schuler_5779_ece_527_report_05

October 7, 2024

1 GMU ECE 527 - Computer Exercise #05 - Report

Stewart Schuler - G01395779
20241007

1.1 Exercise 5.1

Data Preparation The first task after importing the *diabetes* dataset is to scale the features to a roughly common scale. This can be done by two different approaches. *MinMax Normalization* which divides the dataset but the largest value bounding the transformed features between 0 and 1. Or by *Standard Normalization* which subtracks the mean and divides by the variance. The result of apply the two different normalization approaches the the dataset is shown in *Figure 1.* It can be seen that some of the features, namely 5 and 7, have a large number of datapoints which fall very far outside the box plots quartile range. In the case of *Standard Normalization* these point being so far away from the feature average means that post transformation the features cover non-insignificant ranges. Because of this, for this dataset *MinMax Normalization* is the better approach to take. For the remainder of this lab the data being applied to the regression models will be *MinMax* normalized.

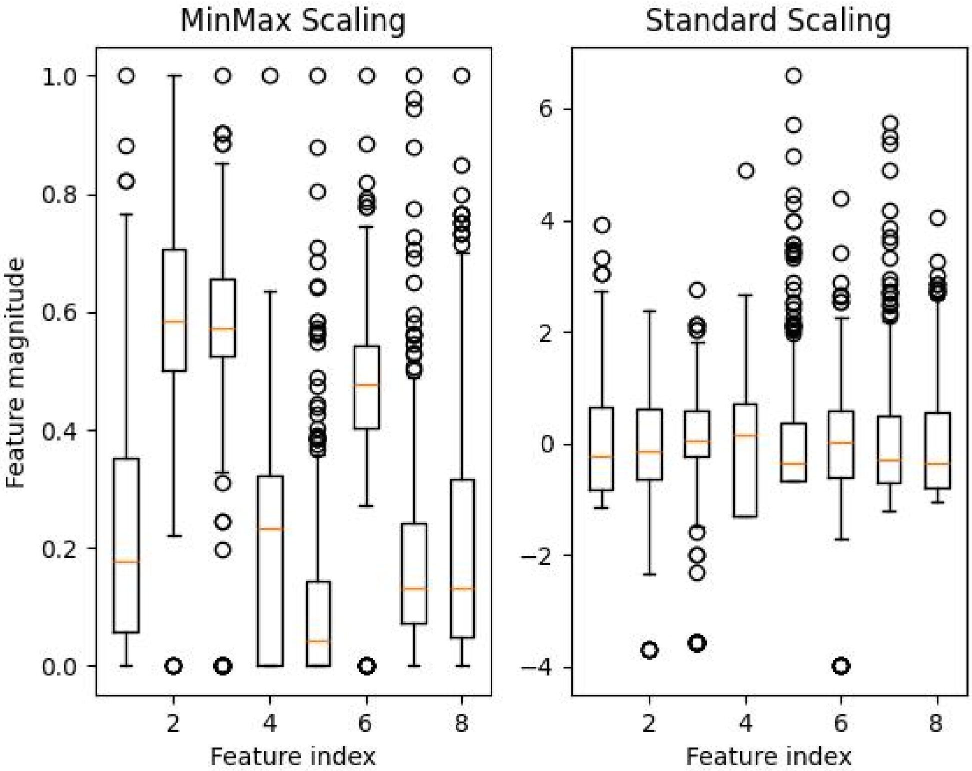

Figure 1. Normalized Dataset Box Plots

Since we are doing regression it is helpful to maintain a separate testset of data to be used to compare the regressors results to. Because we want the test set to be completely isolated from any part of the training procedure, when it comes to normalizing the test set we use the normalization parameters extracted from the training set. In the case of *MinMax Normalization* which we are using, it is possible that a feature value in the test set could be larger or smaller that those in the training set. Which would lead to a value outside of the bounded 0 to 1 range. While not ideal, that must be the case to maintain test set independence.

