Text

Description automatically generated

Toronto, Canada

**A Report on**

Module 2 Project: Building the Car of the Future

Predictive Analytics

(ALY 6020)

Guided by:

Prof. Dr. Harpreet Sharma

|  |  |  |
| --- | --- | --- |
| Name of Student | Submitted By:  NUID | Date of submission |
| Parth Shah | 002956963 | 25th January, 2023 |

**Table of Figures**

Figure 1: Importing Libraries--------------------------------------------------------------------------------3

Figure 2: Importing Dataset----------------------------------------------------------------------------------4

Figure 3: Null Values-----------------------------------------------------------------------------------------5

Figure 4: Unique Values--------------------------------------------------------------------------------------6

Figure 5: Horsepower column with NaN values----------------------------------------------------------6

Figure 6: Filling NaN values with mean in Horsepower-------------------------------------------------7

Figure 7: Distribution of MPG------------------------------------------------------------------------------8

Figure 8: Correlation Plot------------------------------------------------------------------------------------9

Figure 9: Splitting Dataset-----------------------------------------------------------------------------------9

Figure 10: Linear Regression ------------------------------------------------------------------------------10

Figure 11: Accuracy of Model-----------------------------------------------------------------------------12

Figure 12: Regression Model with five variables--------------------------------------------------------12

Figure 13: Regression Model with four attributes--------------------------------------------------------13

**Introduction**

* The automobile dataset is a collection of statistics on different car makes and models. It includes details like model year, year, engine size, cylinder count, horsepower, weight, acceleration, US Made, and miles per gallon (mpg) ratings.
* The mpg of an automobile may frequently be predicted using this dataset, which is frequently used for machine learning and data analysis.

**Part-1**

Text, letter

Description automatically generated

Figure 1: Importing Libraries

* This code imports various libraries and modules that are commonly used in data analysis and machine learning.
* warnings is used to suppress warning messages that may be generated by the code.
* numpy is a library for mathematical and scientific computing and is often used for working with arrays and matrices.
* Pandas are a library for data manipulation and analysis, and is often used for working with data in the form of dataframes.
* matplotlib and seaborn are libraries for data visualization, which are commonly used to create plots and charts.
* statsmodels.api is a library for statistical modeling, which is often used for linear and logistic regression, hypothesis testing, and other statistical analyses.
* sklearn.preprocessing contains classes for data preprocessing, such as StandardScaler and MinMaxScaler which are used to standardize and normalize the data.
* sklearn.model\_selection contains classes for splitting data into training and test sets.
* sklearn.linear\_model contains classes for linear regression model such as LinearRegression.
* sklearn.metrics contains classes for evaluating model performance, such as mean absolute error and mean squared error.
* r2\_score is used to measure the goodness of fit of the model.
* You can use these libraries and modules to perform various data analysis and machine learning tasks on the mpg dataset, such as data cleaning, visualization, model building, and evaluation.

A picture containing text

Description automatically generated

Figure 2: Importing Dataset

* The "car.csv" dataset is read into the variable "data" by this code block from the supplied file path, "D:/PARTH SHAH/R/ALY 6020/Module 2 Project/car.csv." Using the pd.read csv() method, a CSV file may be read into a pandas dataframe.
* Which values should be treated as missing data is determined using the na values argument. If a cell in the CSV file has the value "?" as in this instance, it will be considered as missing data and given the value np.nan (Not a Number) in the output dataframe.
* The dataframe's first five rows are shown using the data.head(5) command. This is a practical method for swiftly scanning the dataframe's structure and contents.

Graphical user interface, text

Description automatically generated

Figure 3: Null Values

* This code counts the number of missing values in each column and looks for any null or missing values in the dataframe.
* If any of the values in the original dataframe are null, the data.isnull() function will produce a dataframe with True in those locations and False in all other locations.

Text, letter

Description automatically generated

Figure 4: Unique Values

* This code block extracts the dataframe's "Horsepower" column, which is then used to check the column's unique values using the unique() method. This will produce an array containing each distinct value found in the column labelled "Horsepower."
* You may determine whether a column has any non-numeric values or empty cells by using the unique() method on the column.

Table

Description automatically generated

Figure 5: Horsepower column with NaN values

* This code block searches the dataframe for any rows with NaN values in the "Horsepower" column. Rows from the dataframe are chosen using the loc function depending on specific criteria.
* In this case, it is looking for rows where the 'Horsepower' column value is 'NaN,' and it will only return those rows where the criteria is satisfied.

Table

Description automatically generated

Figure 6: Filling NaN values with mean in Horsepower

* The 'Horsepower' column's mean is calculated using this code block, and any empty spaces in that column are filled with the result.
* The mean of the "Horsepower" column is determined using the mean() function, and it is then assigned to the variable "average" in the first line of code.
* The next line of code data['Horsepower'].fillna(average, inplace=True) uses the fillna() function to fill any missing values in the 'Horsepower' column with the mean value. The inplace=True parameter is used to modify the dataframe in place, rather than returning a new dataframe with the missing values filled in.

