Project Report

Content

Project Overview: Describe the Project along with Project Outcomes (Explain the Project in your own words in 15 – 20 lines)

The project objective is to create a perform linear regression analysis for the dataset mtcars and forecast on the dependent variable. This is to be accomplished using both R and Azure Machine Learning.

The data was extracted from a motor trend US magazine at year 1974 and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models).

The variables and a short description of them are as such:

1. **mpg - Miles/(US) gallon**
2. **cyl - Number of cylinders**
3. **disp - Displacement (cu.in.)**
4. **hp - Gross horsepower**
5. **drat - Rear axle ratio**
6. **wt - Weight (1000 lbs)**
7. **qsec - 1/4 mile time**
8. **vs - Engine (0 = V-shaped, 1 = straight)**
9. **am - Transmission (0 = automatic, 1 = manual)**
10. **gear - Number of forward gears**
11. **carb - Number of carburetors**

For the first part of the project, R programming is used. Data are required to be first separated accordingly to their own categories. Numeric, independent variables and other categorical data types are then later used to determine the correlation. With only numeric variables in hand, the correlation matrix is plotted to better visualize the correlation weightage between each independent variable. The best fit model is then selected to predict mpg data in Rstudio.

After which we explore Azure machine learning to compare the forecast and understand the functionality and comprehensiveness of both software.

1. **Project Technical Environment: (Describe the project with Tools and algorithm used)**

In a summary, there are a total of 10 activities to complete and achieve the desired prediction outcome of mpg. From activity 1 to 9, this project will utilise R programming to import, transform, explore, and extract the dataset from mtcars.

The dataset will then be used to visualise using the library packages (corrplot, caret, ggplot2, lattice) and to create the linear model for mpg prediction.

The activity 10 will utilise the Azure Machine Learning to train and create model for the mpg prediction model.

<u>Procedure for the project</u>

R Programming:

Step 1: Import dataset mtcars

Step 2: Checking summary and structure of the dataset respectively for distribution and data type

Step 3: Data transformation

Step 4: Separate independent/dependent variables

Step 5: EDA - visual to check the independent columns impacting mpg

Step 6: Check correlation for numeric variables

Step 7: Train model

Step 8: Plotting the model

Step 9: Prediction

Azure Machine Learning End to end solution:

Step 1: Load the dataset mtcars

Step 2: Preprocess the data

Step 3: Define the features

Step 4: Select an algorithm and train the model

Step 5: Score the model

Step 6: Evaluate the model accuracy by measuring the Coefficient of Determination

Activity 1: Define data management structures to align and streamline processes of data ownership, retrieval, combination, and usage (Explain the process of the data flow, retrieval, combination of both R and AML and its usage)

- **As the mtcars dataset is a built-in dataset in R, it can be retrieved either from inbuilt method of data() or by using read.csv.**

Sample Code:

#Load preset datasets

data(mtcars)

#Reading file csv file

mtcars<-read.csv("C:/Users/Downloads/mtcars.csv")

Variable Name		**Description**	**Comments**
mpg		Miles per gallon	Amount of distance travelled per gallon
cyl		Cylinder	Amount of cylinder in an engine
disp		Displacement	Overall volume in the engine
hp		Gross horsepower	Engine output power
drat		Rear axle ratio	Number of turns of drive shaft for every one rotation of wheel axle
wt		Weight	Overall weight of vehicle per 1000lbs
qsec		1/4 mile time	Fastest time to travel 1/4 mile from stationary in seconds
vs		V-shape / Straight line	Arrangement of cylinder in V-shaped = 0, Straight line = 1
am		Transmission type	Automatic = 0, Manual = 1
gear		No. of forward gears	No. of gear in transmission
carb		No. of carburetors	No. of carburetor barrels

Activity 2: Create R code for effective data loading, storage and utilization

Install the required visualisation packages (ggplot2, caret, corrplot, lattice) using the install.package

```r
#Activity 2
install.packages("ggplot2")
install.packages("caret")
install.packages("corrplot")

library(ggplot2)
library(caret)
library(corrplot)

```

The dataset is then analyse by using:

```
R 4.1.3 · ~/R Files/Project module/
>
> ##To get the names of the column
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
>
> ##To view the data type or structure
> str(mtcars)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : Factor w/ 3 levels "4","6","8": 2 2 1 2 3 2 3 1 1 2 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 2 2 2 ...
 $ am  : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
 $ gear: Factor w/ 3 levels "3","4","5": 2 2 2 1 1 1 1 2 2 2 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
>
> ##To view the data layout. By default the function produces the top 6 observation of the dataframe
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
>
> ##To view the data layout. By default the function produces the bottom 6 observation of the dataframe
> tail(mtcars)
                mpg cyl  disp  hp drat    wt qsec vs am gear carb
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
>
> ##To view the summary of the variables in term of min, max, mean and quartile
> summary(mtcars)
      mpg        cyl        disp             hp             drat             wt             qsec        vs      am
 Min.   :10.40   4:11   Min.   : 71.1   Min.   : 52.0   Min.   :2.760   Min.   :1.513   Min.   :14.50   0:18   0:19
 1st Qu.:15.43   6: 7   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1:14   1:13
 Median :19.20   8:14   Median :196.3   Median :123.0   Median :3.695   Median :3.325   Median :17.71
 Mean   :20.09          Mean   :230.7   Mean   :146.7   Mean   :3.597   Mean   :3.217   Mean   :17.85
 3rd Qu.:22.80          3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90
 Max.   :33.90          Max.   :472.0   Max.   :335.0   Max.   :4.930   Max.   :5.424   Max.   :22.90
 gear        carb
 3:15   Min.   :1.000
 4:12   1st Qu.:2.000
 5: 5   Median :2.000
        Mean   :2.812
        3rd Qu.:4.000
        Max.   :8.000
>
> ##To get the dimensions of the dataset in terms of number of rows and columns.
> dim(mtcars)
[1] 32 11
```

names(mtcars) – To retrieve the name of the columns

head(mtcars) – To view the data layout. By default, the function produces the top 6 observation of the dataframe tail(mtcars) – To view the data layout. By default, the function produces the bottom 6 observation of the dataframe

str(mtcars) – To view the data type or structure

summary(mtcars) – To view the summary of the variables in term of min, max, mean and quartile

dim(mtcars) – To get the dimensions of the dataset in terms of number of rows and columns. 32 observations and 11 variables were printed in the console

*Mention the number of records and fields from MTCARS in your project report

Dataset ‘mtcars’ is a data frame with 32 observations (rows) of 11 variables (columns)

Activity 3: Data Preprocessing, handling and updating standards and procedures

Factorise the variable cyl, am, vs and gear

```
#Activity 3
##Factor cyl, vs, am, gear as variables are ordered which can be factorised
mtcars$cyl <- as.factor(mtcars$cyl)
mtcars$am <- as.factor(mtcars$am)
mtcars$gear <- as.factor(mtcars$gear)
mtcars$vs <- as.factor(mtcars$vs)
```

Dropping dependent variable for calculating Multicollinearity(mpg)

```
#Subsetting mtcars without the dependent variable to mtcars_indpt_variables
mtcars_indpt_variables <- subset(mtcars[-1])
```

Display the new data and check if mpg is displayed or not

```
#Checking dataset without mpg
str(mtcars_indpt_variables)
```

```
> str(mtcars_indpt_variables)
'data.frame':   32 obs. of  10 variables:
 $ cyl : Factor w/ 3 levels "4","6","8": 2 2 1 2 3 2 3 1 1 2 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 2 2 2 ...
 $ am  : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
 $ gear: Factor w/ 3 levels "3","4","5": 2 2 2 1 1 1 1 2 2 2 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Activity 4: Identifying numeric variables

Identifying numeric variables using apply function and display

```
#Activity 4
#Identify numeric variables, sapply or lapply can used
#sapply provides a vector output compared to lapply which provides a list output
#create a dataset with only numeric independent variables called mtcars_numeric
mtcars_numeric <- sapply(mtcars_indpt_variables, FUN = is.numeric)
mtcars_numeric
```

```
> mtcars_numeric <- sapply(mtcars_indpt_variables, FUN = is.numeric)
> mtcars_numeric
  cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb 
FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE 
```

Observation is seen that disp, hp, drat, wt, qsec and carb are the numeric variables. It is then extracted to another data called *mtcars_numeric*

```
mtcars_numeric <- mtcars_indpt_variables[mtcars_numeric]
mtcars_numeric
```

```
> mtcars_numeric <- mtcars_indpt_variables[mtcars_numeric]
> mtcars_numeric
                     disp  hp drat    wt  qsec carb
Mazda RX4           160.0 110 3.90 2.620 16.46    4
Mazda RX4 Wag       160.0 110 3.90 2.875 17.02    4
Datsun 710          108.0  93 3.85 2.320 18.61    1
Hornet 4 Drive      258.0 110 3.08 3.215 19.44    1
Hornet Sportabout   360.0 175 3.15 3.440 17.02    2
Valiant             225.0 105 2.76 3.460 20.22    1
Duster 360          360.0 245 3.21 3.570 15.84    4
Merc 240D           146.7  62 3.69 3.190 20.00    2
Merc 230            140.8  95 3.92 3.150 22.90    2
Merc 280            167.6 123 3.92 3.440 18.30    4
Merc 280C           167.6 123 3.92 3.440 18.90    4
Merc 450SE          275.8 180 3.07 4.070 17.40    3
Merc 450SL          275.8 180 3.07 3.730 17.60    3
Merc 450SLC         275.8 180 3.07 3.780 18.00    3
Cadillac Fleetwood  472.0 205 2.93 5.250 17.98    4
Lincoln Continental 460.0 215 3.00 5.424 17.82    4
Chrysler Imperial   440.0 230 3.23 5.345 17.42    4
Fiat 128             78.7  66 4.08 2.200 19.47    1
Honda Civic          75.7  52 4.93 1.615 18.52    2
Toyota Corolla       71.1  65 4.22 1.835 19.90    1
Toyota Corona       120.1  97 3.70 2.465 20.01    1
Dodge Challenger    318.0 150 2.76 3.520 16.87    2
AMC Javelin         304.0 150 3.15 3.435 17.30    2
Camaro Z28          350.0 245 3.73 3.840 15.41    4
Pontiac Firebird    400.0 175 3.08 3.845 17.05    2
Fiat X1-9            79.0  66 4.08 1.935 18.90    1
Porsche 914-2       120.3  91 4.43 2.140 16.70    2
Lotus Europa         95.1 113 3.77 1.513 16.90    2
Ford Pantera L      351.0 264 4.22 3.170 14.50    4
Ferrari Dino        145.0 175 3.62 2.770 15.50    6
Maserati Bora       301.0 335 3.54 3.570 14.60    8
Volvo 142E          121.0 109 4.11 2.780 18.60    2
```

Activity 5: Data management tools / standard for Correlation Matrix and Correlated attributes

Calculating Correlation

```
#Activity 5
#Calculating correlation
mt_correlation <- cor(mtcars_numeric)


#Checking correlation within independent variables
mt_correlation
```

```
> mt_correlation
           disp         hp        drat         wt        qsec       carb
disp  1.0000000  0.7909486 -0.71021393  0.8879799 -0.43369788  0.3949769
hp    0.7909486  1.0000000 -0.44875912  0.6587479 -0.70822339  0.7498125
drat -0.7102139 -0.4487591  1.00000000 -0.7124406  0.09120476 -0.0907898
wt    0.8879799  0.6587479 -0.71244065  1.0000000 -0.17471588  0.4276059
qsec -0.4336979 -0.7082234  0.09120476 -0.1747159  1.00000000 -0.6562492
carb  0.3949769  0.7498125 -0.09078980  0.4276059 -0.65624923  1.0000000
```

Observation shows that there is a strong correlation between wt and disp as a cor value ot 0.8879799 is displayed. There are also some strong correlated variables ranging from 0.71 magnitude onwards.

We can refer to the below table to determine whether the correlation strength.
Variables which have correlation more than .8(-ve or +ve) will be dropped
Correlation checking is for numeric variables, not for factor/categorical

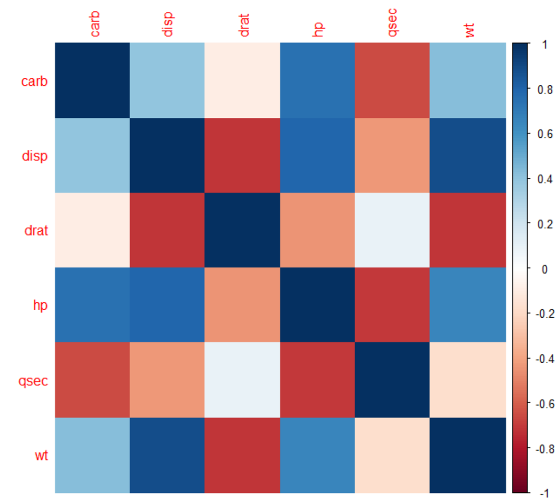
Size of correlation coefficient	Strength of correlation
.91 until 1.00 or -.91 until -1.00	Very strong
.71 until .90 or -.71 until -.90	Strong
.51 until .70 or -.51 until -.70	Moderate
.31 until .50 or -.31 until -.50	Weak
.01 until .30 or -.01until -.30	Very weak
.00	No correlation

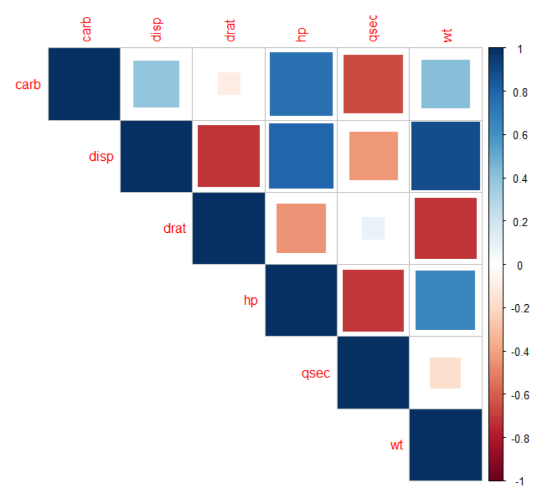
Visualise Correlation matrix

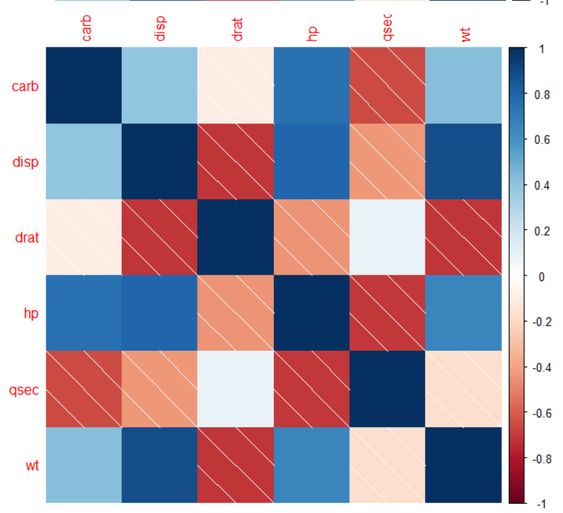
The corrplot() function is applied to allow us to easier identify strong correlated variables through a visual display instead of a numeric table.

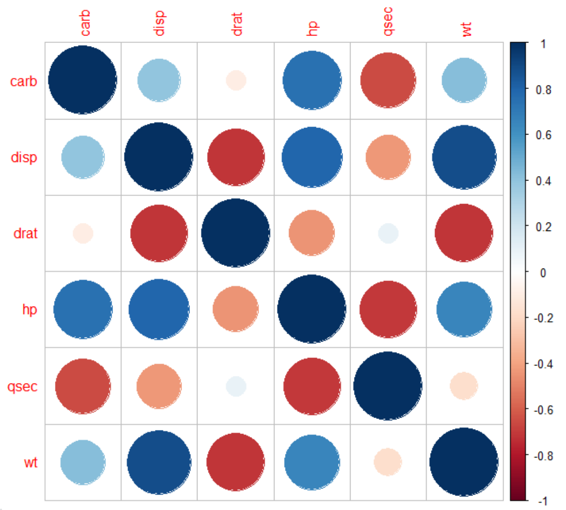
```
#Visualisation of mt_correlation
corrplot(mt_correlation, method = 'color', order = 'alphabet')
corrplot(mt_correlation, method = 'square', order = 'alphabet', type = 'upper')
corrplot(mt_correlation, method = 'shade', order = 'alphabet')
corrplot(mt_correlation, method = 'circle', order = 'alphabet')
```

In the syntax, method represents the kind of shape or representation it will display follow by the order which it will display. As for the second visualisation upper is applied to represent the data in a upper triangular fashion.









Identifying Variable Names of Highly Correlated Variables and print the variables

Using findCorrelation(), the highly correlated variables can be identified by indicating the cutoff value. The cutoff value that we will be using are 0.8 and 0.71.

```
#Identifying Variable Names of Highly Correlated Variables
high_rel_1 <- findCorrelation(mt_correlation, cutoff = 0.80)
high_rel_1

high_rel<-findCorrelation(mt_correlation, cutoff = 0.71)
high_rel

#Getting the column names of the variables
hig_rel_names<- colnames(mtcars_numeric[high_rel_1])
hig_rel_names

hig_rel_n<- colnames(mtcars_numeric[high_rel])
hig_rel_n
```

```
> #Identifying Variable Names of Highly Correlated Variables
> high_rel_1 <- findCorrelation(mt_correlation, cutoff = 0.80)
> high_rel_1
[1] 1
>
> high_rel<-findCorrelation(mt_correlation, cutoff = 0.71)
> high_rel
[1] 2 1 4
>
> #Getting the column names of the variables
> hig_rel_names<- colnames(mtcars_numeric[high_rel_1])
> hig_rel_names
[1] "disp"
>
> hig_rel_n<- colnames(mtcars_numeric[high_rel])
> hig_rel_n
[1] "hp"   "disp" "wt"
```

Remove highly correlated variables and create a new dataset
Create a variable x_new and x_new2 to store and create the new datasets.

```
> #Removing the variable showing high correlation, x_new to be variables without cor value more than 0.80
> x_new <- mtcars_numeric[, -which(colnames(mtcars_numeric) %in% hig_rel_names)]
> x_new
                     hp drat    wt  qsec carb
Mazda RX4           110 3.90 2.620 16.46    4
Mazda RX4 Wag       110 3.90 2.875 17.02    4
Datsun 710           93 3.85 2.320 18.61    1
Hornet 4 Drive      110 3.08 3.215 19.44    1
Hornet Sportabout   175 3.15 3.440 17.02    2
Valiant             105 2.76 3.460 20.22    1
Duster 360          245 3.21 3.570 15.84    4
Merc 240D            62 3.69 3.190 20.00    2
Merc 230             95 3.92 3.150 22.90    2
Merc 280            123 3.92 3.440 18.30    4
Merc 280C           123 3.92 3.440 18.90    4
Merc 450SE          180 3.07 4.070 17.40    3
Merc 450SL          180 3.07 3.730 17.60    3
Merc 450SLC         180 3.07 3.780 18.00    3
Cadillac Fleetwood  205 2.93 5.250 17.98    4
Lincoln Continental 215 3.00 5.424 17.82    4
Chrysler Imperial   230 3.23 5.345 17.42    4
Fiat 128             66 4.08 2.200 19.47    1
Honda Civic          52 4.93 1.615 18.52    2
Toyota Corolla       65 4.22 1.835 19.90    1
Toyota Corona        97 3.70 2.465 20.01    1
Dodge Challenger    150 2.76 3.520 16.87    2
AMC Javelin         150 3.15 3.435 17.30    2
Camaro Z28          245 3.73 3.840 15.41    4
Pontiac Firebird    175 3.08 3.845 17.05    2
Fiat X1-9            66 4.08 1.935 18.90    1
Porsche 914-2        91 4.43 2.140 16.70    2
Lotus Europa        113 3.77 1.513 16.90    2
Ford Pantera L      264 4.22 3.170 14.50    4
Ferrari Dino        175 3.62 2.770 15.50    6
Maserati Bora       335 3.54 3.570 14.60    8
Volvo 142E          109 4.11 2.780 18.60    2
> dim(x_new)
[1] 32  5
```

*disp is removed

```
> #Removing the variable showing high correlation, x_new2 to be variables without cor value more than 0.71
> x_new2 <- mtcars_numeric[, -which(colnames(mtcars_numeric) %in% hig_rel_n)]
> x_new2
                    drat  qsec carb
Mazda RX4           3.90 16.46    4
Mazda RX4 Wag       3.90 17.02    4
Datsun 710          3.85 18.61    1
Hornet 4 Drive      3.08 19.44    1
Hornet Sportabout   3.15 17.02    2
Valiant             2.76 20.22    1
Duster 360          3.21 15.84    4
Merc 240D           3.69 20.00    2
Merc 230            3.92 22.90    2
Merc 280            3.92 18.30    4
Merc 280C           3.92 18.90    4
Merc 450SE          3.07 17.40    3
Merc 450SL          3.07 17.60    3
Merc 450SLC         3.07 18.00    3
Cadillac Fleetwood  2.93 17.98    4
Lincoln Continental 3.00 17.82    4
Chrysler Imperial   3.23 17.42    4
Fiat 128            4.08 19.47    1
Honda Civic         4.93 18.52    2
Toyota Corolla      4.22 19.90    1
Toyota Corona       3.70 20.01    1
Dodge Challenger    2.76 16.87    2
AMC Javelin         3.15 17.30    2
Camaro Z28          3.73 15.41    4
Pontiac Firebird    3.08 17.05    2
Fiat X1-9           4.08 18.90    1
Porsche 914-2       4.43 16.70    2
Lotus Europa        3.77 16.90    2
Ford Pantera L      4.22 14.50    4
Ferrari Dino        3.62 15.50    6
Maserati Bora       3.54 14.60    8
Volvo 142E          4.11 18.60    2
> dim(x_new2)
[1] 32  3
```

*disp, wt and hp is removed from the dataset

Activity 6: Propose Model Creation

Build Linear Regression Model

Build a linear regression model based on the new variables, x_new and x_new2 that was created in activity 5.

Set y with our dependent variable mpg

```
y <- mtcars$mpg
```

```
#Creating model/ building model for x_new

y <- mtcars$mpg

fit_model_1<- lm(y ~.,data=x_new)
fit_model_2<- lm(y ~.,data=x_new2)
```

Checking summary

After creating the linear model, the next step is to check the summary of the linear model by adding summary() with the “fit” variable.

