Multiple Linear Regression

November 15, 2022

1 Multiple Linear Regression

[17]:
```
#Importing the librariesimport pandas as pd
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[18]:
```
#Reading the dataset
dataset = pd.read_csv('Advertising.csv')
dataset.head()
```

[18]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

[19]:
```
#Setting the value for X and Y
x = dataset[['TV', 'Radio', 'Newspaper']]
y = dataset['Sales']
```

[20]:
```
#Splitting the dataset
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,␣
 ↪random_state = 100)
```

[21]:
```
#Fitting the Multiple Linear Regression model
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
mlr.fit(x_train, y_train)
```

[21]:
```
LinearRegression()
```

[22]:
```
#Intercept and Coefficient
print("Intercept: ", mlr.intercept_)
```

```
print("Coefficients:")
list(zip(x, mlr.coef_))
```

```
Intercept:  2.6527896688794996
Coefficients:
```

[22]:
```
[('TV', 0.04542559602399794),
 ('Radio', 0.18975772766893612),
 ('Newspaper', 0.004603078953112024)]
```

[23]:
```
#Prediction of test set
y_pred_mlr= mlr.predict(x_test)

#Predicted values
print("Prediction for test set: \n{}".format(y_pred_mlr))
```

```
Prediction for test set:
[10.62160072 20.00625302 16.91850882 19.17040746 20.94974131 13.12284284
 11.80740696 12.32019766 20.57806782 20.95662688 10.79096475 19.54868702
  6.42403866 15.23133391  8.97226257  7.89897862 16.23599497 12.02636477
 17.09702178 11.26080277 16.97826292  9.75655721 20.82389762 17.20916742
 15.13816239 21.97290698 19.20181841 10.07501899 19.39017185 14.8673761
 14.36798893  7.55604543  9.96742165 14.76342565  7.20995576 13.60003295
  7.49088656 11.70865932 13.46091883 15.2229793  17.18088277 13.56738329
 14.30942267 13.72909849 11.88559349  8.77039705 12.1244102  19.20252289
  9.08376601  5.15367352 16.22852749 18.14111213 12.94835466 16.86274503
 17.86462435 12.33930625  4.3575739  11.25904494 16.11560622 13.56602169]
```

[24]:
```
#Actual value and the predicted value
mlr_diff = pd.DataFrame({'Actual value': y_test, 'Predicted value': y_pred_mlr})
mlr_diff.head()
```

[24]:

	Actual value	Predicted value
126	6.6	10.621601
104	20.7	20.006253
99	17.2	16.918509
92	19.4	19.170407
111	21.8	20.949741