Project Report

on

Evalution of Regression and Classificarion Models

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 **Linear Regression and Decision Tree 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
- Linear Regression
- Decision tree classifier
- Data visualization
- Model evaluation

Alogorithms used :

1.Linear Regression

2.Decision Tree Classifier

1.)Linear Regression：Regression belongs to the class of Supervised Learning tasks where the datasets that are used for predictive modeling contain continous labels.

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

In statistics, linear regression is a linear approach to modeling the relationship between a dependent variable and independent variables. The case of one explanatory variable is called simple linear regression.
Learning/training a linear regression model essentially means estimating the values of the coefficients/parameters used in the representation with the data you have.

2.)Decision Tree Classifer : The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each

branch descending from that node corresponds to one of the possible values for that attribute.
Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable.

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily resembles the human level thinking. That is why decision trees are easy to understand and interpret.

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 Regression model : For Regression algorithm the model evaluation metrics are

1. MSE - mean square error

The average of the square of the difference between the original values and the predicted values.

2. RMSE - root mean square error

Root Mean Square Error (**RMSE**) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; **RMSE** is a measure of how spread out these residuals are.

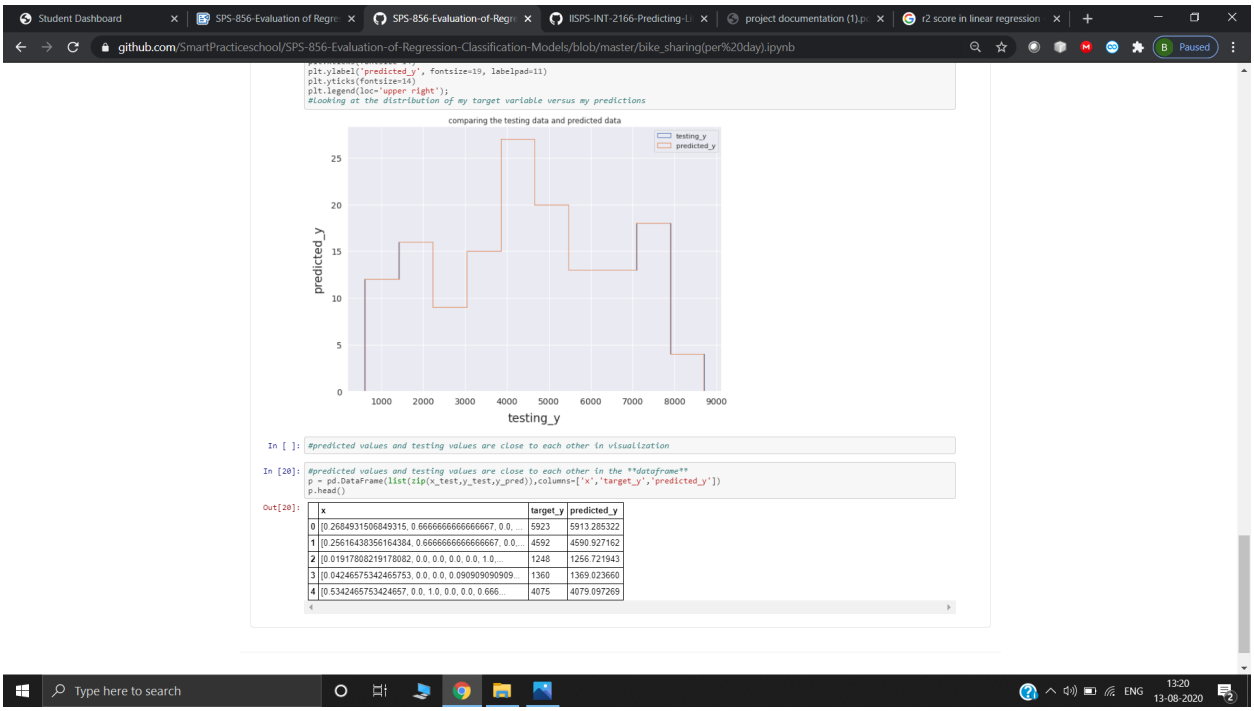
3. R2 score - r square score

R-squared ($\mathbf{R^2}$) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a **regression** model.
With this r2 score , we can evaluate the quality of model.
The values range from 0 to 1.

If the r2 score is 1 , the model is highly accurate but sometimes it leads to over fitting.

```
In [12]: #Linear regression using SKLearn Linear Regression method
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
#Build linear regression model
model = Lasso()
# Train the model, using training data set
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print('Intercept =', model.intercept_)
print('Coefficients = ', model.coef_)
mse = mean_squared_error(y_test, y_pred)
rmse = math.sqrt(  mse)
r2score = r2_score(y_test, y_pred)
print("Mean squared error(MSE) = %.2f" %   mse)
print("Root Mean squared error(RMSE) = %.2f" % rmse)
print('R2 score = %.2f' % r2score)

Intercept = 34.394477016900964
Coefficients =  [ 0.00000000e+00  0.00000000e+00  2.25868938e+00  0.00000000e+00
 -0.00000000e+00  0.00000000e+00 -0.00000000e+00 -0.00000000e+00
  0.00000000e+00  0.00000000e+00 -0.00000000e+00 -0.00000000e+00
  3.38865758e+03  6.90931217e+03]
Mean squared error(MSE) = 39.60
Root Mean squared error(RMSE) = 6.29
R2 score = 1.00
```



Here we can see that predicted values are approximately equal to testing values. Finally we can say that the model is good and accurate .

For Classification model : For Regression algorithm the model evaluation metrics are

1. Accuracy Score

Classification **Accuracy** is what we usually mean, when we use the term **accuracy**. It is the ratio of number of correct predictions to the total number of input samples.

