**MASTER’S OF COMPUTER SCIENCE**

**RESEARCH TOPIC: DESIGN AND IMPLEMENTATION OF PYTHON PROGRAM FOR DATA VISUALIZATION**

**MAJOR: PROGRAMMING WITH PYTHON - DLMDSPWP01**

**NAME: BHAVYASHREE P**

**MATRICULATION: 9210834**

**UNIVERSITY: INTERNATIONAL UNIVERSITY OF APPLIED SCIENCE**

**Thesis Advisor: DR. COSMINA CROITORU**

[**c.croitoru@iubh-fernstudium.de**](mailto:c.croitoru@iubh-fernstudium.de)

**Due Date： APRIL 2023**

|  |  |
| --- | --- |
| **Table of contents** | **page number** |
| introduction | 3 |
| ……Aim | 3 |
| ……Objective | 3 |
| dataset description | 5 |
| ……Data collection | 6 |
| ……Storage of data | 7 |
| ……Data access | 8 |
| methodology | 9 |
| data visualization | 10 |
| Result and evaluation | 14 |
| Conclusion | 17 |
| reference | 17 |
| code deployed | 18 |

# **INTRODUCTION**

The assignment is completed through the implementation of python programming. The usage of the training dataset along with the provision of the accurately defined functions are known to impart the ideal situations. These functions are important to provide accurate results of the components of x and y which appear as a correct fit.

## **Aim**

The designing and development of the codes are to procure the Python Software. The utilization of the training data is subjected to choosing from four functions which are exclusively oriented through the total of 50 functions which were designed for the development of the Python software which has been idealized according to the dataset which has been provided for the completion of this assignment.

The overall aim of this program is to discover python software that has convenient options incorporated within the scope to distinctly diagnose the pairs which are known to consist of the x and y coordinates. Further, the development is also focused on the distinct distribution of all four enlisted functions. The visual portrayal of the data is also provided through this project.

## **Objective**

The programmer must be capable of building a SQL database and fusing it with a single, five-column spreadsheet. Each of the 50 ideal functions must be loaded into a separate database table from a CSV file in order to complete the operation. Test data should be handled line by line after the optimal functions have been loaded. The test data should come from a separate CSV file. The test data should be harmonized in accordance with the function with which it successfully conforms with I. (subcategory above). The outcomes of this harmonization have to be kept in a special, four-column SQLite database table. Upon job completion, a concise presentation of the y values, training data, test data, selected ideal functions, and the accompanying matched/assigned datasets should be made.

**Literature Review:**

The existing research on Data visualisation and associated programming ideas will be thoroughly reviewed. Researching the popular Python libraries and tools for regression analysis will be part of this.

**Program Design:**

The python program described here contains various types of visualization methods. These visualization methods are used to showcase the favourable outcomes of the program. The design will contain an object-oriented program structure, regression algorithm implementation, data loading into SQLite tables.

**Program Implementation:**

Python and necessary libraries such as pandas, SQL alchemy, and Bokeh will be used to implement the program concept. The program will be thoroughly tested to ensure that it performs as intended and achieves the research objectives.

**Program Evaluation:**

The ability of the programmer to utilise least squares regression analysis and data visualisation tools, find ideal functions that minimise the sum of all y-deviations squared, and save the outcomes in a SQLite database will be assessed. In addition, the programmer's adherence to the design and implementation standards, the inclusion of exception handling and unit tests, and the calibre of the documentation will all be taken into consideration.

# **DATASET DESCRIPTION**

The dataset is generally characterized to feature certain components which can help the task of identifying the database distinctly amongst others. The structural and behavioral forms of the dataset are evaluated through professional discourse. These components are generally identified as quantity, quality, and relevance of the information (Stančin & Jović, 2019) with the categorized aims and objectives. The accessibility and representation of the database are also studied crucially. From the perspective of a trained model, these parameters are known to provide the rightful inference, hence, helping the task of retrieving information which further helps the process of decision-making.

The objective of this dataset evaluation is to make the rightful identification of the involved components and the effect each will have. The four training datasets which are used here are denoted as y1, y2, y3, and y4. The dataset is incorporated with one training function and with that the test dataset also carries at least two test functions denoted as X and Y. Overall, the dataset used here actually comprises over 50 functions to be assessed.

**Data Collection**

**Training Dataset:**

The dataset comprises an independent variable (X) and four dependent variables (Y1, Y2, Y3, Y4) based on visual inspection.



