Title:

Identification & comparative evaluation of several machine learning on a specific problem

Word Count: 1452 words (excluding table of content & References)

Table of content:

Abstract:

The purpose of this assignment was to identify & compare evaluation of different machine learning algorithms on a specific problem. The report will represent how to train different machine learning models in the given dataset and get the best Test accuracy & R2 score by applying different scaling features.

Task 2:

Part A

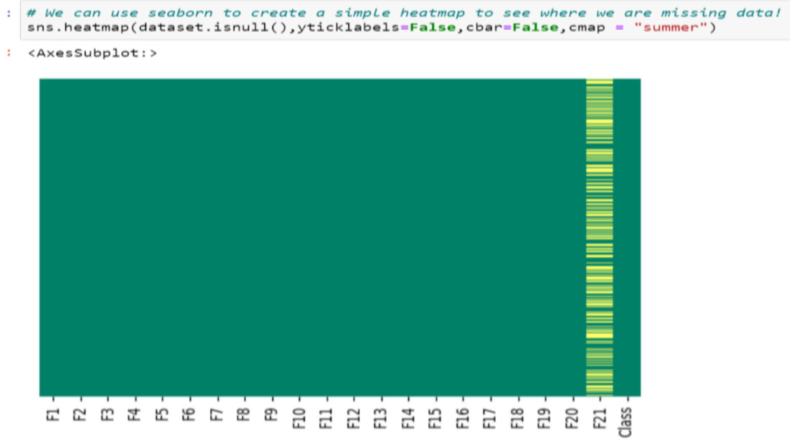
1. **Import Libraries:**
   All the relevant libraries are imported such as pandas, numpy, matplotlib, seaborn. I have imported all the libraries in the starting itself

2. **Importing the data:** The dataset provided is imported using the pandas library.

3. **Data Analysis:**
   Analysis of data is very important before checking for the best fit model. To start with, first I used the **shape** function to check the number of columns and rows. Then class column division is checked, whether it is balanced or not. To show the summary of statistics of data frame, describe function is used. It shows the count, mean, standard deviation, max, min etc. To get the information about data frame info is used, it helps to identify the categorical variable. If any categorical variable is found then it is converted into numerical variable.

   Missing values & treatment: It is required to find the missing values before training the model as it will be difficult to train the model with missing values. I used .isnull().sum() to find the missing values. As a result, it was found that the column F21 has missing values. There are two ways in which missing values can be treated :
   a) Replacement with imputation method like mean, median.
   b) Dropping the table if the missing value is more than 30 percent.

I decided to calculate the percentage of missing value. As the missing percentage was 50%, Figure 1 shows the heatmap of column(F21) with missing values.

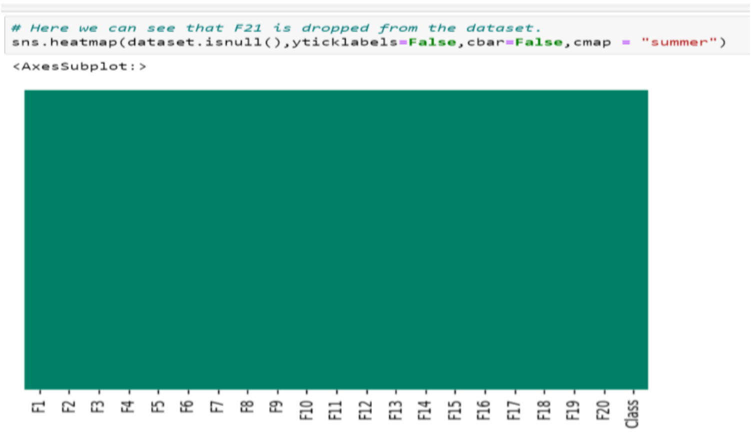


```
: # We can use seaborn to create a simple heatmap to see where we are missing data!
  sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap = "summer")

: <AxesSubplot:>
```

Figure 1: Heatmap with missing values column

As the missing value percentage is 50%, I decided to drop the table as using imputation method will not be a good idea. Figure 2 shows the heatmap without the missing values column(F21).



```
# Here we can see that F21 is dropped from the dataset.
sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap = "summer")
```

```
<AxesSubplot:>
```

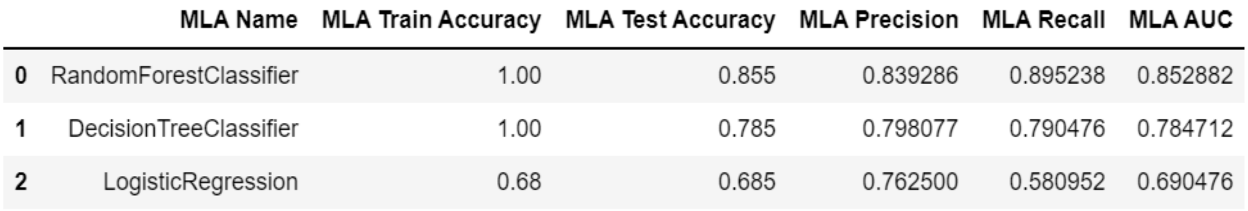
Figure 2: Heatmap without missing values column

4. **Feature Scaling:** As the range of the values a different we have to bring them at same range because data trains well if the values lies in the same range. We can use different scaling features like standard scaler, Robust scaler & MinMax scaler. I have used Robust scaler, as it was giving me better results as compared to the others.

5. **Splitting the data:** Before training the model on the dataset, it needs to be splitted in two different variable train & test or input & output. Independent variables are stored in test & dependent variables are stored in train.

   Split, train & test: Model performance is evaluated on the data which is not trained on it. So for this we split the data into 80:20 ratio. 80% of the data will be used to train the model & 20% will be used to test the model whether is predictions are correct or not.

6. **Training different models:** Once data is splitted different models are trained to get the best fit model. I trained different models for it, but the three models which I considered are Random Forest Classifier, Decision Tree Classifier & Logistic Regression. I used these three models to check the Train accuracy, Test Accuracy, Precision, Recall & AUC. Figure 3 shows a comparative study of the models.



	MLA Name	MLA Train Accuracy	MLA Test Accuracy	MLA Precision	MLA Recall	MLA AUC
0	RandomForestClassifier	1.00	0.855	0.839286	0.895238	0.852882
1	DecisionTreeClassifier	1.00	0.785	0.798077	0.790476	0.784712
2	LogisticRegression	0.68	0.685	0.762500	0.580952	0.690476

Figure 3: Comparative study table of models

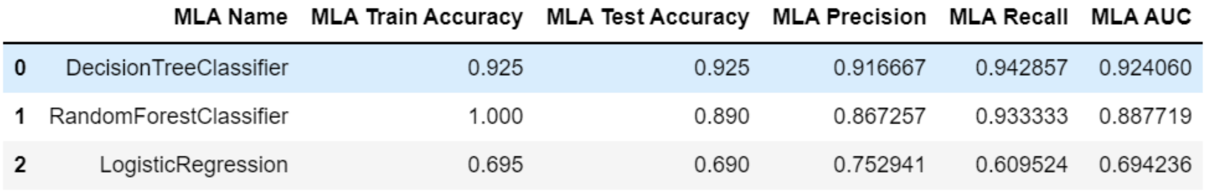
- ***Random Forest Classifier***: After using Robust scaler feature, the model gave the accuracy of 85.5, but train accuracy 1.00 showed that this model was overfitted.

- ***Decision Tree Classifier***: I assumed that this model will give me the good results, but it gave the accuracy of 78.5 which was lesser than the first one and the train accuracy was 1.00, which showed it was also overfitted.

- ***Logistic Regression***: Logistic Regression gave the lowest results. As its accuracy was 68.5, which was lowest among all.

After considering the results, it was clear that it overfitted the data. So I used Hyperparameter Tunning to get the better results.

7. **Hyperparameter Tunning:** There are two methods to tune the models RandomizedSearchCV & GridSearchCV. To improve the performance of the models I have used RandomizedSearchCV. First parameter variable is created, which consists of parameters. Then the parameters & the model related to parameter is passed to the RandomizedSearchCV. RandomizedSearchCV is trained on the dataset & best parameters are obtained for the particular model. The models are again trained after applying the obtained parameters. As a result, it improved the test accuracy of the models. Figure 4 shows the comparative study after tunning.



	MLA Name	MLA Train Accuracy	MLA Test Accuracy	MLA Precision	MLA Recall	MLA AUC
0	DecisionTreeClassifier	0.925	0.925	0.916667	0.942857	0.924060
1	RandomForestClassifier	1.000	0.890	0.867257	0.933333	0.887719
2	LogisticRegression	0.695	0.690	0.752941	0.609524	0.694236

Figure 4: Comparative study table of models after tunning

The above table shows the best fit model is Decision Tree with the Test Accuracy of 92.5 & Train Accuracy of 92.5.

At last, I plotted a confusion matrix to check the type 1 & type 2 error. Figure 5 shows the confusion matrix.

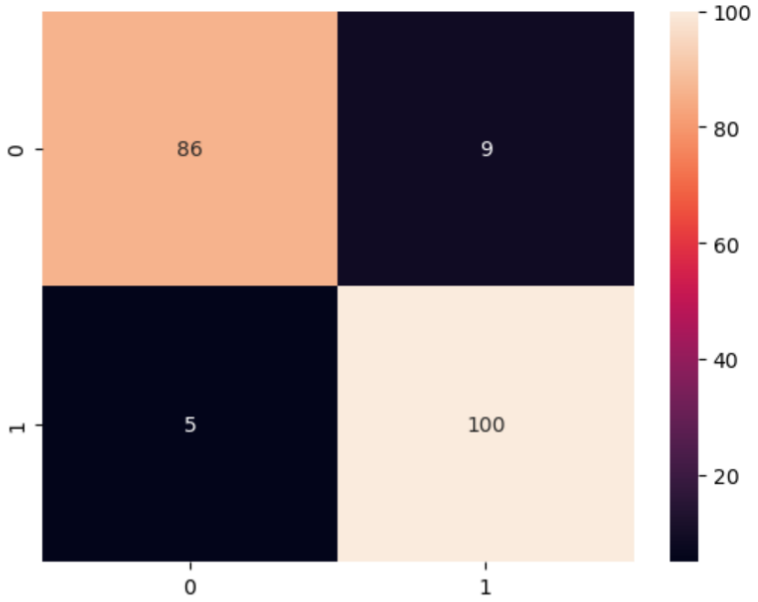


Figure 5: Confusion matrix to check type 1 & type 2 error

Part B:

The above model Decision Tree was trained on 20 columns. As the missing value percentage of F21 was 50%, I dropped that column. I scaled the test data as my model was trained on scaled data. As F21 column was removed in test data because of the pre processing of the data and was given to model for the prediction of the class. The predicted column is again added to the dataset.

Task 3:

Part A

1. **Import Libraries:**
   All the relevant libraries are imported such as pandas, numpy, matplotlib, seaborn. I have imported all the libraries in the starting itself

2. **Importing the data:** The dataset provided is imported using the pandas library.

3. **Data Analysis:**
   Analysis of data is a crucial step before checking for the best fit model. To get the information about data frame info is used, it helps to identify the categorical variable. If any categorical variable is found then it is converted into numerical variable. In the dataset provided it can be seen that column F20 & F27 have categorical variable so that needs to be changed to numerical. Unique function is used on F20 & F27 to find the unique value of the column. F20 is ordinal data & F27 is nominal data, they need to be converted into numerical.

4. **Feature Scaling:** To check the distribution of dataset histogram is plotted. It was analysed that column F14 is left skewed data & F15 is right skewed data. To convert it into normal data in F15 log transform is used & in F14 Square transform is used. Figure 6 shows the skewness of F14 & F15

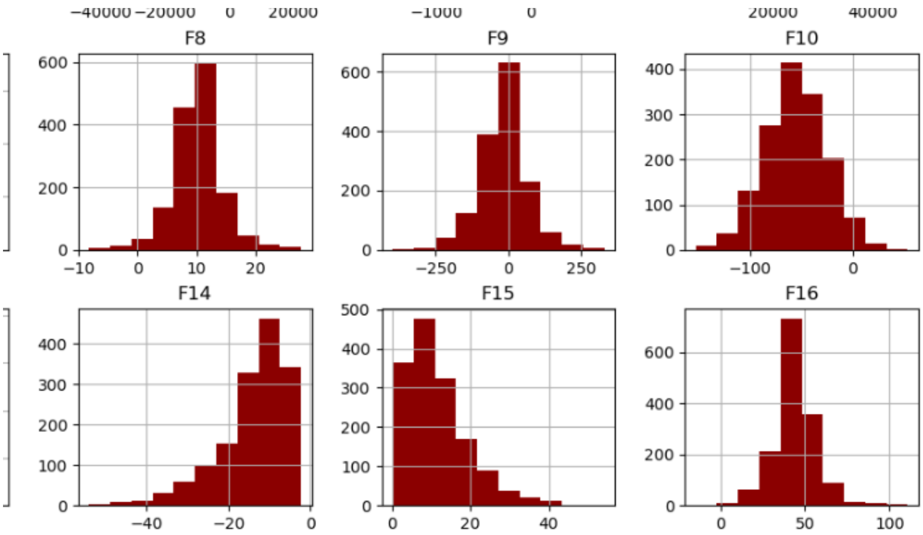
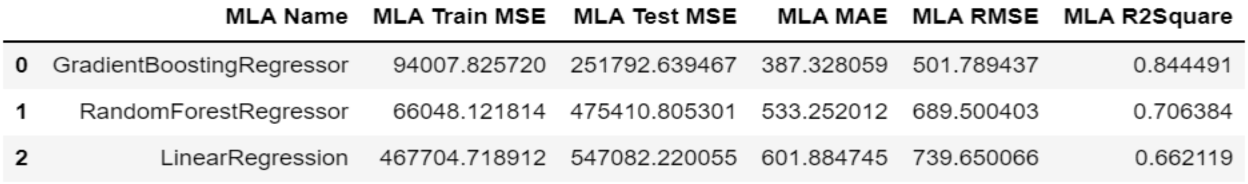


Figure 6: To show the skewness of F14 & F15

5. **Training different models:** After skewing the column F14 & F15, I trained the three different models namely Gradient Boosting Regressor, Linear Regression & Random Forest Regressor. The Target column is predicted after training the dataset on the models. Additionally, the real Target column values with the predicted Target column values using the R2 score are compared. I evaluated the score of the model by seeing the MAE, MSE & RMSE. Figure 7 shows the comparative study of different models



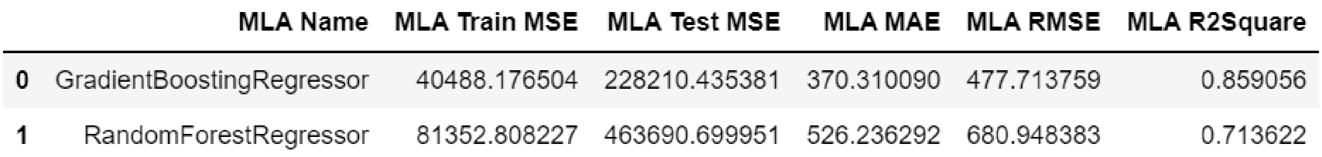
	MLA Name	MLA Train MSE	MLA Test MSE	MLA MAE	MLA RMSE	MLA R2Square
0	GradientBoostingRegressor	94007.825720	251792.639467	387.328059	501.789437	0.844491
1	RandomForestRegressor	66048.121814	475410.805301	533.252012	689.500403	0.706384
2	LinearRegression	467704.718912	547082.220055	601.884745	739.650066	0.662119

Figure 7: Comparative Study of different models

- ***Gradient Boosting Regressor:*** After removing the skewness of the dataset the R2 Score of Gradient Boosting Regressor is 84.4 and the error is 387.32
- ***Random Forest Regressor:*** In this model the R2 score is 70.63 & the error is 533.25, which is quite high from the previous model.
- ***Linear Regression:*** This model shows the lowest R2 Score as compare to other models of 66.21 & the error is quite high 739.65.

After seeing the above data, I evaluated how to get better R2 score & less error, and decided to use hyperparameter tunning.

6. **Hyperparameter Tunning:** After applying Hyperparameter tunning the R2 Score showed little improvement and the error decreased. Figure 8 shows the comparative study after applying Hyperparameter tunning.



	MLA Name	MLA Train MSE	MLA Test MSE	MLA MAE	MLA RMSE	MLA R2Square
0	GradientBoostingRegressor	40488.176504	228210.435381	370.310090	477.713759	0.859056
1	RandomForestRegressor	81352.808227	463690.699951	526.236292	680.948383	0.713622

Figure 8: Comparative study after Hyperparameter Tunning

On the basis of above table, Gradient Boosting Regressor is the best fit model with 85.9 R2 Score & 370.31 error.

Part B:

As per the above table Gradient Boosting Regressor is the best model. It was trained on model after converting the categorical variable into numerical variable. To remove the left & right skewness of the dataset log transform and square transform are used.

Conclusion:

To conclude, as shown in above analysis there are different ways to increase the accuracy & R2 score of the used models. In task 2, Decision Tree Classifier is the best model with the test accuracy of 92.5. Gradient Boosting Regressor is giving the best R2 score of 85.9 & 370.31 error.

References:

Scikit-learn online documentation and tutorials : https://scikit-learn.org

https://stackoverflow.com for guidance on codes

Lecture & Lab notes on machine learning