```
> summary(fit_model_1)

Call:
lm(formula = y ~ ., data = x_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7011 -1.5939 -0.2785  0.9175  5.3075 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 18.97886   10.45652   1.815 0.081073 .  
hp          -0.01409    0.01629  -0.865 0.394817    
drat         1.99965    1.36766   1.462 0.155698    
wt          -3.56476    0.92705  -3.845 0.000699 ***
qsec         0.46303    0.45231   1.024 0.315407    
carb        -0.28748    0.49790  -0.577 0.568646    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.571 on 26 degrees of freedom
Multiple R-squared:  0.8473,	Adjusted R-squared:  0.818 
F-statistic: 28.86 on 5 and 26 DF,  p-value: 7.865e-10
```

```
> summary(fit_model_2)

Call:
lm(formula = y ~ ., data = x_new2)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.0161 -2.4474 -0.1745  1.4097  7.9116 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -4.8172     9.9557  -0.484  0.63224    
drat          7.1501     1.1577   6.176 1.14e-06 ***
qsec          0.2197     0.4572   0.480  0.63463    
carb         -1.6813     0.5058  -3.324  0.00248 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.429 on 28 degrees of freedom
Multiple R-squared:  0.7076,	Adjusted R-squared:  0.6763 
F-statistic: 22.59 on 3 and 28 DF,  p-value: 1.234e-07
```

Extract the coefficient using the summary() function and use the subset to extract the coefficient, “$coeff”.

```
> #Checking coefficients
> summary(fit_model_1$coefficients)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-3.5648 -0.2191  0.2245  2.9292  1.6155 18.9789 
> summary(fit_model_2$coefficients)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-4.8172 -2.4653 -0.7308  0.2178  1.9523  7.1501 
```

Activity 7: Plot model

- **Plot the fit model in a 2*2 matrix using par**

The following task is to plot the model in a 2*2 matrix by using the par() function.

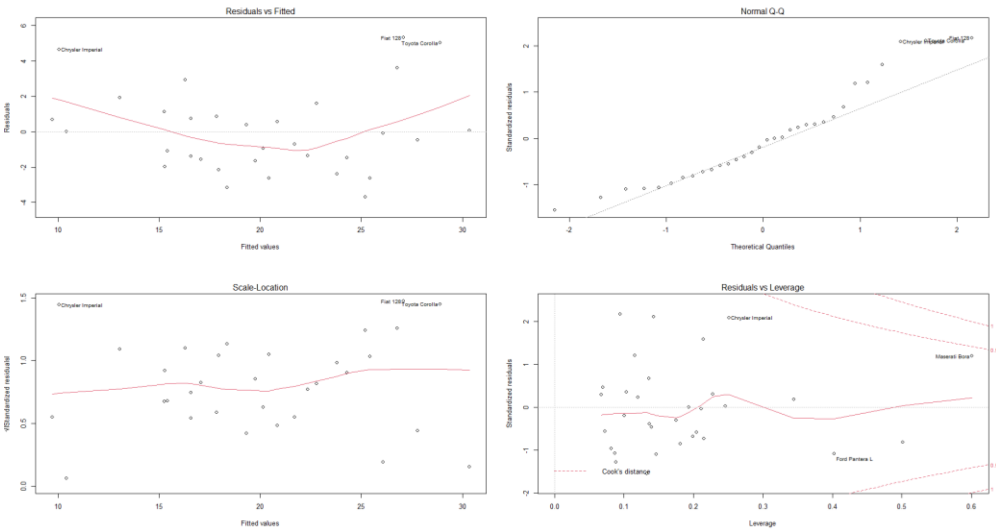
```
##Activity 7
#Plotting the model

saved_par <- par()
par(mfrow=c(2,2))
plot(fit_model_1)
plot(fit_model_2)
```

We first save our default par settings in saved_par in case for easy retrieve.

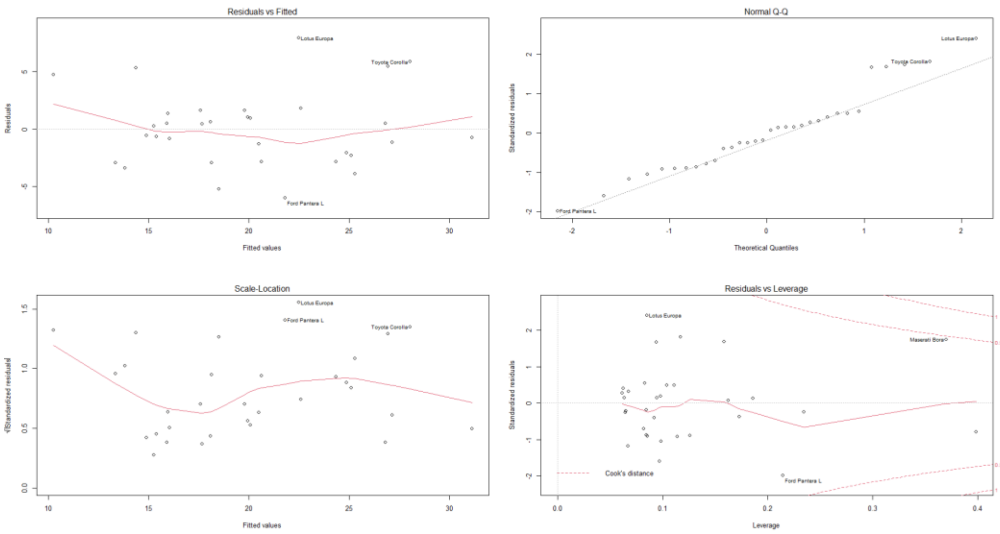
Both fit model are then plot in a 2,2 setting.

This is the plot output after using par and the plot function

Fit_model_1



Fit_model_2



Activity 8: Establish internal processes to Calculating Model Performance, monitor compliance of data with relevant metrics procedure

Extracting R-squared value

In this activity, we will be extracting r-squared from the summary() function with the line, “$r.squared”. The r-squared 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.

```
> summary(fit_model_2)$r.squared
[1] 0.7075948
> #Extracting R Square and adjusted R Square
> summary(fit_model_1)$r.squared
[1] 0.8473426
> summary(fit_model_2)$r.squared
[1] 0.7075948
>
> summary(fit_model_1)$adj.r.squared
[1] 0.8179855
> summary(fit_model_2)$adj.r.squared
[1] 0.6762656
```

Activity 9: Predict mpg

Use cbind to combine original mtcars and predicted values d mpg

Prediction of mpg will be conducted using the predict() function to estimate mpg values based on our two linear models. Next the mpg will be cbind with the prediction values to compare them side by side.

