PolynomialRegression (1)

August 19, 2022

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.metrics import mean_absolute_error, mean_squared_error
```

[2]:
```
data = pd.read_csv('Advertising.csv',index_col=0)
data.head()
```

[2]:

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

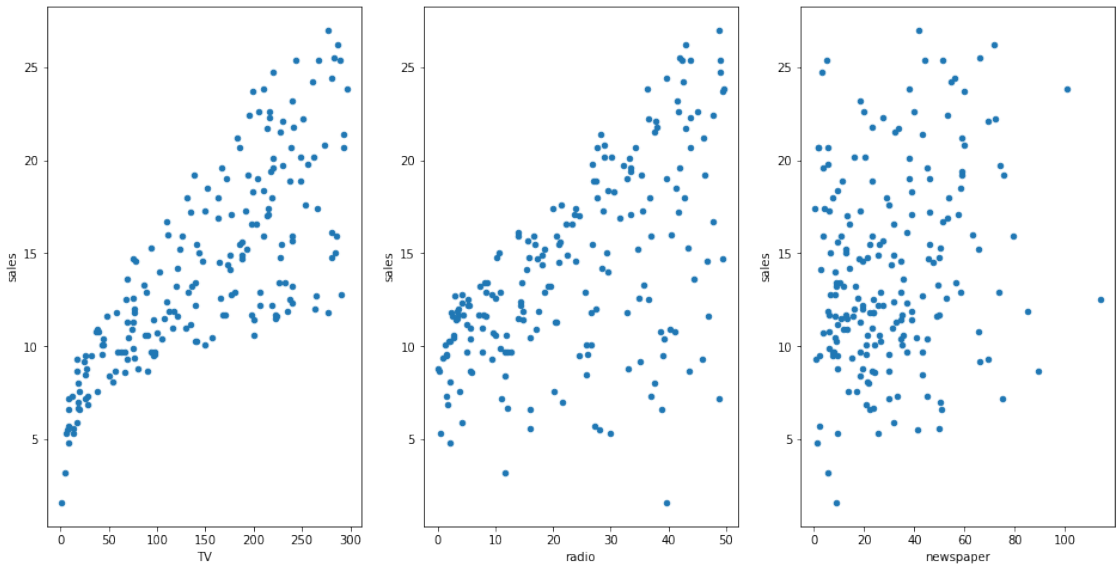
[3]:
```
data.columns
```

[3]:
```
Index(['TV', 'radio', 'newspaper', 'sales'], dtype='object')
```

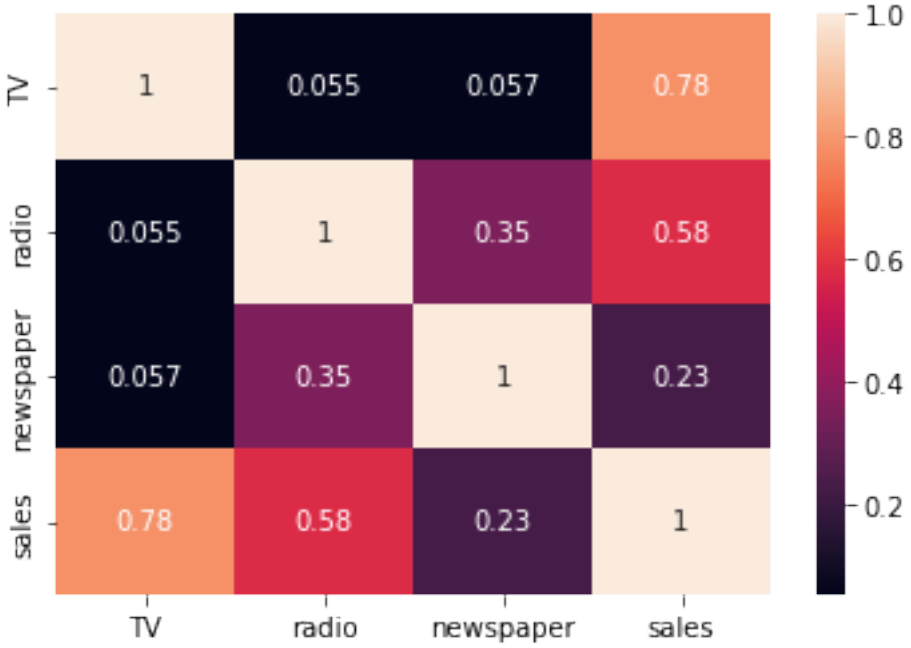
[4]:
```
data.shape
```

[4]:
```
(200, 4)
```

[5]:
```
fig, axs = plt.subplots(1, 3)
data.plot(kind='scatter', x='TV', y='sales', ax=axs[0], figsize=(16,8))
data.plot(kind='scatter', x='radio', y='sales', ax=axs[1])
data.plot(kind='scatter', x='newspaper', y='sales', ax=axs[2]);
```



[6]:
```
sns.heatmap(data.corr(), annot = True);
```



[7]:
```
features = data[['TV', 'newspaper','radio']]
target = data[['sales']].values
```

Convert Features to Polynomial features

[8]:
```
from sklearn.preprocessing import PolynomialFeatures
features_poly = PolynomialFeatures(degree=2)