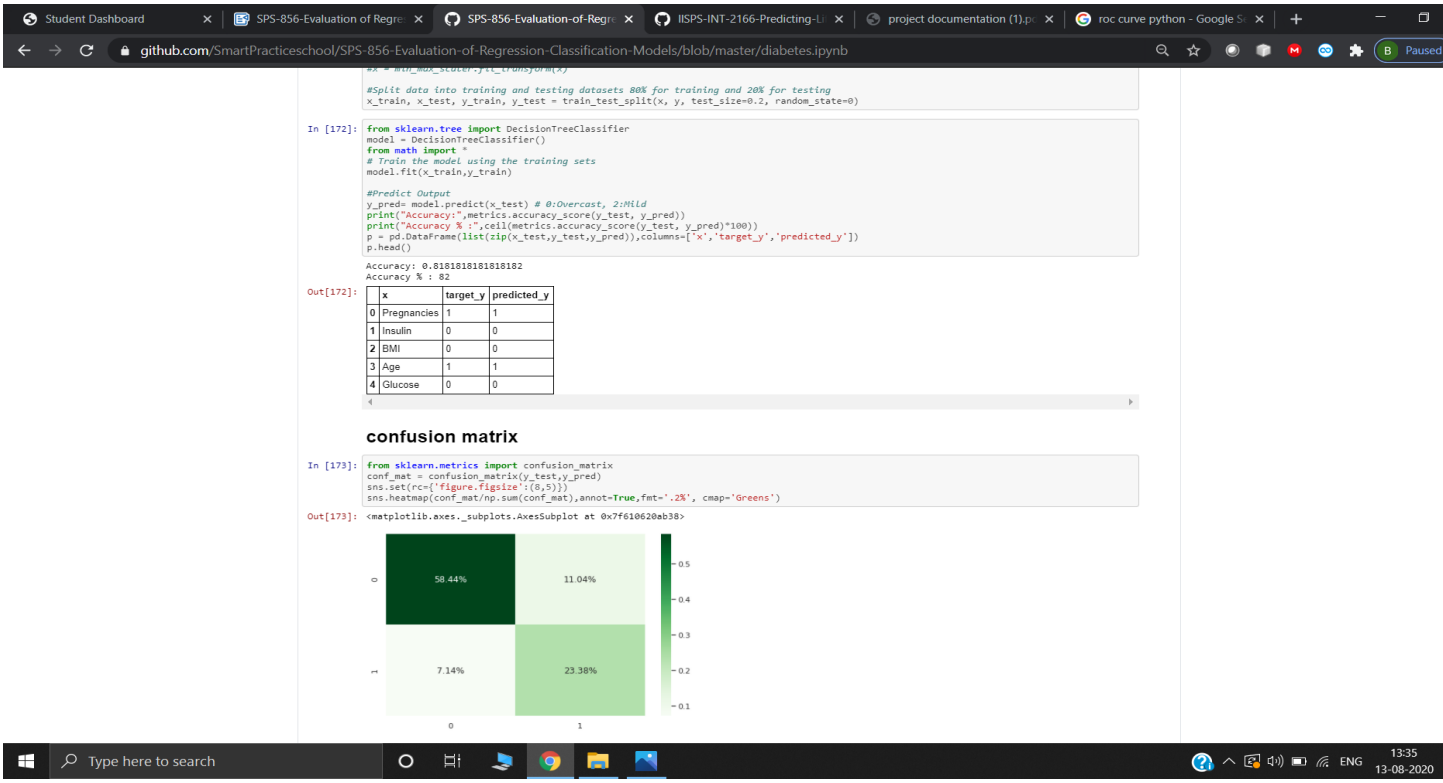
2. Confusion Matrix

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values **are** known.

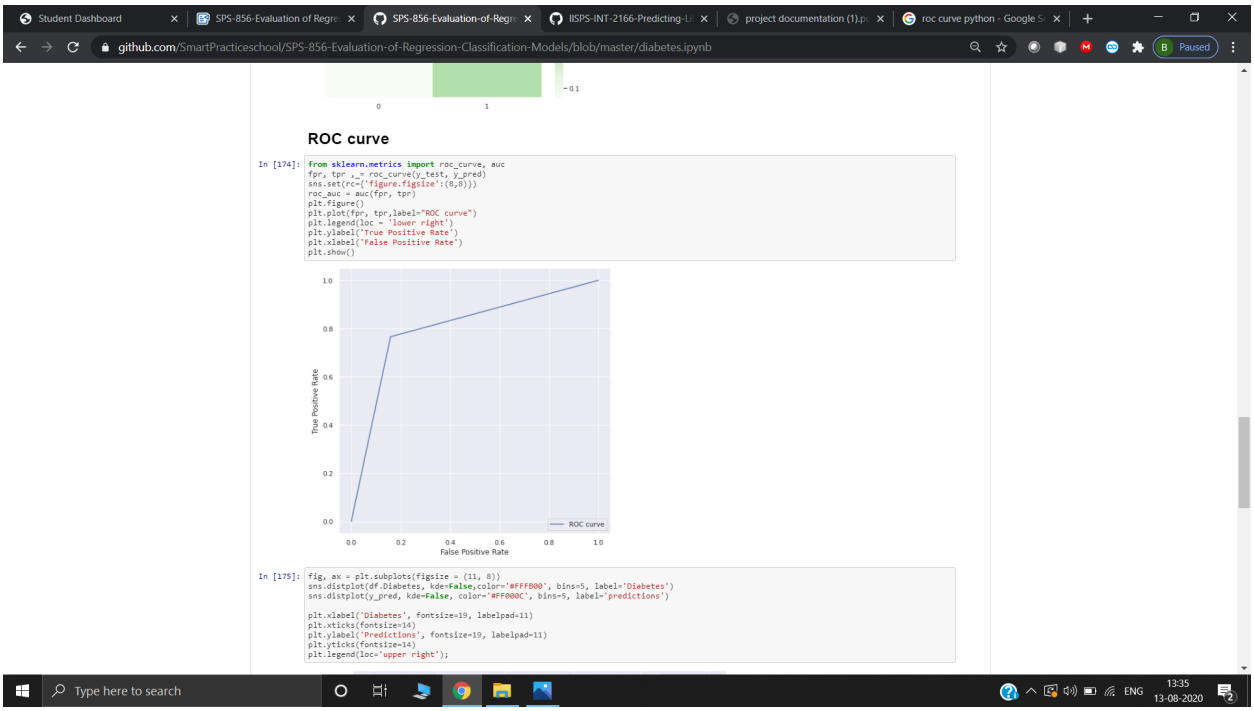
3. ROC curve -

ROC is a plot of signal (True Positive Rate) against noise (False Positive Rate). ... The model performance is determined by looking at the area under the **ROC curve** (or AUC).

Higher the accuracy , more good is the model .
I got the accuracy of 82%



ROC Curve



comparing testing values and predicted values.