Experiment 5.4 Next we compute the correlation between all the features and with the desired result *Outcome*. This is presented in tabular for in *Table 1* and as a heatmap in *Figure 2*.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Table 1. Feature Corrleation

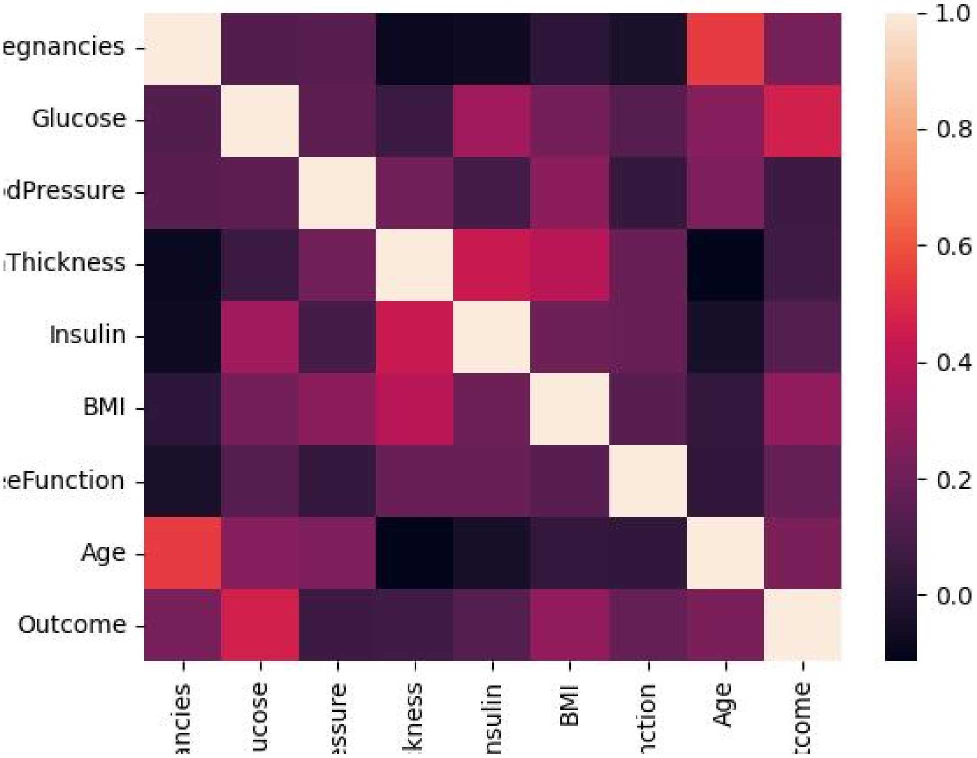

Figure 2. Feature Correlation

From the above correlation we can see that *Age/Pregnancies*, *Insulin/SkinThickness* and *BMI/SkinThickness* have the highest correlation values between features. When correlated the *Outcome*, *Glucose*, *BMI*, and *Age* have the strongest correlation.

Next consider the pair plot in *Figure 3* for the three features identified as having the strongest correlation with *Outcome*. It can be seen that *Glucose* has the strong separation between the two distributions as we would expect given the strongest correlation value. Secondly, the 2D plot of *BMI* and *Glucose* appears to have the strongest separation between feature clusters. That would mean those two features in combination would be a good candidate for a reduced feature space model.

