西南石油大學

NAME: KARIMA ZAFAR

ID: 201739060051

MAJOR: SOFTWARE ENGINEERING

GRADE: 2017

SALES ADVERTISING

The data set contains product sales in 200 different markets, and each sales corresponds to 3 types of advertising media input costs: TV, radio, and newspaper. If we can analyze the relationship between advertising media investment and sales, we can better allocate advertising spending and maximize sales.

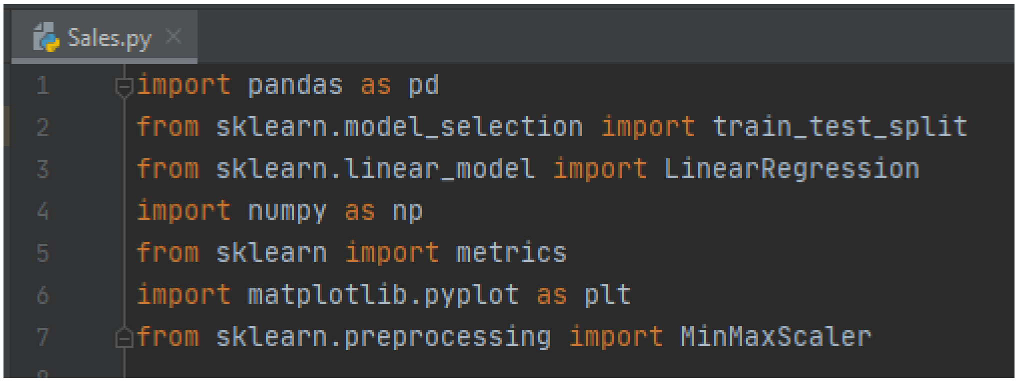
The following experiments are now required:

Use the pandas library to read the data set and get the corresponding matrix. Use matplotlib library to draw: data scatter plot of TV, Radio, Newspaper and product sales. Also, draw test/predict graph.

Specific requirements:

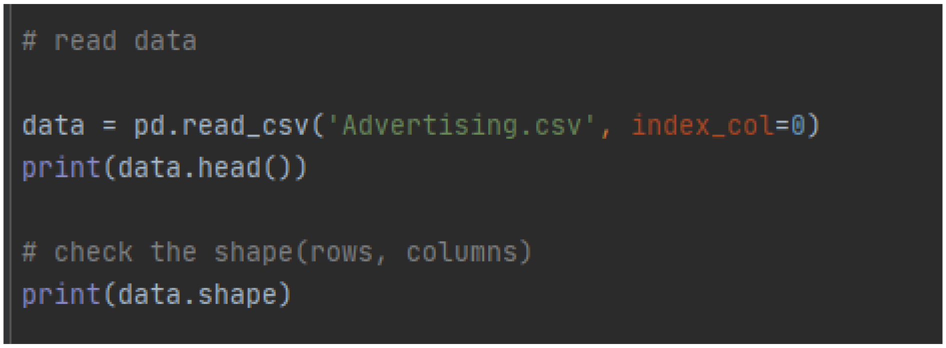
- The result is a picture, TV, radio, and newspaper need to be represented by points of different shapes.
- The X-axis of the graph is advertising spend, and the Y-axis is the value of sales.
- Need to draw grid reference lines in dotted form.

1. Import packages



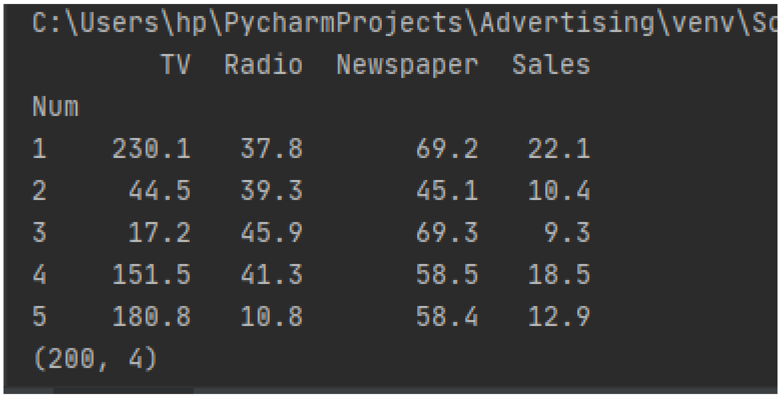
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

