University of Essex

Comparative study

Design and Application of a Machine Learning System for a Practical Problem

Ishita Agarwal
2201110

Word Count:1505

School of Computer Science and Electronic Engineering
CE802 Machine Learning

Introduction

This report will briefly discuss the performance of various models that were used and how among all, the best model that was implemented to predict the solutions for 2&3 part of the study.

Problem Statement

We will look at the performance of each customer and a label representing whether the customer belongs to the class of people who had problems with paying their energy bills during the last few months (the value TRUE in the data means "Yes, they suffered from the increasing energy prices"

Exploratory Data Analysis

To comprehend the data's qualities and relationships, we first analyse it. We start with below **Data Enrichment** steps:

Eliminate redundant observations:

We run df.drop_duplicates() to eliminate duplicate records from the dataset that could bias any analysis's results. But since we have no duplicates in our database we proceed to next cleansing step.

Fix missing data:

df.isnull().sum(): We check if there is any missing values as isnull() will indicate whether the corresponding entry is missing or not s**um()** method will return the number of missing values for each column in the df.

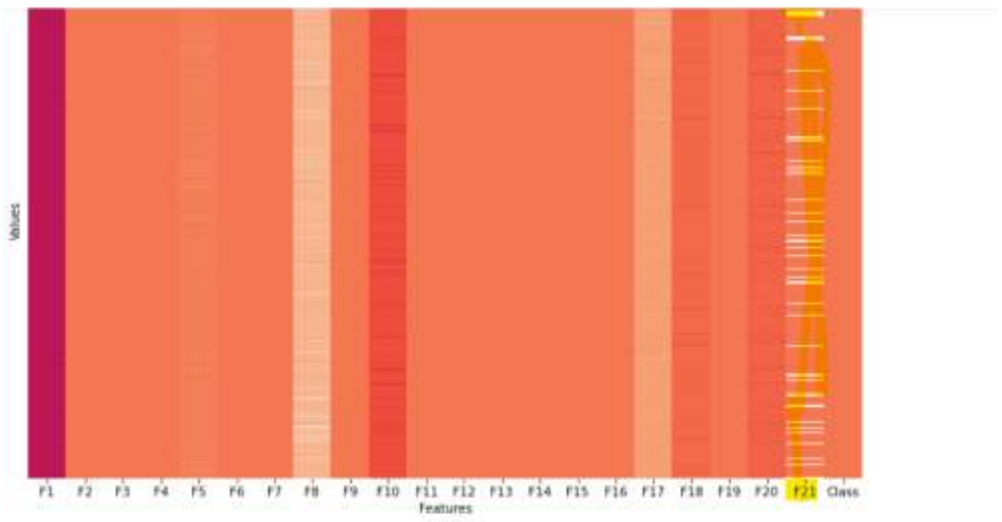
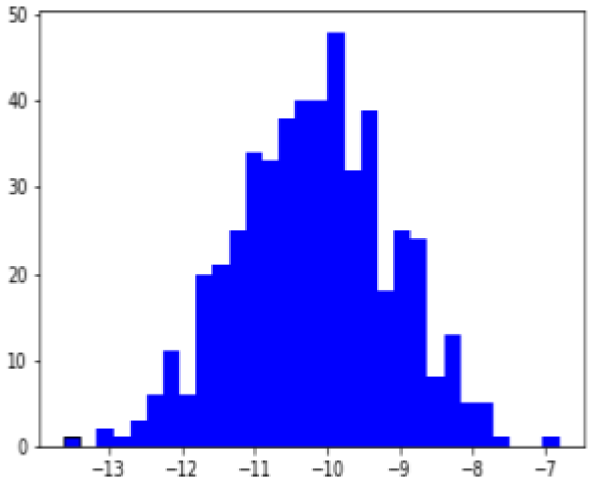
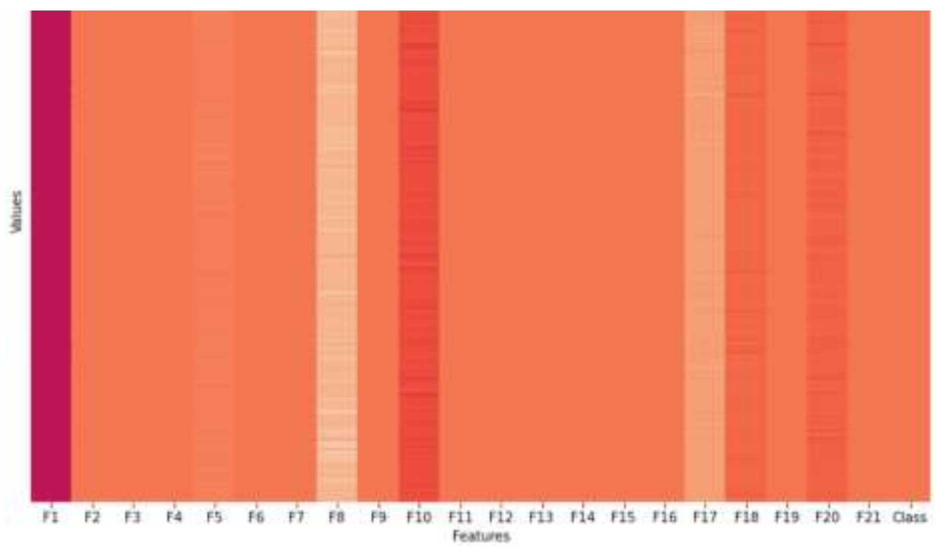