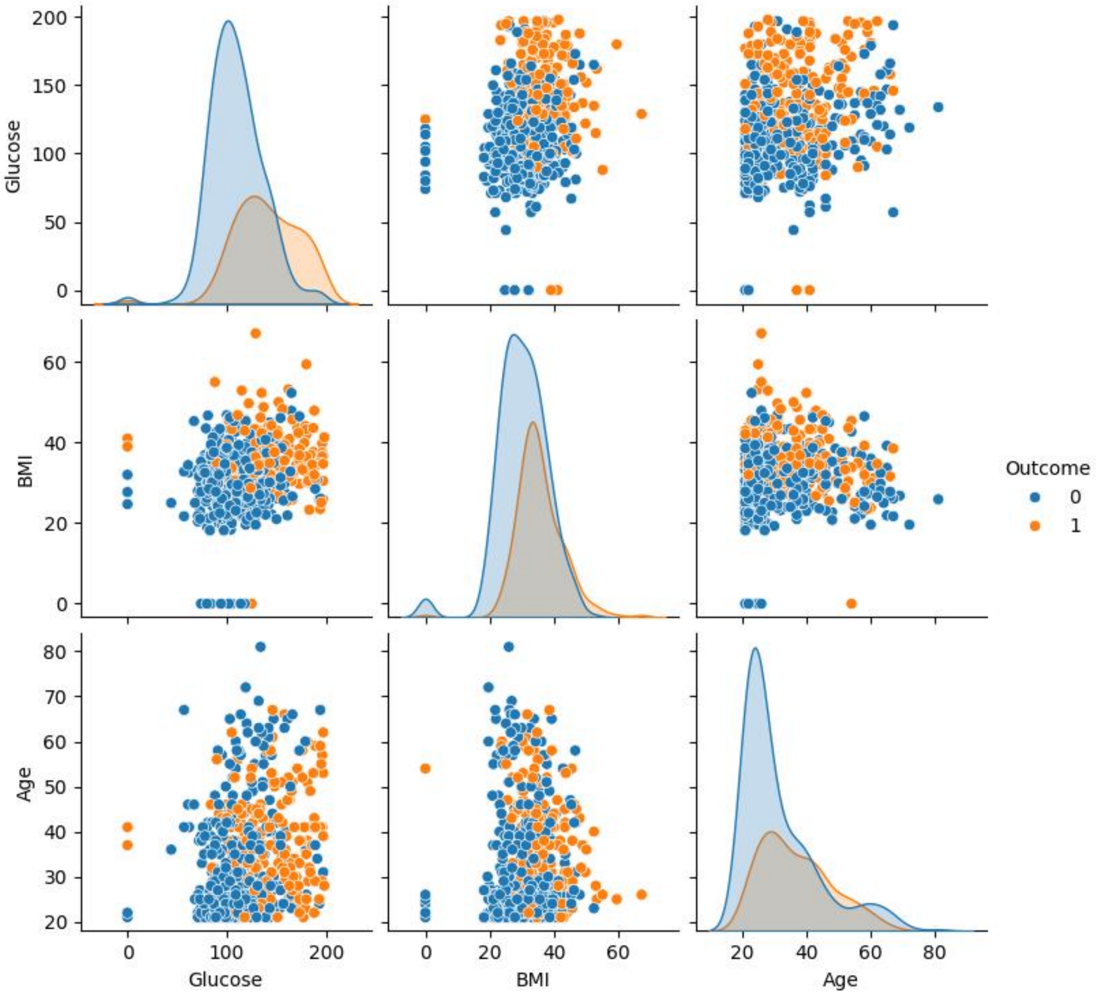

Figure 3. Feature Pair Plot

Experiment 5.5 Next we apply the *LogisticalRegressor scikit-learn* class to the dataset. We initially use all the available features, the result of such a test are shown in *Table 2*. For comparison, included is the results using *Standard Normalization* as well. The results confirm our hunch that *MinMax Normalization* would be better suited for this dataset. However the difference is so small both would likely be acceptable.

Normalization	Accuracy
MinMax	78.57
Standard	77.38

Table 2. Logistical Regression Results - Full Features

Plotting the pseduo-probabilities computed by the classifier in *Figure 4* we can see a trend that the classifier makes predictions with a much higher confidence when guessing an outcome of 0. This can be analytically shown by computing the ratio of high confidence predictions. Computed as follows for the two classes.

$$\frac{\#Pred > 0.8}{\#Pred > 0.5} = 10.0\%$$

$$\frac{\#Pred < 0.2}{\#Pred < 0.5} = 32.0\%$$

Likewise 73% of all predictions fall in this *low confidence* region between 0.2 and 0.8.

