Capstone Project

Diabetes Prediction

Ruth Caswell Smith

December 1, 2020

Problem Context

Diabetes is a serious disease causing many medical complications throughout a person’s life. Screening for it based on contributing factors could lead to earlier detection and therefore better care. The goal of this project was to use data on Pima Indian women containing information on diabetes risk factors and predict whether or not a woman developed diabetes based on those risk factors. The data was labeled with a binary diagnosis code, and so supervised learning models were used.

Data Wrangling

Data Source

The dataset was downloaded from Kaggle as a CSV file containing nearly 800 instances. The following features were available:

- Pregnancies (integer)
- Glucose (continuous decimal)
- BloodPressure (continuous decimal)
- SkinThickness (continuous decimal)
- Insulin (continuous decimal)
- BMI (continuous decimal)
- DiabetesPedigreeFunction (continuous decimal)
- Age (integer)
- Outcome (binary)

Missing Values

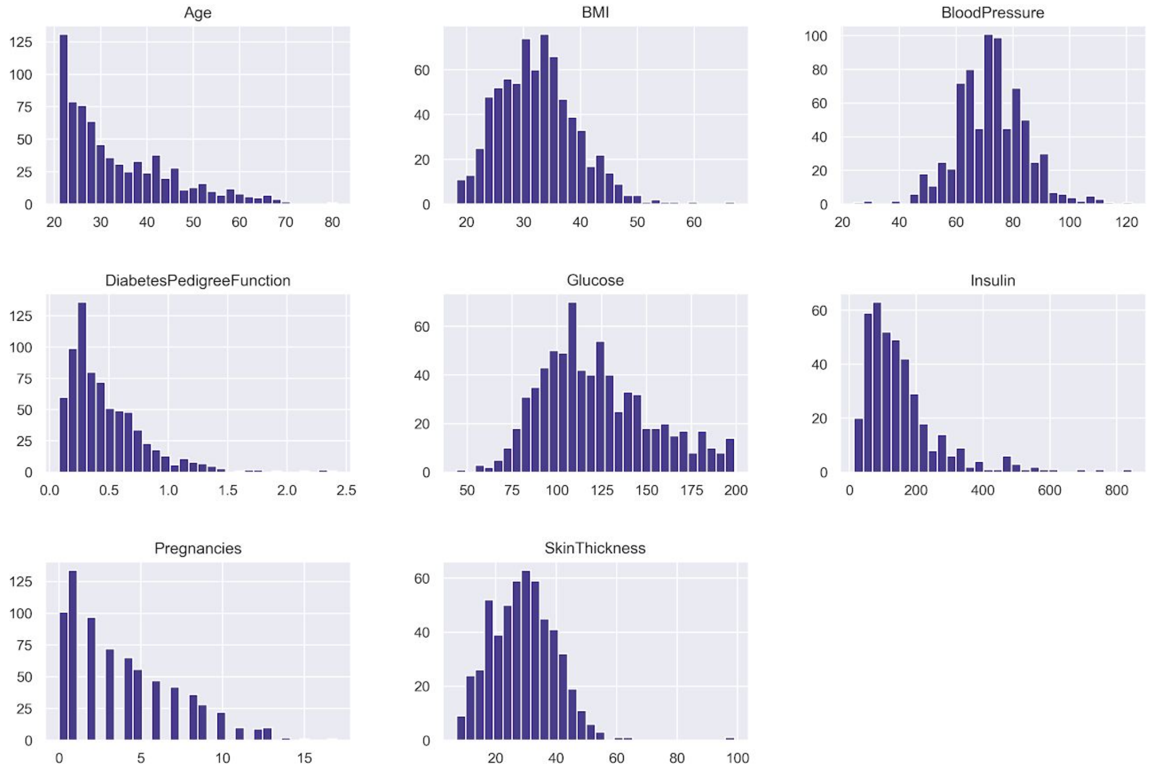
Missing data was encoded as 0, and so it was necessary to be selective about deciding which values of 0 were actually missing data and which were actually 0. This was done based on common sense for the various features.

Less than 5% of the data consisted of instances where 3 or 4 of the features were missing data, and so those instances were dropped from the dataset.

In addition, two of the features, namely SkinThickness and Insulin, were missing a large percentage of values. However, due to the fact that these features could contain valuable information, they were kept for later evaluation.