```
#Activity 9
##dataset mtcars created for prediction
new_data <- mtcars

prediction_model_1 <- predict(fit_model_1, new_data)
new_data$predicted <- prediction_model_1


prediction_model_2 <- predict(fit_model_2, new_data)
#You can also insert the prediction_model_2 by: new_data$predicted_2 <- prediction_model_2

new_data <- cbind(new_data, prediction_model_2)

View(new_data)
```

Print both actual and printed mpg

```
mpg_values <- new_data[,c("mpg", "predicted", "predicted_2")]
mpg_values
```

A subset data called mpg_values is created and printed to display the values for both actual and printed mpg

	mpg	predicted	predicted_2
Mazda RX4	21.0	22.359084	19.95859
Mazda RX4 Wag	21.0	21.709366	20.08160
Datsun 710	22.8	25.426068	25.11740
Hornet 4 Drive	21.4	20.840604	19.79411
Hornet Sportabout	18.7	17.854430	18.08169
Valiant	18.1	19.758979	17.67740
Duster 360	14.3	15.403122	14.88880
Merc 240D	24.4	22.797806	22.59736
Merc 230	22.8	24.278020	24.87892
Merc 280	19.2	20.144732	20.50577
Merc 280C	17.8	20.422549	20.63757
Merc 450SE	16.4	15.266662	15.91180
Merc 450SL	17.3	16.571288	15.95573
Merc 450SLC	15.2	16.578261	16.04360
Cadillac Fleetwood	10.4	10.409035	13.35684
Lincoln Continental	10.4	9.713723	13.82220
Chrysler Imperial	14.7	10.058647	15.37887
Fiat 128	32.4	27.092485	26.95084
Honda Civic	30.4	30.347521	31.13843
Toyota Corolla	33.9	28.886770	28.04632
Toyota Corona	21.5	25.201098	24.35240
Dodge Challenger	15.5	17.072265	15.26018
AMC Javelin	15.2	18.354235	18.14319
Camaro Z28	13.3	15.281350	18.51242
Pontiac Firebird	19.2	16.284616	17.58777
Fiat X1-9	27.3	27.773220	26.82563
Porsche 914-2	26.0	26.083843	27.16358
Lotus Europa	30.4	26.781734	22.48842
Ford Pantera L	15.8	17.960438	21.81610
Ferrari Dino	19.7	19.328934	14.38298
Maserati Bora	15.0	13.070531	10.25058
Volvo 142E	21.4	23.788583	25.29289

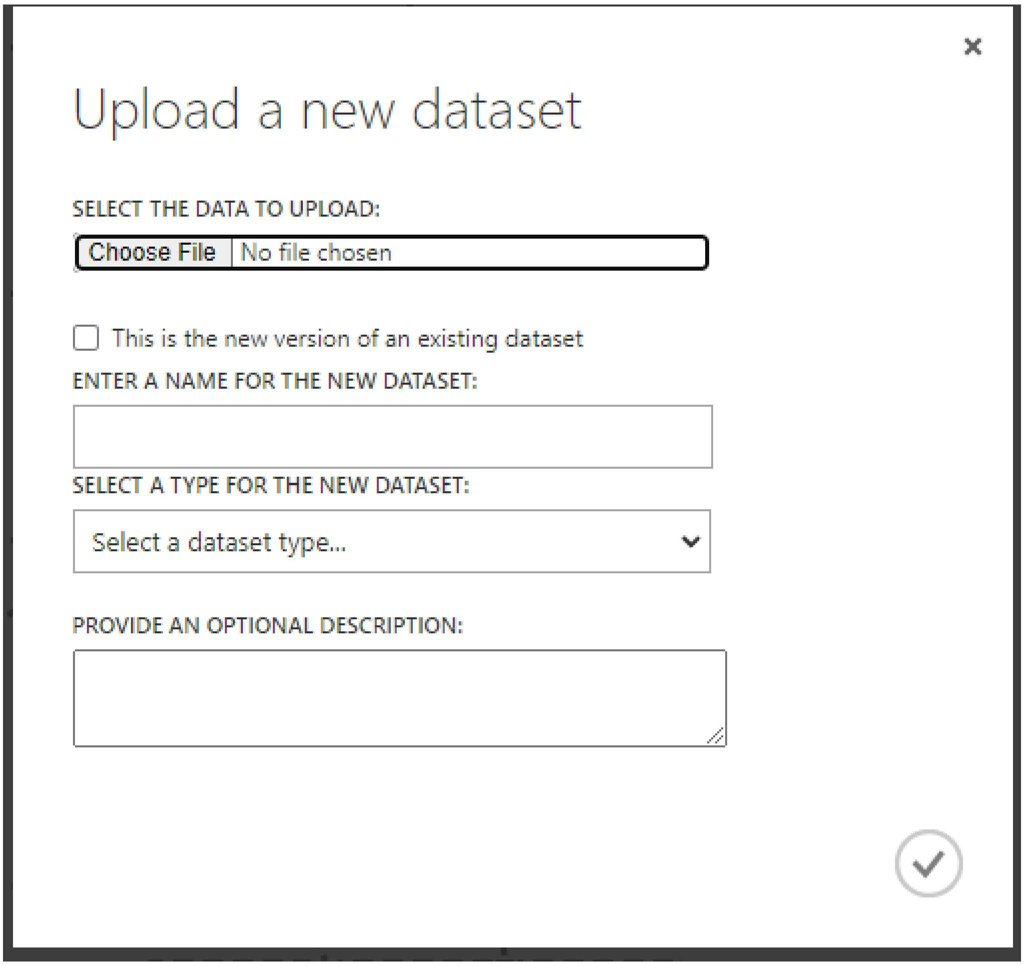
**Actual and printed mpg values*

Activity 10: AML rules and guidelines to ensure proper adoption and adherence of same R program in AML