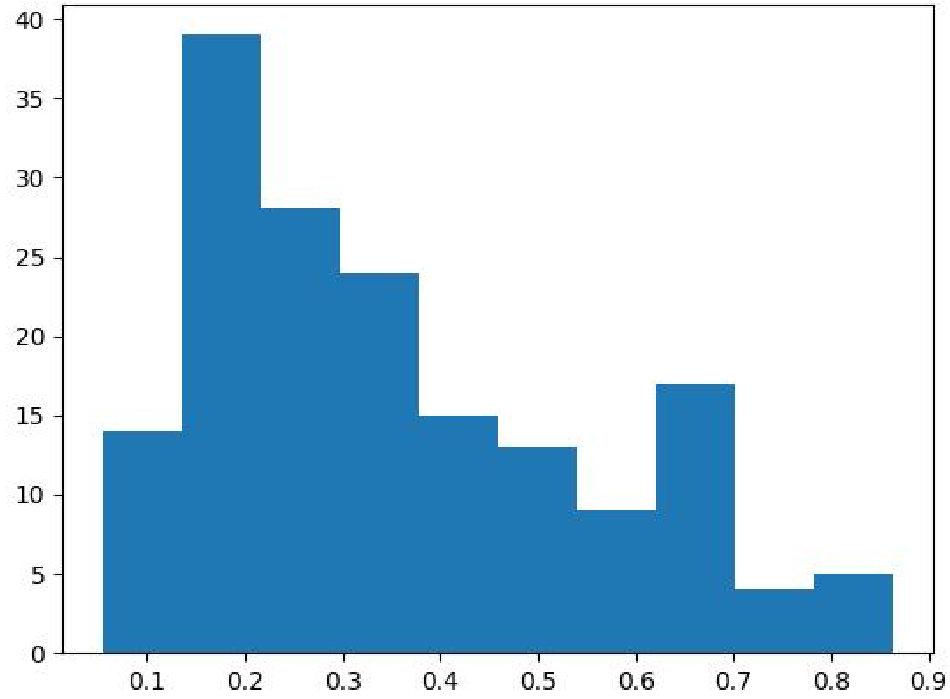

Figure 4. Decision Probability

From these number it can be concluded that the classifier is more confident when predicting an outcome of 0 rather than 1, in the latter case the classifier is not very confident in it's answers. This likely means that the decision boundary being learned by the classifier well encompases the 0 class, but in doing so has a decent ammount of 1 class samples contained in it.

Experiment 5.6 Next we consider the confusion matrix for the classifier. The results it's present appear to line up with our intuition from the *Figure 4*, in that the classifier is more confident about predicting class 0 and less confident about class 1. The can be shown analytically be comaring the inclass accuracies of the two classes. A value of 92.5% for class 0 and only a 53.3% accuracy for class 1.

To interpret the confusion matrix, the sum of each row indicates the number of samples corresponding to the *truth* value for each class. That is there were 108 samples in the test dataset that have an label of 0 and 60 samples that have a label of 1
Likewise the sum of each column represent to predicted labels of the test dataset. That is the classifier predicted 128 samples would be class 0 and only 40 samples for class 1.

It can be clearly seen in *Figure 5* that the confusion matrix is not symmetic. Ideally we would want out classifier to by roughly symmetic, a symmetic matrix means that the classifier performs equivilantly on each of the classes. The imbalance displayed by our matrix indicates it performs significantly better on one class (0) than the other.

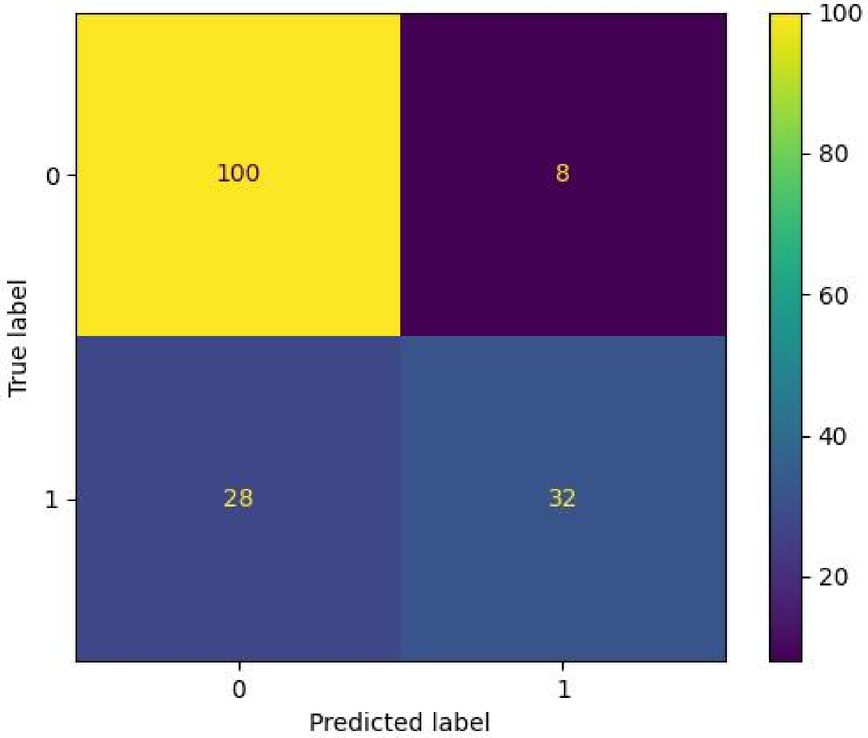

Figure 5. Decision Probability

Next from the confusion matrix values we can compute some additional metics. The *Precision*, *Recall*, and *f1-score* found in *Table 3.*

Metric	Score
Precision	0.80
Recall	0.53
f1-score	0.64

Table 3. Additional Metrics

From this additional discussion of results it can be seen how the raw accuracy value can be misleading about the performance of a classifier. The designed predictor performs well if the goal was only to correctly identify input as class 0, which is the majority class in this dataset leading to an inflated accuracy score. However we have shown that the classifier does a very poor job at correctly predicting samples belonging to class 1, in fact it only slightly out performs a 50-50 guess. Also taking into consideration the medical nature of this dataset, the biggest risk when diagnosing a patient would be a *false negative*, which are abundant in this classifier. From that we can conclude that this classifier is not very good.

Lastly we consider a further advanced metric the *ROC* curve, shown in Figure 6. In the *ROC* curve our classifier with a γ value of 0, lays at the point indicated by the green square. The ideal classifier is a the top left corner of the plot, when *True Positive Rate* is 1, and *False Positive Rate* is 0. With this is mind it make sense to increase the γ value such that the classifier moves closer to that point on the plot.

Also in the *ROC* curve there is the red dashed line, this line indicate a reference classifier for a classifier that chooses 1 for every decision.

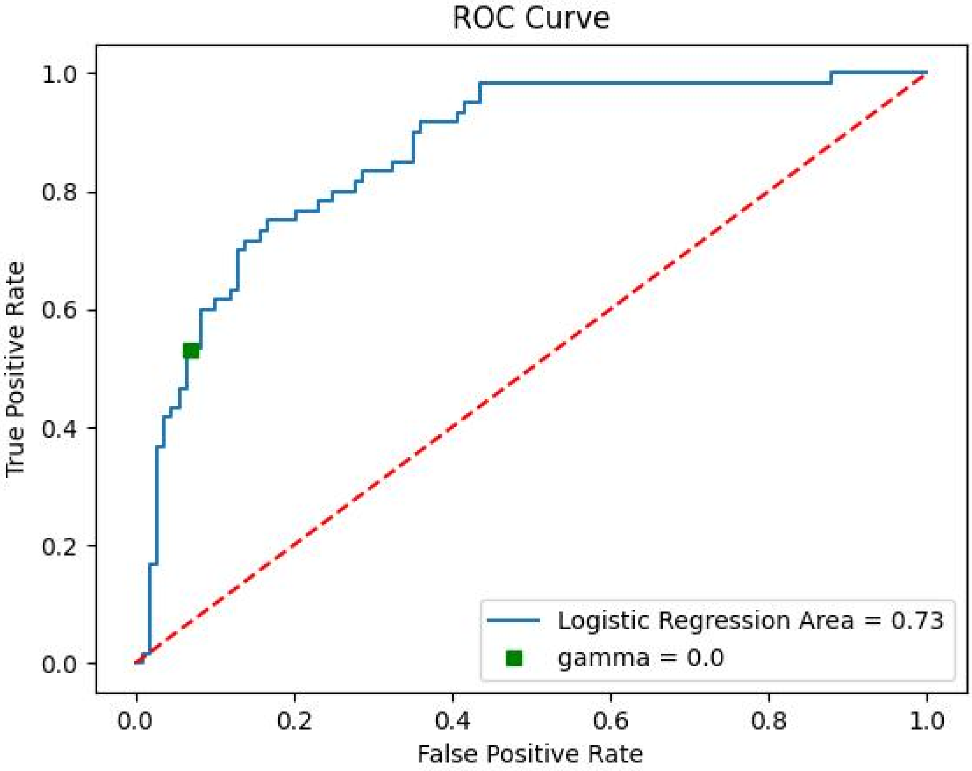

Figure 6. *ROC* Curve

Experiment 5.7 To examine the learned inpact of each feature on the outcomes we can plots the coefficients of the $\mathbf{w}$ vector, as shown in *Figure 7.* From the plot we can see that *Glucose* and *BMI* are the two most significant feature on the outcome. This is inline with what we expect from our previous analysis. Interestingly *BloodPressure* has a negative impact on outcome, that is the higher the blood pressure the more likely the result will be 0.

