Project Report

on

Data classification and analysis

by : H.Bharat Chandra

B.Tech 2nd year (IT) Vignan's Institute of Information Technology, Duvvada.

Email: bharat.chandra200@gmail.com

Scope of project :

After the collection of dataset and performing data cleaning , data processing , and data visualiations , the data sets are trained with machine learning models such as **Logistic Regression and k nearest neighbour classifier** and model is built .

Steps implemented :

MACHINE LEARNING

Python version-3.6
Data collection
Data cleaning
Data processing
Libraries-sklearn,numpy,pandas,math,tensorflow,seaborn,csv
Training
Logistic Regression
k nearest neighbour classifier
Data visualization
Model evaluation

Alogorithms used :

1.Logistic Regression

2.K nearest neighbour Classifier

1.)Logistic Regression : Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

It is used to determine the extent to which there is a linear relationship between a dependent variable and one or more independent variables.

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2.)K Nearest Neighbour Classifer : K-Nearest Neighbors is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

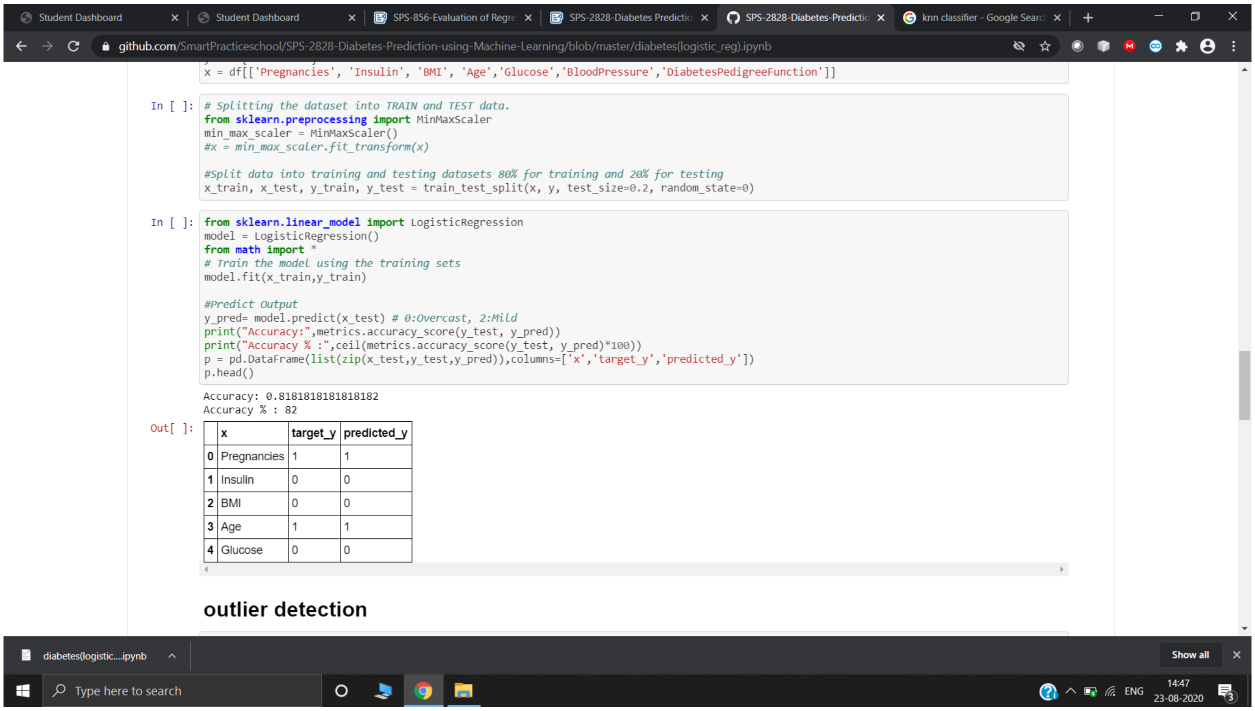
Further steps :

After building the model we need to evaluate the performance / results of the model . for that we use different metrics for different algorithms.

For both the models , they are evaluated based on accuracy score.

