LED DISPLAY DOMAIN

Grande Pranathi.
11801844

Department of computer science and engineering, lovely professional university, Phagwara, India

Abstract

LED display domain is simple domain contains 7 Boolean attributes and 10 classes, the set of decimal digits. Recall that LED displays contain 7 light-emitting diodes. Hence the reason for 7 attributes. The class attribute is an integer ranging between 0 to 9 inclusive, representing the possible digits show on the display. The problem would be easy if not for the introduction of noise. In this In this case, each attribute value has the 10% probability of having its value inverted. It's value inverted. It's valuable to know the optimal Bayes rate for these databases. In this case, the misclassification rate is 26% (74% classification accuracy). Attribute Information *V1-V7 represent each of the 7 LEDs, with values either 0 or 1, According to whether the corresponding light is on or not for the decimal digit. Each has a 10% percent chance of being inverted. We have to develop some preparing data(Feature identification), models and Training models and confusion matrix, prediction and I have collected data set from the UCI machine learning respositary. A led display domain is a flat panel display that uses an array of light-emitting diodes as pixels for video display. Their brightness allows then to be used outdoors where they are visible in the sun for store signs and billboards. In recent years, they have also become commonly used in destination signs on public transport vehicles, as well as variable-message signs on highways. In the LED display domain attributes is of 7 and number of attributes is of 500 rows and the number of features is 8 columns of the data set and the number of the classes is 10 and it is a number of distinct target attributes (if its normal) and the number of numeric attributes is 7 is V1, V2, V3, V4, V5, V6, V7 represents each of the 7 LEDs, with the values of either 0 or 1. The class is the target value in the LED display domain.

INTRODUCTION

In this I have to Preparing the data (Features identification) in which features is an individual measurable property or characteristic of a phenomenon being observed choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression. Features are usually numeric, but structural features such as strings and graphs are used in the syntactic pattern recognition. The Concept of "Feature" is related to that of explanatory variable used in statistical techniques such as linear Regression.

Models I have used are Ridged Regression, Lasso Regression, Decision tree Regression, Random Forest Regression.

Ridged Regression:A Ridged Regression is basically a regularized version of Linear Regressor. i.e to the original cost function of linear regressor we add a regularized term which forces the learning algorithm to fit the data and helps to keep the weights lower as possible. The regularized term has the parameter 'alpha' which controls the regularization of the model. i.e helps in reducing the variance of the estimates. Cost Function for Ridge Regressor.

Lasso Regression: A Lasso regression stands for Least Absolute Shrinkage and Selection Operator. It adds penalty term to the cost function. This term is the absolute sum of the coefficients. As the value of coefficients increase from 0 this term penalizes, cause model, to decrease the value of coefficients in order to reduce loss. The difference between ridge and lasso regression is that it tends to make coefficient to absolute Zero,

Decision tree Regression: A Decision tree builds Regression or classification models in the form of a tree structures. it breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Random Forest Regression: Every decision tree has a high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of classification problem, the final output is taken by using the majority voting classifier. In the case of regression problem, the final output is the mean of all the output.

Confusion matrix:A confusion matrix is a technique for summarizing the performance of an classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.

Prediction: Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days.

LED display domain is simple domain contains 7 Boolean attributes and 10 classes, the set of decimal digits. Recall that LED displays contain 7 light-emitting diodes. Hence the reason for 7 attributes. The class attribute is an integer ranging between 0 to 9 inclusive, representing the possible digits show on the display. The problem would be easy if not for the introduction of noise. In this In this case, each attribute value has the 10% probability of having its value inverted. It's value inverted. It's valuable to know the optimal Bayes rate for these databases. In this case, the misclassification rate is 26% (74% classification accuracy). Attribute Information *V1-V7 represent each of the 7 LEDs, with values either 0 or 1, According to whether the corresponding light is on or not for the decimal digit. Each has a 10% percent chance of being inverted. We have to develop some preparing data(Feature identification), models and Training models and confusion matrix, prediction and I have collected data set from the UCI machine learning respositary. A led display domain is a flat panel display that uses an array of light-emitting diodes as pixels for video display. Their brightness allows then to be used outdoors where they are visible in the sun for store signs and billboards. In recent years, they have also become commonly used in destination signs on public transport vehicles, as well as variable-message signs on highways. In the LED display domain attributes is of 7 and number of attributes is of 500 rows and the number of features is 8 columns of the data set and the number of the classes is 10 and it is a number of distinct target attributes (if its normal) and the number of numeric attributes is 7 is V1, V2, V3, V4, V5, V6, V7 represents each of the 7 LEDs, with the values of either 0 or 1. The class is the target value in the LED display domain.