When comparing two features, *BloodPressure* and *Age*, to determine which feature is more important there are two considerations that must be taken. The first is the absolute value of the $\mathbf{w}$ vector contribution, from that we can be seen that *Age* has a *marginally* higher impact on the result so we might be tempted to say *Age* is the more important feature. The second consideration is that since *BloodPressure* is the only negatively weighted feature it's uniqueness give is more importance. For example, in a normalized random sample with a true label of 1, the contribution of *Age* towards the final result being 1 would be dwarfed by the other features with positive weights. Where as for a sample with a true label of 0, where all the positivly weighted values contribute minimally it is heavily dependent on the *BloodPressure* feature to drive the result down. Therefore, it can be concluded that *BloodPressure* is the more important of the two features.

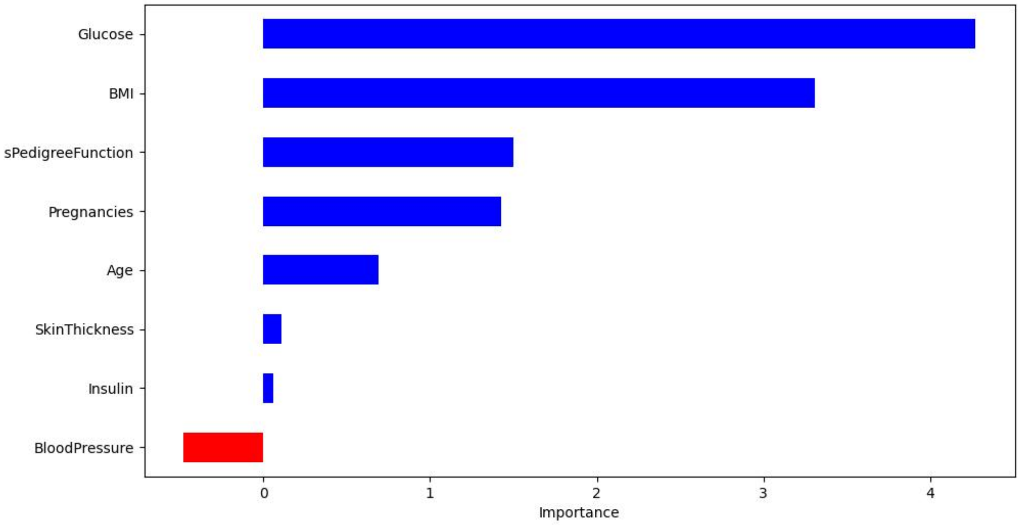

Figure 7. Feature Weights

Experiment 5.8 Next consider a specific input case, defined by the prenomalized values in *Table 4*. For this case the classifier predict a negative diagnosis since the $Pr\{y == 1|\mathbf{x}\} = \mathbf{0.31}$.

Feature	Value
Pregnancies	0
Glucose	130
BloodPressure	125
SkinThickness	30
Insulin	100
BMI	32
DiabetesPedigreeFunction	1.1
Age	25
Predicted Outcome	0.31

Table 4. Specific Case Prediction

Finally we consider the predictor when trained and tested on a single feature at a time. It can be seen in *Figure 8* that those features such as *BMI* and *Glucose* which had stronger weight values (when trained on all features) have a steeper slope and cover the full range of probabilities from 0 to 1. Where as those features which has smaller weight values cover a smaller range with a flatter scope. And Never really approach an outcome probabiltiy of 0 or 1 within a reasonable feature rage.

When trained solely on the *BloodPressure* feature, given the result in *Figure 7*, I would've expected the slope to be negative. However that isn't that case, it does have the flattest slope of all the features is still positive. That must mean for *BloodPressure* to negatively contribute to the outcome it must be coupled with some other feature.

