IBM data science professional certificate (Applied data science capstone final report)

Introduction

Diabetes is one of the major problems that put millions of people lives at risk every day. The good knowledge about the diseases could help preventing more people to be affected and will ultimately save a lot of lives. Therefore, It is very important to know what parameters could be related more to diabetes so people can monitor those physiochemical parameters in order to prevent the diseases. Predictive model could anticipate the risk of having diabetes by having large data set of different features that could be related to the diseases. In this project we will use different machine learning classification methods to predict the chance of having diabetes in the near future.

Data preparation

Pima Indian diabetes dataset from UCI machine learning dataset repository will be used to train the classification model, Then we will use that model to predict the unknown data. This data set has data for nearly 800 people and has 8 features in total. The 8 features are physiological attributes that could be used to build our classification model. The names of those 8 features are labeled as A, B, C, D, E, F, G and H in the original CSV file. Therefore, we do not know exactly what they are and we need to assume that user would know what they are when the predictive model will be used. The file contains 768 observations, which is the good size to use to build our predictive model.

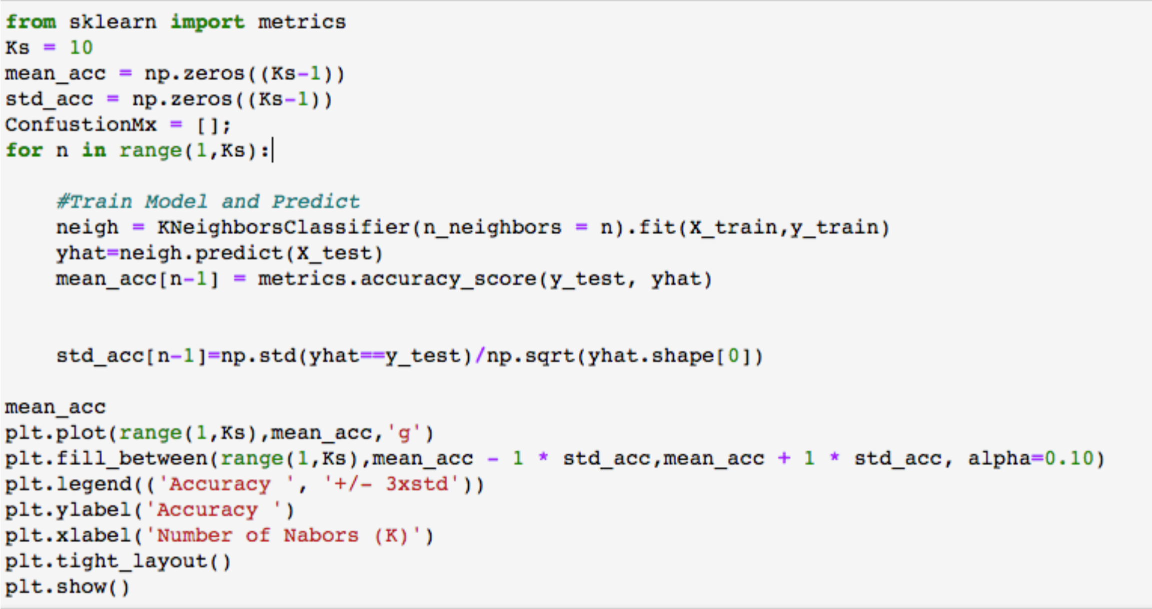
Methodology

We are interested in building the predictive model that will be able to predict if the person has the potential for Diabetes at the moment or in the near future based on measuring some physiological attributes as our inputs. Since we are looking for yes or no answer in this problem, we are dealing with categorical problem. Therefore, classification techniques should be investigated to build a predictive model to predict if the person has potential for Diabetes or not. KNN, decision trees and super vector machine (SVM) are the most common classification techniques, which they will be investigated for our case study. There are 8 features in our CSV file, so all of the 8 features will be used to build our model in order to get better and more robust model.

Results

The three classification models were carried out to build a predictive model, which will be able to predict the chance to have Diabetes based on 8 known most relevant suggested features. Since this is a classification problem, we demonstrated the result of the outcome as 1 and o to represent yes and no for Diabetes respectively.

