

PML LAB 3

G RETHINAGIRI

225229139

```
In [1]: import pandas as ad
```

```
In [4]: a = ad.read_csv("fuel_data.csv")  
a
```

```
Out[4]:
```

	drivenKM	fuelAmount
0	390.00	3600.0
1	403.00	3705.0
2	396.50	3471.0
3	383.50	3250.5
4	321.10	3263.7
5	391.30	3445.2
6	386.10	3679.0
7	371.80	3744.5
8	404.30	3809.0
9	392.20	3905.0
10	386.43	3874.0
11	395.20	3910.0
12	381.00	4020.7
13	372.00	3622.0
14	397.00	3450.5
15	407.00	4179.0
16	372.40	3454.2
17	375.60	3883.8
18	399.00	4235.9

	drivenKM	fuelAmount
0	390.00	3600.0
1	403.00	3705.0
2	396.50	3471.0
3	383.50	3250.5
4	321.10	3263.7
5	391.30	3445.2
6	386.10	3679.0
7	371.80	3744.5
8	404.30	3809.0
9	392.20	3905.0
10	386.43	3874.0
11	395.20	3910.0
12	381.00	4020.7
13	372.00	3622.0
14	397.00	3450.5
15	407.00	4179.0
16	372.40	3454.2
17	375.60	3883.8
18	399.00	4235.9

```
In [5]: a.isnull()
```

Out[5]:

	drivenKM	fuelAmount
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False

In [7]:

a.head()

Out[7]:

	drivenKM	fuelAmount
0	390.0	3600.0
1	403.0	3705.0
2	396.5	3471.0
3	383.5	3250.5
4	321.1	3263.7

In [8]:

a.shape

Out[8]:

(19, 2)

In [9]:

a.size

Out[9]:

38

In [10]:

a.columns

Out[10]:

Index(['drivenKM', 'fuelAmount'], dtype='object')

```
In [11]: type(a)
```

```
Out[11]: pandas.core.frame.DataFrame
```

```
In [12]: a.info
```

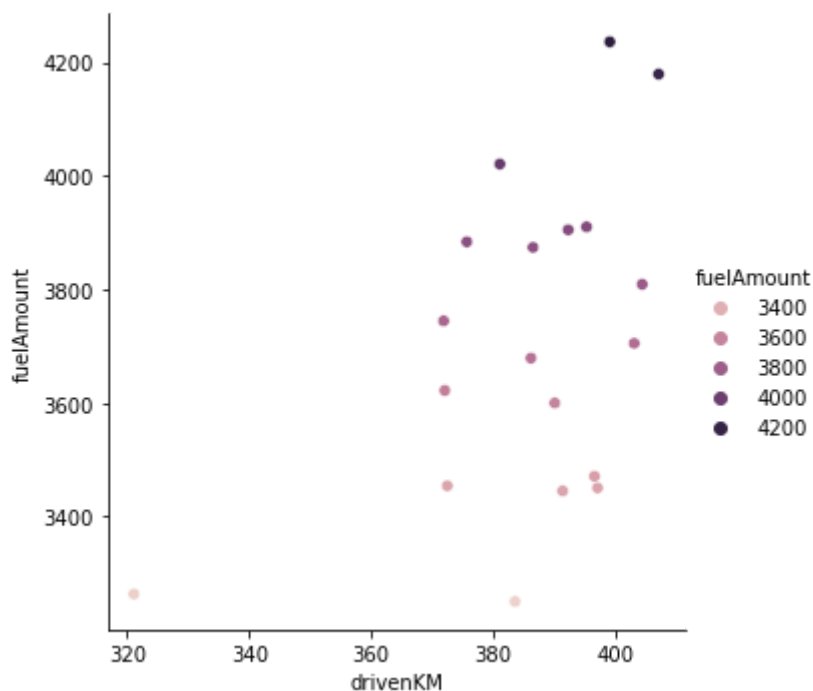
```
Out[12]: <bound method DataFrame.info of      drivenKM  fuelAmount
0      390.00      3600.0
1      403.00      3705.0
2      396.50      3471.0
3      383.50      3250.5
4      321.10      3263.7
5      391.30      3445.2
6      386.10      3679.0
7      371.80      3744.5
8      404.30      3809.0
9      392.20      3905.0
10     386.43      3874.0
11     395.20      3910.0
12     381.00      4020.7
13     372.00      3622.0
14     397.00      3450.5
15     407.00      4179.0
16     372.40      3454.2
17     375.60      3883.8
18     399.00      4235.9>
```

```
In [13]: a.shape[0]
```

```
Out[13]: 19
```