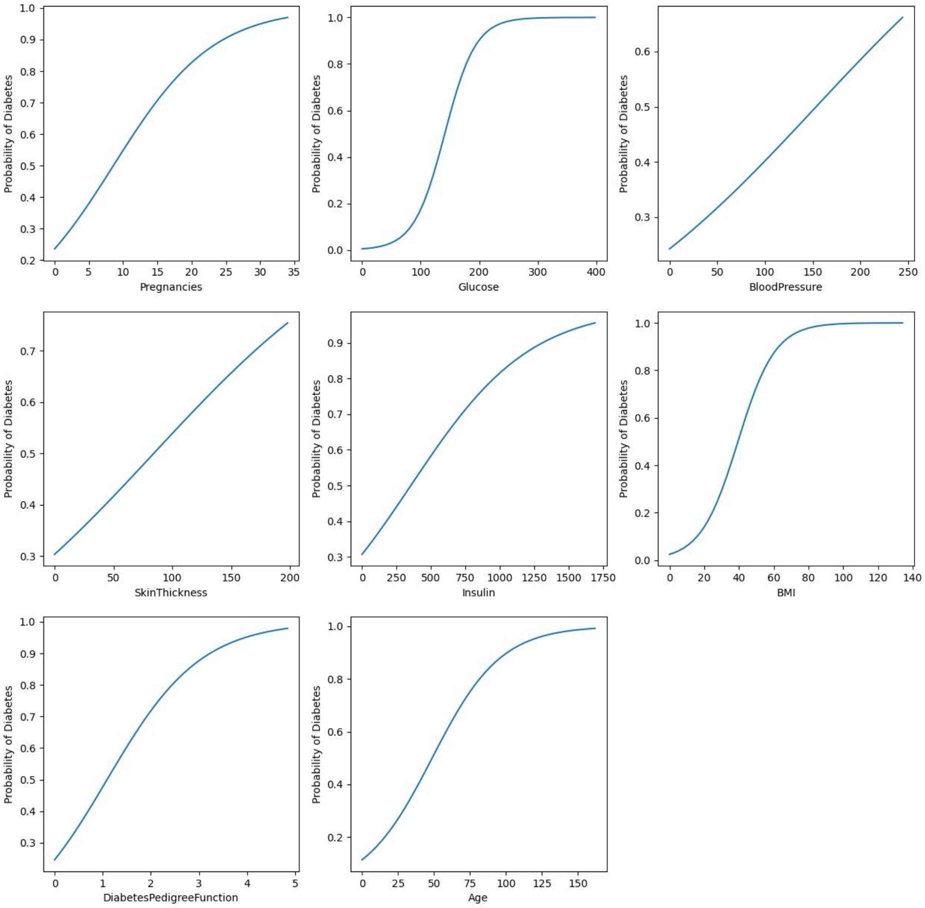

Figure 8. Feature Weights

Extra Credit Consider next the impact of *Age* on specific features. It is known that *Age* is positively correlated with an increased *Outcome* of 1. To visualize how much *Age* modifies this outcome we can plot the probability for a specific feature binned into discrete age ranges. *Figure 9* and *10* show this for *Glucose* and *BMI* respectively.

From *Figure 10* it can be seen how much of an impact *Age* play in the prediction by comparing age ranges at the same *BMI* value. For example, for a *BMI* of 40 there is nearly a 0.4 difference in predicted outcome between the highest and lowest age ranges. Where as in *Figure 9* we see much smaller separation between age ranges for the same value of *Glucose*. Keeping that observation in mind can be a useful dianostic tool.