2. Read data



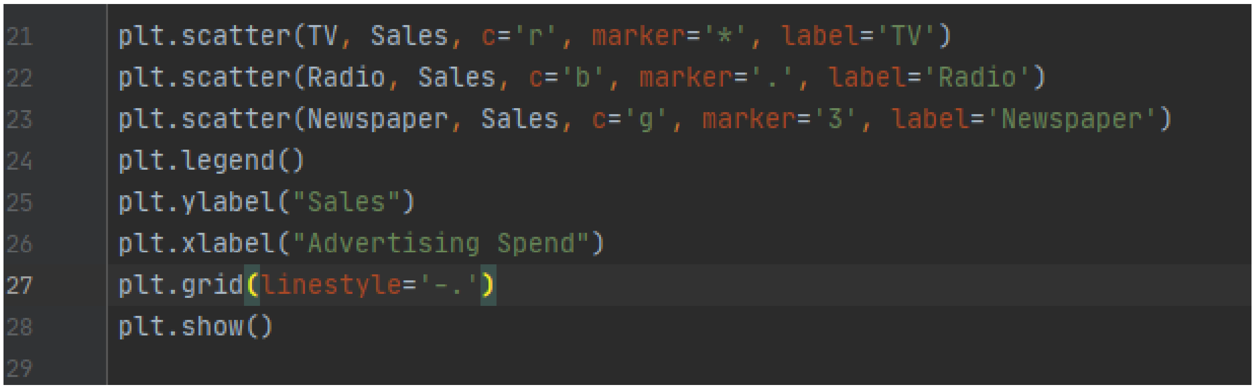
```python
# read data

data = pd.read_csv('Advertising.csv', index_col=0)
print(data.head())

# check the shape(rows, columns)
print(data.shape)
```

3. Print first 5 data and the shape

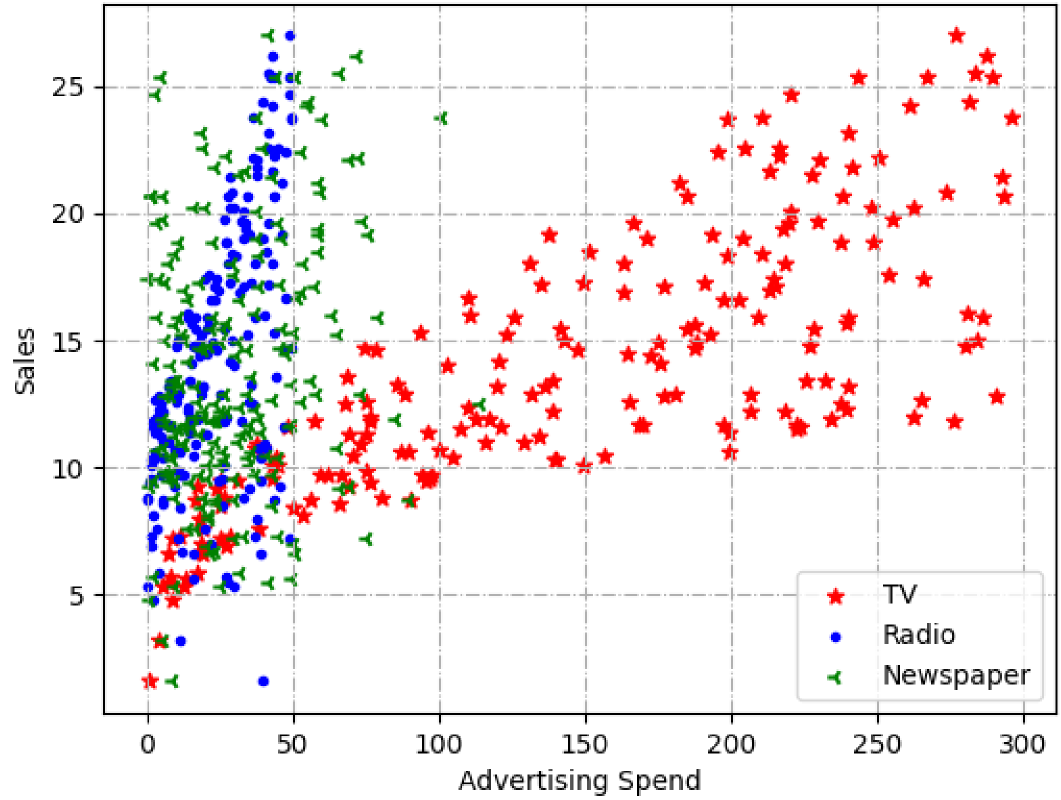


```
C:\Users\hp\PycharmProjects\Advertising\venv\Sc
        TV  Radio  Newspaper  Sales
Num
1    230.1   37.8       69.2   22.1
2     44.5   39.3       45.1   10.4
3     17.2   45.9       69.3    9.3
4    151.5   41.3       58.5   18.5
5    180.8   10.8       58.4   12.9
(200, 4)
```