Chart, histogram

Description automatically generated

Figure 7: Distribution of MPG

* This code block plots the distribution of the dataframe's "MPG" column using the displot() function from the seaborn library. When the kde parameter is set to True, the data density plot is displayed.
* We can see from this plot how the "MPG" column is distributed, including the range of values, the number of observations at each value, and whether the data is symmetric or asymmetric.

A picture containing table

Description automatically generated

Figure 8: Correlation Plot

* This code block shows the dataframe's correlation using the heatmap() function from the Seaborn library. With the exception of NA/null values, the data.corr() method determines the pairwise correlation of columns.
* The correlation plot allows you to see how closely several columns in a dataframe are correlated with one another. Each cell in the grid created by the heatmap() method reflects the correlation between two columns and is color-coded.

**Part - 2**

A picture containing text

Description automatically generated

Figure 9: Splitting Dataset

* This code block is used to split the dataframe into training and testing sets. The goal is to use the training set to train a model, and then use the testing set to evaluate the performance of the model.
* The variables x and y are being defined in this code block. x is assigned the values of all the columns from the dataframe except the column 'MPG' and y is assigned the values of the column 'MPG' from the dataframe. This is done using the iloc[] function which is used to select columns by their index.
* The train\_test\_split() function is used to split the data into training and testing sets. The function takes in four parameters: x\_train, x\_test, y\_train, y\_test. The first two parameters, x\_train and x\_test are assigned the values of the independent variables in the dataframe (x), and the latter two parameters, y\_train and y\_test are assigned the values of the dependent variable in the dataframe (y).
* The test\_size parameter is set to 0.20, which means that 20% of the data will be used.

Text

Description automatically generated

Figure 10: Linear regression

* The code above is creating a linear regression model (lm) using the LinearRegression() function from scikit-learn library. Then, it creates an object (sfs) of the Sequential Feature Selector (SFS) class, which is a feature selection algorithm that selects the best subset of features using a greedy search. The parameters passed to the SFS class are:
* The linear regression model (lm)
* The number of features to select (k\_features = 7)
* The search direction, forward or backward (forward = True)
* Whether to use floating or fixed selection (floating = False)
* The scoring metric to use (scoring = "r2")
* The number of cross-validation folds (cv = 5)
* The number of parallel jobs to run (n\_jobs = -1)
* Then the sfs is fit on the training data (x\_train, y\_train) and the final feature subset is returned.
* Finally, the get\_metric\_dict method is used to get the dictionary containing the performance metric of each feature subset and it is converted to a pandas dataframe and displayed.
* Here Acceleration and Model Year are major factors related to MPG.
* While good acceleration can reduce fuel economy when accelerating, it can also increase fuel efficiency when maintaining speed or making optimum use of the car's power.

Text

Description automatically generated

Figure 11: Accuracy of Model

* The code you provided is training a linear regression model (lm) using the fit() method on the training data (x\_train, y\_train). The fit() method is used to estimate the model parameters using the training data.
* Here, the accuracy of the model is 83 percent.

**Part - 3**

Table

Description automatically generated

Figure 12: Regression Model with five attributes

* After splitting the data, the code is adding a constant term to the input features using the add\_constant() function from the statsmodels library. This is necessary for estimating the model's intercept term.
* Then, the code is fitting an ordinary least squares (OLS) model to the training data using the sm.OLS() function from the statsmodels library. The OLS model is a statistical method for analyzing linear relationships between a response variable and one or more predictor variables. The function takes the target variable (y\_train) and input features (x) as inputs.
* Finally, the code is printing a summary of the model's performance using the summary() method. The summary includes information such as the coefficients of the model, the R-squared value, the adjusted R-squared value (0.829), and the p-values for each feature. The summary can be used to determine which features are most important in predicting the target variable, as well as to assess the overall fit of the model to the data.

Table

Description automatically generated

Figure 13: Regression Model with four variables

* With the four variables, we get r squared value is 0.827 for this model.

**Conclusion**

* In the regression model, we get the r-square value 0.83, which means this model cover 83 percent data points for the dataset.
* The two variables displacement and model year were the only ones that were positively co-related to the dependent variable, according to figure 12.
* Furthermore, if the co-efficients of the variables are examined, it can be concluded that a change in the dependent variable would result in an equivalent change in the co-efficient value of the corresponding independent variable if the independent variable changes by one unit.

**Reference**

* Kumar, A. (2020, July 30). *Sequential Forward Selection - Python Example - Data Analytics*. Data Analytics. Retrieved January 25, 2023, from <https://vitalflux.com/sequential-forward-selection-python-example/>
* *Linear Regression (Python Implementation) - GeeksforGeeks*. (2017, March 19). GeeksforGeeks. Retrieved January 25, 2023, from <https://www.geeksforgeeks.org/linear-regression-python-implementation/>
* Python, R. (n.d.). *NumPy, SciPy, and Pandas: Correlation With Python – Real Python*. NumPy, SciPy, and Pandas: Correlation With Python – Real Python. Retrieved January 25, 2023, from <https://realpython.com/numpy-scipy-pandas-correlation-python/>