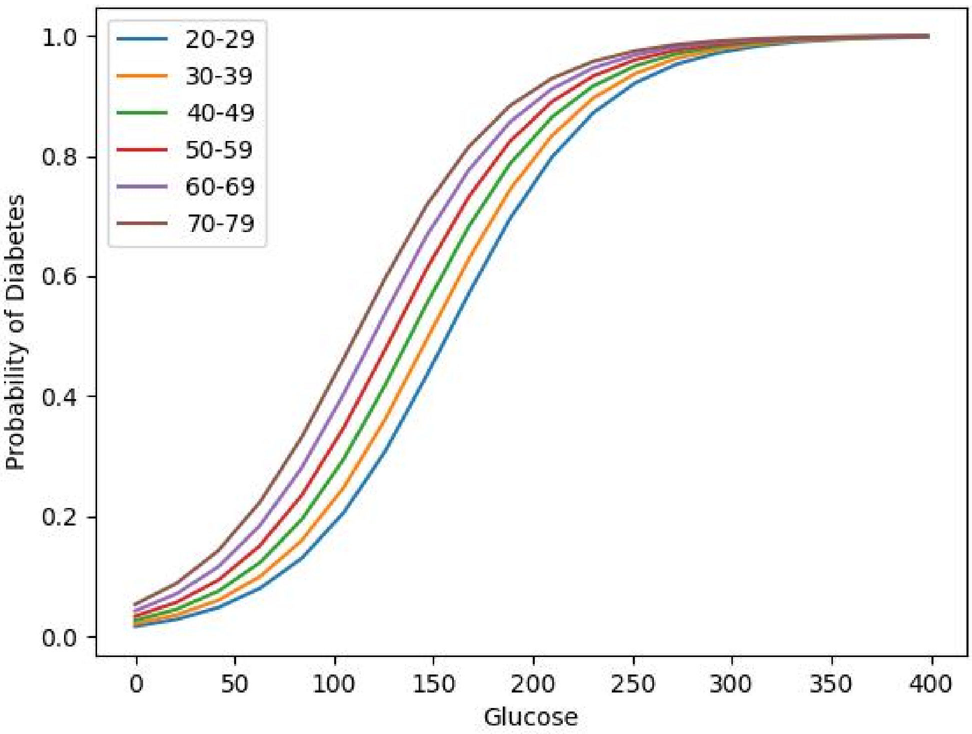

Figure 9. Glucose in Age Bins

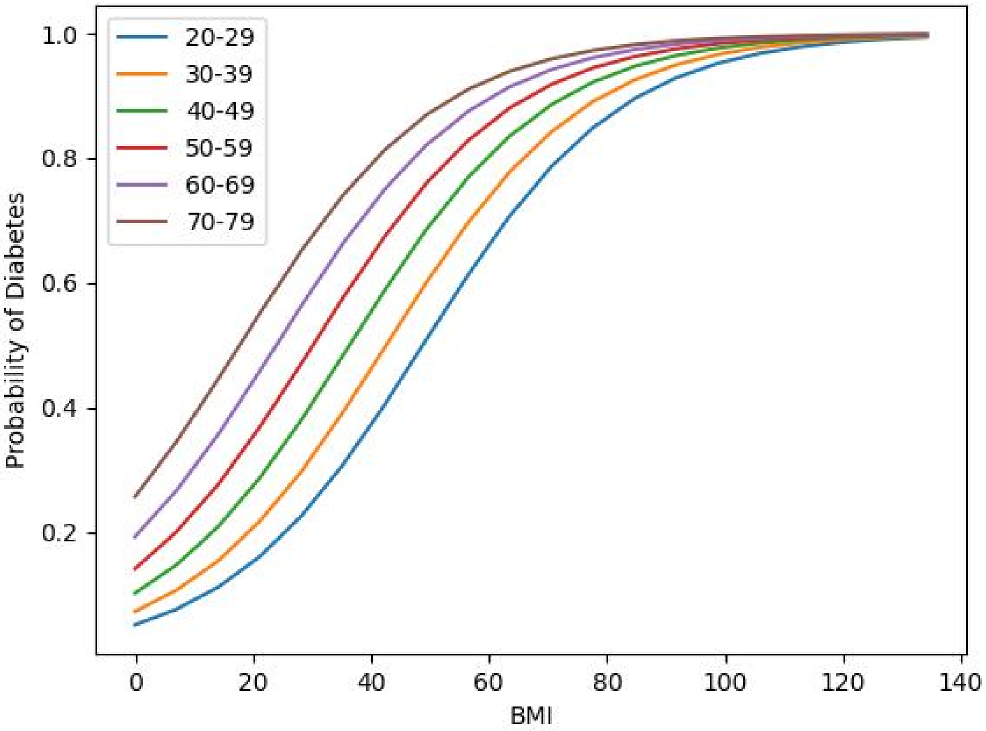

Figure 10. BIM in Age Bins