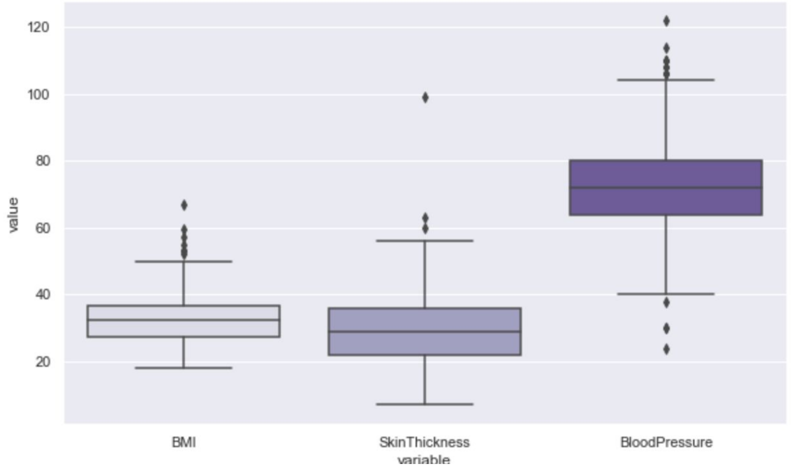
Distributions

Distributions for the various features can be seen in the graph below. Several of these do not appear to be normally distributed, and tend more towards an exponential distribution. This certainly makes sense for features such as age and pregnancies. For some of the other features, a log transform was implemented and they then approximated a normal distribution.



Outliers

Three of the features contained what appeared to be some extreme outliers, as can be seen below.



However, since it was not clear that these values did not represent true values, they were kept in the dataset.

BMI Category

Since BMI is often categorized, the continuous decimal values were binned into the following categories:

< 18.5 underweight

18.5–24.9 normal weight

25.0–29.9 overweight

30.0–34.9 class I obesity

35.0–39.9 class II obesity

≥ 40.0 class III obesity

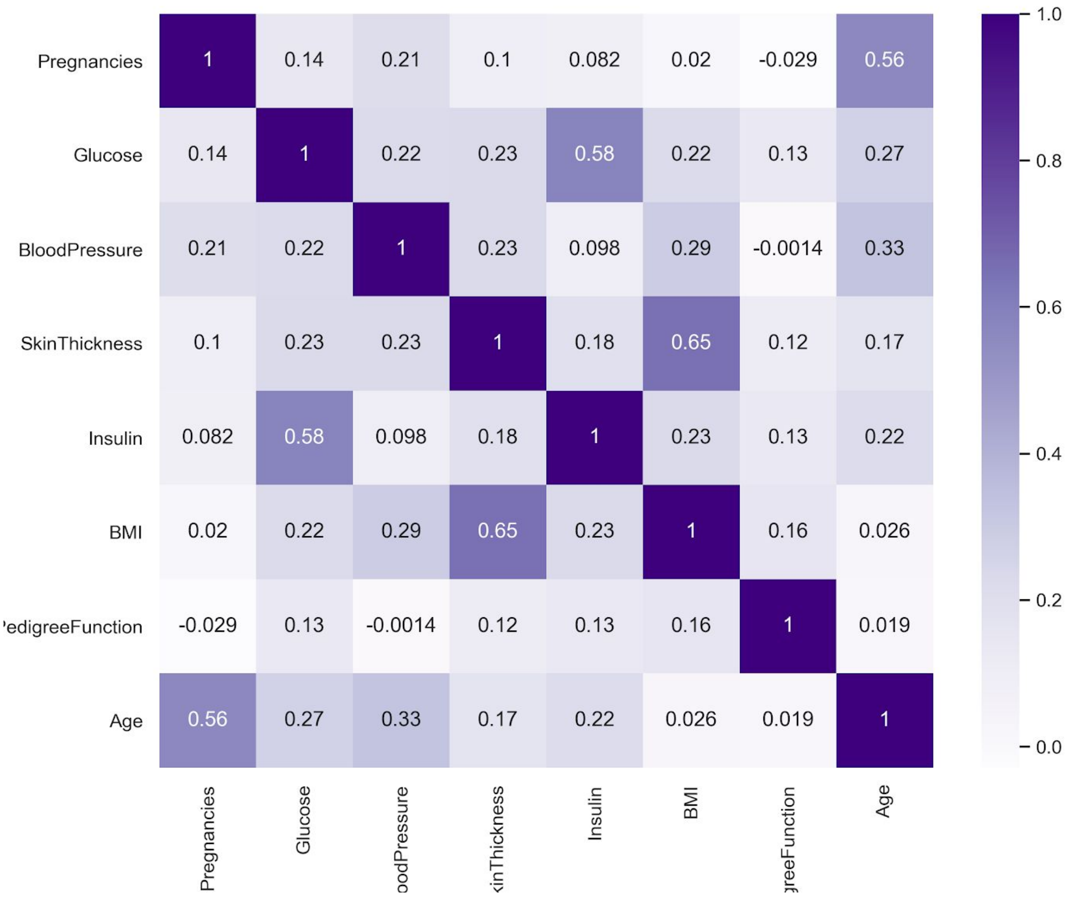
Balance of Dataset

It was noted that the dataset is imbalanced, with approximately ⅓ of the instances having an outcome of 1 and ⅔ having an outcome of 0.

