Without_use_any_Encoding_Technique

April 1, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
df = pd.read_csv('online_profit.csv')
```

[3]:
```
df.head()
```

[3]:

	Marketing Spend	Administration	Transport	Area	Profit
0	114523.61	136897.80	471784.10	Dhaka	192261.83
1	NaN	151377.59	443898.53	Ctg	191792.06
2	153441.51	101145.55	407934.54	NaN	191050.39
3	144372.41	118671.85	383199.62	Dhaka	182901.99
4	142107.34	91391.77	366168.42	Rangpur	166187.94

1 Check Null Values in Dataset

[4]:
```
df.isnull().sum()
```

[4]:
```
Marketing Spend    2
Administration     0
Transport          0
Area               3
Profit             1
dtype: int64
```

2 Replace Null Values

[5]:
```
mean = df['Marketing Spend'].mean()
```

[6]:
```
mean
```

[6]:
```
70691.35312500001
```

[7]:
```
df['Marketing Spend'] = df['Marketing Spend'].fillna(mean)
```

[8]:
```
df.head()
```

[8]:

	Marketing Spend	Administration	Transport	Area	Profit
0	114523.610000	136897.80	471784.10	Dhaka	192261.83
1	70691.353125	151377.59	443898.53	Ctg	191792.06
2	153441.510000	101145.55	407934.54	NaN	191050.39
3	144372.410000	118671.85	383199.62	Dhaka	182901.99
4	142107.340000	91391.77	366168.42	Rangpur	166187.94

[9]:
```
df['Area'] = df['Area'].fillna(method='ffill')
```

[10]:
```
median = df['Profit'].median()
```

[11]:
```
median
```

[11]:
```
107404.34
```

[12]:
```
df['Profit'] = df['Profit'].fillna(median)