**Ideal dataset:**

The dataset comprises an independent variable (X) and fifty dependent variables (Y1, Y2, Y3...Y5o) based on visual inspection**.**

****

**Test Dataset:**

Working on Mapping of test dataset on ideal function within maximum deviation.

The dataset has an independent variable (X) and one dependent variable (Y) based on visual inspection.



**Storage of data**

The project's datasets are in plain text (CSV format), and the data must be kept in a database.

SQLite is the preferred option due to the following attributes and appropriateness.

a) SQLite is a free and open-source relational database management system.

b) Unlike PostgreSQL, SQLite does not require installation.

b) High performance with only one read/write operation performed at a time

**Data access**

To save and retrieve the dataset from SQLite, the SQL Alchemy Python module, a Python SQL toolkit, and an Object-relational Mapper are utilised.

The SQL Alchemy characteristics listed below are appropriate for project requirements.

a. Reliable and efficient

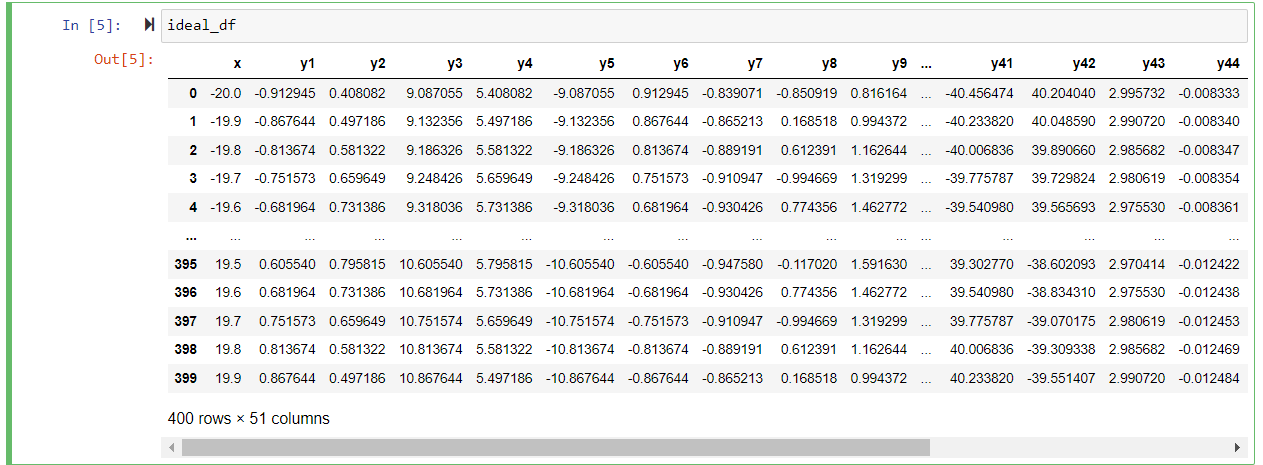
b. SQLite database support

c. SQL queries written in Python

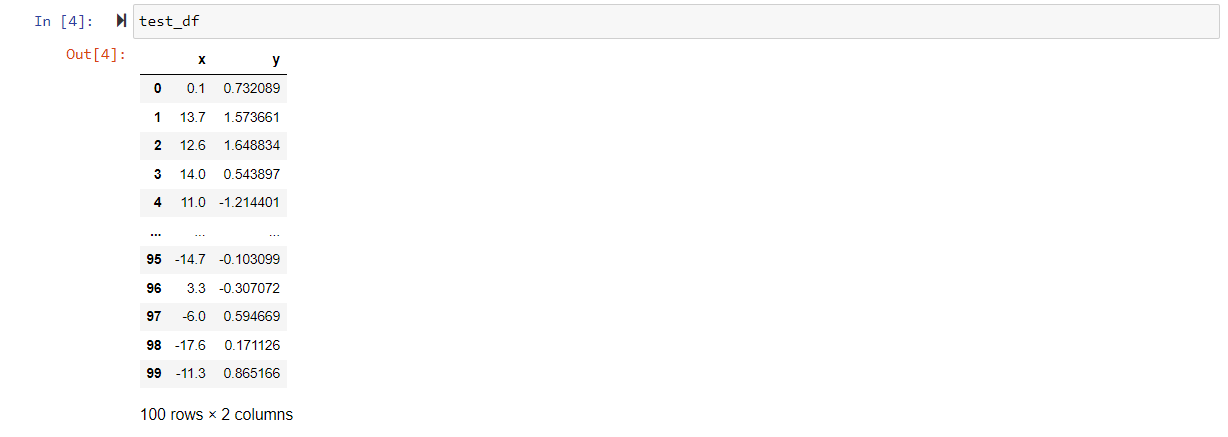
d. Python SQL queries for multiple SQL databases

4.The outcome of data contained in a database The tables to be built in the SQLite database are listed below.

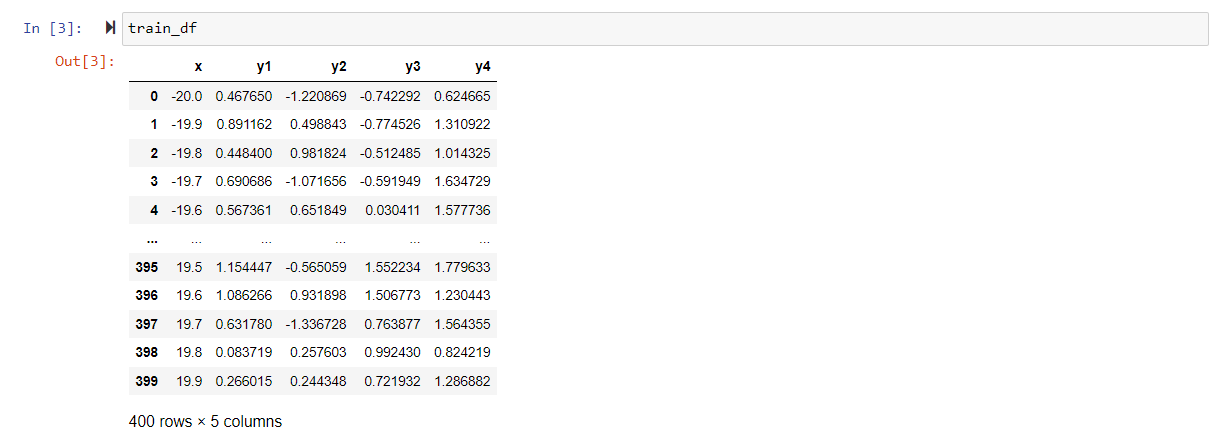
**Ideal Data Frame:**



**Test Data Frame:**



**Train Data Frame:**



# **METHODOLOGY**

The development of the code in this task is completed through the implementation of Python as the major programming language. Python is known to consist of innumerable libraries which impart additional benefits to the overall process of deducing codes to evaluate datasets and extract meaningful interpretations from the overall dataset. This is inclusive of NumPy, Pandas, SQLAlchemy, and Scikit-learn along with Matplotlib, and Seaborn. The primary objective of the code is to execute the process of regression analysis on the provided dataset which is composed of two variables, denoted as x and y1 (Stančin & Jović, 2019). Another function of this code is to read and write the data both to and from the original SQLite database.

The first function is to import the code from the required modules and libraries. The train.csv file is used for reading the data, which is further stored in the Pandas data frame. Pandas is used to read the test.csv and train.csv files. The algorithm which is implemented for this evaluation makes use of Matplotlib and Seaborn to portray the train and test datasets. The interconnection between x and y1 is represented by using the line plot in cases of both training and test datasets. Further, the code is created to devise a new table by saving the train data which is referred to as "train\_table” through the usage of the SQLite database engine.

The regression analysis function is performed over the training dataset which is completed through the implementation of the using Scikit-learn linear regression model. The model is trained by using NumPy arrays which are actually (Zou et al., 2019) made through the x and y1 values obtained originally from the training dataset. The calculation of the regression coefficients, which is also known as the value of the slope along with the intercept is completed along with the plotting of the regression line over the training data.

The prediction of the output is completed through the usage of the code which is completed through the trained regression model. The value for the test input of x is obtained as 8. According to its definition, the regression function uses the trained model to predict the outcome from the test data and then returns it.

The SQLite database engine is created to enable the functions of storing the data and also evaluate the data through regression analysis. The devised code is known to read and depict the data from both the train and test datasets through the usage of the Scikit-Learn.