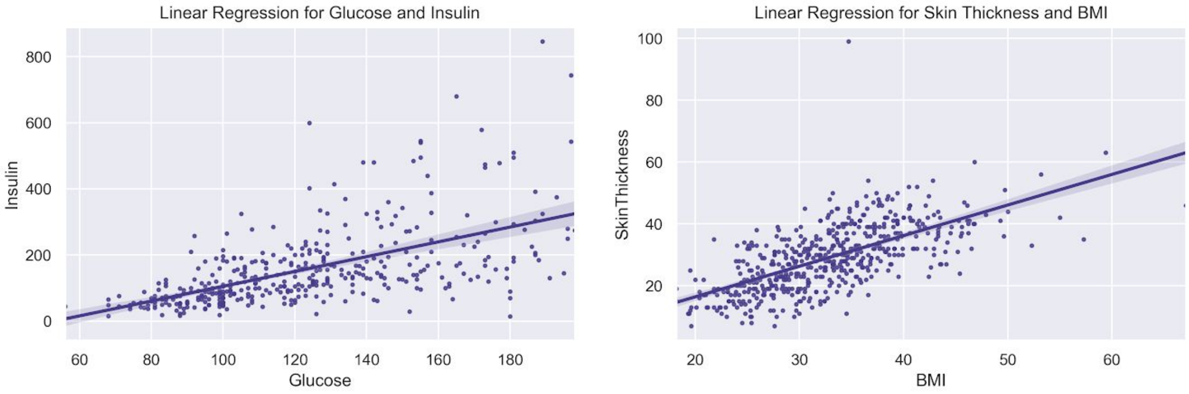
Exploratory Data Analysis

Correlation Between Features

A heatmap showed that none of the variables showed particularly high correlation with each other, which is good,especially for using logistic regression. The ones that were most highly correlated were age and pregnancies, insulin and glucose, and BMI and Skin Thickness.

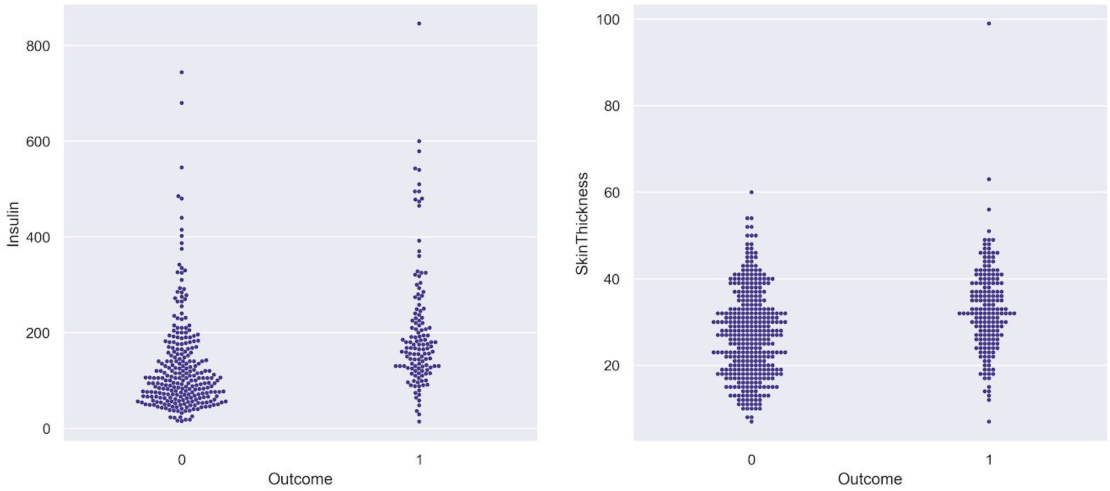


In particular, the correlation between insulin and glucose and skin thickness and BMI were looked at more closely, especially since both insulin and skin thickness had a fair number of missing values.   Linear regression showed a fairly strong correlation, which indicates that it may make sense to not include these two features in the models.



Predictive Power of Features

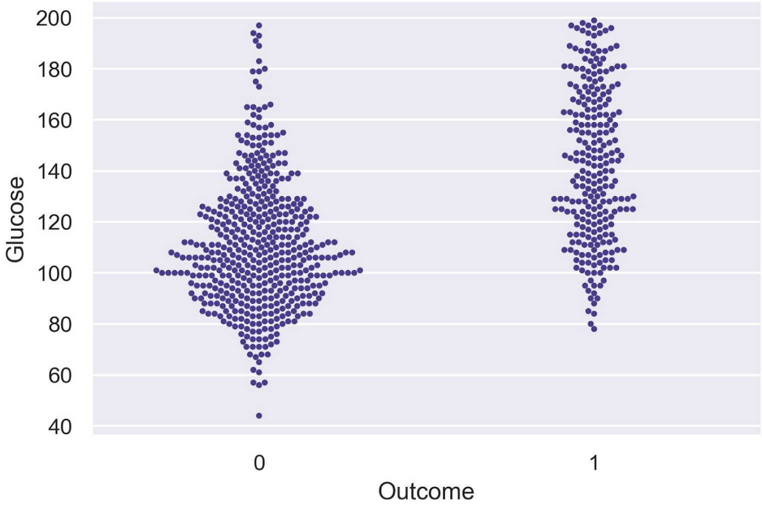
Next, the relationship between the various features and the recorded outcome was explored. This was to gain an understanding of the predictive power of these features. For each of these features, a statistical test was implemented to understand the significance of the feature's influence.