```

[13]:
```
df.head()
```

[13]:

	Marketing Spend	Administration	Transport	Area	Profit
0	114523.610000	136897.80	471784.10	Dhaka	192261.83
1	70691.353125	151377.59	443898.53	Ctg	191792.06
2	153441.510000	101145.55	407934.54	Ctg	191050.39
3	144372.410000	118671.85	383199.62	Dhaka	182901.99
4	142107.340000	91391.77	366168.42	Rangpur	166187.94

3 Encoding

[14]:
```
df['Area'] = df['Area'].replace(['Dhaka','Ctg','Rangpur'],[3,2,1])
```

[15]:
```
df.head()
```

[15]:

	Marketing Spend	Administration	Transport	Area	Profit
0	114523.610000	136897.80	471784.10	3	192261.83
1	70691.353125	151377.59	443898.53	2	191792.06
2	153441.510000	101145.55	407934.54	2	191050.39
3	144372.410000	118671.85	383199.62	3	182901.99
4	142107.340000	91391.77	366168.42	1	166187.94

[16]:
```
x = df.drop(['Profit'], axis=1)
```

[17]:
```
y = df['Profit']
```

[18]:
```
x.head()
```

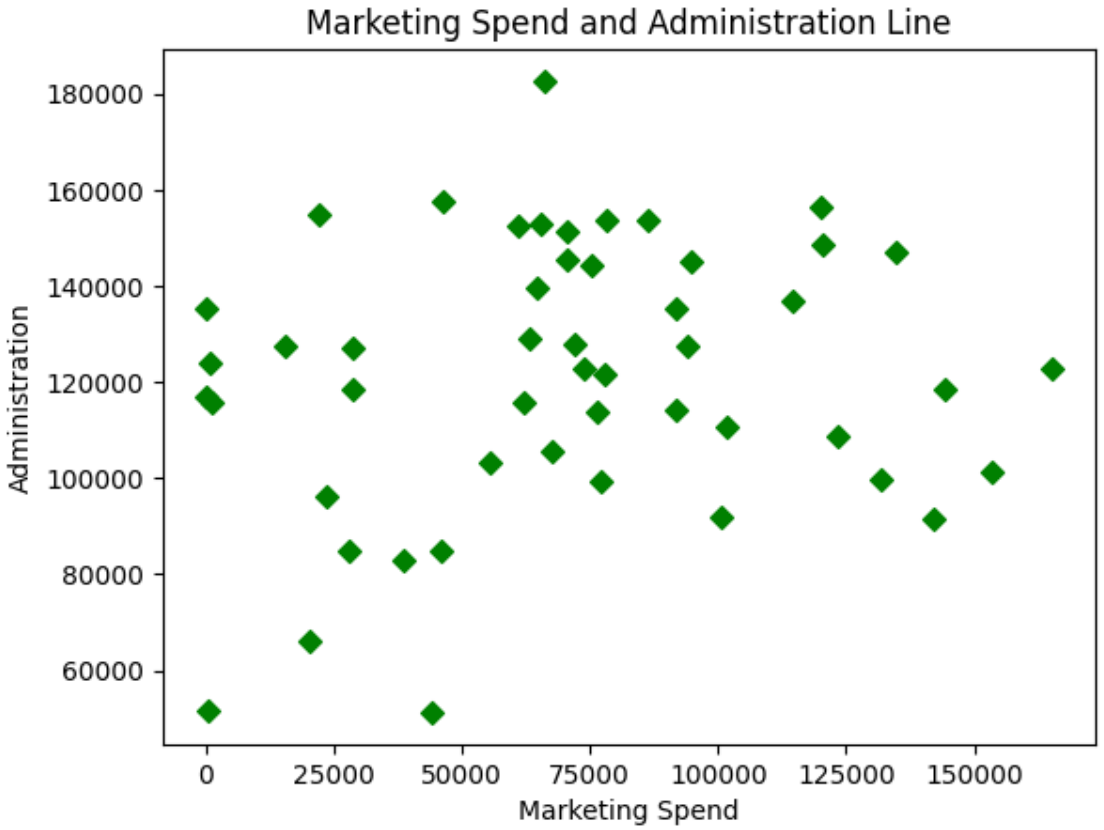
[18]:

	Marketing Spend	Administration	Transport	Area
0	114523.610000	136897.80	471784.10	3
1	70691.353125	151377.59	443898.53	2
2	153441.510000	101145.55	407934.54	2
3	144372.410000	118671.85	383199.62	3
4	142107.340000	91391.77	366168.42	1

4 Visualization

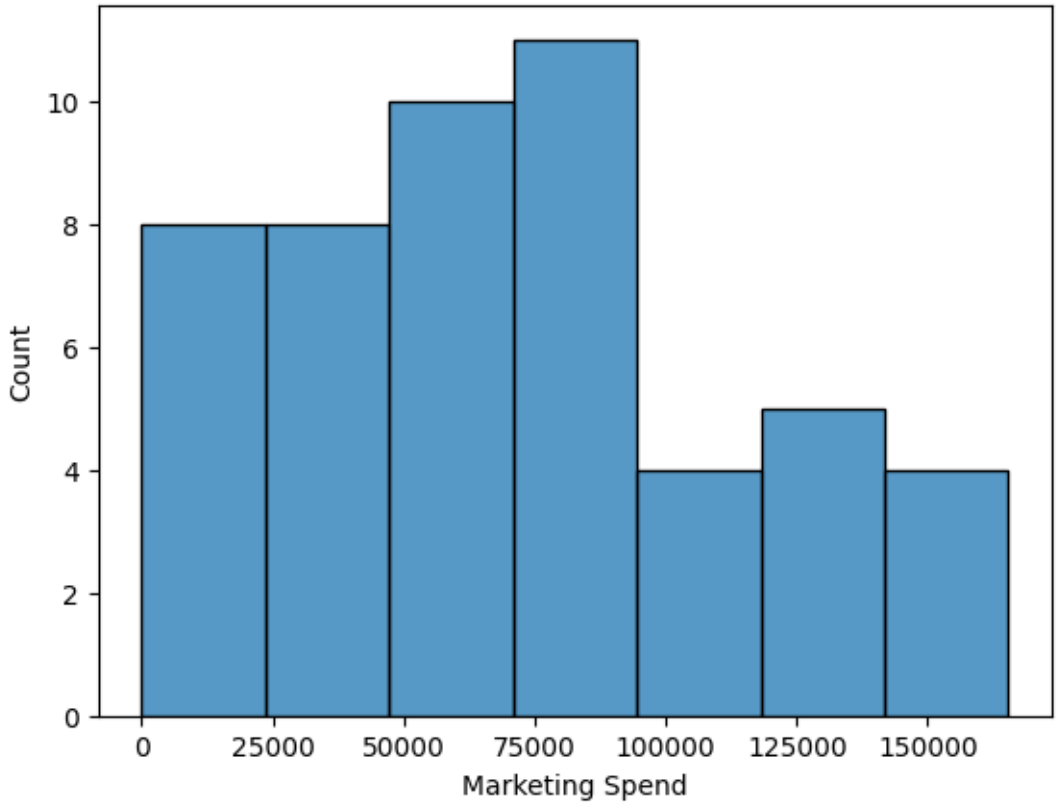
[19]:
```
plt.title("Marketing Spend and Administration Line")
plt.xlabel("Marketing Spend")
plt.ylabel("Administration")
plt.scatter(df['Marketing Spend'],df['Administration'],marker="D",color="Green")