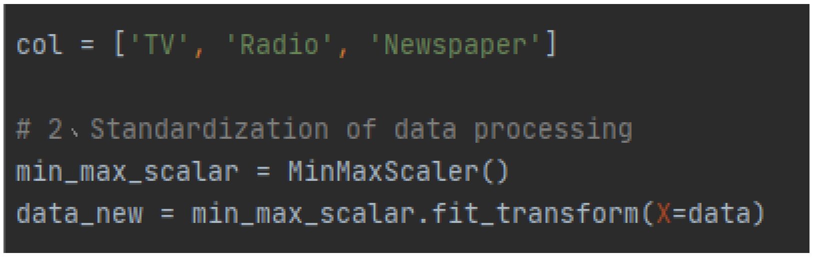
4. Plot scatter graph



```python
plt.scatter(TV, Sales, c='r', marker='*', label='TV')
plt.scatter(Radio, Sales, c='b', marker='.', label='Radio')
plt.scatter(Newspaper, Sales, c='g', marker='3', label='Newspaper')
plt.legend()
plt.ylabel("Sales")
plt.xlabel("Advertising Spend")
plt.grid(linestyle='-.')
plt.show()

```



5. Standardize data



```
col = ['TV', 'Radio', 'Newspaper']

# 2、Standardization of data processing
min_max_scalar = MinMaxScaler()
data_new = min_max_scalar.fit_transform(X=data)
```

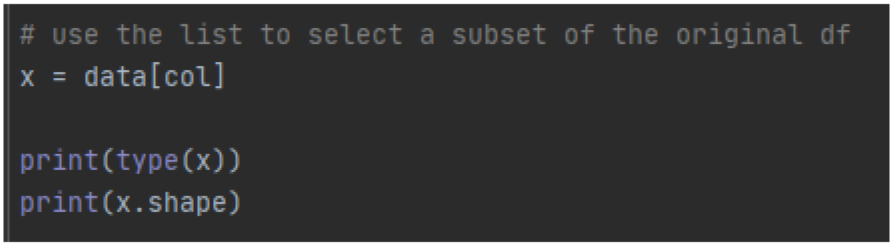
After standardizing the data, a polynomial fitting model in linear regression is established. Finally, draw a drawing based on the predicted result and the actual result.

Specific requirements:

1. Take 25% for the test set and 75% for the training set. Because there are three data features (TV, Radio, Newspaper), it is impossible to draw a two-dimensional graph of features and prediction results. Therefore, the X axis is replaced by the test sample index, and the Y axis is the product sales.

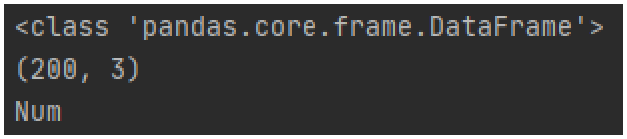
In the graph, the blue line represents the predicted sales of the model for the test set, and the red line represents the actual product sales corresponding to the test set. The title of the figure indicates the degree of the linear model polynomial.

6. Select a subset of the original dataframe



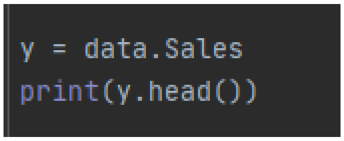
```
# use the list to select a subset of the original df
x = data[col]

print(type(x))
print(x.shape)
```

7. Print type and shape of the data again



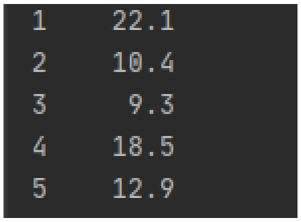
```
<class 'pandas.core.frame.DataFrame'>
(200, 3)
Num
```

8. Print first 5 data for Sales (y)



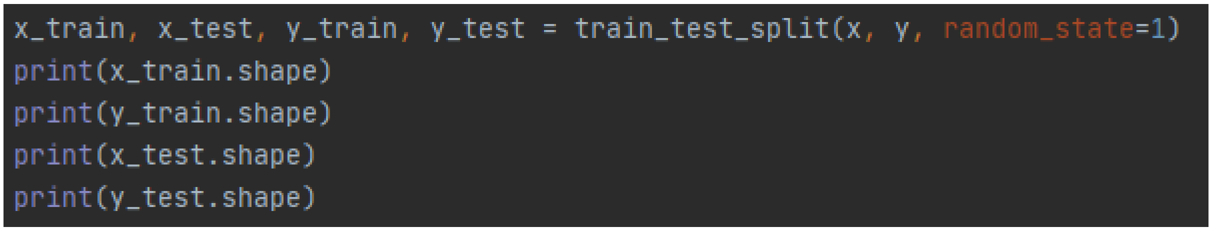
```
y = data.Sales
print(y.head())
```

9. Output



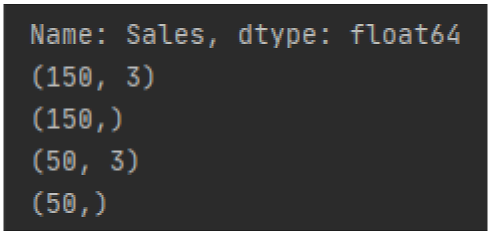
```
1    22.1
2    10.4
3     9.3
4    18.5
5    12.9
```

10. Split dataset into test and train data



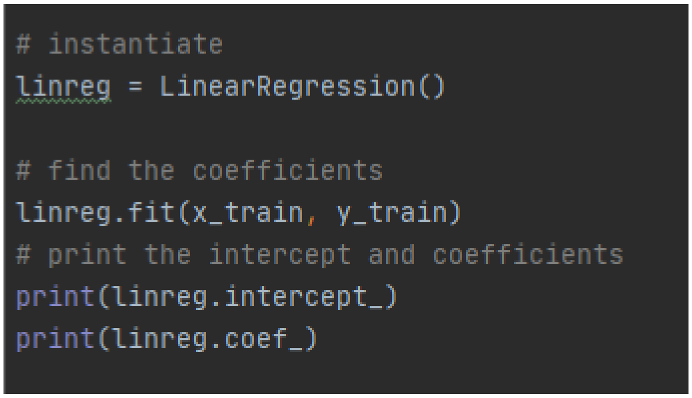
```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

11. Print train and test data’s shape and type



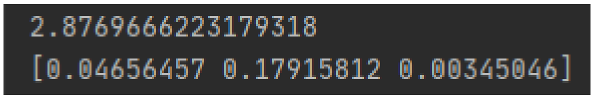
```
Name: Sales, dtype: float64
(150, 3)
(150,)
(50, 3)
(50,)
```

12. Instantiate Linear Regression



```
# instantiate
linreg = LinearRegression()

# find the coefficients
linreg.fit(x_train, y_train)
# print the intercept and coefficients
print(linreg.intercept_)
print(linreg.coef_)
```

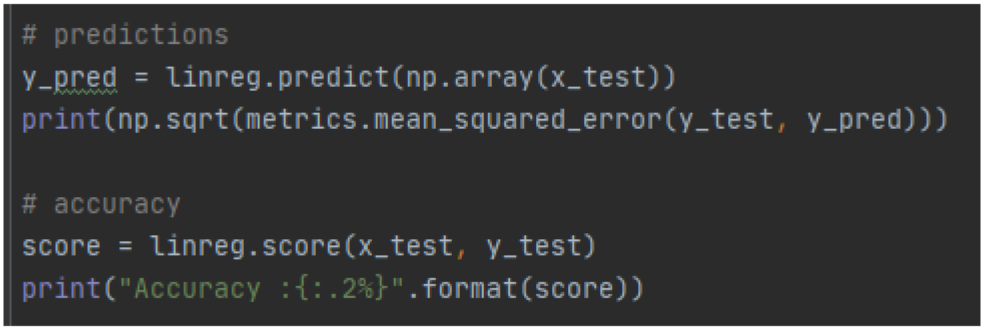
13. Print intercept and co-efficient.