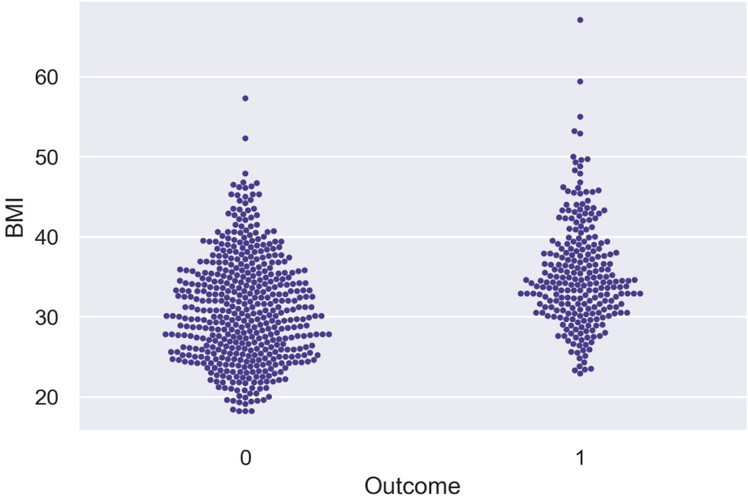
As seen in these swarm plots, it is hard to distinguish visually whether skin thickness and insulin are correlated with the outcome, however both a rank sums test and a permutation test indicated that these features did have some predictive power. For both of these tests, the median was employed as a test statistic in order to minimize the influence of outliers.

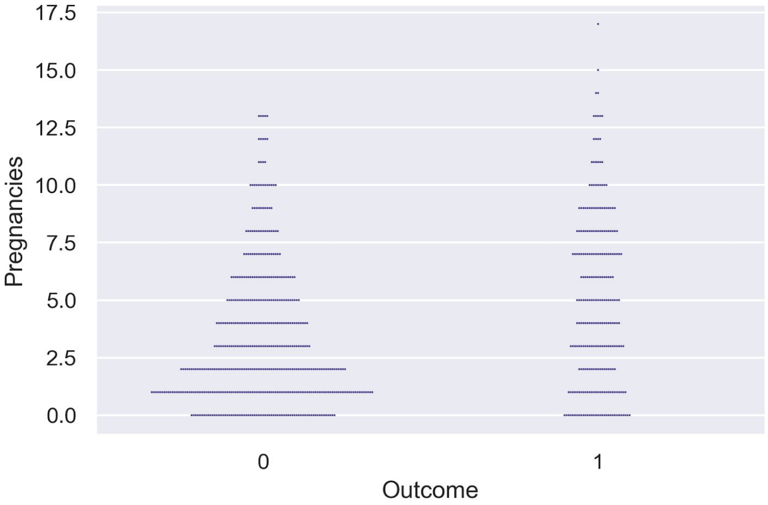
This indicates that it may indeed be useful to include these features in the models, even considering the large percentage of missing values.

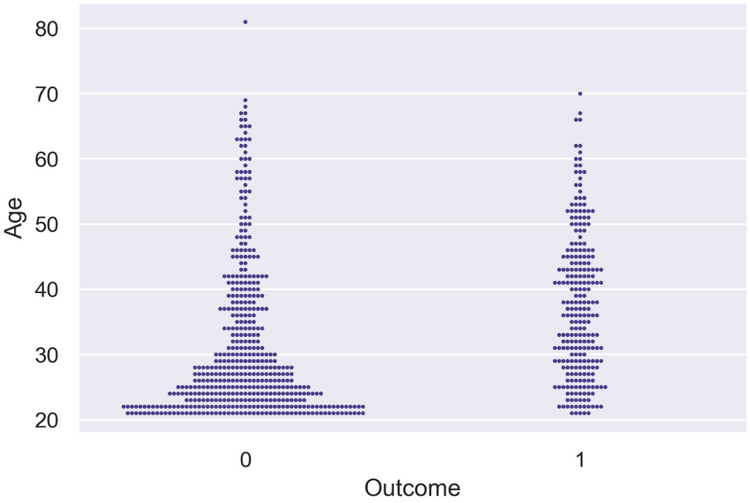
The next feature explored was glucose, and it is apparent even from a visual perusal of the swarm plot that it does indeed influence outcome.   In agreement with this observation, the p-value for the permutation test was very small.