IMPLEMENTATION OF MODELS

Models I have used are Ridged Regression, Lasso Regression, Decision tree Regression, Random Forest Regression.

Ridged Regression:A Ridged Regression is basically a regularized version of Linear Regressor. i.e to the original cost function of linear regressor we add a regularized term which forces the learning algorithm to fit the data and helps to keep the weights lower as possible. The regularized term has the parameter 'alpha' which controls the regularization of the model. i.e helps in reducing the variance of the estimates. Cost Function for Ridge Regressor.

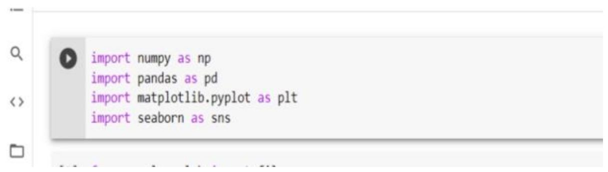
Lasso Regression: A Lasso regression stands for Least Absolute Shrinkage and Selection Operator. It adds penalty term to the cost function. This term is the absolute sum of the coefficients. As the value of coefficients increase from 0 this term penalizes, cause model, to decrease the value of coefficients in order to reduce loss. The difference between ridge and lasso regression is that it tends to make coefficient to absolute Zero,

Decision tree Regression: A Decision tree builds Regression or classification models in the form of a tree structures. it breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Random Forest Regression: Every decision tree has a high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of classification problem, the final output is taken by using the majority voting classifier. In the case of regression problem, the final output is the mean of all the output.

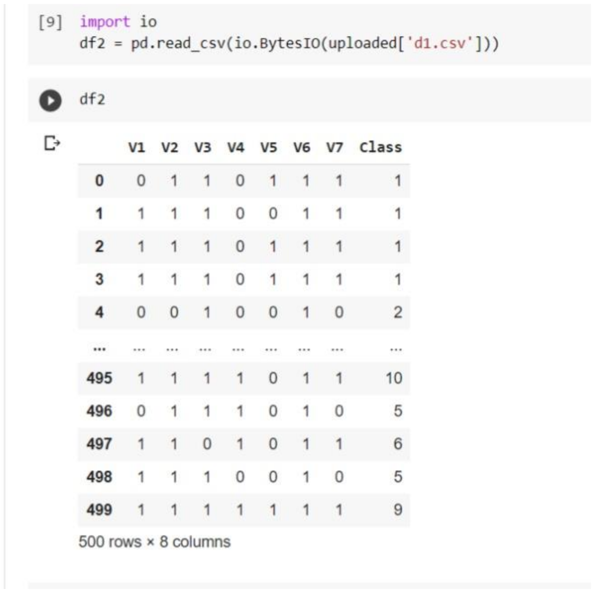
Result and discussion

Step-1: This is the program which i have implemented in the program importing all the important libraries like numpy, pandas, seaborn.



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Step-2: After that i am read.csv file and i will get output these are data values which consists of 500 rows and 8 columns.



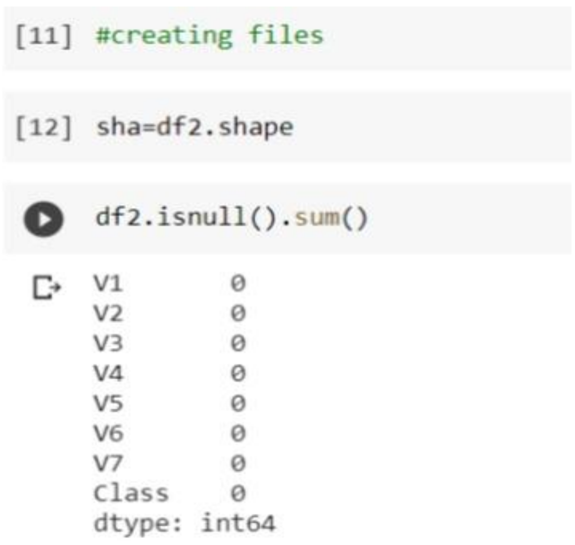
```
[9] import io
    df2 = pd.read_csv(io.BytesIO(uploaded['d1.csv']))
```

```
df2
```

	V1	V2	V3	V4	V5	V6	V7	Class
0	0	1	1	0	1	1	1	1
1	1	1	1	0	0	1	1	1
2	1	1	1	0	1	1	1	1
3	1	1	1	0	1	1	1	1
4	0	0	1	0	0	1	0	2
...	...	...	...	...	...	...	...	...
495	1	1	1	1	0	1	1	10
496	0	1	1	1	0	1	0	5
497	1	1	0	1	0	1	1	6
498	1	1	1	0	0	1	0	5
499	1	1	1	1	1	1	1	9

500 rows × 8 columns

Step-3: After that this data has interpting the values as the null values which as the null value are not.



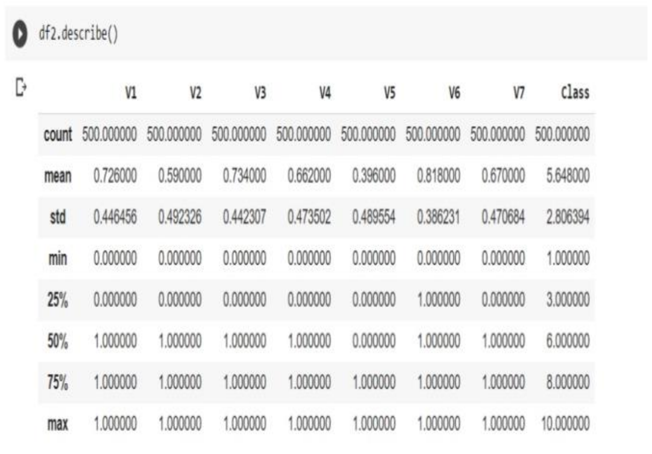
```
[11] #creating files

[12] sha=df2.shape

df2.isnull().sum()

V1       0
V2       0
V3       0
V4       0
V5       0
V6       0
V7       0
Class    0
dtype: int64
```

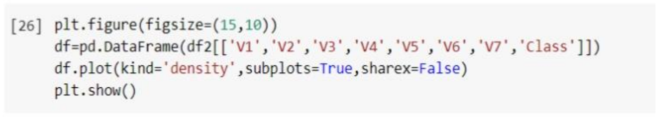
Step-4: After that i am describing the dataset like count ,mean of the rows, std, min, 25percent, 50 percent, 75percent max using the describe() (describe function).



```
df2.describe()
```

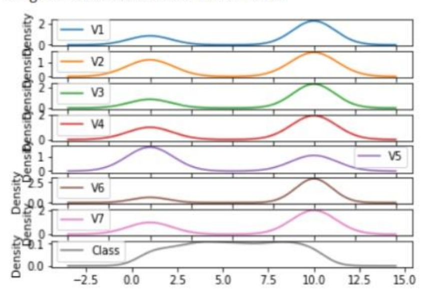
	V1	V2	V3	V4	V5	V6	V7	Class
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	0.726000	0.590000	0.734000	0.662000	0.396000	0.818000	0.670000	5.648000
std	0.446456	0.492326	0.442307	0.473502	0.489554	0.386231	0.470684	2.806394
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	3.000000
50%	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	6.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	8.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	10.000000

Step-5: After that here we are plotting the graph like density graph values ranging and how the values are varying from one row to another row which as the highest density values as 5 something.

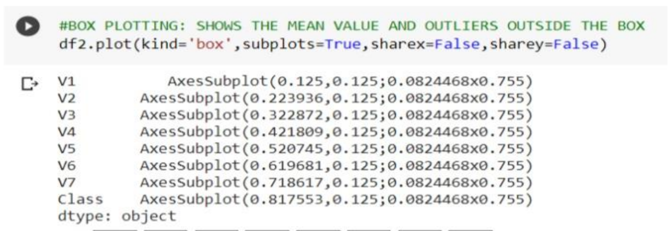


```
[26] plt.figure(figsize=(15,10))
     df=pd.DataFrame(df2[['V1','V2','V3','V4','V5','V6','V7','Class']])
     df.plot(kind='density',subplots=True,sharex=False)
     plt.show()
```

```
<Figure size 1080x720 with 0 Axes>
```

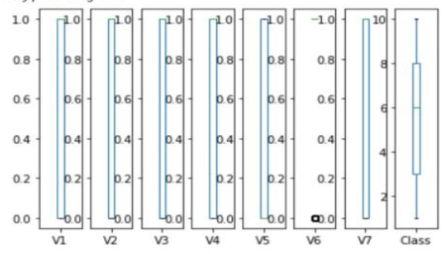


Step-6: After that i have done the box plotting which represent by the if there is out layers are not it shows that 0 to 1 percent represents the mean value like middle value 7 shoes the mean value and there is one out Layer in the V6 two out layer are more than we are going to remove in this graph there is output layer.

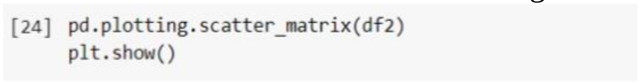


```
#BOX PLOTTING: SHOWS THE MEAN VALUE AND OUTLIERS OUTSIDE THE BOX
df2.plot(kind='box',subplots=True,sharex=False,sharey=False)
```

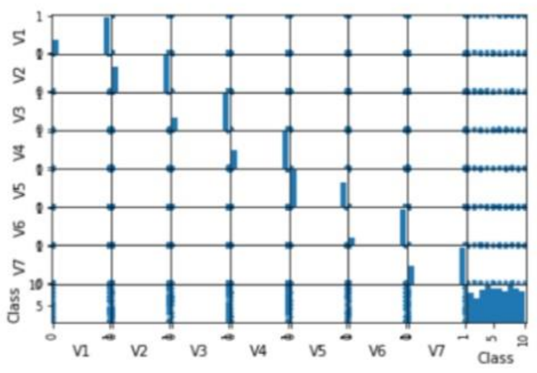
```
V1          AxesSubplot(0.125,0.125;0.0824468x0.755)
V2       AxesSubplot(0.223936,0.125;0.0824468x0.755)
V3       AxesSubplot(0.322872,0.125;0.0824468x0.755)
V4       AxesSubplot(0.421809,0.125;0.0824468x0.755)
V5       AxesSubplot(0.520745,0.125;0.0824468x0.755)
V6       AxesSubplot(0.619681,0.125;0.0824468x0.755)
V7       AxesSubplot(0.718617,0.125;0.0824468x0.755)
Class    AxesSubplot(0.817553,0.125;0.0824468x0.755)
dtype: object
```



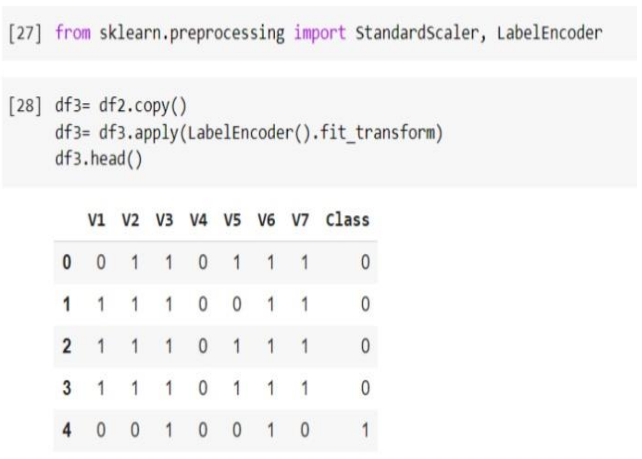
Step-7: After that we have to plot the scatter plotting which shows that how the values are one values to another values are scattering.



```
[24] pd.plotting.scatter_matrix(df2)
     plt.show()
```



Step-8: In which we have used Standard Scaler, Label Encoder.

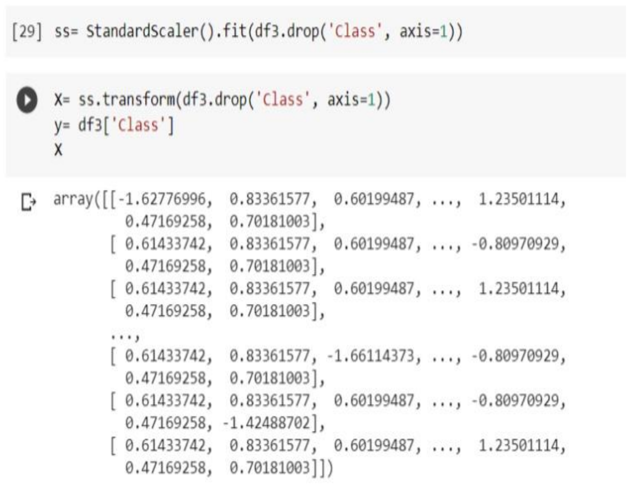


```
[27] from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```
[28] df3= df2.copy()
     df3= df3.apply(LabelEncoder().fit_transform)
     df3.head()
```

	V1	V2	V3	V4	V5	V6	V7	Class
0	0	1	1	0	1	1	1	0
1	1	1	1	0	0	1	1	0
2	1	1	1	0	1	1	1	0
3	1	1	1	0	1	1	1	0
4	0	0	1	0	0	1	0	1

Step-9: In which we have used Standard Scaler, Label Encoder in which have done the class.

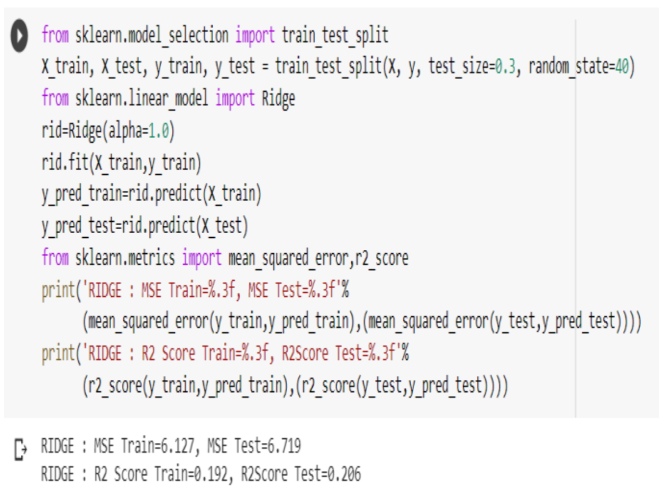


```
[29] ss= StandardScaler().fit(df3.drop('Class', axis=1))
```

```
X= ss.transform(df3.drop('Class', axis=1))
y= df3['Class']
X
```

```
array([[-1.62776996,  0.83361577,  0.60199487, ...,  1.23501114,
         0.47169258,  0.70181003],
       [ 0.61433742,  0.83361577,  0.60199487, ..., -0.80970929,
         0.47169258,  0.70181003],
       [ 0.61433742,  0.83361577,  0.60199487, ...,  1.23501114,
         0.47169258,  0.70181003],
       ...,
       [ 0.61433742,  0.83361577, -1.66114373, ..., -0.80970929,
         0.47169258,  0.70181003],
       [ 0.61433742,  0.83361577,  0.60199487, ..., -0.80970929,
         0.47169258, -1.42488702],
       [ 0.61433742,  0.83361577,  0.60199487, ...,  1.23501114,
         0.47169258,  0.70181003]])
```

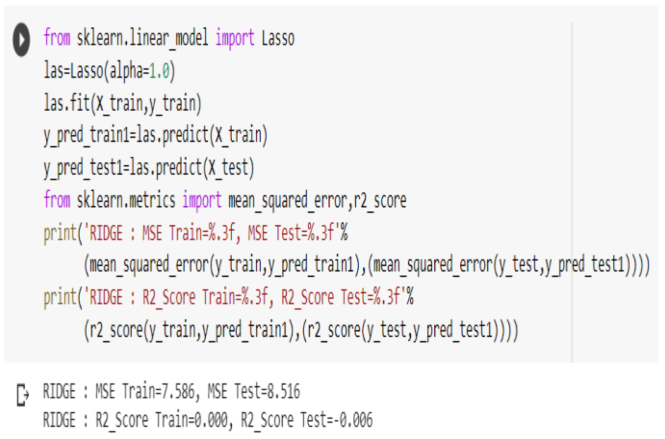
Step-10: In this code we have used Ridged Regression model. In which I have got Ridge MSE Train =6.719, MSE test is 6.719, R2 Score train is0.192 and R2 Score Test=0.206.



