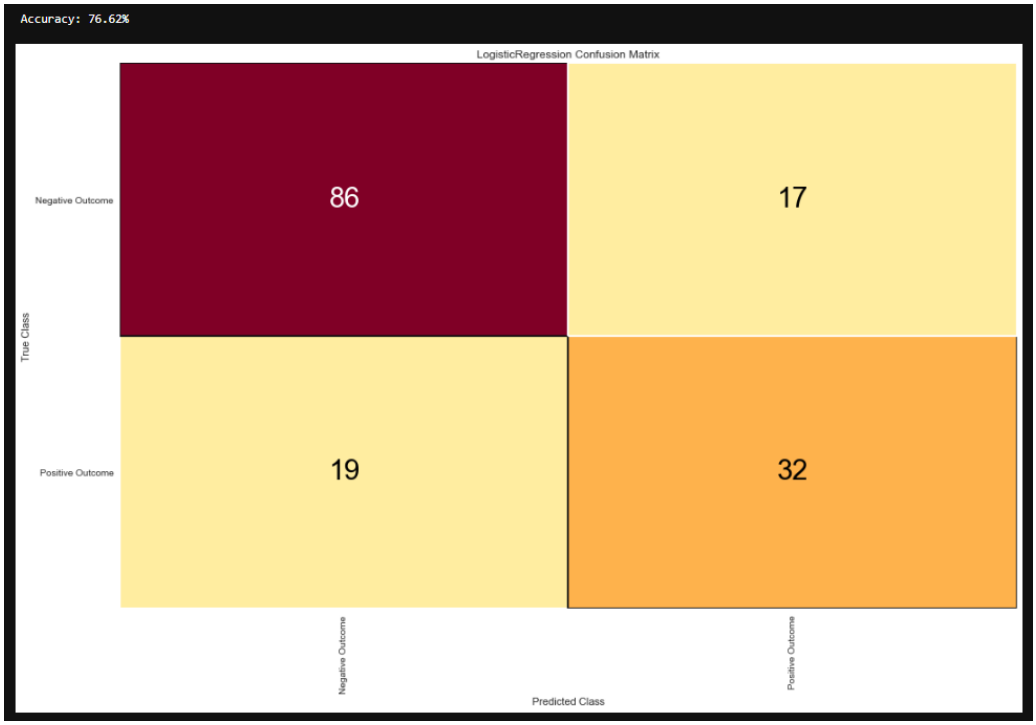


Fig (15): Confusion Matrix on Logistic Regression (All Data)

```
plt.figure(figsize=(25,15))
# ROC curve and AUC score
lr_roc = ROCAUC(lr, encoder={0 : 'Negative Outcome', 1: 'Positive Outcome'})
lr_roc.fit(X_test, y_test)

lr_roc.score(X_test, y_test)
lr_roc.show()
```

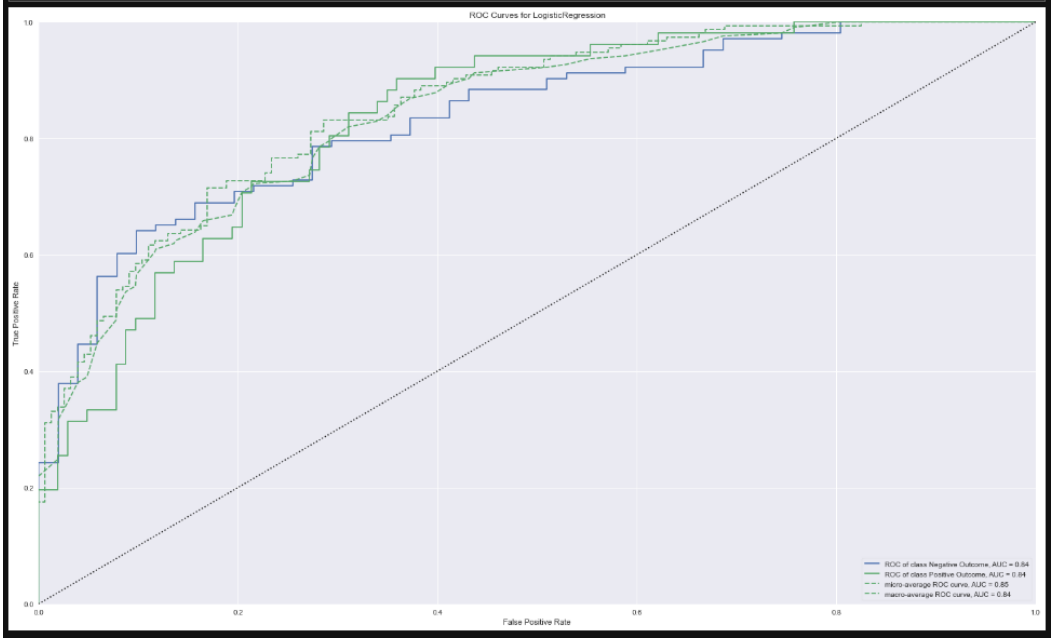


Fig (16): ROC curve for Logistic Regression (All Data)

```
plt.figure(figsize=(25,15))
# Classification Report
lr_cr = ClassificationReport(lr, encoder={0 : 'Negative Outcome', 1: 'Positive Outcome'})
lr_cr.fit(X_train, y_test)
lr_cr.score(X_test, y_test)

# Confusion matrix vizualization
for l in lr_cr.ax.texts:
    l.set_size(30)
lr_cr.show()
```

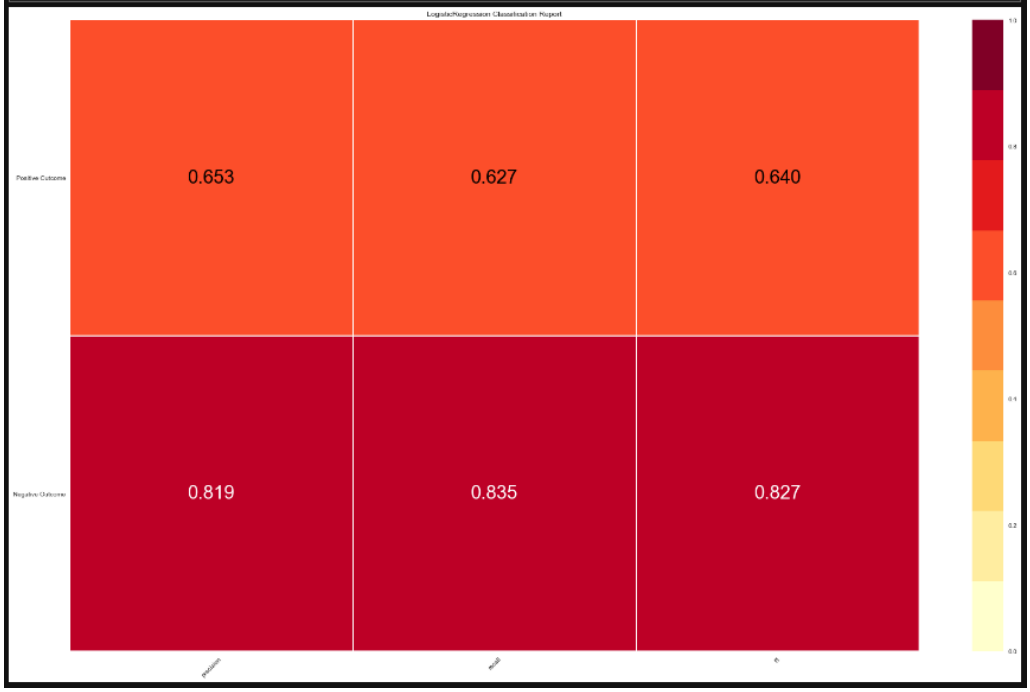


Fig (17): Classification report for Logistic Regression (All Data)

5.2 Results Interpretation

The process has been applied to Logistic Regression, Random Forest Classifier, and Decision Tree Classifier through 2 phases. The first phase was applying the models to all the variables within the dataset, I used `OG` to identify the models applied to all the variables. The second phase was applying the models to the highly correlated data that is shown in the EDA completed above with spearman correlation, I used HC to identify the models applied to the highly correlated variables only. The results are shown in Fig (18-19).

	Model	Accuracy	F1_Positive	F1_Negative	AUC
1	Random_Forest_OG	78.57	0.697	0.834	0.84
0	Logistic_Regression_OG	76.62	0.640	0.827	0.84
3	Logistic_Regression_HC	75.32	0.620	0.817	0.81
4	Random_Forest_HC	74.68	0.642	0.804	0.81
2	Decision_Tree_OG	69.48	0.624	0.743	0.75
5	Decision_Tree_HC	69.48	0.624	0.743	0.75

Fig (18): Results for all models

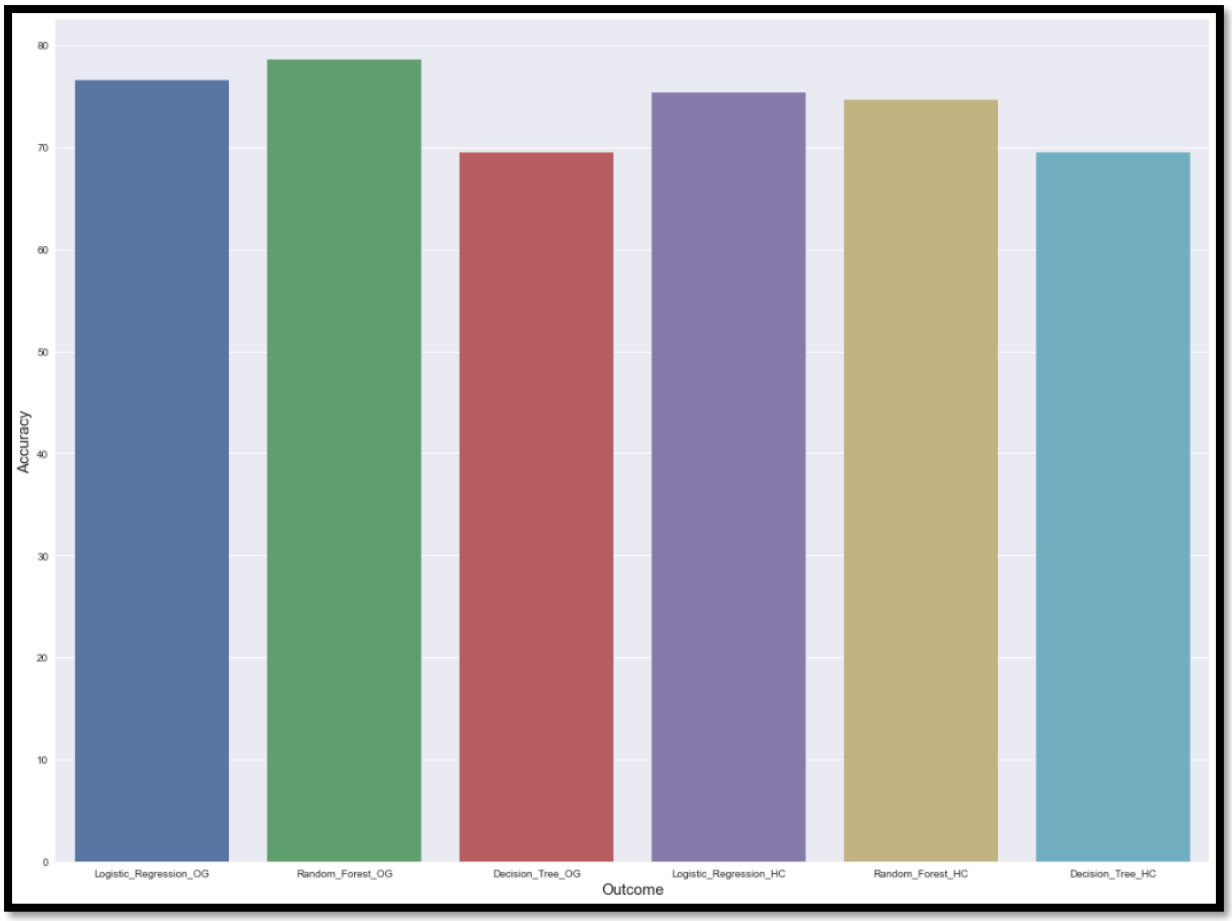


Fig (19): Bar chart showing the accuracy of each model

The results determine that applying random forest to all the data gives the most accuracy, the best F1 score, and the highest AUC score compared to the other models, followed by logistic regression, which gave results similar to random forest. 78.57% accuracy is considered a good model to predict the outcome of diabetes with the features given. I would argue that more detail and other variables could help boost the model's accuracy. The closer the F1 is to 1, the better the model is, and as we can see, `Random_Forest_OG` has the highest F1 score on both positive and negative outcomes. The AUC score is the area under the curve in the ROC curve. Therefore, the higher the AUC the better the performance of the model at distinguishing between the positive and negative classes. The table shows how the model is performing:

Table (1): AUC Score Range

AUC Score Range	Interpretation
0.5-0.7	Poor
0.7-0.8	Acceptable
0.8-0.9	Excellent
>0.9	Outstanding

By looking at the results of the AUC for all the models, we can determine that Random Forest and Logistic Regression are showing excellent discrimination on the data. Decision Tree models are shown to be acceptable.

5.3 Conclusion and Recommendations

Since there is a pregnancy column, we can conclude that the dataset is targeting females only. Therefore, the models are factoring in pregnancies when predicting the outcome. If males to be added to the dataset, then another classification should be added to determine if the patient is a male or a female, which may need to result in retraining the models.

Applying Random Forest or Logistic Regression would be the models of choice based on the results found. Since the results of both are close, I would recommend using them to predict the outcome of diabetes based on the variables given. Both models have an excellent chance of predicting accurate outcomes. If the data can be updated to show more detail, I believe the model can be improved in terms of accuracy, since we determined that applying the models to the highly correlated variables only creates a negative impact on the accuracy of the model.