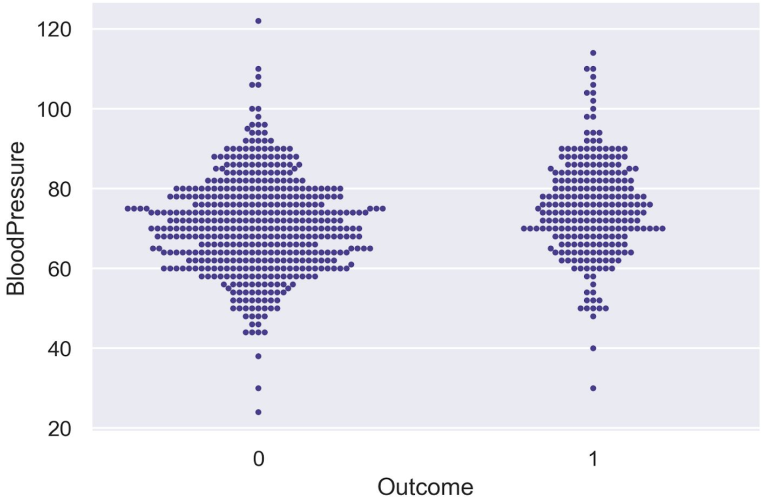


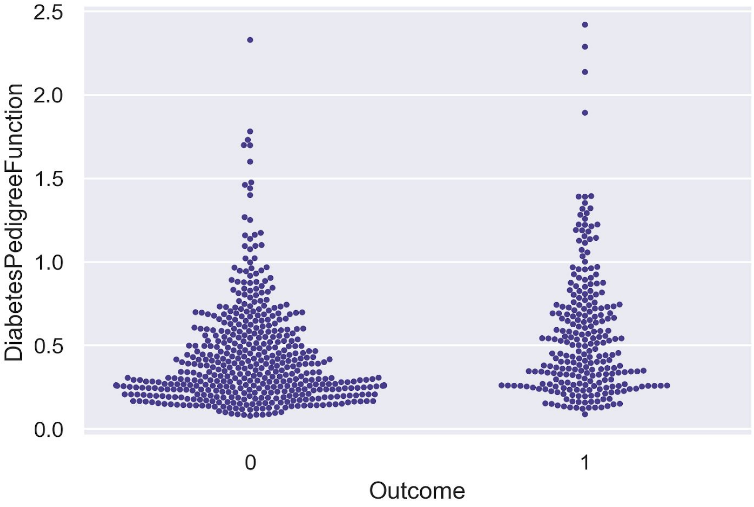
Each of our features was explored in turn, and although it was not always visually apparent from the swarm test, statistical analysis did indeed show that each of the features was significant in terms of determining outcome.



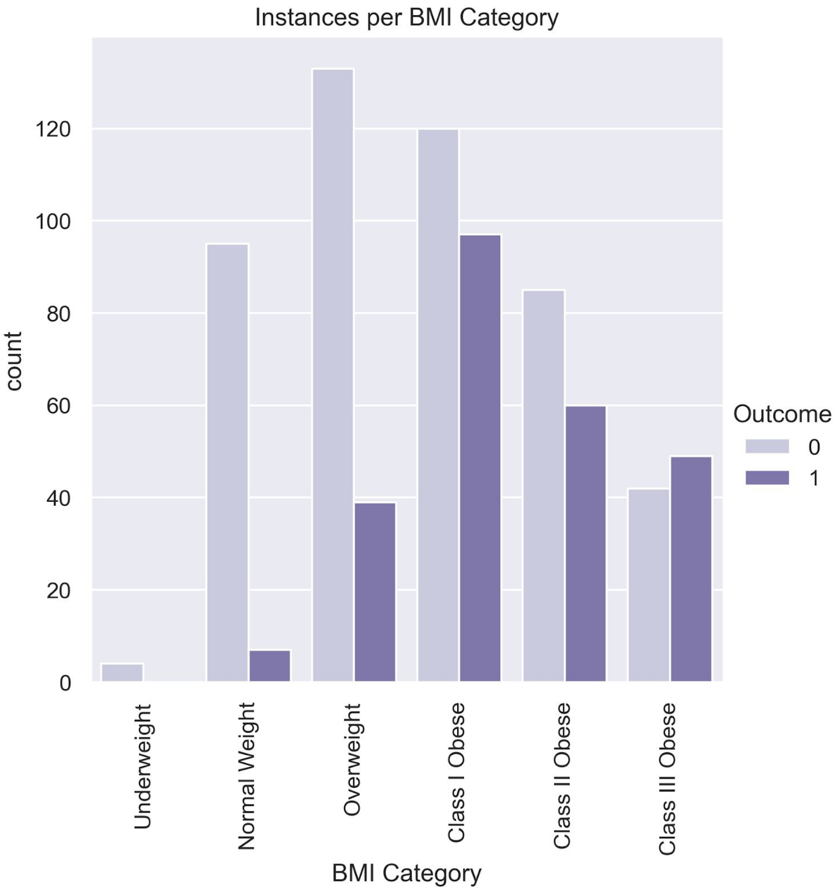

17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Pregnancies
0
1
Outcome


80
70
60
50
40
30
20
Age
0
1
Outcome


120
100
80
60
40
20
BloodPressure
0
1
Outcome

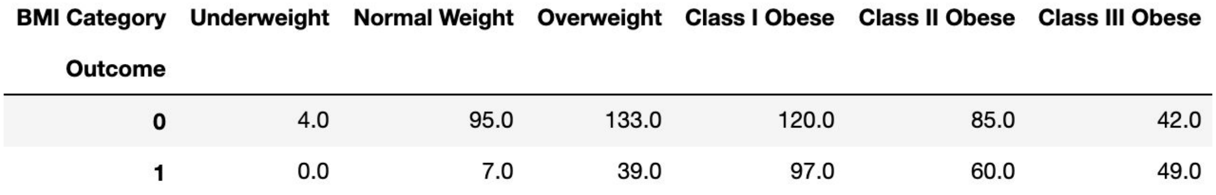

2.5
2.0
1.5
1.0
0.5
0.0
DiabetesPedigreeFunction
0
1
Outcome

Below is a graph illustrating outcome per BMI category. From just a visual inspection it is clear that BMI is a factor in the outcome.



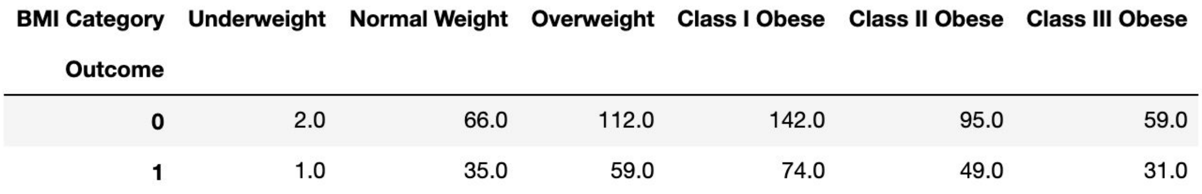
Since the feature we had engineered, BMI Category, was categorical, a chi-squared test was done to determine whether BMI Category would be a good predictor of outcome. Indeed, the chi-squared test showed that the difference between outcome for the various categories was statistically significant.

Our observed instances per category:



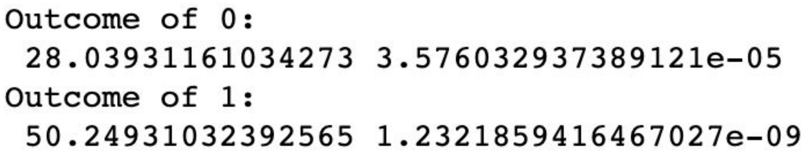
BMI Category	Underweight	Normal Weight	Overweight	Class I Obese	Class II Obese	Class III Obese
Outcome						
0	4.0	95.0	133.0	120.0	85.0	42.0
1	0.0	7.0	39.0	97.0	60.0	49.0

And our expected distribution of instances:



BMI Category	Underweight	Normal Weight	Overweight	Class I Obese	Class II Obese	Class III Obese
Outcome						
0	2.0	66.0	112.0	142.0	95.0	59.0
1	1.0	35.0	59.0	74.0	49.0	31.0

The p-values for both of these outcomes was very small:



```
Outcome of 0:
 28.03931161034273 3.576032937389121e-05
Outcome of 1:
 50.24931032392565 1.2321859416467027e-09
```

Pre-Processing

Note that some of these steps were implemented as part of our pipeline.

Categorical Variable BMI

The variable BMI Category was one-hot-encoded using the pandas method get_dummies with drop_first set to True. This ensured that no duplicate information was added to the dataframe.

Train / Test Split

The dataset was split into a training and testing set, with a random seed set to ensure reproducibility. It was found during this project that the accuracy of the predictive models were sensitive to the train / test split, probably due to the small size of the dataset, so for this reason the same train / test split was used to train all the models.

Scaling

The data was scaled using sklearn’s Standard Scaler. Although technically not necessary for decision tree, random forest, or gradient boost algorithms, it was configured as part of the pipeline and so was used for all classifiers.

Imputation

KNN Imputer was used instead of a simple imputer. The idea behind using this imputer was to use a more discerning scheme than a simple mean or median due to the large percentage of missing values for some of our features. This imputer looks at similar instances and imputes with the corresponding values.   This imputer was tuned with a hyperparameter which determined the number of neighbors to be used.