The first classification method was K nearest neighbor (KNN) method. It is well known that choosing the best K value is critical to have better model and prediction. Therefore, we investigated the accuracy of the model based on multiple K values from 1 to 10. The main part of the code is shown here:



```
from sklearn import metrics
Ks = 10
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfustionMx = [];
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)


    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

mean_acc
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()
```

The result indicated 7 as the best value to build KNN classification model.

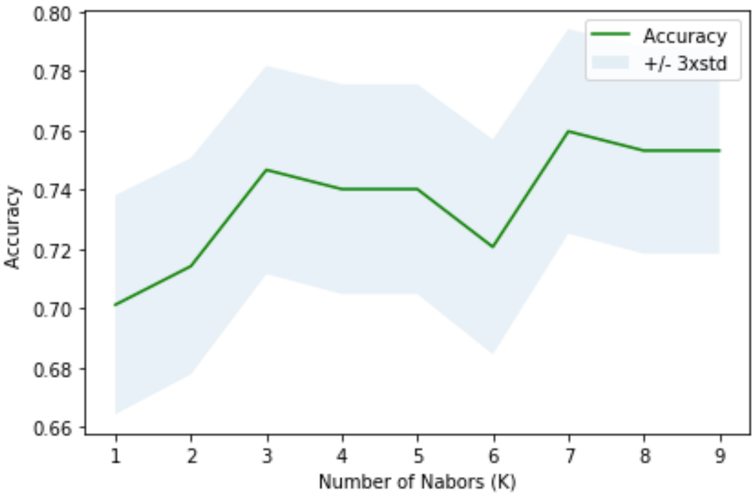
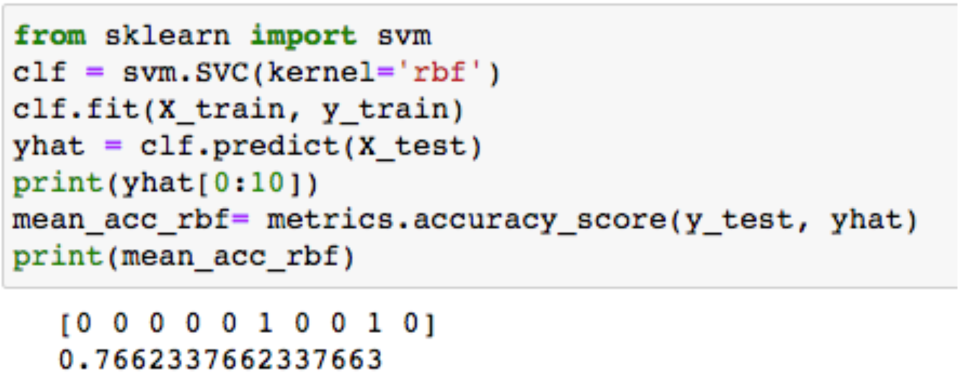


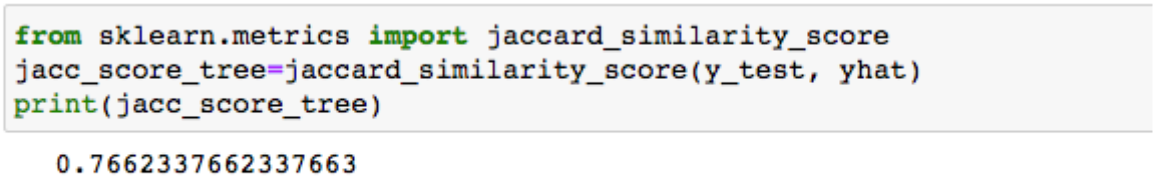
Figure 1. The accuracy plot for different K in KNN algorithm

The next step is to compare this result with support vector machine (SVM) classifier. Radial basis function and linear kernel were used to separately build the SVM model. The results indicate that the linear kernel will result in better accuracy. Therefore, that kernel was used for comparison with the KNN result.



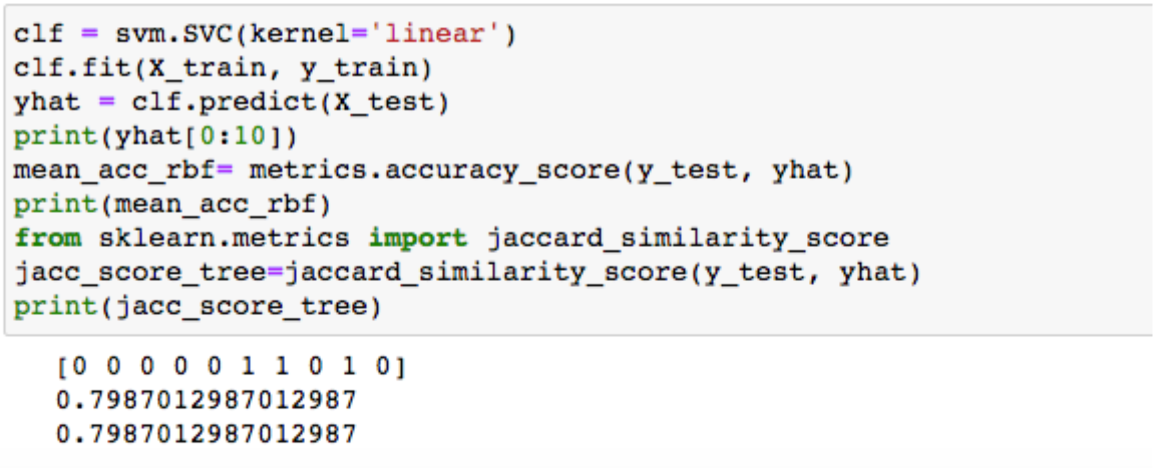
```
from sklearn import svm
clf = svm.SVC(kernel='rbf')
clf.fit(X_train, y_train)
yhat = clf.predict(X_test)
print(yhat[0:10])
mean_acc_rbf= metrics.accuracy_score(y_test, yhat)
print(mean_acc_rbf)
```

```
[0 0 0 0 0 1 0 0 1 0]
0.7662337662337663
```



```
from sklearn.metrics import jaccard_similarity_score
jacc_score_tree=jaccard_similarity_score(y_test, yhat)
print(jacc_score_tree)
```

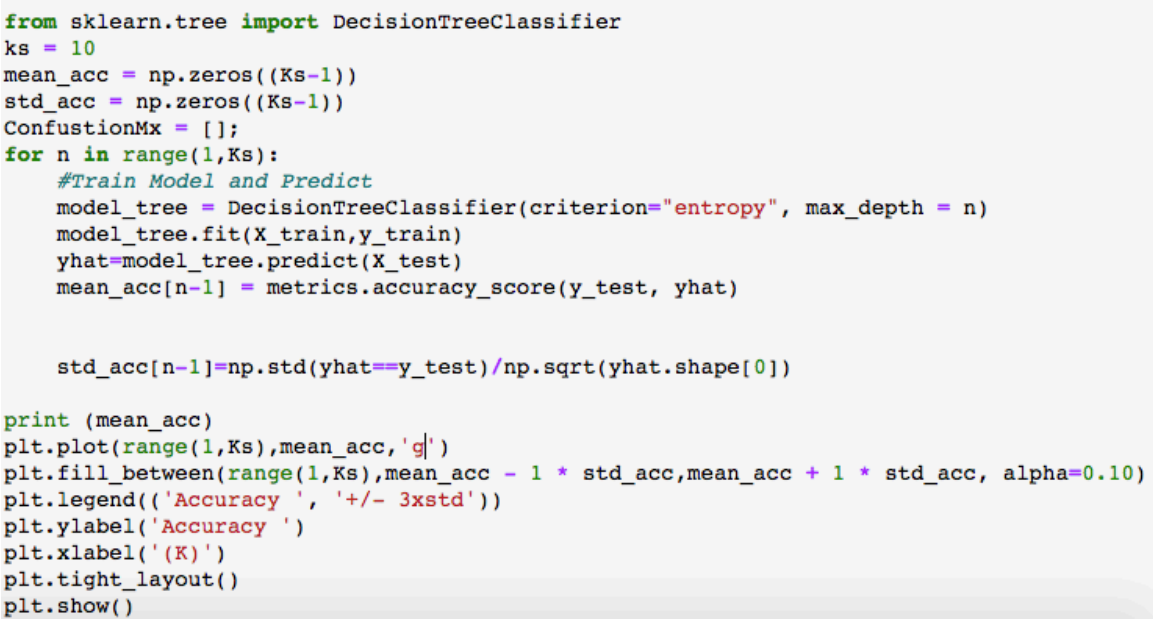
```
0.7662337662337663
```



```
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)
yhat = clf.predict(X_test)
print(yhat[0:10])
mean_acc_rbf= metrics.accuracy_score(y_test, yhat)
print(mean_acc_rbf)
from sklearn.metrics import jaccard_similarity_score
jacc_score_tree=jaccard_similarity_score(y_test, yhat)
print(jacc_score_tree)
```

```
[0 0 0 0 0 1 1 0 1 0]
0.7987012987012987
0.7987012987012987
```