TV_poly = features_poly.fit_transform(features[['TV']])
newspaper_poly = features_poly.fit_transform(features[['newspaper']])
radio_poly=features_poly.fit_transform(features[['radio']])
```

[9]:
```
poly_features = pd.concat([pd.DataFrame(TV_poly),pd.
↪DataFrame(newspaper_poly),pd.DataFrame(radio_poly)], axis = 1)
```

[10]:
```
poly_features
```

[10]:

	0	1	2	0	1	2	0	1	2
0	1.0	230.1	52946.01	1.0	69.2	4788.64	1.0	37.8	1428.84
1	1.0	44.5	1980.25	1.0	45.1	2034.01	1.0	39.3	1544.49
2	1.0	17.2	295.84	1.0	69.3	4802.49	1.0	45.9	2106.81
3	1.0	151.5	22952.25	1.0	58.5	3422.25	1.0	41.3	1705.69
4	1.0	180.8	32688.64	1.0	58.4	3410.56	1.0	10.8	116.64
..	…	…	…	…	…	…	…	…	…
195	1.0	38.2	1459.24	1.0	13.8	190.44	1.0	3.7	13.69
196	1.0	94.2	8873.64	1.0	8.1	65.61	1.0	4.9	24.01
197	1.0	177.0	31329.00	1.0	6.4	40.96	1.0	9.3	86.49
198	1.0	283.6	80428.96	1.0	66.2	4382.44	1.0	42.0	1764.00
199	1.0	232.1	53870.41	1.0	8.7	75.69	1.0	8.6	73.96

[200 rows x 9 columns]

[11]:
```
from sklearn.model_selection import train_test_split
```

[12]:
```
X_train, X_test, y_train, y_test = train_test_split(poly_features, target,␣
↪random_state = 6)
```

[13]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(150, 9)
(50, 9)
(150, 1)
(50, 1)
```

[14]:
```
from sklearn.linear_model import LinearRegression
```

[15]:
```
my_model = LinearRegression()
my_model.fit(X_train, y_train)
```

[15]:
```
LinearRegression()
```

Prediction and Evaluation on Train Data

[16]:
```
preds_train = my_model.predict(X_train)
print('MAE : ', mean_absolute_error(y_train, preds_train))
print('MSE : ', mean_squared_error(y_train, preds_train))
print('R2 Score : ', my_model.score(X_train, y_train))
```

```
MAE :  1.202893489619854
MSE :  2.488776665854576
R2 Score :  0.9139135358655949
```

Prediction and Evaluation on Test Data

[17]:
```
preds = my_model.predict(X_test)
print('MAE : ', mean_absolute_error(y_test, preds))
print('MSE : ', mean_squared_error(y_test, preds))
print('R2 score : ', my_model.score(X_test, y_test))
```

```
MAE :  0.9912741510674818
MSE :  1.5270219257565474
R2 score :  0.9293009924314467
```

[20]:
```
#For Degree 3
from sklearn.preprocessing import PolynomialFeatures
poly_feature1=PolynomialFeatures(degree=3)
TV_poly1=poly_feature1.fit_transform(features[['TV']])
NP_poly1=poly_feature1.fit_transform(features[['newspaper']])
radio_poly1=poly_feature1.fit_transform(features[['radio']])
Fetaure_transformed=pd.concat([pd.DataFrame(TV_poly1),pd.DataFrame(NP_poly1),pd.
 ↪DataFrame(radio_poly1)],axis=1)
x_train1,x_test1,y_train1,y_test1=train_test_split(Fetaure_transformed,target,random_state=6)
lr1=LinearRegression()
lr1.fit(x_train1,y_train1)
pred1 = lr1.predict(x_test1)
print('MAE : ', mean_absolute_error(y_test1, pred1))
print('MSE : ', mean_squared_error(y_test1, pred1))
print('R2 score : ', lr1.score(x_test1, y_test1))
```

```
MAE :  1.0118905313835787
MSE :  1.6566344481057027
R2 score :  0.9233001115377409
```

[22]:
```
#For Degree 100
from sklearn.preprocessing import PolynomialFeatures
poly_feature2=PolynomialFeatures(degree=100)
TV_poly2=poly_feature2.fit_transform(features[['TV']])
```

```
print(TV_poly2)
NP_poly2=poly_feature2.fit_transform(features[['newspaper']])
radio_poly2=poly_feature2.fit_transform(features[['radio']])
Fetaure_transformed2=pd.concat([pd.DataFrame(TV_poly2),pd.
 ↪DataFrame(NP_poly2),pd.DataFrame(radio_poly2)],axis=1)
x_train2,x_test2,y_train2,y_test2=train_test_split(Fetaure_transformed2,target,random_state=6)
lr2=LinearRegression()
lr2.fit(x_train2,y_train2)
pred2 = lr2.predict(x_test2)
print('MAE : ', mean_absolute_error(y_test2, pred2))
print('MSE : ', mean_squared_error(y_test2, pred2))
print('R2 score : ', lr2.score(x_test2, y_test2))
```

```
[[1.00000000e+000 2.30100000e+002 5.29460100e+004 … 2.93649070e+231
  6.75686510e+233 1.55475466e+236]
 [1.00000000e+000 4.45000000e+001 1.98025000e+003 … 3.46163343e+161
  1.54042688e+163 6.85489960e+164]
 [1.00000000e+000 1.72000000e+001 2.95840000e+002 … 1.20722382e+121
  2.07642497e+122 3.57145096e+123]
 …
 [1.00000000e+000 1.77000000e+002 3.13290000e+004 … 2.00161296e+220
  3.54285493e+222 6.27085323e+224]
 [1.00000000e+000 2.83600000e+002 8.04289600e+004 … 2.31851654e+240
  6.57531290e+242 1.86475874e+245]
 [1.00000000e+000 2.32100000e+002 5.38704100e+004 … 6.85746761e+231
  1.59161823e+234 3.69414591e+236]]
MAE :  3.6754203555320073
MSE :  20.17402020773462
R2 score :  0.06597070854228848
```