SMOTE

In order to overcome the imbalance in the dataset, SMOTE was used to oversample the training data and provide a balanced dataset for the model to fit.

Modeling

Pipelines

Initially, an sklearn pipeline was set up with the following steps:

- Standard scaler
- KNN imputer
- Classifier

Once SMOTE was implemented, an imblearn pipeline needed to be used so that SMOTE was applied only to the training data and not to the test data. The steps for this pipeline were:

- Standard scaler
- KNN imputer
- SMOTE
- Classifier

CV Grid Search

Cross-validaiton and hyperparameter optimization were implemented using sklearn's CVGridSearch.

5-fold cross validation was used, and hyperparameters were tuned for the KNN Imputer as well as for the various classifiers.

Feature Selection

Based on our EDA, the first model (which was a KNN Classifier) was trained on just Glucose and BMI Category. However, after adding in the additional features (and imputing those with missing values), it was apparent that better results would be achieved from using all available features, and so this was done for all subsequent models.

Scoring Metrics

There are many metrics that can be used for binary classification problems. In this instance, the two primary metrics used were overall accuracy (defined as the number of correct classifications divided by the total classifications), and recall score (defined as the number of true positives divided by all positives).

Recall score is particularly important for problems such as disease prediction because it is preferable to err on the side of more false positives than false negatives. A recall score of 1 means that every positive case we correctly identified, and so maximizing the recall score is a means of reducing the incidence of false negatives. Due to the imbalanced nature of the dataset, it would be possible to maximize accuracy while allowing a large percentage of false negatives to occur. For this reason, the scoring metric to maximize as specified to GridSearchCV was the recall score.

In addition, the entire confusion matrix was considered for each classifier, so that it was clear how the classifier was performing in terms of false positives and false negatives.

Classifiers

The following types of classifiers were used in the pipeline: KNN, Decision Tree, Random Forest, Gradient Boost, Logistic Regression, and Support Vector. All were tuned with regards to their various hyperparameters.

Once SMOTE was applied, only Random Forest and the Support Vector were trained again.

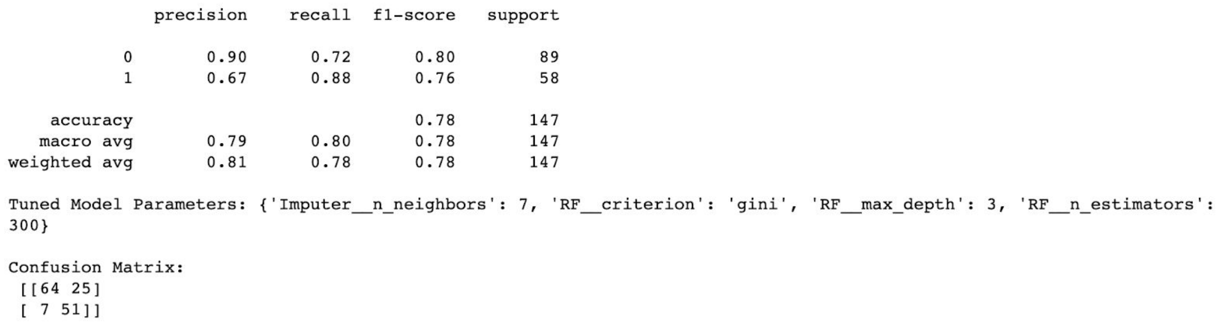
Classifer	**Hyperparameters**	**Best Accuracy Score**	**Best Recall Score**
KNN	nearest neighbors, distance metric	.79	.62
Decision Tree	entropy/gini criterion, max depth of tree	.75	.64
Random Forest	entropy/gini criterion, max depth of trees, number of estimators	.76	.57
Gradient Boost	max depth, number of features, learning rate, and number of estimators	.78	.60
Logistic Regression	Inverse of regularization strength		
Support Vector	trained with rbf kernel and gamma set to 'scale', optimized for cost function	.76	.67
With SMOTE			
Random Forest[1]	same as above	.78	.88
Support Vector	kernel of rbf or poly, gamma of auto or scale, range forInverse of regularization strength	.76	.67
Logistic Regression	same as above but used a log transform on some of the features	.78	.83

[1] Even better recall scores were achieved by using a continuous BMI feature instead of the categorical one; however it was determined that since BMI is usually categorized in the medical world, the model would use that.