SVM method with linear kernel performed better than KNN method to predict the test data set outcome. Therefore, we will surely ignore KNN method and will compare the result of linear SVM with the last classification method, which is a decision trees method. Same procedure was used to choose the best value of max_depth in decision trees method as it was used to determine the best K in KNN classification method. Values from 1 t 10 were selected and results indicated that 3 is going to be the best number to represent the model:



```
from sklearn.tree import DecisionTreeClassifier
ks = 10
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfustionMx = [];
for n in range(1,Ks):
    #Train Model and Predict
    model_tree = DecisionTreeClassifier(criterion="entropy", max_depth = n)
    model_tree.fit(X_train,y_train)
    yhat=model_tree.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)


    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

print (mean_acc)
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('(K)')
plt.tight_layout()
plt.show()
```

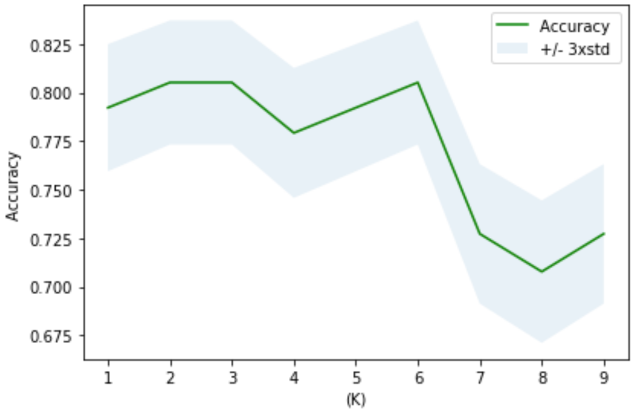


Figure 2. Accuracy plot to determine the best K for decision trees model

The model with K=3 resulted in almost %80 accuracy, which was the highest among all of the classification methods that were performed in our analysis. Therefore, the decision trees model will be used to predict the chance of having Diabetes for the patient by measuring 8 features that we used to train our models. The results prove the importance and the power of statistical modeling and machine learning techniques to appropriately predict important phenomena such as sever and deadly diseases like Diabetes.

Discussion

We investigated our classification problem with the three most common classification machine learning algorithms. It should be noted that we could use more elaborated techniques like artificial neural network (ANN) to build our model and predict the future outcome. Our dataset had 768 observations and 8 features; therefore the methods that we used could be acceptable. However, in situation that we have a lot of features and more observations, it will be much better use more complex technique like artificial neural network and deep learning python libraries to build our model. There are nice python library packages such as Kera, Tensorflow and Pytorch to build more elaborated regression and classification models.

Conclusion

In this report, we investigated different machine learning algorithms to build the predictive model to predict the condition of the patient based on 8 physiological measurements and tell us if the patient has Diabetes or not. Three classification methods were carried out for our study. Results indicated that decision trees classification method with k=3 was able to provide highest accuracy compared to support vector machine (SVM) and K-nearest neighbor (KNN) method. In overall, this report demonstrated the importance and strength of the machine learning techniques to predict the vital and important phenomena related to human lives.