60004210210

Amartya Mishra

COMPS – C31

ML Experiment 1

60004210210
Amartya Mishra
COMPS - C3)

ML - Experiment - 1

Aim: To perform Data cleaning & preprocessing on the Dataset & implement Linear Regression

Theory:
Data cleaning is the process of fixing or removing ~~incomplete~~ incorrect, corrupted, incorrectly formatted, duplicate or incomplete data within a dataset.

While Integrating Multiple data sources, discrepancy is possible which makes outcomes & algorithms unreliable.

Steps to clean Data:
→ Remove irrelevent data
→ Remove duplicate & incomplete cases
→ Fix Structural Errors
→ Deal with missing Data
→ Identify & review outliers
→ Filter out outliers
→ Encode categorical data
→ Splitting the Dataset.
→ feature Scaling.

Linear Regression:
It is a special type of machine learning algorithm more specifically a supervised Algorithm. It learns from the labbled dataset & maps the data points to the most optimized linear functions

It computes the linear relationship between a dependent variable & one or more independent features.

Simple Linear Regression :

$y_i = \beta_0 + \beta_1 x$

Y: dependent variable
x: Independent variable
β_0: Intercept
β_1: Slope

The goal of Algorithm is to find the best fit line equation that can predict the value based on independent variables

$$\text{cost function } (J) = \frac{1}{n} \sum_{i}^{n} (\hat{y}_i - y_i)^2$$

n → No. of datapoints
y_i → actual value
$\hat{y}_i$ → Predicted value

Conclusion: Thus we understood steps of Data pre-processing & cleaning & there after implemented linear regression.

Implementation: Done in 3 ways, firstly on given data using direct formulas.

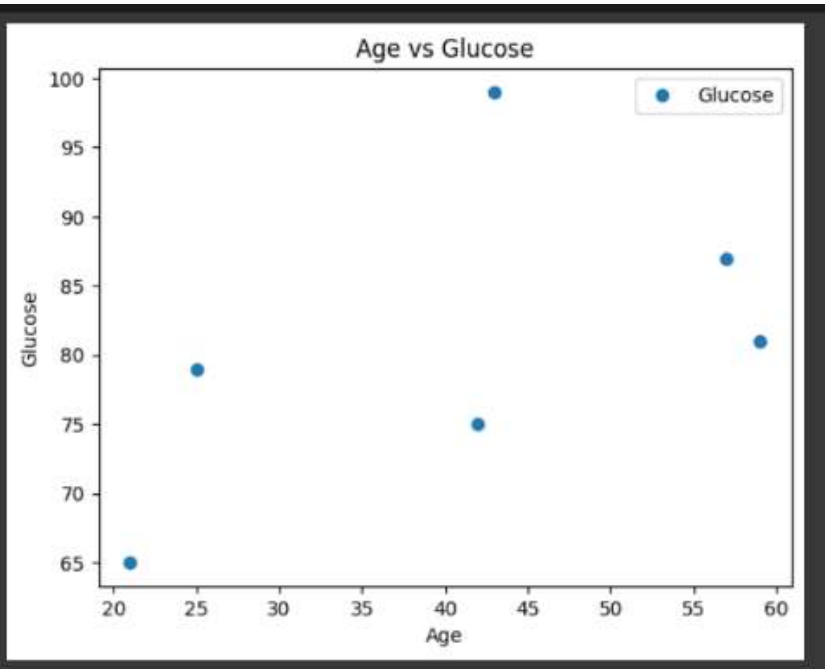
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
[ ] d = {'Age': [43, 21, 25, 42, 57, 59], 'Glucose': [99, 65, 79, 75, 87, 81]}
```

```
[ ] n = len(d['Age'])
    print(n)

    6
```

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

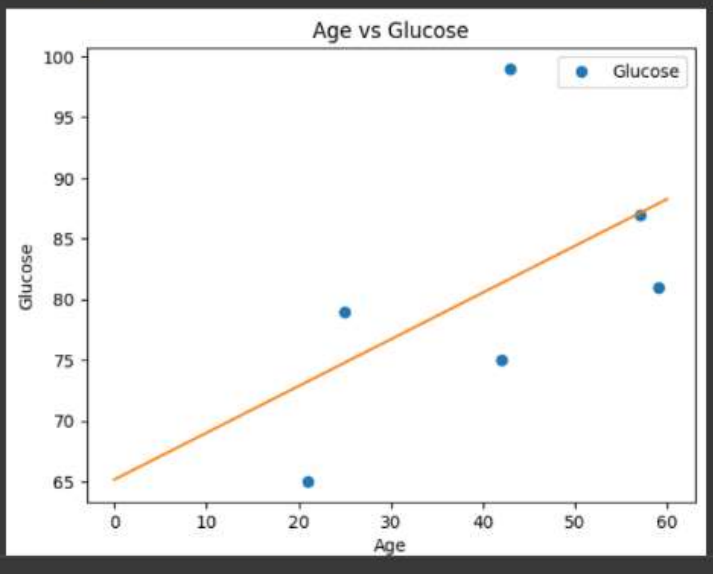
```
df.plot(x='Age', y='Glucose', style='o')
plt.xlabel('Age')
plt.ylabel('Glucose')
plt.title('Age vs Glucose')
plt.show()
```




```
[ ] xs=ys=xy=x=y=0
```

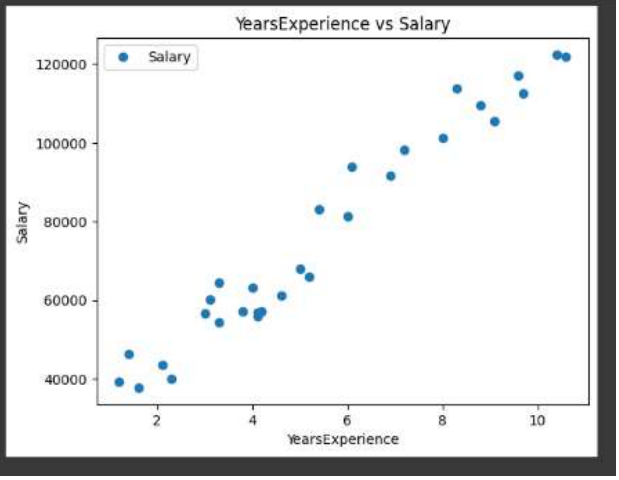
```
[ ] for i in range(0,n):
        x += d['Age'][i]
        y += d['Glucose'][i]
        xs += d['Age'][i]**2
        ys += d['Glucose'][i]**2
        xy += d['Age'][i]*d['Glucose'][i]
```

```
[ ] bo = ((y*xs)-(x*xy))/((n*xs)-(x**2))

[ ] b1 = ((n*xy)-(x*y))/((n*xs)-(x**2))

df.plot(x='Age', y='Glucose', style='o')
plt.plot([0, 60], [bo + b1*0, bo + b1*60])
plt.xlabel('Age')
plt.ylabel('Glucose')
plt.title('Age vs Glucose')
plt.show()
```



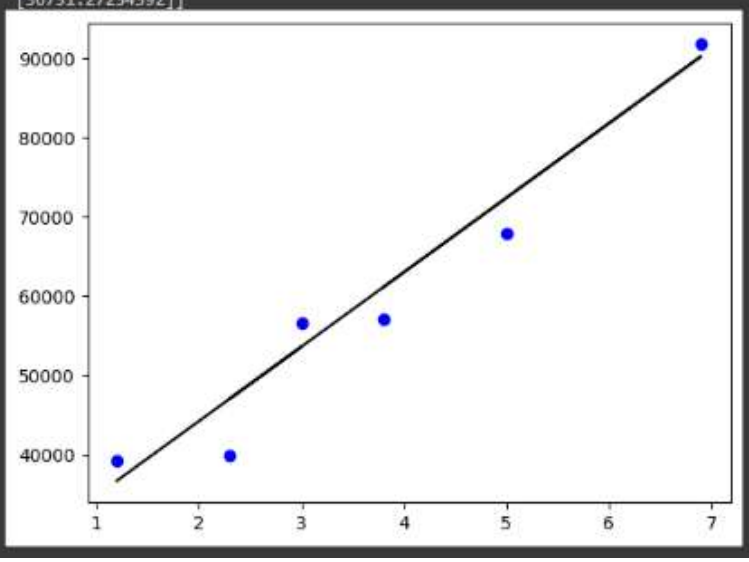
Then on an imported dataset using inbuilt functions and libraries

```
[ ] ycap = bo + (55*b1)
    print(ycap)

    86.32894560107455

[ ] df2 = pd.read_csv('Salary_dataset.csv')

df2.plot(x='YearsExperience', y='Salary', style='o')
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.title('YearsExperience vs Salary')
plt.show()
```



```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression
    X = np.array(df2['YearsExperience']).reshape(-1, 1)
    y = np.array(df2['Salary']).reshape(-1, 1)

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

[ ] regr = LinearRegression()
    regr.fit(X_train, y_train)
    print(regr.score(X_test, y_test))

    0.9450507958680934

    y_pred = regr.predict(X_test)
    plt.scatter(X_test, y_test, color ='b')
    plt.plot(X_test, y_pred, color ='k')
    print(X_test)
    print(y_pred)
    plt.show()
```

```
    print(y_pred)
    plt.show()

    [[3.8]
     [6.9]
     [5. ]
     [2.3]
     [3. ]
     [1.2]]
    [[61140.05516025]
     [90242.83467203]
     [72405.64722933]
     [47058.0650739 ]
     [53629.66044753]
     [36731.27234392]]
```



And then the same procedure is implemented from scratch: