GEN 511: Machine Learning Assignment 1 Report

Daggubati Siri Chandana[1], Soorya Peter[2], and Dhvani katkoria[3]

[1]MS2019005
[2]MS2019020
[3]MS2019007

September 13, 2019

1 Problem Statement

Build a machine learning model to accurately classify whether or not the patients in the data set have diabetes or not.

1.1 Missing data handling

We tried the following ways for handling the missing data:

- Discarding the rows with missing data

  On analysis, we found that we lose up to 11% of the data if we discard the rows with missing values. And as the data set available was small we could not afford that.

- Removing insignificant features

  To check if any insignificant features could be discarded, we used the extra tree classifier as shown below:

```
print("\nranking features: ")
input = data[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']]
input.to_numpy()
model = ExtraTreesClassifier()
model.fit(input, list(data['Outcome']))
# display the relative importance of each attribute
print("\nsignificance of exsisting features:")
print(['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age'])
print(model.feature_importances_)
```

But the output clearly showed that the features were all equally important.
Output:

```
ranking features:

significance of exsisting features:
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
[0.11152084 0.22394677 0.10351914 0.08907296 0.08217969 0.13180858
 0.12081022 0.1371418 ]
```

- Imputing the missing values

  The missing data in the data set was replaced with the median value of each column. Median and mean gave the same values, but we chose median over mean as mean can be affected by the outliers, but not median.

```
Mean values:
 Pregnancies                   3.816722
Glucose                     119.905014
BloodPressure                68.605819
SkinThickness                20.111729
Insulin                      78.359467
BMI                          31.703168
DiabetesPedigreeFunction      0.469499
Age                          33.423353
Outcome                       0.346154
dtype: float64
```

```
Median values:
 Pregnancies                   3.000
Glucose                     117.000
BloodPressure                72.000
SkinThickness                22.000
Insulin                      20.000
BMI                          32.000
DiabetesPedigreeFunction      0.364
Age                          29.000
Outcome                       0.000
dtype: float64
```

1.2 Data Transformation

Data transformation converts data from one format or structure into another format or structure. One of the technique for it is normalization of data. Normalization brings all the columns to the same scale so that the data is not skewed towards large scale numerical values.

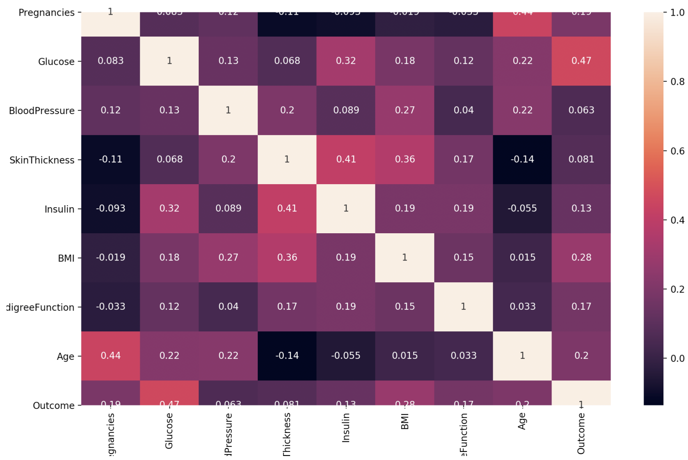
We tried normalizing the data using Min-Max normalization instead of Standard Scaling(making mean zero) and Robust Scaling(making median zero) as the data is not normally distributed.

```
#scaling using min max as we positive values
norm = MinMaxScaler()
norm.fit(train_data)
train_norm= norm.transform(train_data)
test_norm = norm.transform(test_data)
```

1.3 Features Selection

As discussed, the data handling extra tree classifier states that all the features have equal significance and the correlation between any of the features is not greater than 0.54, which tells that the features are not redundant and that all features contribute to the output.

Hence we didn't go for dimensionality reduction and selected all the existing features.



1.4 Model Building & Perfomance Evaluation

For model building we split the data set into 80% train set and 20% test set. We tried the following models and checked with which we get the maximum accuracy.

- Logistic Regression
- Naive Bayesian Classification
- K Nearest Neighbors Classifier
- Stochastic Gradient Descent
- Support Vector Classification
- Random Forest Classifier
- Decision Tree Classifier

We calculated the accuracy and confusion matrix for performance evaluation of all the models and SVC gives the best results in terms of accuracy(80%).