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=40)
from sklearn.linear_model import Ridge
rid=Ridge(alpha=1.0)
rid.fit(X_train,y_train)
y_pred_train=rid.predict(X_train)
y_pred_test=rid.predict(X_test)
from sklearn.metrics import mean_squared_error,r2_score
print('RIDGE : MSE Train=%.3f, MSE Test=%.3f'%
      (mean_squared_error(y_train,y_pred_train),(mean_squared_error(y_test,y_pred_test))))
print('RIDGE : R2 Score Train=%.3f, R2Score Test=%.3f'%
      (r2_score(y_train,y_pred_train),(r2_score(y_test,y_pred_test))))
```

```
RIDGE : MSE Train=6.127, MSE Test=6.719
RIDGE : R2 Score Train=0.192, R2Score Test=0.206
```

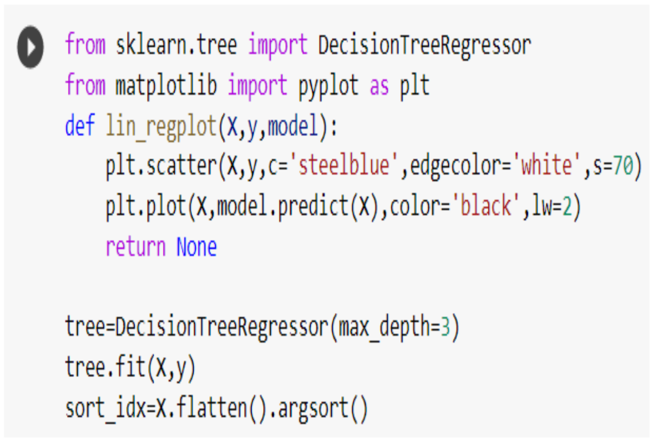
Step-11: In this code we have used Lasso Regression model. In which I have got Ridge MSE Train is 7.586 and MSE test is 8.516, Rigid R2 Score Train is 0.000 and R2 Score test is 0.006.



```
from sklearn.linear_model import Lasso
las=Lasso(alpha=1.0)
las.fit(X_train,y_train)
y_pred_train1=las.predict(X_train)
y_pred_test1=las.predict(X_test)
from sklearn.metrics import mean_squared_error,r2_score
print('RIDGE : MSE Train=%.3f, MSE Test=%.3f'%
      (mean_squared_error(y_train,y_pred_train1),(mean_squared_error(y_test,y_pred_test1))))
print('RIDGE : R2_Score Train=%.3f, R2_Score Test=%.3f'%
      (r2_score(y_train,y_pred_train1),(r2_score(y_test,y_pred_test1))))
```

```
RIDGE : MSE Train=7.586, MSE Test=8.516
RIDGE : R2_Score Train=0.000, R2_Score Test=-0.006
```

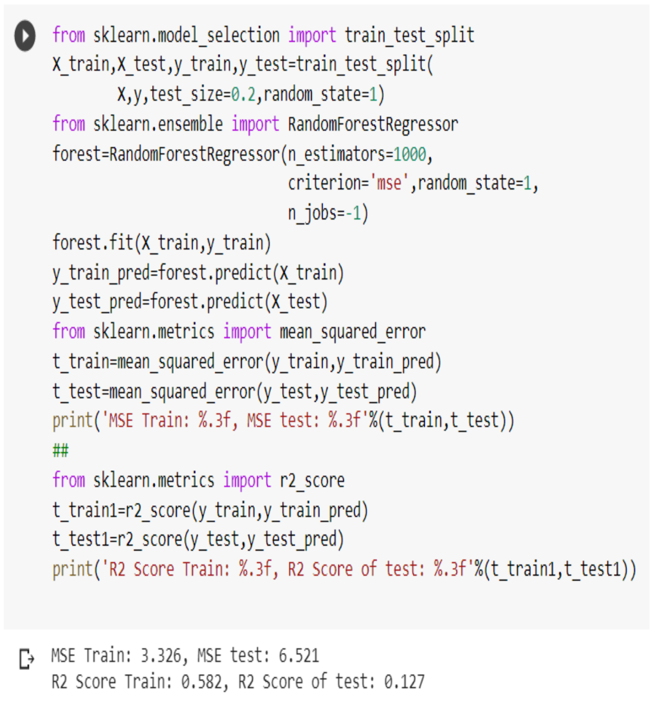
Step-12:
In which I have used decision tree Regression.



```
from sklearn.tree import DecisionTreeRegressor
from matplotlib import pyplot as plt
def lin_regplot(X,y,model):
    plt.scatter(X,y,c='steelblue',edgecolor='white',s=70)
    plt.plot(X,model.predict(X),color='black',lw=2)
    return None

tree=DecisionTreeRegressor(max_depth=3)
tree.fit(X,y)
sort_idx=X.flatten().argsort()
```

Step: 13: In which I have worked on Random forest Regression in which I have got MSE Train is 3.326, MSE test is 6.521, R2 score Train is 0.582 and R2 Score of test is 0.127.



```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(
        X,y,test_size=0.2,random_state=1)
from sklearn.ensemble import RandomForestRegressor
forest=RandomForestRegressor(n_estimators=1000,
                             criterion='mse',random_state=1,
                             n_jobs=-1)
forest.fit(X_train,y_train)
y_train_pred=forest.predict(X_train)
y_test_pred=forest.predict(X_test)
from sklearn.metrics import mean_squared_error
t_train=mean_squared_error(y_train,y_train_pred)
t_test=mean_squared_error(y_test,y_test_pred)
print('MSE Train: %.3f, MSE test: %.3f'%(t_train,t_test))
##
from sklearn.metrics import r2_score
t_train1=r2_score(y_train,y_train_pred)
t_test1=r2_score(y_test,y_test_pred)
print('R2 Score Train: %.3f, R2 Score of test: %.3f'%(t_train1,t_test1))
```

```
MSE Train: 3.326, MSE test: 6.521
R2 Score Train: 0.582, R2 Score of test: 0.127
```

Step-14: After Regressions we have to do confusion matrix for all the regression that I have used in the project.

Step-15: After that I have done the used Non linear SVM Regression and then I have done confusion matrix for that.

Step-16: After that I have done the prediction matrix.

Conclusion

The Scope of this project is to apply machine learning concepts taught in class on our data set. We made use of open dataset. We are able to apply several classification machine learning methods on the Led display domain data set to achieve the goal of the project.
The goal of the project is to model the data, so as to accurately predict the stability of Led display domain. Also, infer the relationship between predictors and the response. We achieved these goals by using several classification methods and in this project we have download data sets in the UCI and we have used the data sets in the form of .csv file in this. We have applied the concepts of the machine learning and we have learned that the LED display domain is simple domain contains 7 Boolean attributes and 10 classes, the set of decimal digits. Recall that LED displays contain 7 light-emitting diodes. Hence the reason for 7 attributes. The class attribute is an integer ranging between 0 to 9 inclusive , representing the possible digits show on the display. The problem would be easy if not for the introduction of noise. In this In this case, each attribute value has the 10% probability of having its value inverted. It’s value inverted. It’s valuable to know the optimal Bayes rate for these databases. In this case, the misclassification rate is 26% (74% classification accuracy). Attribute Information *V1-V7 represent each of the 7 LEDs, with values either 0 or 1 and the class is the target value by using the data sets. I have learnt so many libraries how to use in this machine learning project. I have done in this project by Preparing data (Feature Identification) , models like Riged Regression, Lasso Regression, Decision tree Regression and Random forest, confusion matrix and prediction. in this project I have used this algorithms in this project and I have the model of the data and accurately predict the Led display domain.

Acknowledgement

I would like to express my special thanks of gratitude to my faculty who gave me the opportunity to do this project on the topic Project: Led display domain. which also helped me in doing a lot of work. I am really grateful to them. Secondly I would like to thank my friends and the google. Last but not the least I would like to thank the almighty who gave me strength to complete my Project.

References

UCI machine learning repositary:
https://archive.ics.uci.edu/ml/datasets/LED+Display+Domain

https://en.wikipedia.org/wiki/LED_display.

https://www.openml.org/d/40496
https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/

The Project Colab link

The colab file name is: Grande Pranathi_11801844_B36_CA-2_Sem_6

https://colab.research.google.com/drive/1GSHMPjTplAYjIZ40tuDfxzoAg-JcmoJQ?usp=sharing#scrollTo=uI66tTGKr0zo