- **Login to AML studio and Upload the dataset in AML studio**

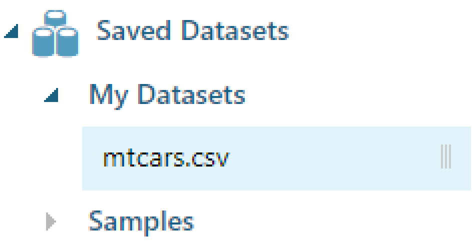
After logging in to AML studio, the first step is to upload the dataset.

This can be accomplished by using the “+new” button



Select the file in the local drive and it will be uploaded with the file type it is in.

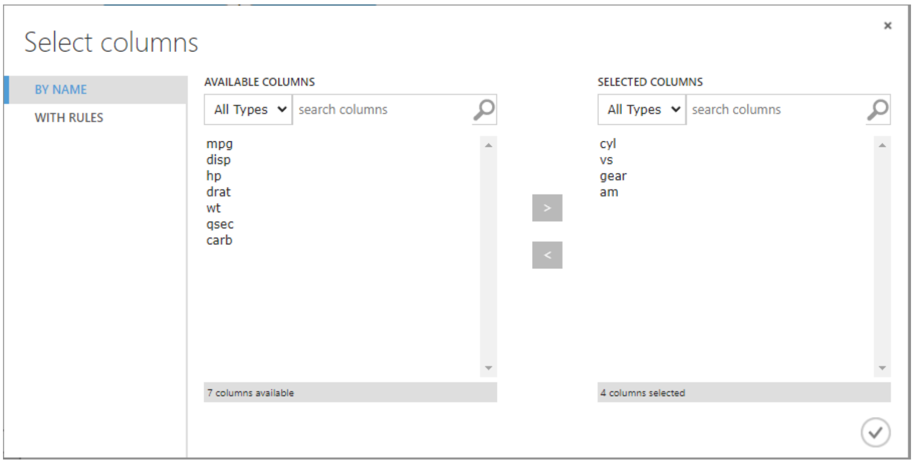
After uploading the dataset , it will reflected the uploaded dataset under “my dataset”.

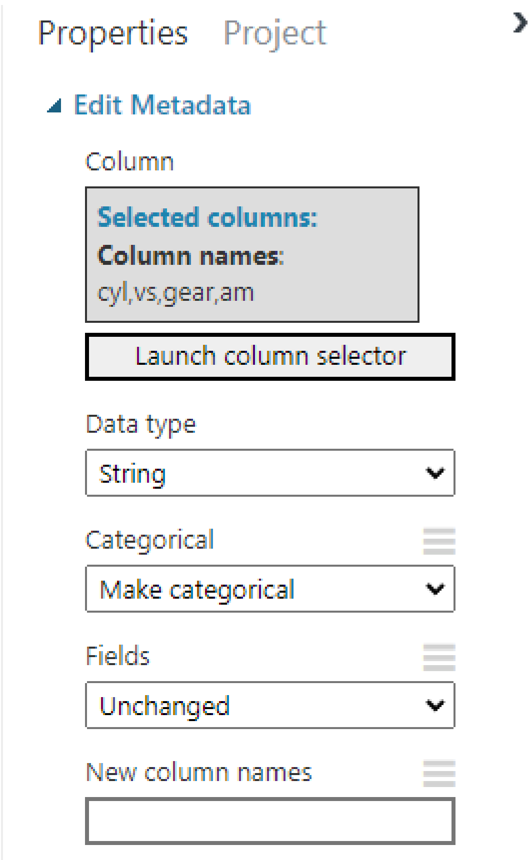


- **Use edit metadata to make cyl,vs,am,gear fields categorical**

After the data has loaded, it will load as a module. The next following step is similar to ‘r’ but in this case, we can search for edit metadata from the search bar and drag it to the module.

After dragging the module, we can include the variables that we would like to change to categorical from the side bar.

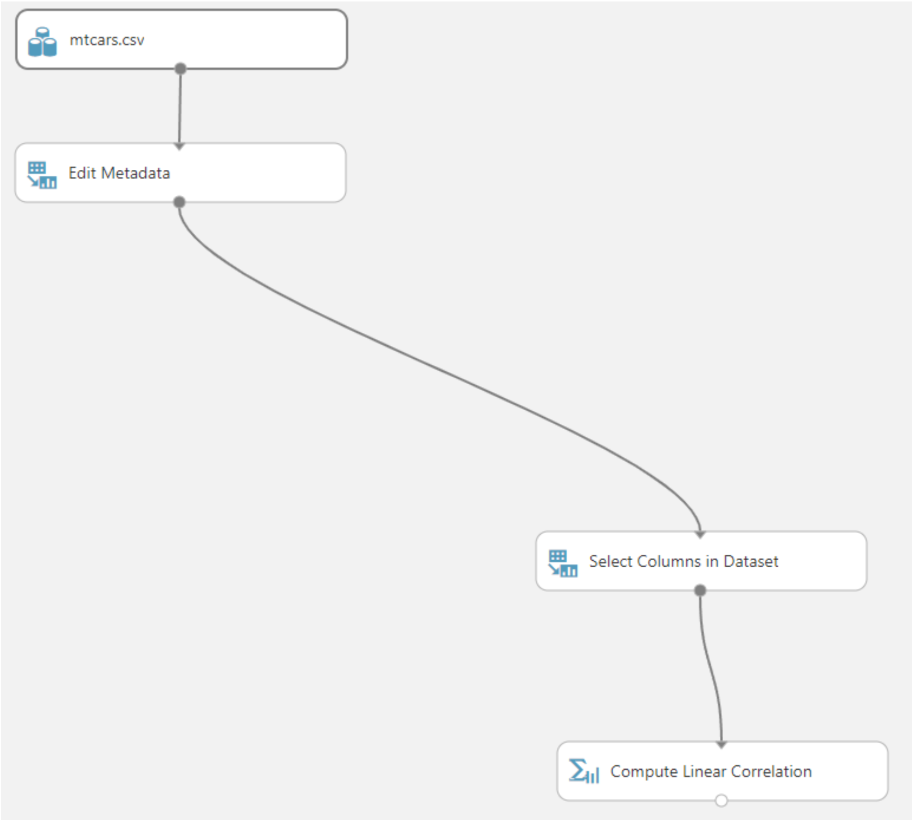




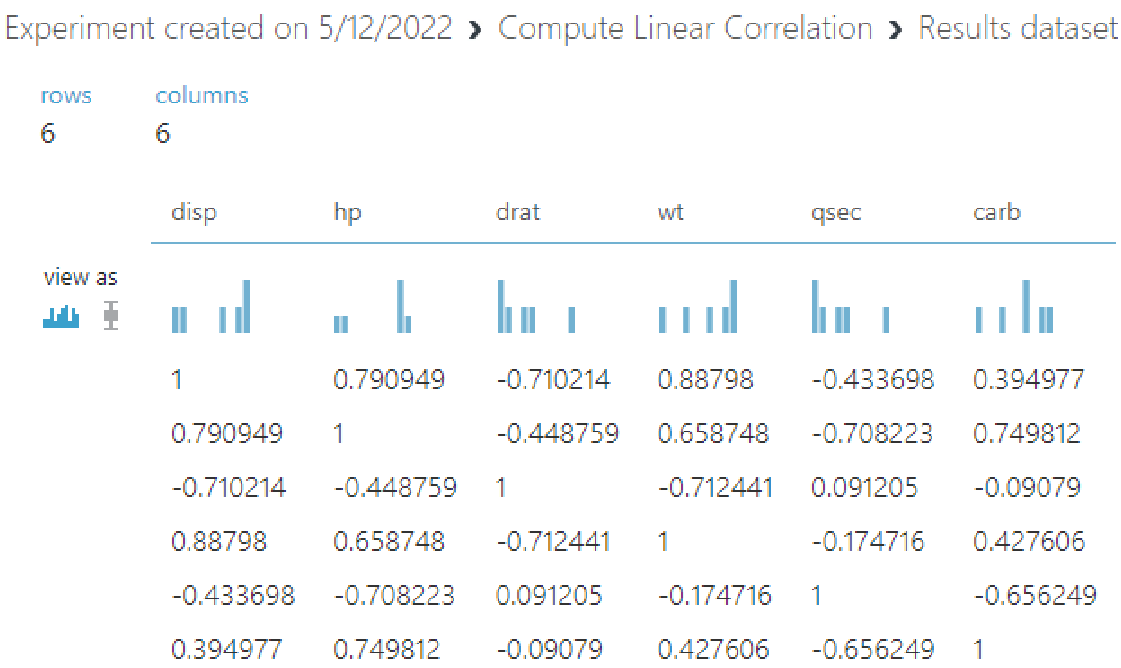
Launch column selector and select the desired columns, from the categorical field select Make categorical and save the changes.

- **Perform Compute Linear Correlation**

  To compute the linear correlation, we must first search for “Select Columns in Dataset”. After which numeric independent columns are then selected and the compute linear correlation is dragged out to place on the tray canvas. From there, we must link it from the edit metadata to “compute line correlation”.



We then run the linear correlation to find out that disp hp, and wt are highly correlated.



Experiment created on 5/12/2022 › Compute Linear Correlation › Results dataset

rows 6
columns 6

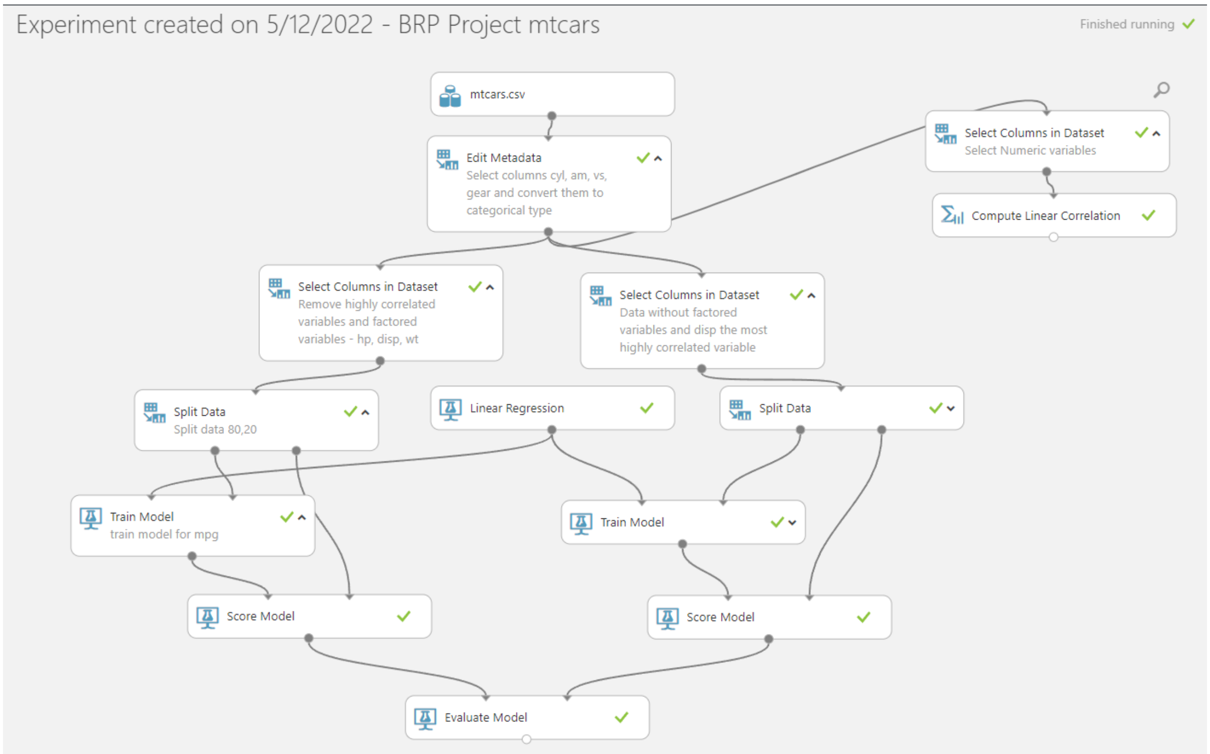
view as

disp	hp	drat	wt	qsec	carb
1	0.790949	-0.710214	0.88798	-0.433698	0.394977
0.790949	1	-0.448759	0.658748	-0.708223	0.749812
-0.710214	-0.448759	1	-0.712441	0.091205	-0.09079
0.88798	0.658748	-0.712441	1	-0.174716	0.427606
