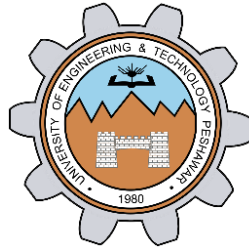


LAB # 08



Fall 2021

Data Analytics Lab

Submitted by: **Shah Raza**

Registration No.: **18PWCSE1658**

Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: _____

Submitted to:

Engr. Mian Ibad Ali Shah

February 13, 2022

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Lab Objectives:

- Training Multi Linear Regression Model.

TASK:

```
import numpy as np
import pandas as pd
import os
```

```
df = pd.read_csv('car selling dataset.csv')
df.head()
```

```
[:
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXi BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	11.5@ 4,500(kgm@ rpm)	5.0

```
df.columns
```

```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
      'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
      'seats'])
```

```
df.columns
```

```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
      'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
      'seats'],
      dtype='object')
```

```
df['transmission'].value_counts()
```

```
Manual      7078
Automatic   1050
Name: transmission, dtype: int64
```

```
import numpy as np
import pandas as pd
import os
```

```
df = pd.read_csv('car selling dataset.csv')
df.head()
```

```
]:
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	11.5@ 4,500(kgm@ rpm)	5.0

```
df.columns
```

```
]:
```

```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner', 'mileage', 'engine', 'max_power', 'torque',
       'seats'],
      dtype='object',
      name='columns')
```

```
df.isnull().sum()
```

```
]:
```

```
name          0
year          0
selling_price  0
km_driven     0
fuel          0
seller_type   0
transmission  0
owner         0
mileage       221
engine        221
max_power     215
torque        222
seats         221
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.isnull().sum()
```

```
]:
```

```
name          0
year          0
selling_price  0
km_driven     0
fuel          0
seller_type   0
transmission  0
owner         0
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power
count	7906.000000	7906.000000	7.906000e+03	7.906000e+03	7906.000000	7906.000000	7906.000000	7906.000000	7906.000000	7906.000000	7906.000000
mean	0.000126	2013.983936	6.498137e+05	6.918866e+04	0.006577	0.14002	0.868328	0.659626	0.002150	0.002277	0.011384
std	0.011247	3.863695	8.135827e+05	5.679230e+04	0.080838	0.34703	0.338155	0.473865	0.046324	0.047664	0.106092
min	0.000000	1994.000000	2.999900e+04	1.000000e+00	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	2012.000000	2.700000e+05	3.500000e+04	0.000000	0.00000	1.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	2015.000000	4.500000e+05	6.000000e+04	0.000000	0.00000	1.000000	1.000000	0.000000	0.000000	0.000000
75%	0.000000	2017.000000	6.900000e+05	9.542500e+04	0.000000	0.00000	1.000000	1.000000	0.000000	0.000000	0.000000
max	1.000000	2020.000000	1.000000e+07	2.360457e+06	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000	1.000000

```
➤ X = df.drop('selling_price',axis=1)
  y = df['selling_price']
```

```
➤ from sklearn.model_selection import train_test_split
  X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=1)
```

```
➤ len(y_test)
```

```
] : 1582
```

```
➤ from sklearn.linear_model import LinearRegression
  lr = LinearRegression()
  lr = lr.fit(X_train,y_train)
  pred = lr.predict(X_test)
```

```
➤ len(pred)
```

```
] : 1582
```

```
➤ from sklearn.metrics import r2_score
  r2_score(y_test,pred)
```

```
] : 0.9662063787501225
```