Final Model Selection

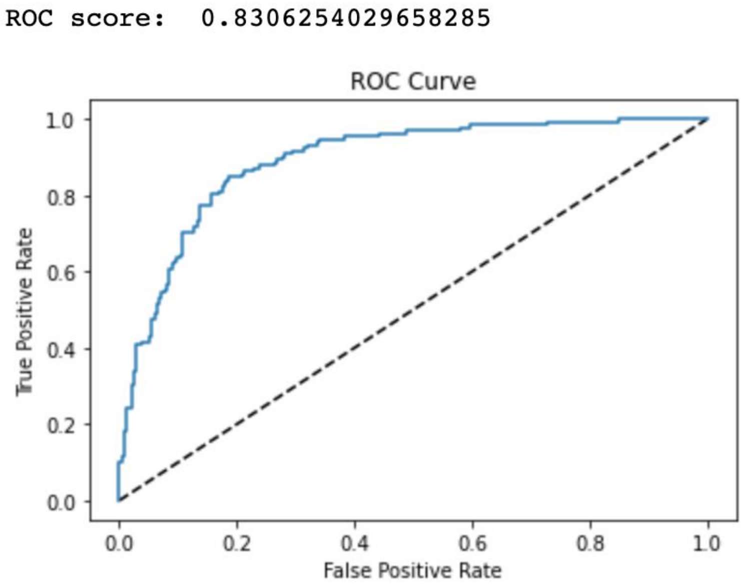
It was difficult to determine the best classifier because there were so many combinations of things to try. In the end, it was determined that the random forest classifier using SMOTE outperformed all other classifiers in terms of maximizing the recall score.

The final parameters and scores for this classifier were:



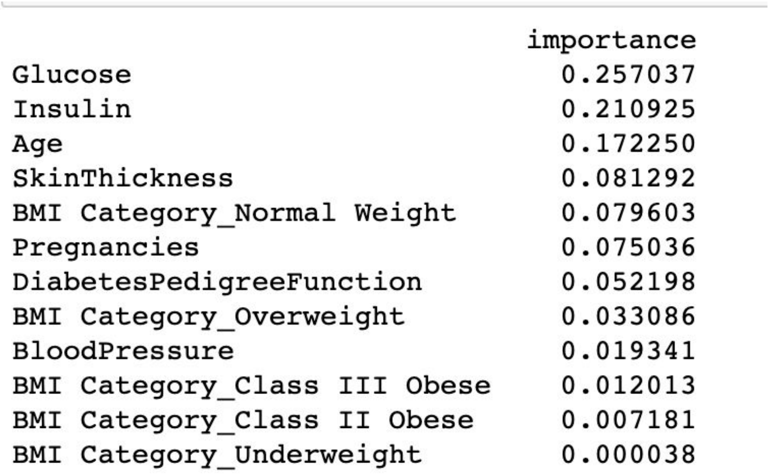
```
              precision    recall  f1-score   support

           0       0.90      0.72      0.80        89
           1       0.67      0.88      0.76        58

    accuracy                           0.78       147
   macro avg       0.79      0.80      0.78       147
weighted avg       0.81      0.78      0.78       147

Tuned Model Parameters: {'Imputer__n_neighbors': 7, 'RF__criterion': 'gini', 'RF__max_depth': 3, 'RF__n_estimators':
300}

Confusion Matrix:
 [[64 25]
 [ 7 51]]
```

This model had an ROC curve and score as shown:



```
ROC score:  0.8306254029658285
```

In addition, it could be determined which features were most important in predicting outcome:



```
                              importance
Glucose                         0.257037
Insulin                         0.210925
Age                             0.172250
SkinThickness                   0.081292
BMI Category_Normal Weight      0.079603
Pregnancies                     0.075036
DiabetesPedigreeFunction        0.052198
BMI Category_Overweight         0.033086
BloodPressure                   0.019341
BMI Category_Class III Obese    0.012013
BMI Category_Class II Obese     0.007181
BMI Category_Underweight        0.000038
```

Conclusions and Final Thoughts

In the end, a predictive model was developed which could accurately forecast the occurrence of diabetes given a set of values. This model was not optimized for accuracy, but was instead, optimized for recall score, so that the incidence of false negatives was minimized. Even so, it performed with an overall accuracy of almost 80% and a recall score of almost 90%,

Several drivers for diabetes were determined, with the top three being glucose levels, age, and skin thickness.

It was interesting to build a pipeline and see how easy it was to implement all of the different classifiers, however it was daunting to see how many possible combinations of things could be tried in order to find the best-performing classifier.

Further work could focus on feature selection in terms of implementing continuous BMI vs categorical BMI, or categorizing Insulin levels as well. In addition, instead of one-hot-encoding BMI, it might be interesting to bin it as an ordinal feature, with values from 1 to 6. It would also be interesting to further explore the role of outliers, and whether treating outliers in a different manner might result in better prediction. Another thing to try would be to compare imputation methods for missing data, and see if the KNN Imputer is actually resulting in better results than a simple imputation using mean or median.

Final model selection is also interesting. Choosing a model that is more intuitive to understand, such as a decision-tree based model or logistic regression, has an advantages in terms of explaining it to your clients. If I were to spend more time on this project, I would try to optimize a logistic regression classifier so that it would be possible to compare predicted probabilities with respect to changes in values features.

Another area of further work could also be to implement primary component analysis and understand the intrinsic dimension of the eight features.