-0.433698	-0.708223	0.091205	-0.174716	1	-0.656249
0.394977	0.749812	-0.09079	0.427606	-0.656249	1

We will be doing 2 analysis.

First: select columns in dataset and exclude disp, hp and wt as they are highly corelated along with the factored variables as they are not applicable for the analysis.

Second: select columns in dataset and exclude only factorised variables (am,vs,cyl,gear) and the most highly correlated variable.



Select Columns in Dataset

Select columns

Selected columns:
All columns
Exclude column names:
cyl,disp,hp,wt,vs,am,gear

Launch column selector

START TIME	5/12/2022 ...
END TIME	5/12/2022 ...
ELAPSED TIME	0:00:01.594
STATUS CODE	Finished
STATUS DETAILS	None

View output log

Properties Project

Select Columns in Dataset

Select columns

Selected columns:
All columns
Exclude column names:
cyl,vs,am,gear,disp

Launch column selector

START TIME	5/13/2022 ...
END TIME	5/13/2022 ...
ELAPSED TIME	0:00:01.688
STATUS CODE	Finished
STATUS DETAILS	None

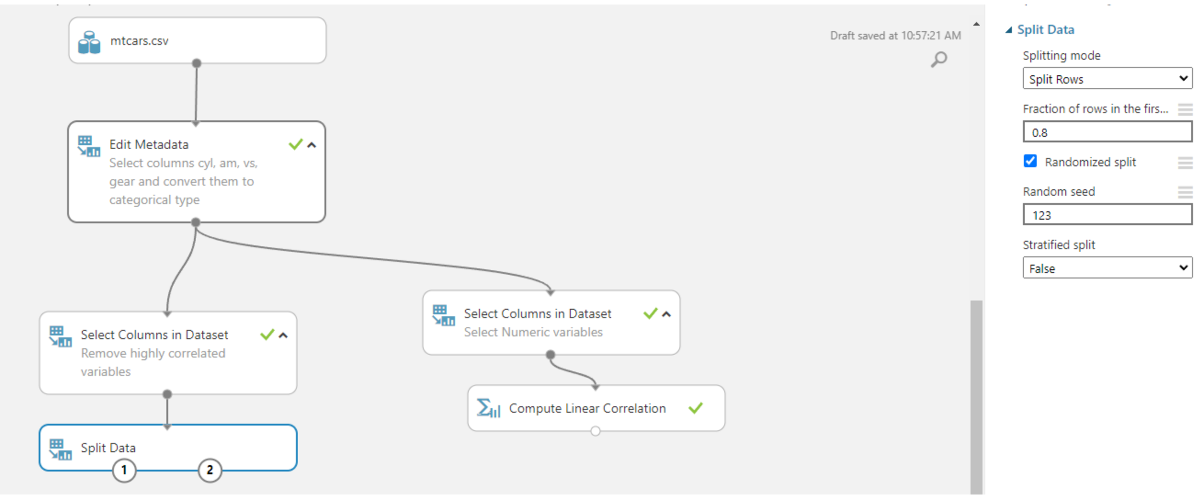
View output log

An output of 4 columns versus 6 columns is observed.

- **Split the data to 80:20 and train the model using Linear regression to predict MPG field**

Drag and drop the "split data" into the canvas. Input 0.8 as the 80% split data between train model data (consists of 26 rows, 4 columns) and apply to the 20% test data in the next score model (consists of 6 rows, 4 columns).

Random seed of 123 is input to ensure same value every time you run the above code.

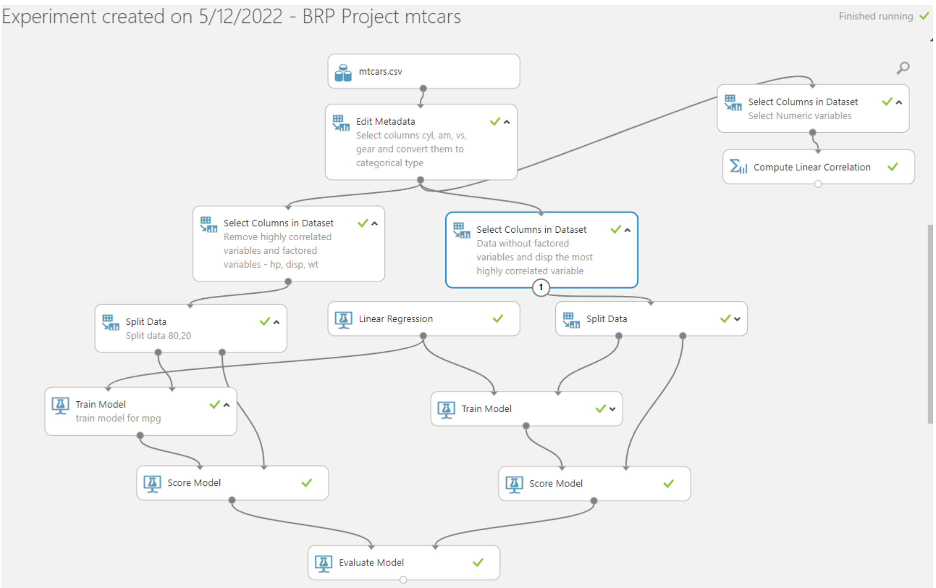


Next, it is to predict the mpg. We search for linear progression and train model under the search bar and dragged it to the grey canvas. From there we will link the first point of the split data to the train model and the point from linear regression to the train model.

- **Score model and take screenshot of the predicted values**

Now, we then score the model, by doing the same step, search and drag the score model from the search. We will then have to link the 2nd output to the score module and linked the 80% trained model to the score output.

In the Train model, *mpg* is selected.



- **Score model**

Analysis 1:

Experiment created on 5/12/2022 - BRP Project mtcars › Score Model

rows	columns
6	5

mpg	drat	qsec	carb	Scored Labels
21	3.9	17.02	4	19.771964
30.4	4.93	18.52	2	30.097285
22.8	3.85	18.61	1	24.618333
24.4	3.69	20	2	21.923423
15.5	2.76	16.87	2	15.581908
33.9	4.22	19.9	1	27.143529

Analysis 2:

Experiment created on 5/12/2022 - BRP Project mtcars › Score Model › Scored dataset

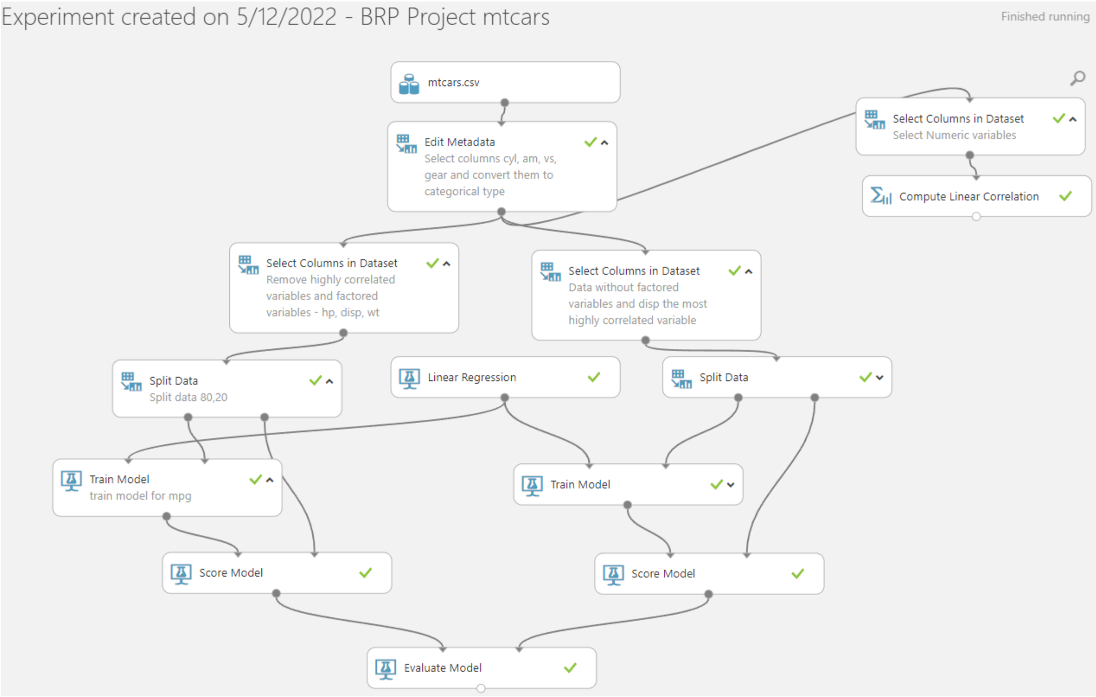
rows	columns
6	7

mpg	hp	drat	wt	qsec	carb	Scored Labels
21	110	3.9	2.875	17.02	4	21.672939
30.4	52	4.93	1.615	18.52	2	29.274455
22.8	93	3.85	2.32	18.61	1	25.062137
24.4	62	3.69	3.19	20	2	22.377905
15.5	150	2.76	3.52	16.87	2	17.872775
33.9	65	4.22	1.835	19.9	1	27.955666

We can infer that the scored labels are predicting a similar score to the actual mpg values.

- **Evaluate model**

The final step is to evaluate the model. We will have to step for evaluate model, dragged to the grey canvas and then linked from the score model to the evaluate model.



Experiment created on 5/12/2022 - BRP Project mtcars › Evaluate Model › Evaluation results

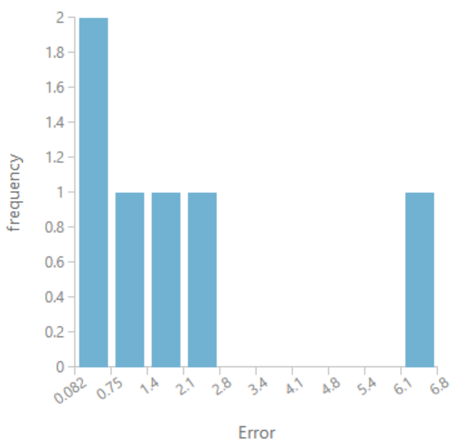
Metrics

Mean Absolute Error	2.110673
Root Mean Squared Error	3.073979
Relative Absolute Error	0.423075
Relative Squared Error	0.258705
Coefficient of Determination	0.741295

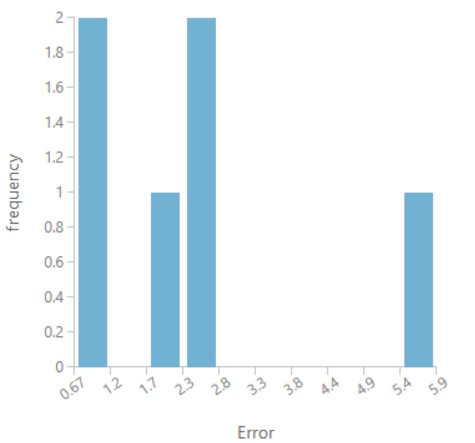
Metrics

Mean Absolute Error	2.399971
Root Mean Squared Error	2.940833
Relative Absolute Error	0.481063
Relative Squared Error	0.236779
Coefficient of Determination	0.763221

Error Histogram



Error Histogram



We manage to achieve 74.1% (Without highly correlated variables) versus 76.3% (With highly correlated variables) accuracy by using this linear model, as illustrated in the Coefficient of Determination.