```

[19]: <matplotlib.collections.PathCollection at 0x26d6d51e800>



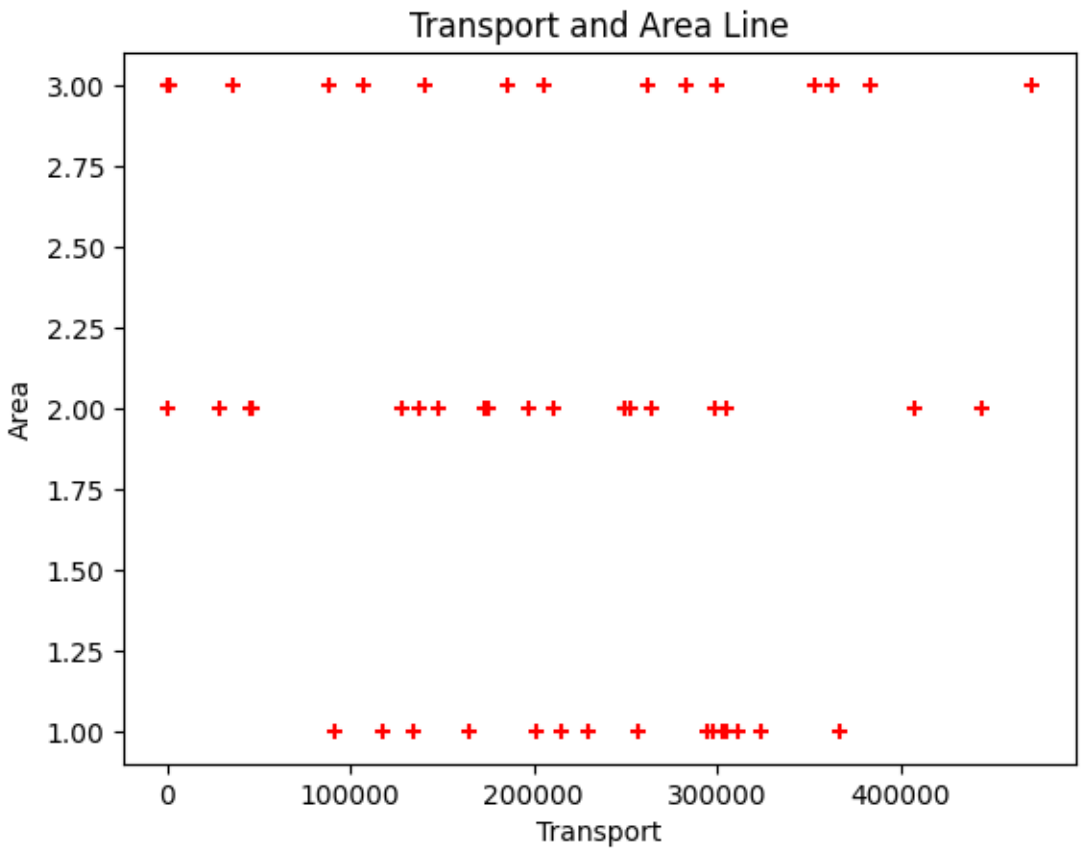
[20]:
```
sns.histplot(df['Marketing Spend'])
```

[20]: <AxesSubplot: xlabel='Marketing Spend', ylabel='Count'>



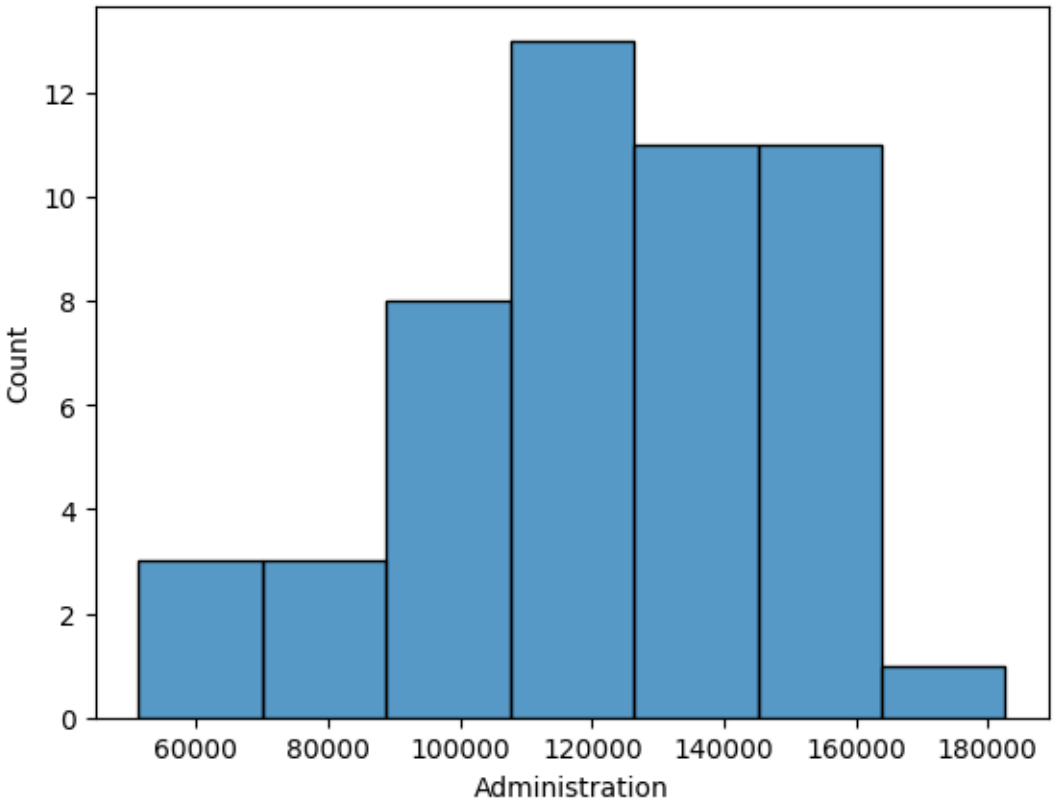
[21]:
```
plt.title("Transport and Area Line")
plt.xlabel("Transport")
plt.ylabel("Area")
plt.scatter(df['Transport'],df['Area'],marker="+",color="Red")
```

[21]: <matplotlib.collections.PathCollection at 0x26d6d5df550>



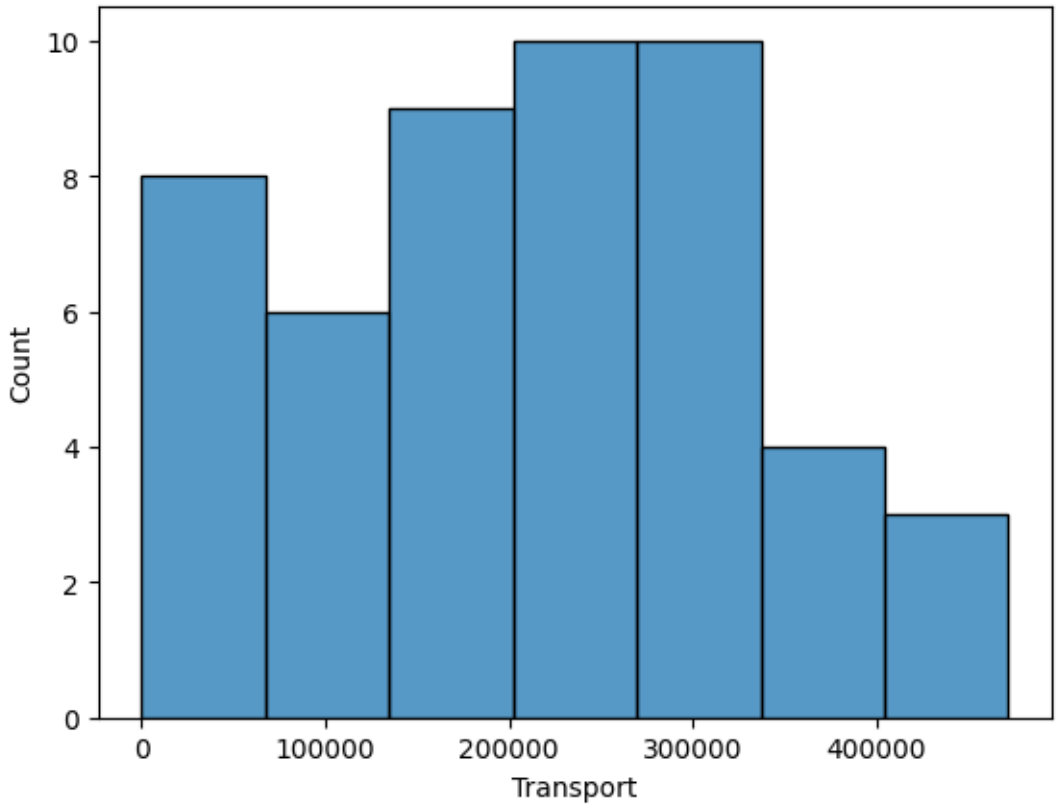
[22]:
```
sns.histplot(df['Administration'])
```

[22]: <AxesSubplot: xlabel='Administration', ylabel='Count'>



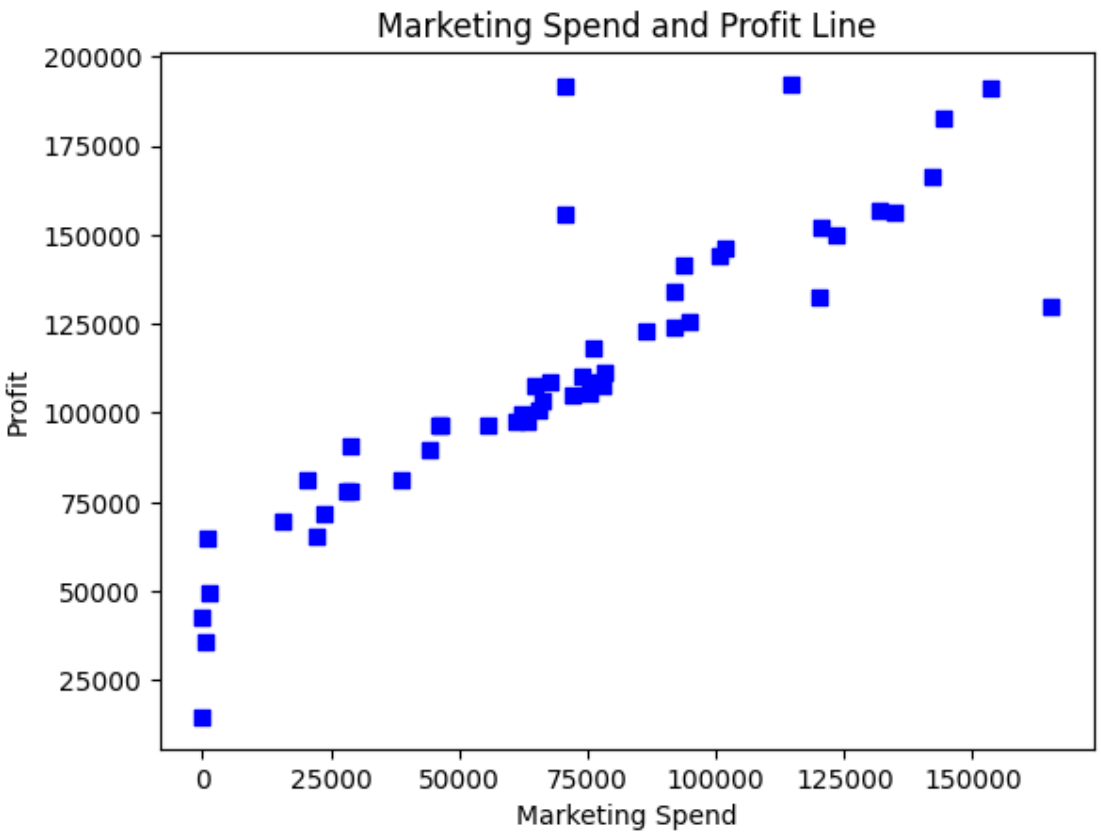
[23]:
```
sns.histplot(df['Transport'])
```

[23]:
```
<AxesSubplot: xlabel='Transport', ylabel='Count'>
```



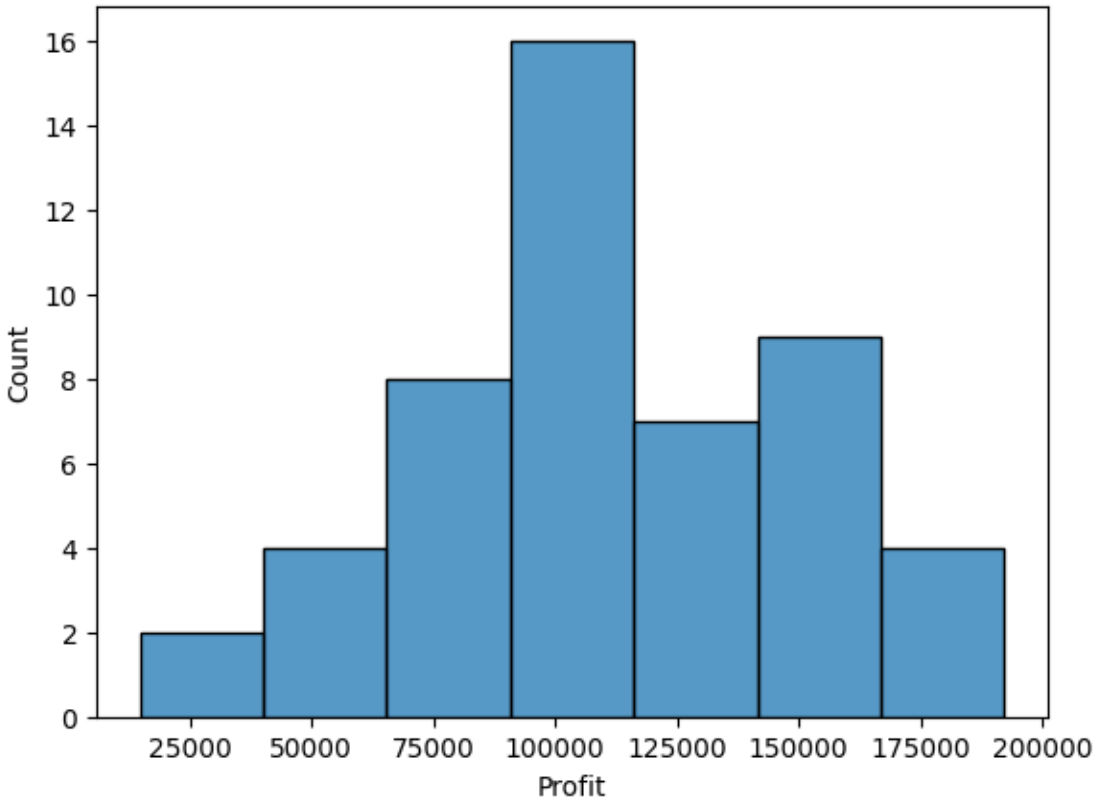
[24]:
```
plt.title("Marketing Spend and Profit Line")
plt.xlabel("Marketing Spend")
plt.ylabel("Profit")
plt.scatter(df['Marketing Spend'],df['Profit'],marker="s",color="Blue")
```

[24]: <matplotlib.collections.PathCollection at 0x26d6f9d2800>



[25]:
```
sns.histplot(df['Profit'])
```

[25]:
```
<AxesSubplot: xlabel='Profit', ylabel='Count'>
```



5 Training Data

[26]:
```
from sklearn.model_selection import train_test_split
```

[27]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x,y,train_size=.
↪70,random_state=42)
```

[28]:
```
xtrain.shape
```

[28]: (35, 4)

[29]:
```
xtest.shape
```

[29]: (15, 4)

[30]:
```
ytrain.shape
```

[30]: (35,)

[31]:
```
ytest.shape
```

[31]:
```
(15,)
```

6 Linear Regression

[32]:
```
from sklearn.linear_model import LinearRegression
```

[33]:
```
reg = LinearRegression()
```

[34]:
```
reg.fit(xtrain,ytrain)
```

[34]:
```
LinearRegression()
```

[35]:
```
reg.predict(xtest)
```

[35]:
```
array([129627.84571793,  79308.29450855,  84903.98868084,  39815.94545506,
       137680.28312165,  27168.58144403, 103544.69121221,  97879.00965629,
        81744.73452037,  92329.81919023, 128888.833587  , 166782.11206859,
        80383.83379185, 156140.70576909, 176144.45317228])
```

[36]:
```
ytest
```

[36]:
```
13    134307.35
39     81005.76
30     99937.59
45     64926.08
17    125370.37
48     35673.41
26    105733.54
25    107404.34
32     97427.84
19    122776.86
12    141585.52
4     166187.94
37     89949.14
8     152211.77
3     182901.99
Name: Profit, dtype: float64
```

[37]:
```
reg.score(xtest.values,ytest)
```

[37]:
```
0.8726448123190611
```

[38]:
```
reg.coef_
```

[38]:
```
array([5.60631094e-01, 1.67194619e-01, 1.49138930e-01, 4.95687455e+02])
```

[39]:
```
reg.intercept_
```

[39]: 16726.45274193675

[40]:
```
reg.predict([[142107.34,91391.77,366168.42,1]])
```

[40]: array([166782.11206859])