```
2.8769666223179318
[0.04656457 0.17915812 0.00345046]
```

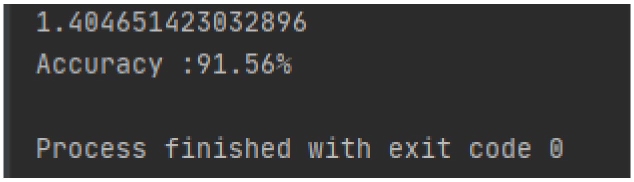
14. Make y predictions and compute the Root Mean Squared Error for the Sales predictions

Print accuracy of the predictions



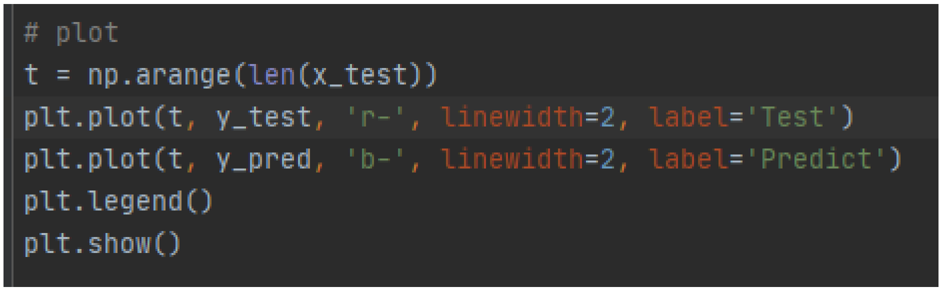
```
# predictions
y_pred = linreg.predict(np.array(x_test))
print(np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

# accuracy
score = linreg.score(x_test, y_test)
print("Accuracy :{:.2%}".format(score))
```

15. Output



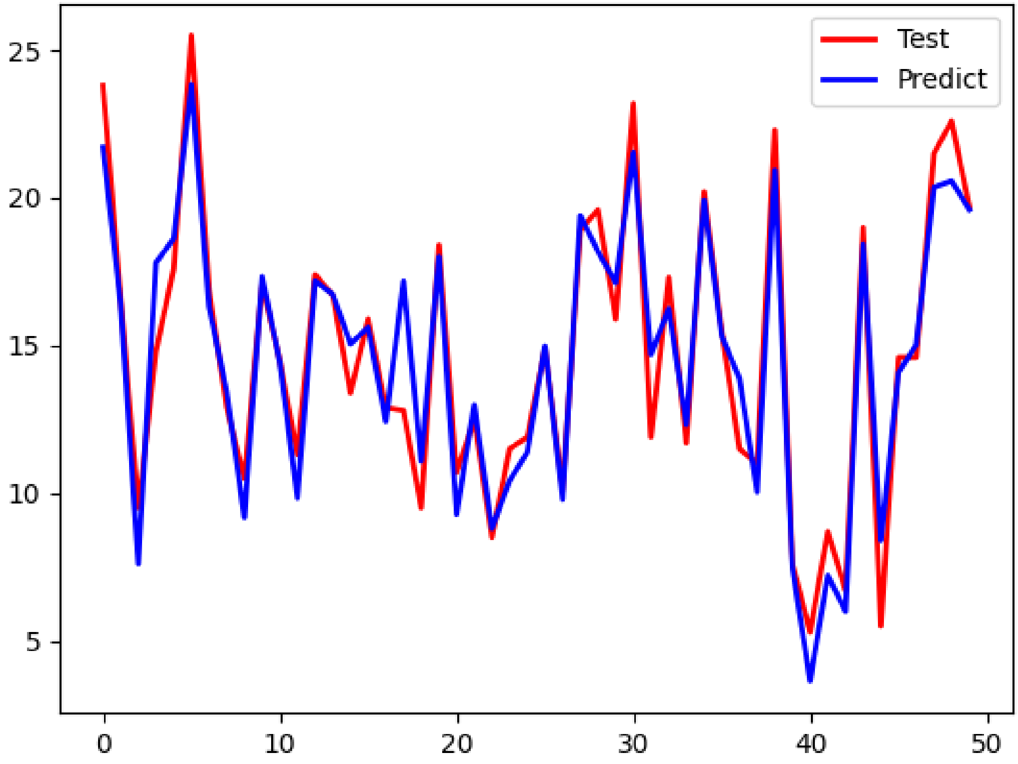
```
1.404651423032896
Accuracy :91.56%

Process finished with exit code 0
```

16. Plot graph



```
# plot
t = np.arange(len(x_test))
plt.plot(t, y_test, 'r-', linewidth=2, label='Test')
plt.plot(t, y_pred, 'b-', linewidth=2, label='Predict')
plt.legend()
plt.show()
```

17. Output



18. Conclusion:

The meaning of the coefficients corresponding to each feature:

Given the advertising investment of Radio and Newspaper, if one additional unit is invested in TV advertising, the corresponding sales will increase by 0.0466 units. It is to join the other two media with fixed investment. For every $1,000 increase in TV advertising (because the unit is $1,000), sales will increase by 46.6 (because the unit is 1,000). But everyone should pay attention to the fact that the coefficient of the newspaper here is negative, so we can consider not using the newspaper feature.