Lab4 – Predictive Analysis

In this lab, you will learn simple and multiple linear regression for predicting automobile price. By the end of the lab you will learn to import data file into Jupyter Notebook, use simple or multiple linear regression to predict the automobile price. The dataset (named as automobile.csv) is used for this lab.

Software: Jupyter Notebook

Procedure

1. In the desktop directory upload Lab 4 file (Lab4 – Predictive Analysis – Participant Copy.ipynb) to the Jupyter Notebook.

2. In the desktop directory upload the data file (automobile.csv) to the Jupyter Notebook.

2. Click the Lab4 – Predictive Analysis – Participant Copy.ipynb file to start the lab.

Import dataset

Import pandas and numpy libraries and load data to data frame as `df`.

In [ ]:

```
#Import pandas and numpy libraries

import pandas as pd

import numpy as np

#load data to dataframe as df

df = pd.read_csv('automobile.csv')

df.head()
```

Out [ ]:

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...	9.0	154.0	5000.0	19	26
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...	10.0	102.0	5500.0	24	30
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...	8.0	115.0	5500.0	18	22

5 rows × 29 columns

p-value

It is the probability value that the correlation between these two variable is statistically significant. A significant level of 0.05 means that we are 95% confident that we are 95% confident that the correlation between the variables are significant.

p-value < 0.001 means strong evidence that the correlation is significant.

p-value < 0.05 means moderate evidence that the correlation is significant.

p-value < 0.1 means weak evidence that the correlation is significant.

p-value > 0.1 means no evidence that the correlation is significant.

The p-value can be obtained from stats module in scipy library.

In [ ]:

```
from scipy import stats

#Pearson_coefficient of wheel-base and price

pearson_coef, p_value = stats.pearsonr(df['wheel-base'], df['price'])

print("The Pearson Correlation Coefficient for wheel-base vs price is", pearson_coef, " with a P-value of P =", p_value)
```

Out[ ]:

```
The Pearson Correlation Coefficient for wheel-base vs price is  0.5846418222655081  with a p-value of p = 8.076488270732989e-20
```

Since the p-value is less than 0.001, the correlation between `wheel-base` and `price` is statistically significant although the linear relationship is not extremely strong (0.5846).

Exercise 4.1

Write the code to print the Pearson Correlation Coefficient and p-value of the following variables vs `price`:

- `horsepower`
- `length`
- `width`
- `curd-weight`
- `engine-size`
- `bore`
- `city-mpg`
- `highway-mpg`

In [ ]:

```
#Write the code and press shift + enter
```

Out[ ]:

```
The Pearson Correlation Coefficient for horsepower vs price is 0.8095745670081455  with a p-value of p =  6.369057414856692e-48
The Pearson Correlation Coefficient for length vs price is 0.6906283810037583  with a p-value of p =  8.016476289929682e-30
The Pearson Correlation Coefficient for width vs price is 0.7512653454356577  with a p-value of p = 9.200331125593175e-38
The Pearson Correlation Coefficient for curb-weight vs priceis 0.8344145257702846  with a p-value of p =  2.1895772388936914e-5
3
The Pearson Correlation Coefficient for engine-size vs price is 0.8723351674455185  with a p-value of p = 9.265491622198389e-64
The Pearson Correlation Coefficient for bore vs price is 0.5431553832623068  with a p-value of p =   8.049189484376619e-17
The Pearson Correlation Coefficient for city-mpg vs price is -0.6865710067844677  with a p-value of p =  2.321132065567674e-29
The Pearson Correlation Coefficient for highway-mpg vs price is -0.7046922650589529  with a p-value of p =  1.7495471144477352e
-31
```

Summarize the results in the following table:

	Pearson Correlation Coefficient	P-value	Significant of correlation
'wheel-base' vs 'price'	0.5846	<0.001	Strong
'horsepower' vs 'price'			
'length' vs 'price'			
'width' vs 'price'			
'curd-weight' vs 'price'			
'engine-size' vs 'price'			
'bore' vs 'price'			
'city-mpg' vs 'price'			
'highway-mpg' vs 'price'			

ANOVA - Analysis of Variance

It is a statistical method used to test whether there are significant differences between the means of two or more groups of categorical variables and it returns F-test score. A large difference means there is a larger difference between the means.

We used the ANOVA to test the groups (`'rwd'`, `'fwd'` and `'4wd'`) in drive-wheels.

In [ ]:

```
df_gptest = df[['drive-wheels','body-style','price']]

grouped_test=df_gptest[['drive-wheels',
'price']].groupby(['drive-wheels'])

df_gptest
```

Out[ ]:

	drive-wheels	body-style	price
0	rwd	convertible	13495
1	rwd	convertible	16500
2	rwd	hatchback	16500
3	fwd	sedan	13950
4	4wd	sedan	17450
...	...	...	...
196	rwd	sedan	16845
197	rwd	sedan	19045
198	rwd	sedan	21485
199	rwd	sedan	22470
200	rwd	sedan	22625

201 rows × 3 columns

In [ ]:

```
# ANOVA test for rwd vand 4wd

f_val, p_val = stats.f_oneway(grouped_test.get_group('fwd')['price'],
grouped_test.get_group('rwd')['price'],
grouped_test.get_group('4wd')['price'])

print( "ANOVA results - rwd and 4wd: F=", f_val, ", P =", p_val)

# ANOVA test for fwd vand rwd

f_val, p_val = stats.f_oneway(grouped_test.get_group('fwd')['price'],
grouped_test.get_group('rwd')['price'])

print( "ANOVA results - fwd and rwd: F=", f_val, ", P =", p_val )

#ANOVA test for 4wd and rwd

f_val, p_val = stats.f_oneway(grouped_test.get_group('4wd')['price'],
grouped_test.get_group('rwd')['price'])

print( "ANOVA results - 4wd and rwd: F=", f_val, ", P =", p_val)

#ANOVA test for 4wd and fwd

f_val, p_val = stats.f_oneway(grouped_test.get_group('4wd')['price'],
grouped_test.get_group('fwd')['price'])

print("ANOVA results - 4wd and rwd: F=", f_val, ", P =", p_val)
```

Out[ ]:

```
ANOVA results - rwd and 4wd: F= 67.95406500780399 , P = 3.3945443577151245e-23
ANOVA results - fwd and rwd: F= 130.5533160959111 , P = 2.2355306355677845e-23
ANOVA results - 4wd and rwd: F= 8.580681368924756 , P = 0.004411492211225333
ANOVA results - 4wd and fwd: F= 0.665465750252303 , P = 0.41620116697845666
```

After performing the analysis, we find that the following variables are important for prediction of 'price':

- 'length'
- 'width'
- 'curb-weight'
- 'engine-size'
- 'horsepower'

- 'city-mpg'
- ' highway-mpg'
- 'wheel-base'
- 'bore'
- 'drive-wheels' (categorical variables)

Simple/Multiple Linear Regression Prediction Model

In [ ]:

```
from sklearn.linear_model import LinearRegression

from sklearn import metrics

from sklearn.metrics import r2_score

lm=LinearRegression()

X=df[['highway-mpg']]

Y=df['price']

lm.fit(X,Y)

Y_pred=lm.predict(X)

print("Gradient: ",lm.coef_)

print("Intercept:", lm.intercept_)

print("Coefficient of Determination:", r2_score(Y,Y_pred))

df = pd.DataFrame({'Actual': Y, 'Predicted': Y_pred})

df
```

Out[ ]:

```
Gradient:  [-821.73337832]
Intercept: 38423.305858157386
Coefficient of Determination: 0.4965911884339175
```

	Actual price	Predicted price
0	13495	16236.504643
1	16500	16236.504643
2	16500	17058.238022
3	13950	13771.304508
4	17450	20345.171535
...	...	...
196	16845	15414.771265
197	19045	17879.971400
198	21485	19523.438157
199	22470	16236.504643
200	22625	17879.971400

201 rows × 2 columns

The simple linear regression equation to predict `'price'`,

`'price'` = - 821.733 * `'highway-mpg'` + 38423.305.

This is not an accurate simple linear regression as the coefficient of determination is not high (0.497).

Exercise 4.2

Modify the code to find the coefficient of determination for the simple linear regression model for `'length'`, `'width'`, `'curb-weight'`, `'engine-size'`, `'horsepower'`, `'city-mpg'`, `'wheel-base'` and `'bore'`. Record the gradient, intercept and coefficient of determination of the variables:

	Gradient	Intercept	Coefficient of Determination
`'wheel-base'`			
`'horsepower'`			
`'length'`			
`'width'`			
`'curb-weight'`			
`'engine-size'`			
`'bore'`			
`'city-mpg'`			
`'highway-mpg'`	-821.733	38423.305	0.497

Note: Execute the code from the start after modifying the code for difference variable.

Best simple linear regression prediction model?____

____

____

____

____

Multiple Linear Regression Prediction Model

Modify the above code for multiple regress prediction model and execute the code from the start to obtain best multiple linear regression prediction model.

Best Coefficient of Determination obtained?_____________________________________

Best multiple linear regression prediction model?________________________________

____

____

____

--- End of Lab4 ---