central test linear regression

March 20, 2024

[39]:

```
# importing the necesary libraries
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression,Ridge
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,r2_score
import matplotlib.pyplot as plt
```

[40]:

```
# Loading the dataset
data=pd.read_csv('C:\\Users\\lynda\\Desktop\\company.csv')
data.head()
```

[40]:

```
   R&D Spend  Administration  Marketing Spend       State     Profit
0  165349.20       136897.80        471784.10    New York  192261.83
1  162597.70       151377.59        443898.53  California  191792.06
2  153441.51       101145.55        407934.54     Florida  191050.39
3  144372.41       118671.85        383199.62    New York  182901.99
4  142107.34        91391.77        366168.42     Florida  166187.94
```

[41]:

```
#checking for null values
data.isna().sum()
```

[41]:

```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

[42]:

```
# defining the x(independent valu)
x=data[['R&D Spend','Administration','Marketing Spend']]
x.head()
```

[42]:

```
   R&D Spend  Administration  Marketing Spend
0  165349.20       136897.80        471784.10
1  162597.70       151377.59        443898.53
2  153441.51       101145.55        407934.54
```

```
3  144372.41        118671.85        383199.62
4  142107.34         91391.77        366168.42
```

[43]:
```
# y(dependent values)
y=data['Profit']
y.head()
```

[43]:
```
0    192261.83
1    191792.06
2    191050.39
3    182901.99
4    166187.94
Name: Profit, dtype: float64
```

[44]:
```
# spliting my data into training and testing set
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.2,␣
 ↪random_state=42)
```

[45]:
```
# Building my Ordinary Linear regression
model = LinearRegression()
model.fit(x_train, y_train)
```

[45]:
```
LinearRegression()
```

[47]:
```
#calculating accuracy score
y_pred = model_ord.predict(x_test)
print('y_pred_ord',y_pred_ord)
```

```
y_pred_ord [126703.02716461  84894.75081556  98893.41815974  46501.70815036
 129128.39734381  50992.69486261 109016.5536578  100878.4641454
  97700.59638629 113106.15292226]
```

[49]:
```
accuracy = r2_score(y_test,y_pred)

print('accuracy:',accuracy)
```

```
accuracy: 0.900065308303732
```

[54]:
```
# Optimization of Linear Regression
scaler = StandardScaler()
x_train_opt = scaler.fit_transform(x_train)
x_test_opt = scaler.transform(x_test)
# using ridge
ridge_model=Ridge(alpha=0.5)
ridge_model.fit(x_train_opt, y_train)
y_pred_ridge= ridge_model.predict(x_test_opt)
accuracy2= r2_score(y_test,y_pred_ridge)
print('accuracy2',accuracy2)
```

```
accuracy2 0.8983198084871222
```

[55]:

```
if accuracy>accuracy2:
    print('The ordinary model performs better than the optimized model')
elif accuracy < accuracy2:
    print('The ordinary model performs better than the optimized')
else:
    print('Both modes have the same accuracy')
```

```
The ordinary model performs better than the optimized model
```

[ ]:

```
```

[ ]:

```
```

[ ]:

```
```