I got the accuracy of 82% for logistic regression and 79% for KNN classifier.

for logistic regression :


Student Dashboard
Student Dashboard
SPS-856-Evaluation of Regre
SPS-2828-Diabetes Predictio
SPS-2828-Diabetes-Predictio
knn classifier - Google Searc
github.com/SmartPracticeschool/SPS-2828-Diabetes-Prediction-using-Machine-Learning/blob/master/diabetes(logistic_reg).ipynb
diabetes(logistic....ipynb
Show all
Type here to search
ENG
14:47
23-08-2020

```
x = df[['Pregnancies', 'Insulin', 'BMI', 'Age','Glucose','BloodPressure','DiabetesPedigreeFunction']]
```

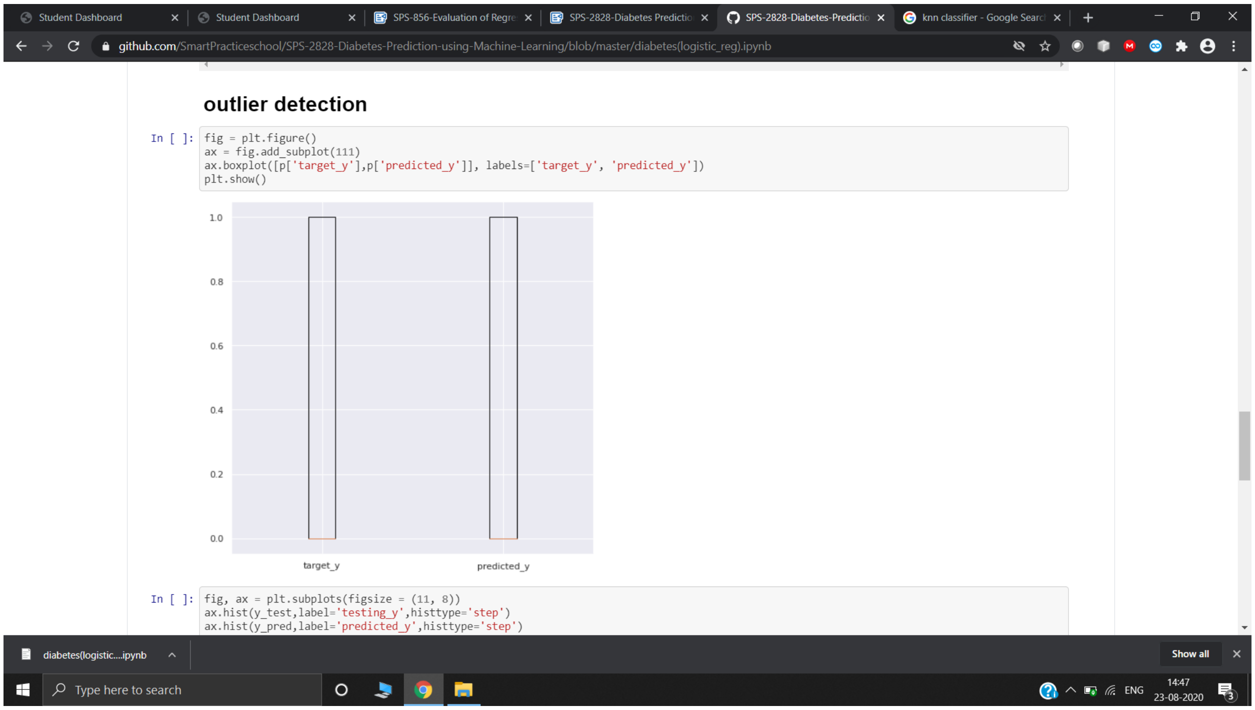
```
In [ ]: # Splitting the dataset into TRAIN and TEST data.
        from sklearn.preprocessing import MinMaxScaler
        min_max_scaler = MinMaxScaler()
        #x = min_max_scaler.fit_transform(x)

        #Split data into training and testing datasets 80% for training and 20% for testing
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

```
In [ ]: from sklearn.linear_model import LogisticRegression
        model = LogisticRegression()
        from math import *
        # Train the model using the training sets
        model.fit(x_train,y_train)

        #Predict Output
        y_pred= model.predict(x_test) # 0:Overcast, 2:Mild
        print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
        print("Accuracy % :",ceil(metrics.accuracy_score(y_test, y_pred)*100))
        p = pd.DataFrame(list(zip(x_test,y_test,y_pred)),columns=['x','target_y','predicted_y'])
        p.head()

