Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B.Tech (Computer Engineering)
Course:	Data Mining and Warehouse Laboratory
Course Code:	DJ19CEL501
Experiment No.:	04

AIM: Implementation of Linear Regression for Single Variate and Multi-variate

THEORY:
Linear regression predicts outcomes based on one variable in single-variate and multiple variables in multi-variate. It finds the best-fit line for the data to make predictions.

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

     from sklearn.datasets import fetch_california_housing
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import mean_squared_error
     from sklearn.metrics import accuracy_score,classification_report, r2_score

     from ucimlrepo import fetch_ucirepo
```

Univariate

Dataset - Auto MPG (miles per gallon)

```
[ ]  # fetch dataset
     df = fetch_ucirepo(id=9)

     # data (as pandas dataframes)
     x = df.data.features
     y = df.data.targets
```

```
[ ]  df1 = pd.DataFrame(x)
     df2 =  pd.DataFrame(y)
```

```
[ ]  df1
```

	displacement	cylinders	horsepower	weight	acceleration	model_year	origin
0	307.0	8	130.0	3504	12.0	70	1
1	350.0	8	165.0	3693	11.5	70	1
2	318.0	8	150.0	3436	11.0	70	1
3	304.0	8	150.0	3433	12.0	70	1
4	302.0	8	140.0	3449	10.5	70	1
...	...	...	...	...	...	...	...
393	140.0	4	86.0	2790	15.6	82	1
394	97.0	4	52.0	2130	24.6	82	2
395	135.0	4	84.0	2295	11.6	82	1
396	120.0	4	79.0	2625	18.6	82	1
397	119.0	4	82.0	2720	19.4	82	1

398 rows × 7 columns

```
df2
```

	mpg
0	18.0
1	15.0
2	18.0
3	16.0
4	17.0
...	...
393	27.0
394	44.0
395	32.0
396	28.0
397	31.0

398 rows × 1 columns

```
[ ] df2["mpg"].unique()
```

```
array([18. , 15. , 16. , 17. , 14. , 24. , 22. , 21. , 27. , 26. , 25. ,
       10. , 11. ,  9. , 28. , 19. , 12. , 13. , 23. , 30. , 31. , 35. ,
       20. , 29. , 32. , 33. , 17.5, 15.5, 14.5, 22.5, 24.5, 18.5, 29.5,
       26.5, 16.5, 31.5, 36. , 25.5, 33.5, 20.5, 30.5, 21.5, 43.1, 36.1,
       32.8, 39.4, 19.9, 19.4, 20.2, 19.2, 25.1, 20.6, 20.8, 18.6, 18.1,
       17.7, 27.5, 27.2, 30.9, 21.1, 23.2, 23.8, 23.9, 20.3, 21.6, 16.2,
       19.8, 22.3, 17.6, 18.2, 16.9, 31.9, 34.1, 35.7, 27.4, 25.4, 34.2,
       34.5, 31.8, 37.3, 28.4, 28.8, 26.8, 41.5, 38.1, 32.1, 37.2, 26.4,
       24.3, 19.1, 34.3, 29.8, 31.3, 37. , 32.2, 46.6, 27.9, 40.8, 44.3,
       43.4, 36.4, 44.6, 40.9, 33.8, 32.7, 23.7, 23.6, 32.4, 26.6, 25.8,
       23.5, 39.1, 39. , 35.1, 32.3, 37.7, 34.7, 34.4, 29.9, 33.7, 32.9,
       31.6, 28.1, 30.7, 24.2, 22.4, 34. , 38. , 44. ])
```

```
[ ] df
```

```
{'data': {'ids':                    car_name
  0    chevrolet,chevelle,malibu
  1            buick,skylark,320
  2           plymouth,satellite
```

```
df1["mpg"] = df2["mpg"]
```

```
[ ] df1
```

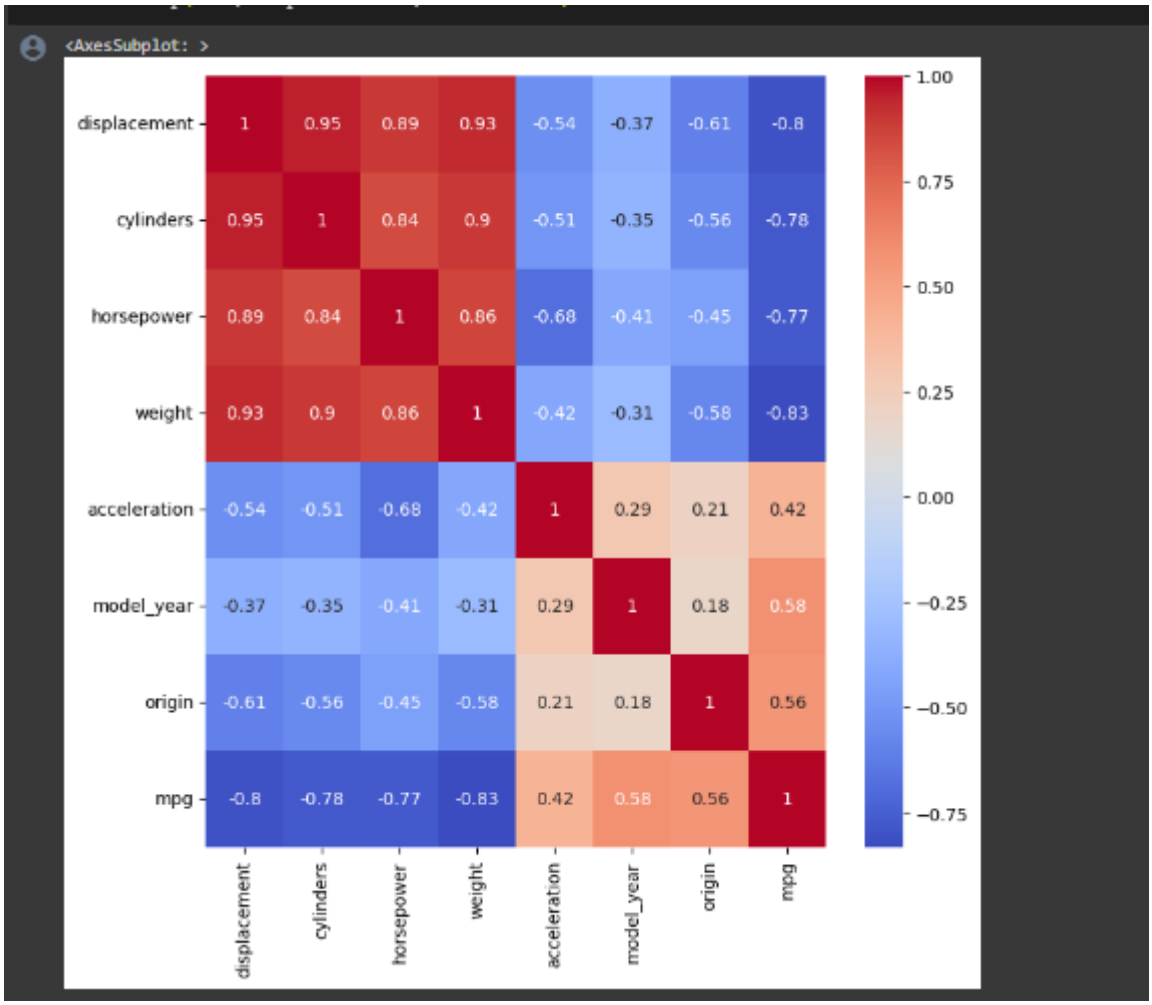
	displacement	cylinders	horsepower	weight	acceleration	model_year	origin	mpg
0	307.0	8	130.0	3504	12.0	70	1	18.0
1	350.0	8	165.0	3693	11.5	70	1	15.0
2	318.0	8	150.0	3436	11.0	70	1	18.0
3	304.0	8	150.0	3433	12.0	70	1	16.0
4	302.0	8	140.0	3449	10.5	70	1	17.0
...	...	...	...	...	...	...	...	...
393	140.0	4	86.0	2790	15.6	82	1	27.0
394	97.0	4	52.0	2130	24.6	82	2	44.0
395	135.0	4	84.0	2295	11.6	82	1	32.0
396	120.0	4	79.0	2625	18.6	82	1	28.0
397	119.0	4	82.0	2720	19.4	82	1	31.0

398 rows × 8 columns