Fig: Depicting missing values in F21

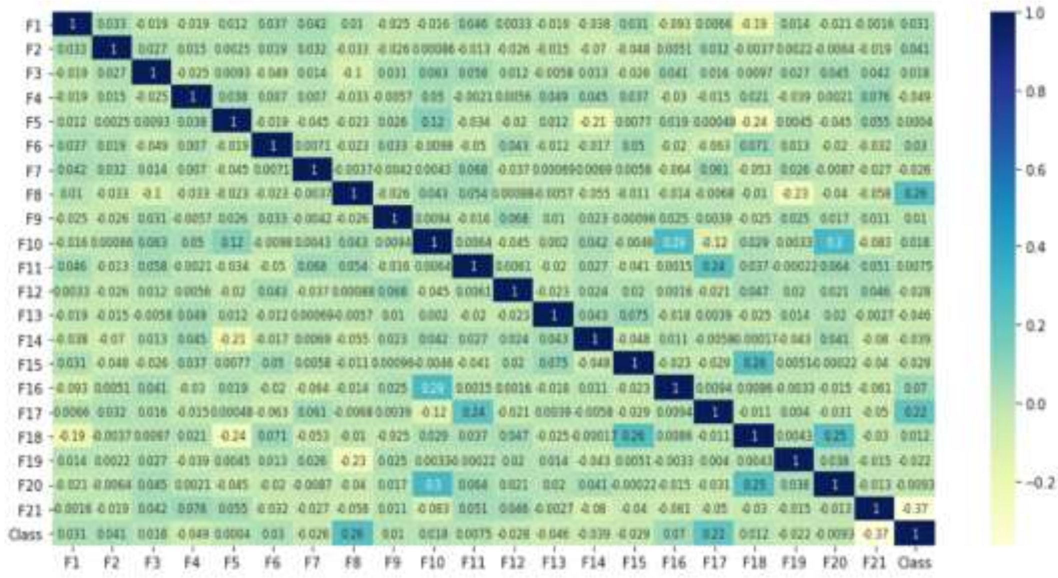
- **Mean Imputation**: Before Imputing we try to understand the distribution of a dataset.



We then replace the missing values data with mean using SimpleImpute a pre-processing tool.



- **<u>Visualization of Missing values</u>:**Using heatmap we infer that all features are negatively co-related with the target variable.



- **<u>Encoding:</u>**We replace the values of target variable from boolean to binary values to 1 and 0

Splitting the dataset

We divide the data into train and test using the train_test_split function, in order to get a more accurate estimate of its performance on unseen data. The train_test_split function will divide the dataset into 80% train dataset and 20% test data set. A random state will help us randomly select the samples for the test set.

Part 2a.

DECISION TREES

Model Selection

We procced to create a DecisionTreeClassifier which will be useful in our classification problem. Among the several parameters we have used below parameters to control the behaviour of the algorithm.

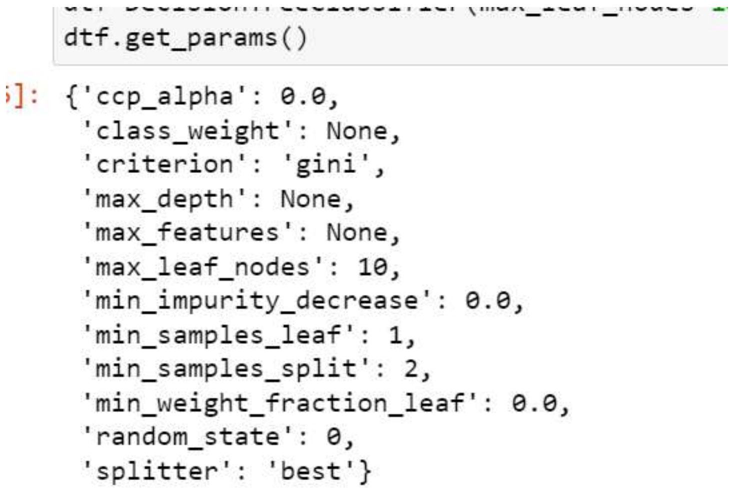
Optimal parameters

max_leaf_nodes: The maximum number of leaf nodes in the tree.

random_state: The seed used by the random number generator.

We then call   fit method to train the decision tree classifier on the provided training data and target labels and creates a decision tree model that can be used to make predictions on new data.

We then to examine the probability for each sample in test using predict_proba method.



```
dtf.get_params()

{'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': None,
 'max_leaf_nodes': 10,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'random_state': 0,
 'splitter': 'best'}
```

Model Evaluation

The score method in the DecisionTreeClassifier class's score method to assess how well the trained model performed on a test dataset.Score closer to 1 denotes perfect accuracy.

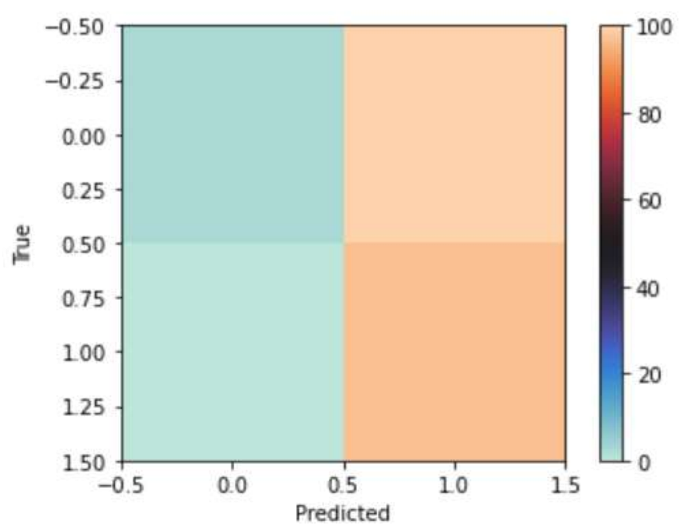
Decision Tree Classifier Score: 0.895 89.5% of the samples are correctly predicted.

Precision Score: 0.8877551020408163 is a metric that measures the proportion of true positive predic tions out of all positive predictions made by the model. A high precision score indicates that the model h as a low number of false positives.

Recall Score: 0.8969072164948454: measures the proportion of true positive predictions out of all act ual positive samples. A high recall score indicates that the model has a low number of false negatives.

Confusion Matrix

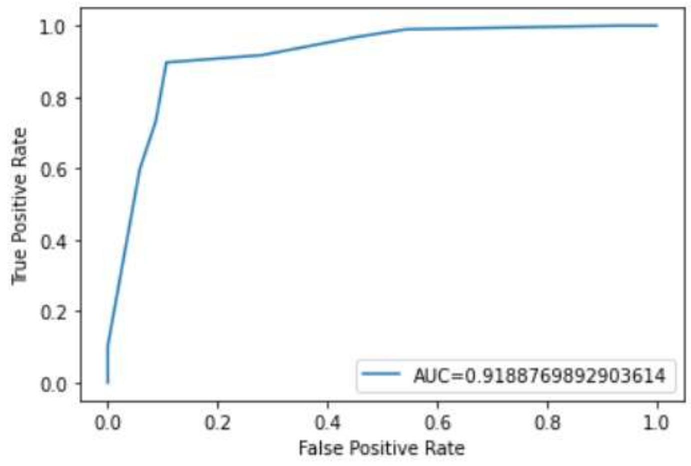
A confusion matrix is a table that is used to define the performance of a classification algorithm



ROC Curve & AUC Value

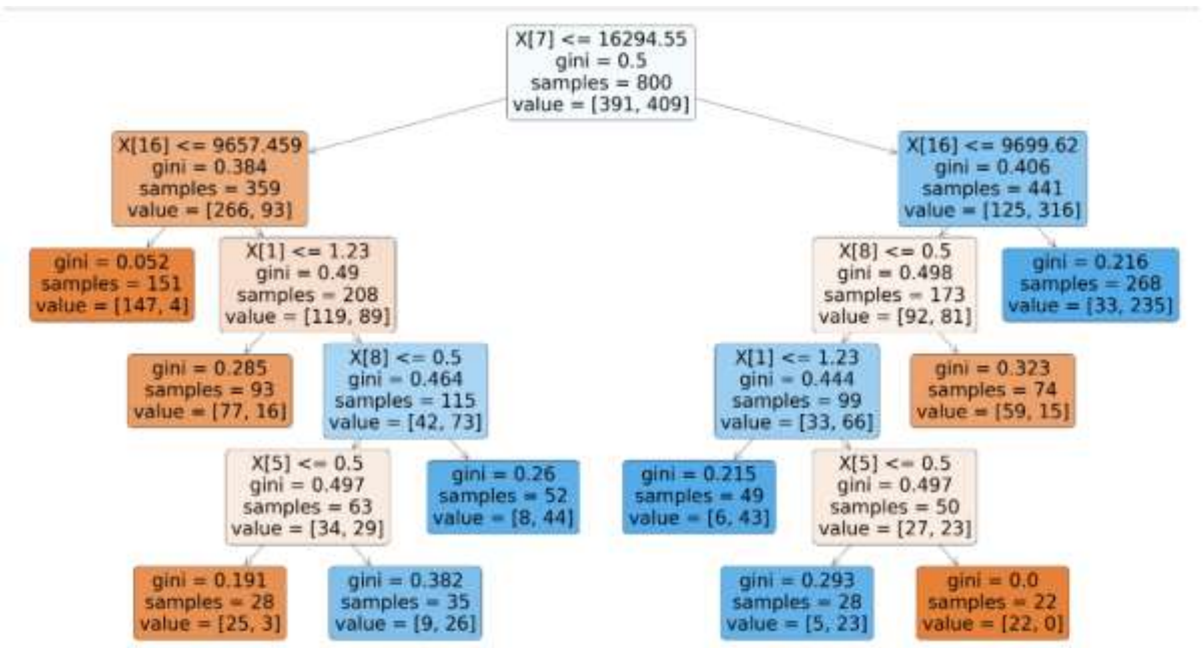
We visualize the performance using ROC curve which plots true positive rate against the false positive rate.

We also achieve the value Area under the curve closer to 1 representing a perfect classifier.



Model Visualization

We then visualize the decision tree using the plot tree function where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label.



2B. Model Selection

RANDOM FOREST

Performing cross validation using the the GridSearchCV class where the combination of parameters to return the best set of parameters that will yield the highest performance.

- **cv:** Number of folds in k-fold cross-validation.
- **verbose**: Controls the verbosity: the higher, the more messages.
- **n_jobs :** Number of CPU cores used when parallelizing over classes if multi-class.

Hyperparameters

- The number of trees in the forest (**n_estimators)** is set to a range of values from 200 to 1000, with 10 increments.
- The number of features to consider at each split (**max_features**) is set to either 'auto' or 'sqrt'. The maximum number of levels in the tree (**max_depth)** is set to 2 and 4.
- The minimum number of samples required to split a node (**min_samples_split**) is set to 2 and 5,
- the minimum number of samples required at each leaf node (**min_samples_leaf**) is set to 1 and 2.
- The method of selecting samples for training each tree (**bootstrap)** is set to either True or False.

- These settings are then combined into a dictionary called **'random_grid'** which will be used later to search for the optimal set of hyperparameters for the mode

Optimal parameters:

rf_random.best_params_ is an attribute the best hyperparameters found by the randomized search.

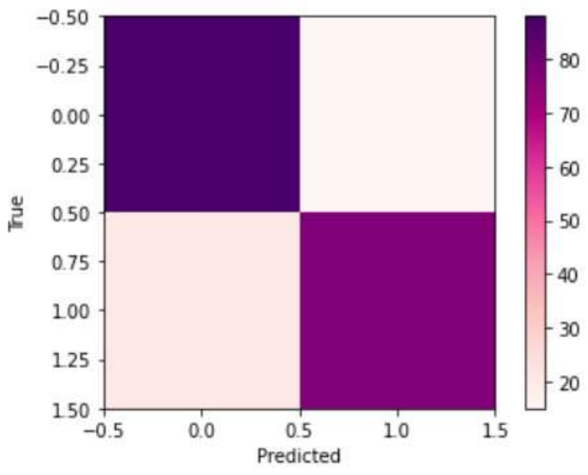
```
{'bootstrap': False,
 'max_depth': 4,
 'max_features': 'auto',
 'min_samples_leaf': 2,
 'min_samples_split': 5,
 'n_estimators': 822}
```

Model Evaluation:

Random Forest Classifier Score: A score of 0.825 means that the model is able to correctly predict the class labels of new instances with an accuracy of 82.5%.

Precision Score: 0.8369565217391305 83.7% of them are actually positive

Recall Score: 0.7938144329896907 79.4% of the actual positive instances as positive.

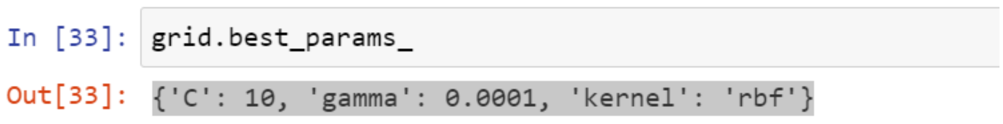


SUPPORT VECTOR MACHINE:

We perform cross validation using the the GridSearchCV class where the combination of parameters to return the best set of parameter that will yield the highest performance.

The param_grid in this case includes the following parameters:
- C: Range of values to choose the best margin value
- gamma : Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
- kernel: Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid'

Optimal parameters:



```
In [33]: grid.best_params_
Out[33]: {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
```

Model Evaluation:

SVM Score: 0.5 Indicates that the model correctly classifies half of the data points.

Precision Score: 0.492 :49% of the data points that the model classified as positive are actually positive.

Recall Score: 1.0: That it is good at identifying all of the positive examples, but is not very good at distinguishing between positive and negative examples.

CONCLUSION:

We have trained our dataset which each model and evaluated its performance, therefore infering that Decision Tree is the best performing model to predict our problem statement. This model was then successfully run to predict our test data.

Part 3

Problem Statement

To predict the variation in the annual expenditure that a customer is going to have due to the increase of the energy cost (in £/year, positive if next year the customer is going to spend more, negative if they are going to spend less).

Exploratory Data Analysis:

Categorical Variable to Numerical Variable

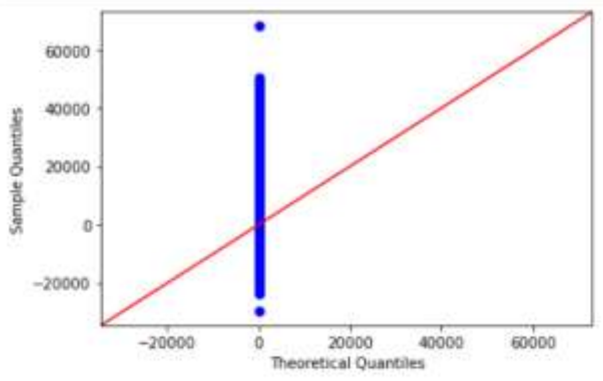
The pd.get_dummies() function is used to convert categorical variables in the columns specified, F34, into numerical variables by creating a new binary column for each unique category in the original column.

And for F5, I have replaced the values 'Very low', 'Low', 'Medium', 'High', and 'Very high' with 0, 1, 2, 3, and 4 respectively.

The drop_first=True argument drops the first level of each categorical variable, reducing the number of columns created by one and avoiding the creation of a perfect multicollinearity.

Q-Q plot

. A Q-Q plot is a graphical tool for checking if data is normally distributed. A 45-degree line is added to the plot using the line='45' argument. If the points in the Q-Q plot fall approximately on the 45-degree line, it suggests that the data is approximately normally distributed



Scaling of Features

I have used **score** function standardize the data in the DataFrame df. It uses standard deviations a value.

LINEAR REGRESSION

Model Evaluation

R-squared score (R2_score) A value of 0.715 is achieved inferring that the model explains 71.5% of the variance in the data.

The mean squared error (MSE) is 0.26204503318207095 calculating the average squared difference between the predicted values and the actual values.

The mean absolute error (MAE) is 0.4128957436887381 is a measure of the average difference between the predicted values and the actual values.

3B. Model Tuning

LASSO REGRESSION

Hyperparameters Tuning:

Performing cross validation using the the GridSearchCV class where the combination of parameters to return the best set of parameters that will yield the highest performance.

- cv: Number of folds in k-fold cross-validation as 5
- alpha: Using range of alpha values to find the best one for Lasso regression

Optimal parameters:

- The best alpha value achieved is 0.02

Model Evaluation:

The R-squared score (R2_score) :A value of 0.7026102192733805 means that the model explains 70.26% of the variance in the data.

The mean squared error (MSE) is 0.2736100731390656.

The mean absolute error (MAE) is 0.421961440458994 14

RIDGE REGRESSION

Hyperparameters Tuning:

Performing cross validation using the the GridSearchCV class where the combination of parameters to return the best set of parameters that will yield the highest performance.

- cv: Number of folds in k-fold cross-validation as 5
- alpha: Using range of alpha values to find the best one for Ridge regression

Optimal parameters:

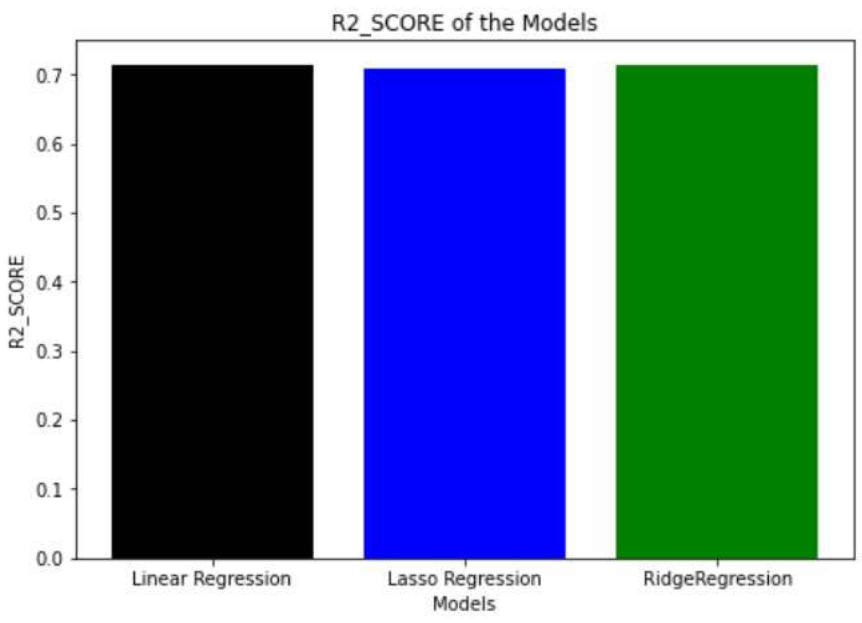
- The best alpha value achieved is 30

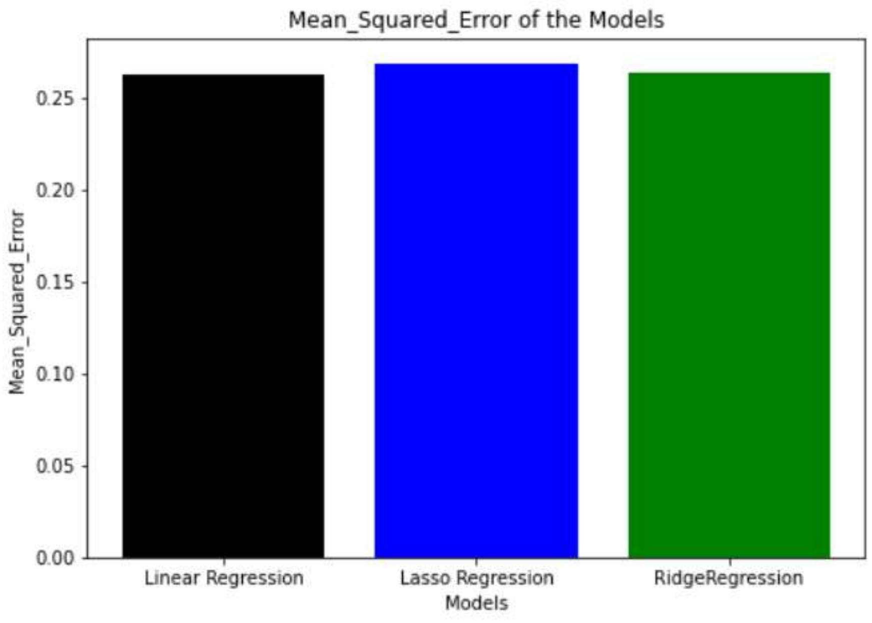
Model Evaluation:

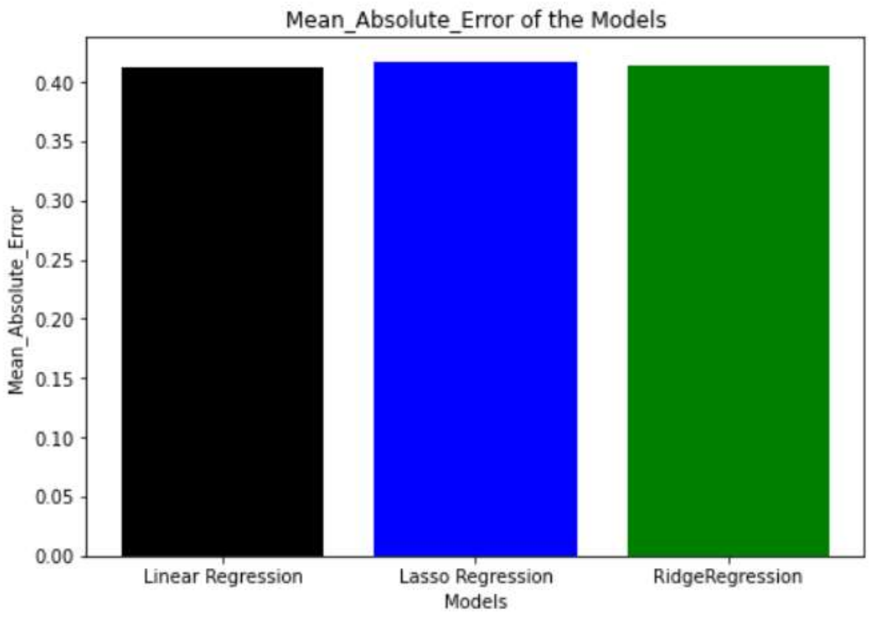
The R-squared score (R2_score) :A value (0.712) 71.23% means that the model explains 70.26% of the variance in the data.

The mean squared error (MSE) is 0.2646

The mean absolute error (MAE) is 0.415


R2_SCORE of the Models
R2_SCORE
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Linear Regression
Lasso Regression
RidgeRegression
Models


Mean_Squared_Error of the Models
Mean_Squared_Error
0.00
0.05
0.10
0.15
0.20
0.25
Linear Regression
Lasso Regression
RidgeRegression
Models


Mean_Absolute_Error of the Models
Mean_Absolute_Error
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Linear Regression
Lasso Regression
RidgeRegression
Models

<u>CONCLUSION:</u>

R-squared and Mean Squared Error (MSE) scores have been used to compare and contrast the three models. Since the Linear Regression model has the highest R-squared of 0.7, we will implement the same model on our test data set.