        Accuracy: 0.8181818181818182
        Accuracy % : 82
```

Out[ ]:

	x	target_y	predicted_y
0	Pregnancies	1	1
1	Insulin	0	0
2	BMI	0	0
3	Age	1	1
4	Glucose	0	0

outlier detection



Student Dashboard
Student Dashboard
SPS-856-Evaluation of Regre
SPS-2828-Diabetes Predictio
SPS-2828-Diabetes-Predictio
knn classifier - Google Search
github.com/SmartPracticeschool/SPS-2828-Diabetes-Prediction-using-Machine-Learning/blob/master/diabetes(logistic_reg).ipynb

outlier detection

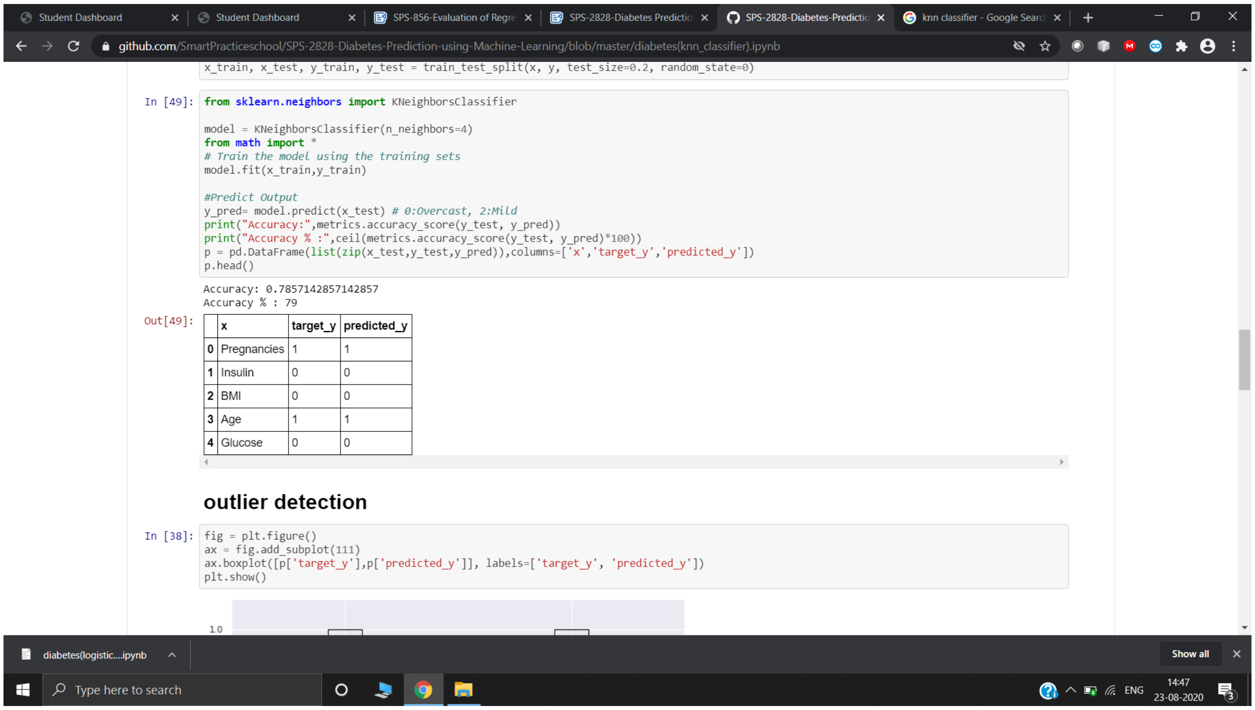
```
In [ ]: fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.boxplot([p['target_y'],p['predicted_y']], labels=['target_y', 'predicted_y'])
        plt.show()
```

1.0
0.8
0.6
0.4
0.2
0.0
target_y
predicted_y

```
In [ ]: fig, ax = plt.subplots(figsize = (11, 8))
        ax.hist(y_test,label='testing_y',histtype='step')
        ax.hist(y_pred,label='predicted_y',histtype='step')
```

diabetes(logistic....ipynb
Show all
Type here to search
ENG
14:47
23-08-2020

For KNN :



Student Dashboard
Student Dashboard
SPS-856-Evaluation of Regre
SPS-2828-Diabetes Predictio
SPS-2828-Diabetes-Predictio
knn classifier - Google Search
github.com/SmartPracticeschool/SPS-2828-Diabetes-Prediction-using-Machine-Learning/blob/master/diabetes(knn_classifier).ipynb

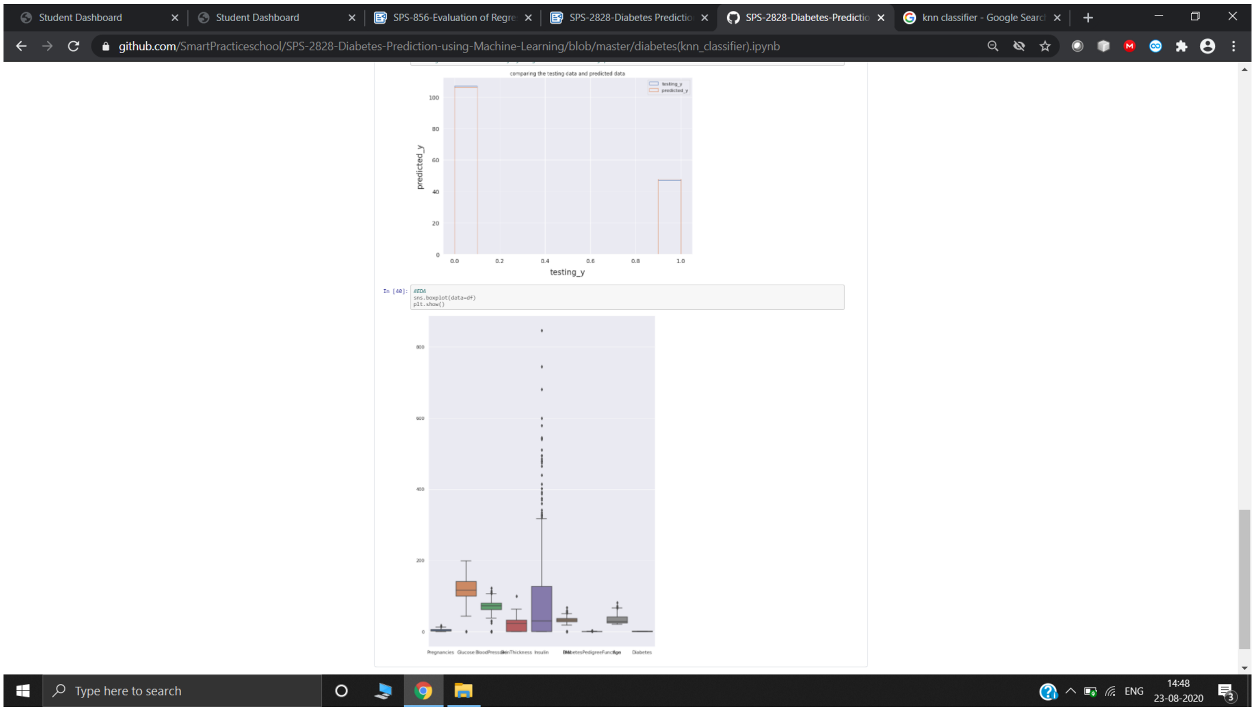
```
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

In [49]: from sklearn.neighbors import KNeighborsClassifier

         model = KNeighborsClassifier(n_neighbors=4)
         from math import *
         # Train the model using the training sets
         model.fit(x_train,y_train)

         #Predict Output
         y_pred= model.predict(x_test) # 0:Overcast, 2:Mild
         print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         print("Accuracy % :",ceil(metrics.accuracy_score(y_test, y_pred)*100))
         p = pd.DataFrame(list(zip(x_test,y_test,y_pred)),columns=['x','target_y','predicted_y'])
         p.head()

         Accuracy: 0.7857142857142857
         Accuracy % : 79
```

Out[49]:

	x	target_y	predicted_y
0	Pregnancies	1	1
1	Insulin	0	0
2	BMI	0	0
3	Age	1	1
4	Glucose	0	0

outlier detection

```
In [38]: fig = plt.figure()
         ax = fig.add_subplot(111)
         ax.boxplot([p['target_y'],p['predicted_y']], labels=['target_y', 'predicted_y'])
         plt.show()
```


Student Dashboard
Student Dashboard
SPS-856-Evaluation of Regre
SPS-2828-Diabetes Predictio
SPS-2828-Diabetes-Predictio
knn classifier - Google Search
github.com/SmartPracticeschool/SPS-2828-Diabetes-Prediction-using-Machine-Learning/blob/master/diabetes(knn_classifier).ipynb
comparing the testing data and predicted data
testing_y
predicted_y
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
testing_y
predicted_y

In [40]:

```
#EDA
sns.boxplot(data=df)
plt.show()
```

800
600
400
200
0
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Diabetes
Type here to search
ENG
14:48
23-08-2020