```
[ ] df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   displacement  398 non-null    float64
 1   cylinders     398 non-null    int64
 2   horsepower    392 non-null    float64
 3   weight        398 non-null    int64
 4   acceleration  398 non-null    float64
 5   model_year    398 non-null    int64
 6   origin        398 non-null    int64
 7   mpg           398 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 25.0 KB
```

```
[ ] df1['horsepower'] = df1['horsepower'].fillna(df1['horsepower'].mean(), inplace=False)

[ ] df1.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 398 entries, 0 to 397
    Data columns (total 8 columns):
     #   Column        Non-Null Count  Dtype
    ---  ------        --------------  -----
     0   displacement  398 non-null    float64
     1   cylinders     398 non-null    int64
     2   horsepower    398 non-null    float64
     3   weight        398 non-null    int64
     4   acceleration  398 non-null    float64
     5   model_year    398 non-null    int64
     6   origin        398 non-null    int64
     7   mpg           398 non-null    float64
    dtypes: float64(4), int64(4)
    memory usage: 25.0 KB

[ ] f, ax = plt.subplots(figsize=(8, 8))
    corr = df1.corr()
    sns.heatmap(corr, cmap="coolwarm", annot = True)
```



Regression Analysis for model_year

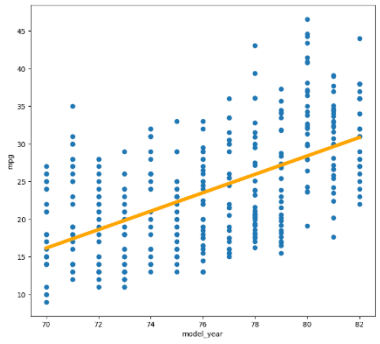
```
[ ] model_year = df1[['model_year']]
    mpg = df2['mpg']

[ ] lr1 = LinearRegression()
    lr1.fit(model_year, mpg)

    ▾ LinearRegression
    LinearRegression()

[ ] print(lr1.coef_)
    print(lr1.intercept_)
    print(mean_squared_error(mpg, lr1.predict(model_year)))

    [1.22444564]
    -69.55560174195769
    40.48897938876626
```

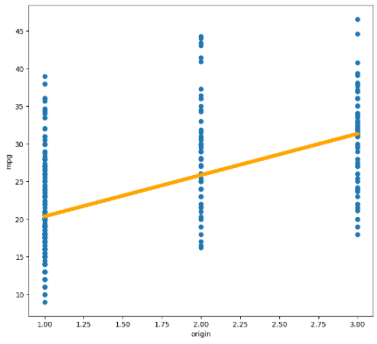
Regression Analysis for origin

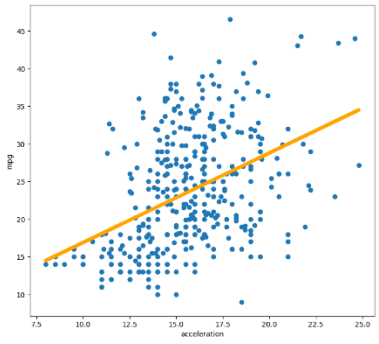
```
[ ] origin = df1[['origin']]
    mpg = df2['mpg']

[ ] lr2 = LinearRegression()
    lr2.fit(origin, mpg)

    ▾ LinearRegression
    LinearRegression()

[ ] print(lr2.coef_)
    print(lr2.intercept_)
    print(mean_squared_error(mpg, lr2.predict(origin)))

    [5.49079532]
```

Regression Analysis for acceleration

```
[ ] acceleration = df1[['acceleration']]
    mpg = df2['mpg']

[ ] lr3 = LinearRegression()
    lr3.fit(acceleration, mpg)

    ▾ LinearRegression
    LinearRegression()

[ ] print(lr3.coef_)
    print(lr3.intercept_)
    print(mean_squared_error(mpg, lr3.predict(acceleration)))

    [1.19120453]
    4.969793004253912
    50.172194407701355

[ ] f = plt.figure()
    f, ax = plt.subplots(1, 3, figsize=(30, 8))

    ax = plt.subplot(1, 3, 1)
    plt.ylabel('mpg')
    plt.xlabel('model_year')
    ax = plt.scatter(model_year, mpg)
    ax = plt.plot(model_year, lr1.predict(model_year), linewidth=5.0, color='orange')

    ax = plt.subplot(1, 3, 2)
    plt.ylabel('mpg')
    plt.xlabel('origin')
    ax = plt.scatter(origin, mpg)
    ax = plt.plot(origin, lr2.predict(origin), linewidth=5.0, color='orange')

    ax = plt.subplot(1, 3, 3)
    plt.ylabel('mpg')
    plt.xlabel('acceleration')
    ax = plt.scatter(acceleration, mpg)
    ax = plt.plot(acceleration, lr3.predict(acceleration), linewidth=5.0, color='orange')

    ax = plt.show()
```







Multi variate

Dataset - California Housing

```
[ ] df = fetch_california_housing()

[ ] df
```

```
{'data': array([[   8.3252    ,   41.        ,    6.98412698, ...,    2.55555556,
          37.88    , -122.23      ],
        [   8.3014    ,   21.        ,    6.23813708, ...,    2.10984183,
          37.86    , -122.22      ],
        [   7.2574    ,   52.        ,    8.28813559, ...,    2.80225989,
          37.85    , -122.24      ],
        ...,
        [   1.7       ,   17.        ,    5.20554273, ...,    2.3256351 ,
          39.43    , -121.22      ],
        [   1.8672    ,   18.        ,    5.32951289, ...,    2.12320917,
          39.43    , -121.32      ],
        [   2.3886    ,   16.        ,    5.25471698, ...,    2.61698113,
          39.37    , -121.24      ]]),
 'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]),
 'frame': None,
 'target_names': ['MedHouseVal'],
 'feature_names': ['MedInc',
  'HouseAge',
  'AveRooms',
  'AveBedrms',
  'Population',
  'AveOccup',
  'Latitude',
  'Longitude'],
 'DESCR': '.. _california_housing_dataset:\n\nCalifornia Housing dataset\n--------------------------\n\n**Data Set Characteristics:**\n\n    :Number of Instances: 20640\n\n    :Number of Attributes: 8 numeric, predictive attributes and the target\n\n    :Attribute Information:\n        - MedInc        median income in block group\n        - HouseAge      median house age in block group\n        - AveRooms      average number of rooms per household\n        - AveBedrms     average number of bedrooms per household\n        - Population    block group population\n        - AveOccup      average number of household members\n        - Latitude      block group latitude\n        - Longitude     block group longitude\n\n    :Missing Attribute Values: None\n\nThis dataset was obtained from the StatLib repository.\nhttps://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html\n\nThe target variable is the median house value for California districts,\nexpressed in hundreds of thousands of dollars ($100,000).\n\nThis dataset was derived from the 1990 U.S. census, using one row per census\nblock group. A block group is the smallest geographical unit for which the U.S.\nCensus Bureau publishes sample data (a block group typically has a population\nof 600 to 3,000 people).\n\nAn household is a group of people residing within a home. Since the average\nnumber of rooms and bedrooms in this dataset are provided per household, these\ncolumns may take surprisingly large values for block groups with few households\nand many empty houses, such as vacation resorts.\n\nIt can be downloaded/loaded using the\n:func:`sklearn.datasets.fetch_california_housing` function.\n\n.. topic:: References\n\n  - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,\n    Statistics and Probability Letters, 33 (1997) 291-297\n'}
```

```
[ ] dataset = pd.DataFrame(df.data)

[ ] dataset.columns = df.feature_names

[ ] dataset
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25
...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24

20640 rows × 8 columns

```
[ ] x = dataset
    y = df.target

[ ] x
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25

```
[ ] y

    array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894])

[ ] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=42)

[ ] model = LinearRegression()

[ ] model.fit(x_train, y_train)

    ▾ LinearRegression
    LinearRegression()

[ ] y_pred = model.predict(x_test)

[ ] score = r2_score(y_test, y_pred)

[ ] score

    0.5957702326061662

[ ] print(model.coef_)
    print(model.intercept_)

    [ 4.45822565e-01  9.68186799e-03 -1.22095112e-01  7.78599557e-01
     -7.75740400e-07 -3.37002667e-03 -4.18536747e-01 -4.33687976e-01]
    -37.05624133152514
```