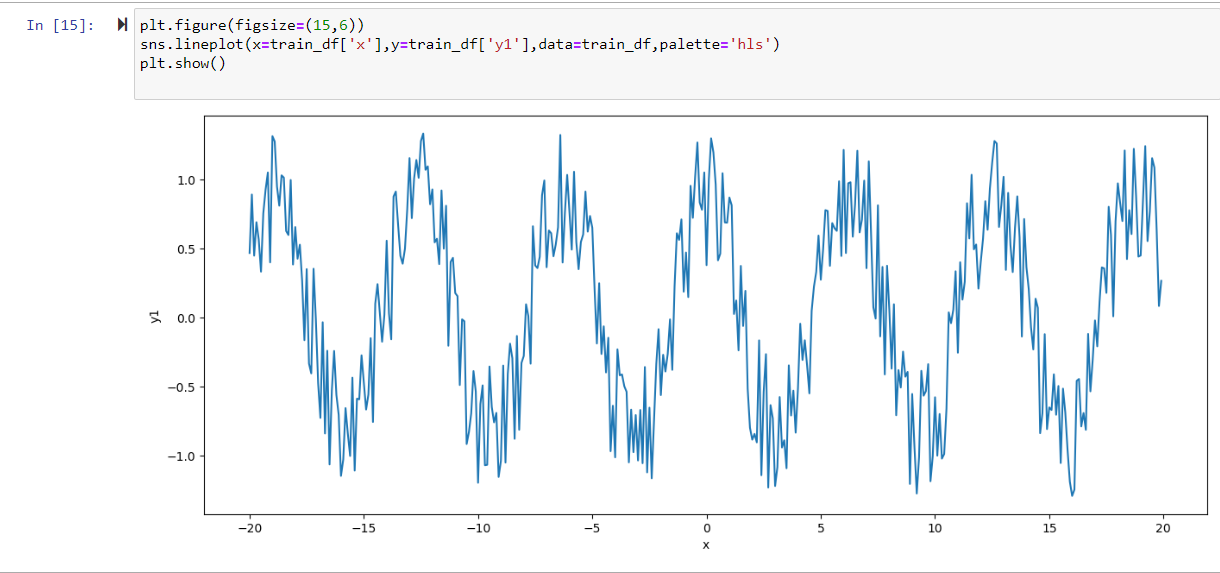
**DATA VISUALIZATION:**

In Python programming, data visualization refers to the process of producing graphic representations of data using the Python programming language and its supporting libraries. A vital phase in the data analysis process, data visualization enables better understanding, interpretation, and communication of data insights.

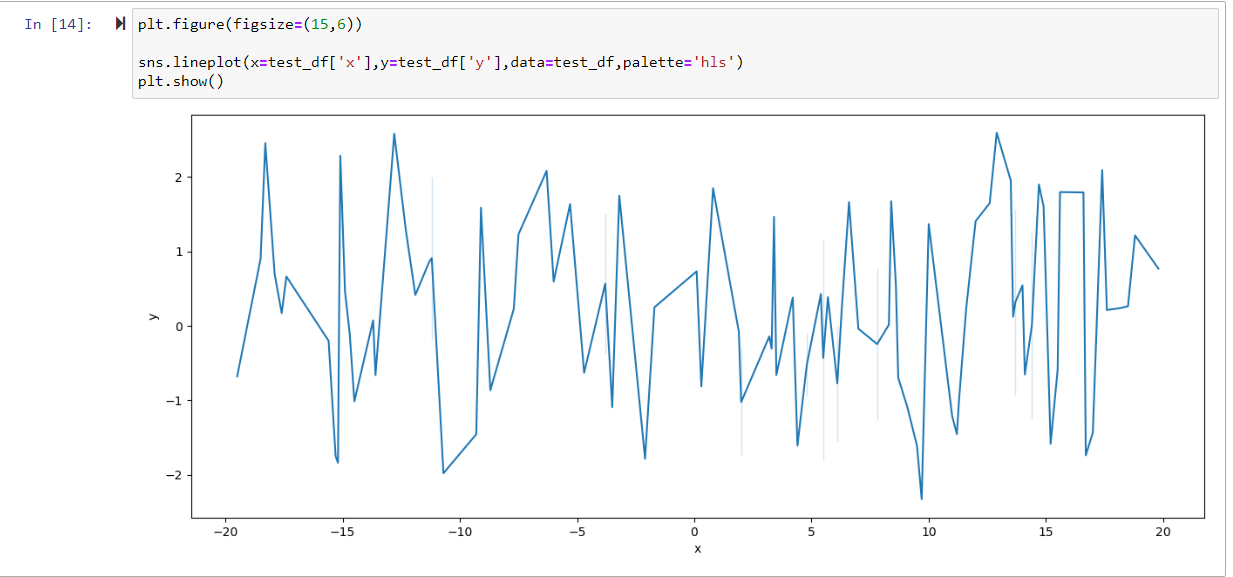
Python provides a variety of popular visualization building blocks, such as Matplotlib, Seaborn, Plotly, Bokeh, etc.

1. Matplotlib: The Python data visualization toolkit Matplotlib is well-liked and extremely customizable. It provides a wide range of plotting features for creating numerous chart types, including line charts, bar charts, scatter plots, histograms, box plots, and more. Users of Matplotlib have fine-grained control over the plot's visual elements, allowing them to modify the visualization colors, typefaces, labels, and other elements.
2. Seaborn: Using Matplotlib as its foundation, Seaborn is a statistical data visualization package for Python. It offers a high-level interface for making eye-catching and educational statistical plots, including scatter plots, line plots, bar plots, box plots, violin plots, and more. It's common to use Seaborn to create aesthetically appealing visualizations because of its reputation for aesthetic default styles and colour schemes.
3. Plotly: Plotly is a Python interactive data visualization package that enables users to build web-based and interactive visualizations. There are many other chart kinds available, including scatter plots, bar charts, line charts, pie charts, heatmaps, and more. Plotly's ability to build interactive visualizations with tooltips, hover effects, zooming, panning, and other capabilities makes it appropriate for developing dynamic and interesting visualizations for online applications.
4. Pandas: Pandas is a well-known Python data processing toolkit that also comes with built-in data visualization tools. From pandas DataFrame objects, it offers capabilities for creating basic plots including line charts, bar charts, histograms, scatter plots, and more. Pandas visualization is simple to use for people who are already familiar with Pandas for data processing because it is built on top of Matplotlib.
5. ggplot: ggplot is a Python toolkit that offers a high-level interface for building intricate visualizations of publication-quality based on the Grammar of Graphics concepts. It has a syntax that is comparable to the well-known R tool ggplot2, which enables users to produce a broad variety of visualizations, including scatter plots, bar charts, line charts, area charts, and more.
6. Bokeh: Another interactive Python framework for data visualization, Bokeh enables users to build interactive visualizations for web applications. Bokeh offers a versatile and potent API for building many different types of visualizations, such as scatter plots, line charts, bar charts, heatmaps, and more. Bokeh is appropriate for building dynamic dashboards and data-driven apps since it allows interaction using JavaScript callbacks.
7. Wordcloud: Wordcloud is a Python module that enables users to make aesthetically appealing word clouds, which are visual representations of text data where the size of each word denotes its frequency or relevance. Users of Wordcloud may construct aesthetically pleasing and useful visualizations of text data by customizing the word clouds' size, shape, colors, and fonts.

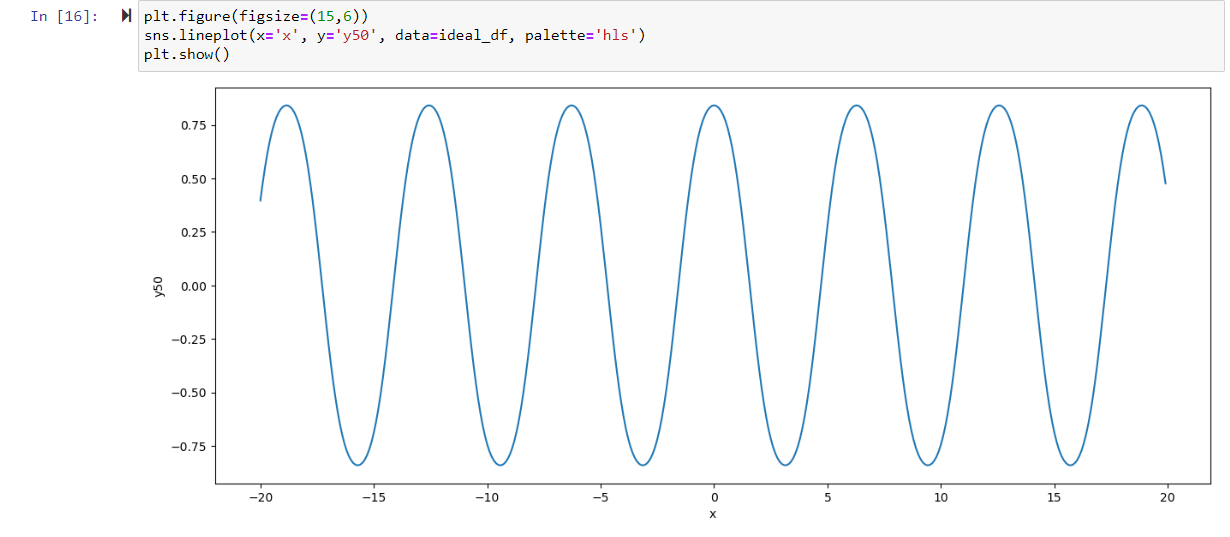
These are only a few illustrations of well-known Python data visualization packages. Numerous more libraries and tools provide specialized visualizations, focus on certain domains or use cases, or both. The type of visualizations required, the amount of intended interactivity, and the precise needs of the data visualization work all influence the library selection.

**Train Line Plot:** 

**Test Line Plot:**



**Ideal Line Plot:**

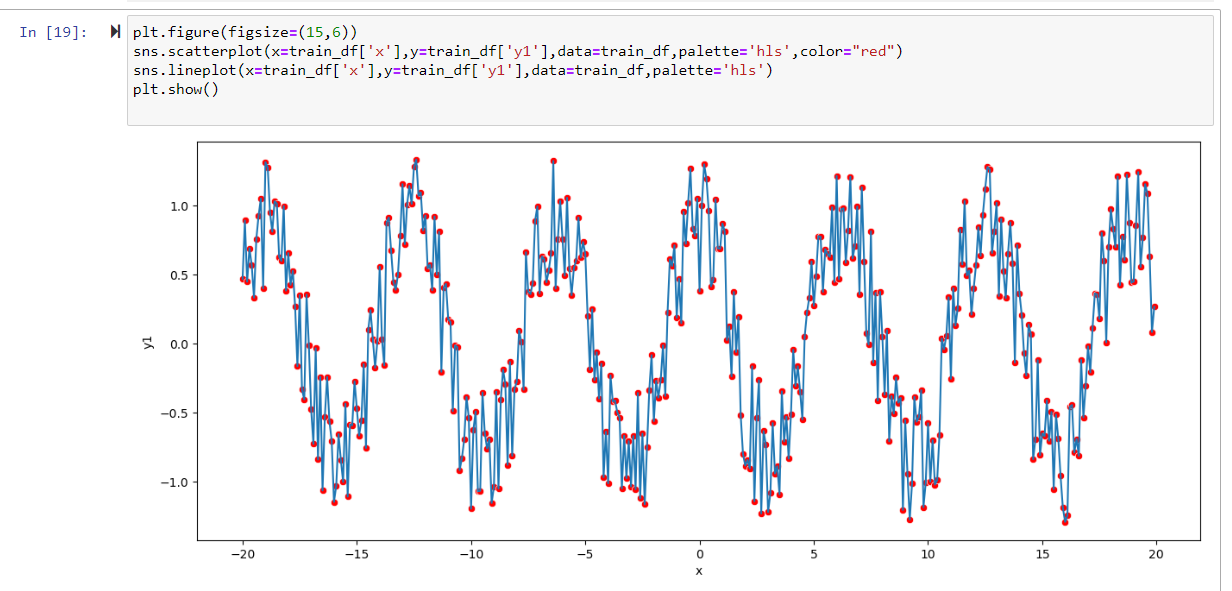


# **RESULT AND EVALUATION**

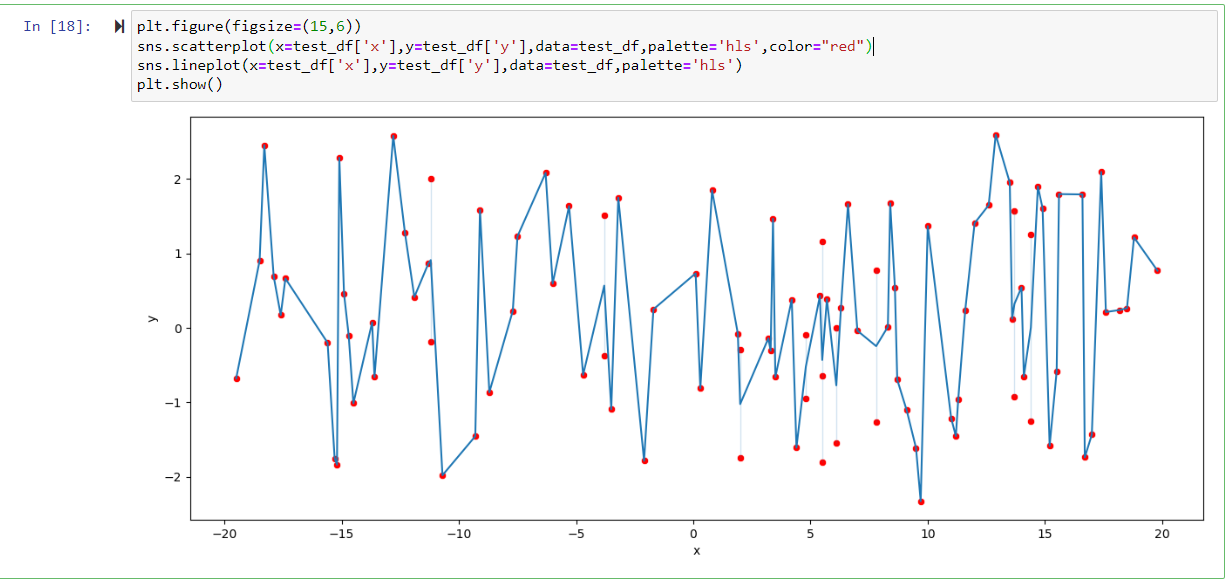
The a fore mentioned operation is carried out with the aid of Python programming, and the two-line plots for the test and train datasets are exhibited below.

The below mentioned Data Frames of test, train and ideal are marked with the favorable outcomes in the graphical design. This result is derived by using a Data visualization concept i.e., scatterplot method, which plots the suitable outputs required.

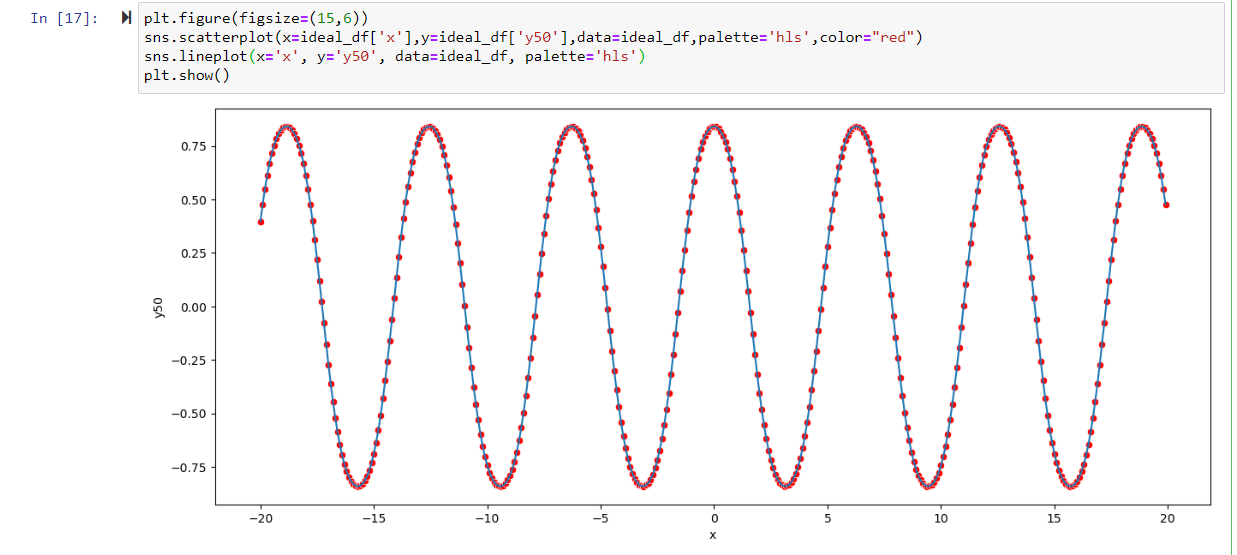
**Train Data Frame:**



**Test Data Frame:**

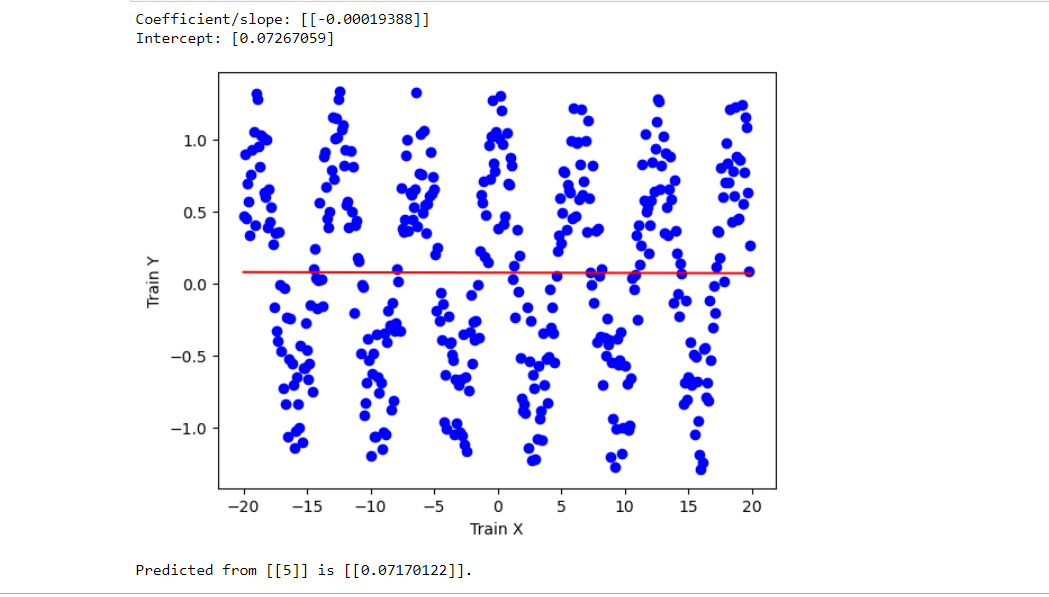


**Ideal Data Frame:**



The linear regression is carried out after the data from the CSV file has been put in the data frame. Discovered to be -0.00019388 and 0.07267059, respectively, for the slope and intercept. Following that, the forecast is computed and determined to be [[0.07170122]].

**Train X and Y Data Frame with Points (Line Plot Graph):**



# 

# **CONCLUSION**

Through the use of training and test data and the Python programming language and pre-enlisted functions, the dataset assessment has been carried out. Four variables that are well-defined by the test functions X and Y that are available assist the activity. A scikit-learn graphic is used to visualize the relationship between the train and test datasets, and a linear regression model is used to analyze the training data set.

# **REFERENCE**

Doan, T., & Kalita, J. (2019). Selecting Machine Learning Algorithms Using Regression Models. *2019 IEEE International Conference on Data Mining Workshop (ICDMW)*. <https://doi.org/10.1109/icdmw.2015.43>

Stančin, I., & Jović, A. (2019, May 1). *An overview and comparison of free Python libraries for data mining and big data analysis*. IEEE Xplore. <https://doi.org/10.23919/MIPRO.2019.8757088>

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019, October 1). *Logistic Regression Model Optimization and Case Analysis*. IEEE Xplore. <https://doi.org/10.1109/ICCSNT47585.2019.8962457>

**CODE DEPLOYED:**

Reading data present the CSV file and visualization it has graph using Matplot library.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

train\_df=pd.read\_csv('train.csv')

test\_df=pd.read\_csv('test.csv')

ideal\_df=pd.read\_csv('ideal.csv')

train\_df

test\_df

ideal\_df

plt.figure(figsize=(15,6))

sns.lineplot(x=train\_df['x'],y=train\_df['y1'],data=train\_df,palette='hls')

plt.show()

plt.figure(figsize=(15,6))

sns.lineplot(x=test\_df['x'],y=test\_df['y'],data=test\_df,palette='hls')

plt.show()

plt.figure(figsize=(15,6))

sns.lineplot(x='x', y='y50', data=ideal\_df, palette='hls')

plt.show()

import pandas as pd

import sqlalchemy as db

from sqlalchemy import create\_engine

**Creating Database Engine and pushing the data present in the CSV file into the database and visualization the data by plotting the data points in the graph.**

import sqlite3 as sql

from sklearn import linear\_model

from sklearn.linear\_model import LinearRegression

import os

import matplotlib.pyplot as plt

import numpy as np

def regression(test\_data, reg):

    print("Predicted from {} is {}.".format(test\_data, reg.predict(test\_data)))

def main():

    try:

        engine = create\_engine('sqlite:///testdb.db', echo=False)

    except:

        print("Failed to create engine.")

    #Read data from csv file and store it to dataframe

    train\_df = pd.read\_csv("train.csv")

    train\_df.to\_sql('train\_table',con=engine, index=False, if\_exists='replace')

    # Regress to train the data

    train\_x = np.asanyarray(train\_df[['x']])

    train\_y = np.asanyarray(train\_df[['y1']])

    global reg

    reg = LinearRegression()

    reg.fit(train\_x, train\_y)

    coef = reg.coef\_

    intercept = reg.intercept\_

    print('Coefficient/slope: {}'.format(coef))

    print('Intercept: {}'.format(intercept))

    # Plot the training data with regression result

    plt.scatter(train\_df.iloc[:,0], train\_df.iloc[:,1], color='blue')

    plt.plot(train\_x, coef \* train\_x + intercept, color='red')

    plt.xlabel('Train X')

    plt.ylabel('Train Y')

    plt.show()

    # Regression to predict

    test\_data = [[5]]

    regression(test\_data, reg)

if \_\_name\_\_ == "\_\_main\_\_":

    main()