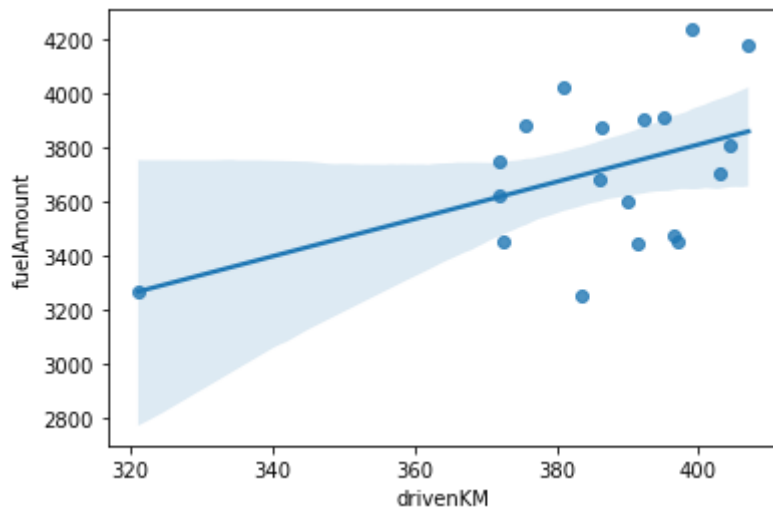
```
In [14]: import seaborn as sns
sns.relplot(data=a,x="drivenKM",y="fuelAmount",hue="fuelAmount")
```

```
Out[14]: <seaborn.axisgrid.FacetGrid at 0x19bb3e43310>
```



```
In [15]: sns.regplot(data=a,x="drivenKM",y="fuelAmount")
```

```
Out[15]: <AxesSubplot:xlabel='drivenKM', ylabel='fuelAmount'>
```



```
In [19]: x=ad.DataFrame(a['drivenKM'])  
y=ad.DataFrame(a['fuelAmount'])  
print(x)  
print(y)
```

```
drivenKM  
0      390.00  
1      403.00  
2      396.50  
3      383.50  
4      321.10  
5      391.30  
6      386.10  
7      371.80  
8      404.30  
9      392.20  
10     386.43  
11     395.20  
12     381.00  
13     372.00  
14     397.00  
15     407.00  
16     372.40  
17     375.60  
18     399.00  
fuelAmount  
0      3600.0  
1      3705.0  
2      3471.0  
3      3250.5  
4      3263.7  
5      3445.2  
6      3679.0  
7      3744.5  
8      3809.0  
9      3905.0  
10     3874.0  
11     3910.0  
12     4020.7  
13     3622.0  
14     3450.5  
15     4179.0  
16     3454.2  
17     3883.8  
18     4235.9
```

```
In [20]: type(x)
```

```
Out[20]: pandas.core.frame.DataFrame
```

```
In [21]: type(y)
```

```
Out[21]: pandas.core.frame.DataFrame
```

```
In [23]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=42)
x_train.shape
```

```
Out[23]: (15, 1)
```

```
In [24]: y_train.shape
```

```
Out[24]: (15, 1)
```

```
In [25]: x_test.shape
```

```
Out[25]: (4, 1)
```

```
In [26]: y_test.shape
```

```
Out[26]: (4, 1)
```

```
In [28]: from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

```
Out[28]: LinearRegression()
```

```
In [29]: pred_800_KM=reg.predict([[800]])
print("Deisel price for 800KM:",pred_800_KM[0])
```

```
Deisel price for 800KM: [6905.64571567]
```

```
C:\Users\elcot\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(
```

```
In [31]: y_pred=reg.predict(x_test)
y_pred
```

```
Out[31]: array([[3775.81615646],
               [3785.74000628],
               [3815.51155575],
               [3875.05465468]])
```

```
In [32]: import sklearn.metrics as metrics
mse=metrics.mean_squared_error(y_test,y_pred)
r2=metrics.r2_score(y_test,y_pred)
print("MSE: ",mse)
print("R2: ",r2)
print("\n")
print("Model parameters:")
print("coefficient:",reg.coef_)
print("Intercept:",reg.intercept_)
```

MSE: 46181.36710639155
R2: -0.6180990161577022

Model parameters:
coefficient: [[7.63373063]]
Intercept: [798.6612099]

In []: