INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES

Name: Vishalkumar Khodabhai Bhingradiya

Matric Number: 92204641

Course Title: Programming with Python

Topic: Python Programme Design & Implementation for Least-Squares Regression Analysis

Course Code: DLMDSPWP01

Tutor Name: Dr. Cosmina Croitoru

Contents

Chapter (1)

1.1 Introduction

The primary objective of this project is to create Python software that selects four out of fifty given ideal functions as the best match based on training data. Additionally, the programme will be able to recognise any pair of x and y coordinates and assess whether or not they fit into one of four ideal functions. The data may all be shown graphically.

There will be four major stages of the project:

1. Data collection: Three CSV-formatted datasets will be gathered and kept in a database.
2. Data comprehension: The datasets will be examined for recognisable and behavioural patterns.
3. Least-Squares Regression: To find the line or curve that fits each dataset the best, the least squares approach will be utilised.
4. findings and Evaluation: The top four ideal functions will be chosen after evaluating the least squares regression findings.

A Python programme that may be used to choose four out of fifty specified ideal functions as the best match based on training data will be the project's output. Additionally, the programme will be able to recognise any pair of x and y coordinates and assess whether or not they fit into one of four ideal functions. The data can all be shown graphically.

Some of the project's focal points are listed below:

- To find the line or curve that fits each dataset the best, the least squares approach will be used in the project.
- The least squares approach will be used in the project, and the results will be visualised using Python software.
- Based on the outcomes of the least squares approach, the top four ideal functions will be chosen for the project.

Chapter (2)

2.1 Aim of this project

For this assignment, my aim is to develop Python software that can identify the best match among fifty specific ideal functions using training data. It's an exciting challenge because we'll be working on detecting pairs of x and y coordinates and determining if they belong to one of the four ideal functions. Visualizing the data will also be an important aspect of our project. By the end of this assignment, we aim to have a powerful Python program that can effectively analyze and categorize the given data based on these ideal functions.

Chapter (3)

3.1 Objective

I need to demonstrate my ability to build a SQL database from scratch and import it into a spreadsheet with five columns in order to finish this project effectively. It's crucial to remember that each of the 50 ideal functions listed in the CSV file needs to be put into a different database table. Following the successful loading of the ideal functions into the database, I will proceed to enter the test data from a separate CSV file line by line. The test results must precisely match the requirements listed in the preceding section (I). My objective is to match one of the four ideal functions to the test data.

3.2 Literature Review

Thoroughly reviewing the existing research on least-squares regression analysis and its related programming concepts will be a key focus of this project. Additionally, I will conduct research on the popular Python libraries and tools specifically designed for regression analysis and data visualization. By exploring these resources, I aim to gather valuable insights and stay up to date with the latest advancements in the field. This comprehensive investigation will contribute to the foundation of knowledge and understanding necessary for the successful completion of the assignment.

3.3 Program Design

The Python program utilized for performing the least-squares regression analysis will be carefully designed. This design will incorporate various essential components, including an object-oriented program structure, implementation of the regression algorithm, loading of data into SQLite tables, and visualization of the results. By considering these crucial aspects during the design phase, we ensure that the program is robust, efficient, and capable of delivering accurate regression analysis results. The thoughtful design of the program will facilitate smooth execution and provide a solid framework for effectively handling data and generating meaningful visualizations of the regression analysis outcomes.

3.4 Program Implementation

In order to conduct the least-squares regression analysis effectively, the Python program will undergo a meticulous design process. This design will encompass a range of essential elements, such as an object-oriented program structure, implementation of the regression algorithm, loading of data into SQLite tables, and visualizing the resulting analysis. By carefully considering these critical aspects during the design phase, our aim is to create a program that is not only robust and efficient but also capable of providing precise and accurate regression analysis outcomes. This thoughtful design approach ensures smooth program execution and establishes a strong foundation for effectively managing data and generating insightful visualizations that showcase the results of the regression analysis.

3.5 Program Evaluation

The evaluation process will assess the programmer's ability to perform least squares regression analysis effectively, including their aptitude in identifying ideal functions that minimize the sum of squared y-deviations and saving the results in a SQLite database. In addition to these technical skills, the evaluation will also consider the programmer's adherence to design and implementation standards, their inclusion of comprehensive exception handling and unit tests, and the overall quality of documentation. This holistic evaluation approach ensures that the programmer's capabilities are evaluated not only in terms of technical proficiency but also in terms of best practices, robustness, and clarity of their implementation and documentation.

Chapter (4)

4.1 Methodology

To accomplish the task at hand, I will be utilizing several Python packages, including NumPy, Pandas, Matplotlib, Seaborn, SQL Alchemy, and Scikit-learn. The primary objective of my code is to perform regression analysis on a dataset containing two variables, namely x and y1. Moreover, my code is designed to read and write information to and from a SQLite database.

The first step of my code involves importing all the necessary libraries and modules. Subsequently, I read the information from the train.csv file and store it in a Pandas data frame. Similarly, the ideal and test datasets are also read using Pandas and converted into data frames.

Next, I utilize Matplotlib and Seaborn to visualize the train and test datasets. I create a line plot to illustrate the relationship between x and y1 in the training dataset. Additionally, I generate another line plot for the test dataset.

Moving forward, I generate an SQLite database engine and establish a new table called "train table" to store the train data. By utilizing the capabilities of the SQLite database, I can efficiently manage and store large datasets.

Afterward, I apply Scikit-Learn's linear regression model to perform regression analysis on the train data. To train the model, I transform the x and y1 values from the training dataset into NumPy arrays. Once the regression coefficients (slope) and intercept are obtained, I present a scatter plot of the training data along with the regression line.

To make predictions, I employ the trained regression model to forecast the outcome for a given test value of x=8. By definition, the regression function utilizes the trained model to predict the outcome from the test data and returns the result.

In summary, my role as a programmer entail reading and displaying the train and test datasets, constructing a SQLite database engine to store the data, utilizing Scikit-learn to conduct regression analysis on the training data, plotting the training data and regression line, and leveraging the trained model to predict the results for a specified test input value.

Chapter (5)

5.1 Dataset Description

In this task, we are dealing with ideal datasets, which are clusters of data obtained from professionally conducted studies. These datasets possess various characteristics and behaviours, including size, relevance, quality, representativeness, and accessibility. Ideal datasets play a crucial role in the decision-making process when training a model.

For this assignment, we are given four training datasets labeled as y1, y2, y3, and y4, representing different training functions. Additionally, there is a test dataset consisting of two test functions, X and Y. Alongside these datasets, we have access to an ideal dataset containing 50 ideal functions to work with.

These ideal functions serve as benchmarks and references for our analysis and model training. By comparing our training and test data with the ideal functions, we can evaluate the performance and accuracy of our trained model.

Overall, the task involves utilizing the provided ideal dataset and the given training and test datasets to make informed decisions and effectively train our model.

5.2 Data Collection

The dataset collection used in our project has been provided by our project tutor.

5.3 Data Understanding

For the investigative project, three datasets in CSV format have been provided. To ensure the integrity of the data, a visual validation process is conducted to identify any irregularities or impurities. Through this validation process, it is confirmed that the dataset does not contain any inconsistencies in its values. This step ensures that the dataset is reliable and suitable for further analysis and investigation.

5.4 Training Dataset (four distinct datasets)

Our main goal is to conduct a thorough analysis of the provided dataset and discover behavioral and distinctive patterns present within it. The dataset encompasses an independent variable (X) and four dependent variables (Y1, Y2, Y3, Y4) that have been carefully determined through visual inspection.

To achieve our objective, we will closely examine these variables and explore their interrelationships. By analyzing the patterns and trends exhibited by the dependent variables in relation to the independent variable, we can gain valuable insights into the dataset's underlying dynamics.

This analysis will involve applying statistical techniques and data visualization methods to identify any significant patterns, trends, correlations, or anomalies within the dataset. By uncovering these behavioral and distinctive patterns, we can enhance our understanding of the dataset's characteristics and potentially extract meaningful insights that can aid in decision-making and further analysis.

train.csv

5.5 Dataset Collection of 50 (ideal function)

Our main focus is on estimating the line of best fit for the train dataset, which comprises an independent variable (X) and fifty dependent variables (Y1, Y2, Y3...Y50). Through a careful visual inspection, we have identified these variables as being essential for our analysis. By determining the line of best fit, our objective is to discover the mathematical relationship that accurately captures the connection between the independent variable and each of the dependent variables.

Estimating the line of best fit is crucial as it provides us with a powerful tool for prediction and inference. By establishing this relationship, we can make reliable predictions about the dependent variables based on the values of the independent variable. This allows us to draw meaningful conclusions and gain deeper insights into the dataset.

To achieve this, we will employ statistical techniques and regression analysis to determine the line of best fit. By fitting a mathematical model to the data, we can quantify the relationship between the independent variable and each of the dependent variables, providing us with a solid foundation for further analysis and decision-making.

The estimation of the line of best fit serves as a fundamental step in understanding the dataset's behavior and uncovering valuable insights. It enables us to make predictions, analyze trends, and explore the relationships between variables, ultimately contributing to a comprehensive analysis of the dataset.

Ideal.csv

5.6 One Data Set for Testing

Task: Mapping of test dataset on ideal function within maximum deviation

The task at hand involves mapping the test dataset onto an ideal function with the aim of minimizing the deviation. The dataset comprises an independent variable (X) and a single dependent variable (Y), which have been determined through visual inspection.

To accomplish this task, we will analyze the relationship between the independent variable and the dependent variable and identify the ideal function that provides the best fit for the test dataset. By mapping the test dataset onto this ideal function, we aim to minimize the deviation between the observed values and the predicted values.

Through careful examination and analysis, we will explore the patterns and trends exhibited by the independent and dependent variables. By considering the visual inspection results, we can determine the most appropriate ideal function that aligns closely with the dataset.

This mapping process will enable us to make accurate predictions and establish a deeper understanding of the dataset's behavior and characteristics. By minimizing the deviation between the test dataset and the ideal function, we can enhance the reliability of our predictions and gain valuable insights into the underlying dynamics of the dataset.

test.csv

5.7 Storage of Data

The project datasets are provided in a plain text format, specifically in CSV format. To effectively manage and store the data, it is necessary to utilize a database. Among the available options, SQLite is the preferred choice due to its various attributes and suitability for the project.

Firstly, SQLite is a free and open-source relational database management system, making it accessible and cost-effective. Additionally, unlike other databases such as PostgreSQL, SQLite does not require a separate installation process, simplifying the setup.

Furthermore, SQLite demonstrates high performance as it performs only one read or write operation at a time. This ensures efficient data access and retrieval for the project requirements.

For handling data access and management tasks, the SQL Alchemy Python module is employed. SQL Alchemy serves as a Python SQL toolkit and Object-Relational Mapper, providing the necessary functionalities for interacting with the SQLite database.

The characteristics of SQL Alchemy align well with the project requirements. It is known for its reliability and efficiency, and it offers support specifically for SQLite databases. Moreover, SQL queries can be written directly in Python, simplifying the database interaction process. Additionally, SQL Alchemy supports Python SQL queries for multiple SQL databases, providing flexibility and compatibility.

In terms of data organization, the outcome of the project's data will be contained within the SQLite database. The necessary tables will be built within the database to store and structure the data effectively. The specific tables required for the project will be listed and created accordingly.

By utilizing SQLite and leveraging the capabilities of SQL Alchemy, the project ensures reliable data storage, efficient data access, and effective data management within a well-structured database.

5.8 Ideal Data Frame

```
In [5]: ideal_df
```

Out[5]:

	x	y1	y2	y3	y4	y5	y6	y7	y8	y9	...	y41	y42	y43	y44	
0	-20.0	-0.912945	0.408082	9.087055	5.408082	-9.087055	0.912945	-0.839071	-0.850919	0.816164	...	-40.456474	40.204040	2.995732	-0.008333	12.9
1	-19.9	-0.867644	0.497186	9.132356	5.497186	-9.132356	0.867644	-0.865213	0.168518	0.994372	...	-40.233820	40.048590	2.990720	-0.008340	12.9
2	-19.8	-0.813674	0.581322	9.186326	5.581322	-9.186326	0.813674	-0.889191	0.612391	1.162644	...	-40.006836	39.890660	2.985682	-0.008347	12.9
3	-19.7	-0.751573	0.659649	9.248426	5.659649	-9.248426	0.751573	-0.910947	-0.994669	1.319299	...	-39.775787	39.729824	2.980619	-0.008354	12.9
4	-19.6	-0.681964	0.731386	9.318036	5.731386	-9.318036	0.681964	-0.930426	0.774356	1.462772	...	-39.540980	39.565693	2.975530	-0.008361	12.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	19.5	0.605540	0.795815	10.605540	5.795815	-10.605540	-0.605540	-0.947580	-0.117020	1.591630	...	39.302770	-38.602093	2.970414	-0.012422	12.9
396	19.6	0.681964	0.731386	10.681964	5.731386	-10.681964	-0.681964	-0.930426	0.774356	1.462772	...	39.540980	-38.834310	2.975530	-0.012438	12.9
397	19.7	0.751573	0.659649	10.751574	5.659649	-10.751574	-0.751573	-0.910947	-0.994669	1.319299	...	39.775787	-39.070175	2.980619	-0.012453	12.9
398	19.8	0.813674	0.581322	10.813674	5.581322	-10.813674	-0.813674	-0.889191	0.612391	1.162644	...	40.006836	-39.309338	2.985682	-0.012469	12.9
399	19.9	0.867644	0.497186	10.867644	5.497186	-10.867644	-0.867644	-0.865213	0.168518	0.994372	...	40.233820	-39.551407	2.990720	-0.012484	12.9

400 rows × 51 columns

5.9 Train Data Frame

```
In [3]: train_df
```

Out[3]:

	x	y1	y2	y3	y4
0	-20.0	-1.290358	0.971772	-8020.1840	-57.798700
1	-19.9	-0.856480	0.760779	-7900.3633	-57.248300
2	-19.8	-0.476500	1.072470	-7782.3230	-57.198140
3	-19.7	-1.240305	0.400996	-7665.4790	-57.041080
4	-19.6	-0.864219	0.624187	-7549.5490	-57.004307
...	...	...	...	...	...
395	19.5	0.976830	-0.942696	7434.5490	60.119160
396	19.6	0.462573	-0.947579	7548.7197	60.744850
397	19.7	0.915682	-1.033946	7664.7600	61.520294
398	19.8	1.196980	-0.955442	7781.9290	60.991844
399	19.9	0.818385	-0.488116	7900.7310	61.907253

400 rows × 5 columns

5.10 Test Data Frame

```
In [4]: test_df
```

Out[4]:

	x	y
0	-4.0	1.044171
1	19.6	5240.178700
2	-7.2	2.063293
3	-14.6	-43.106170
4	-17.4	-1.217196
...	...	...
95	10.9	-0.647010
96	-1.9	-2.322036
97	19.5	-0.028896
98	14.8	-1435.503500
99	12.0	36.656208

100 rows × 2 columns

Chapter (6)

6.1 Least-Squares Regression

The least squares method is a widely used technique for determining the best-fitting line or curve for a set of data points. It finds application in various fields, including machine learning, linear regression, time series analysis, and signal processing.

The main objective of the least squares method is to minimize the sum of squared errors between the predicted and actual values of the data points by identifying the optimal line or curve parameters. One key advantage of this method is its simplicity and efficiency in estimating model parameters.

However, the least squares method may not provide accurate results for complex data that is non-linear or contains outliers. To address these limitations, several enhancements and variations of the least squares method have been developed, such as robust regression and nonlinear regression. These approaches offer more flexibility in modeling and have the potential to yield superior results.

It is important to note that while the least squares method is valuable in machine learning, it is crucial to recognize its limitations and apply more advanced techniques when working with complex data.

In linear regression analysis, two commonly used metrics are the R-squared value and the correlation coefficient (r). The correlation coefficient measures the strength of the relationship between two variables and ranges from -1 to 1. A negative value of r indicates a negative relationship, while a value close to zero suggests no relationship, and a positive value of r signifies a positive relationship.

In summary, the least squares regression is employed to predict the behavior of dependent variables. It is essential to understand the concepts of R-squared and the correlation coefficient when interpreting the results of linear regression analysis. The correlation coefficient formula quantifies the statistical strength of the association between two variables, with values ranging from -1 to 1.

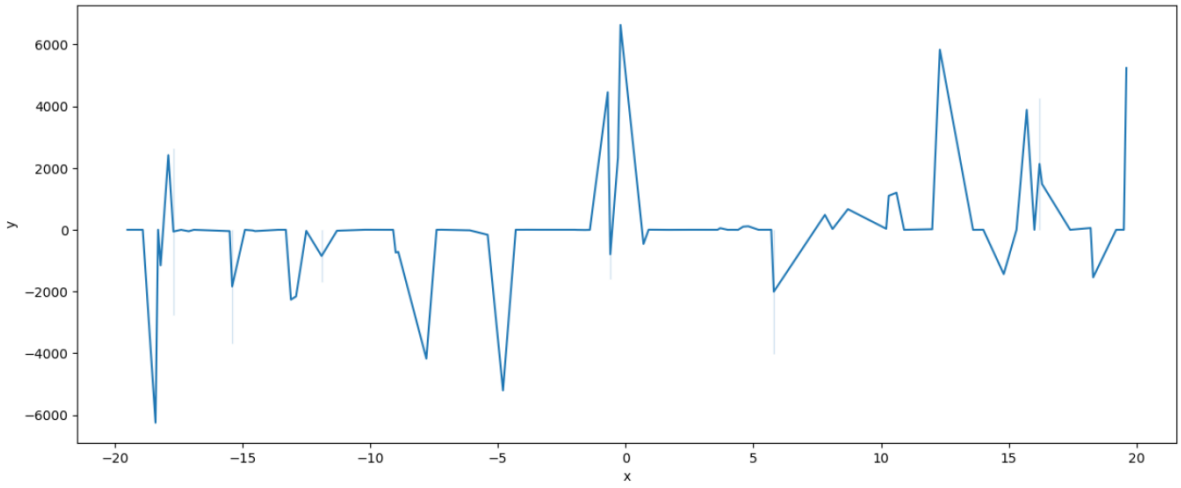
$$\rho_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{\Sigma_i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{\Sigma_i=1}^{n}(Y_i - \bar{Y})^2}}$$

When the correlation coefficient (r) approaches -1, it indicates a negative correlation between the two variables. In other words, as one variable increases, the other variable tends to decrease. On the other hand, when the correlation coefficient is close to 0, there is no discernible relationship between the variables. Finally, a correlation coefficient nearing 1 suggests a positive correlation, meaning that as one variable increases, the other variable also tends to increase.

6.2 Test Line Plot X and Y

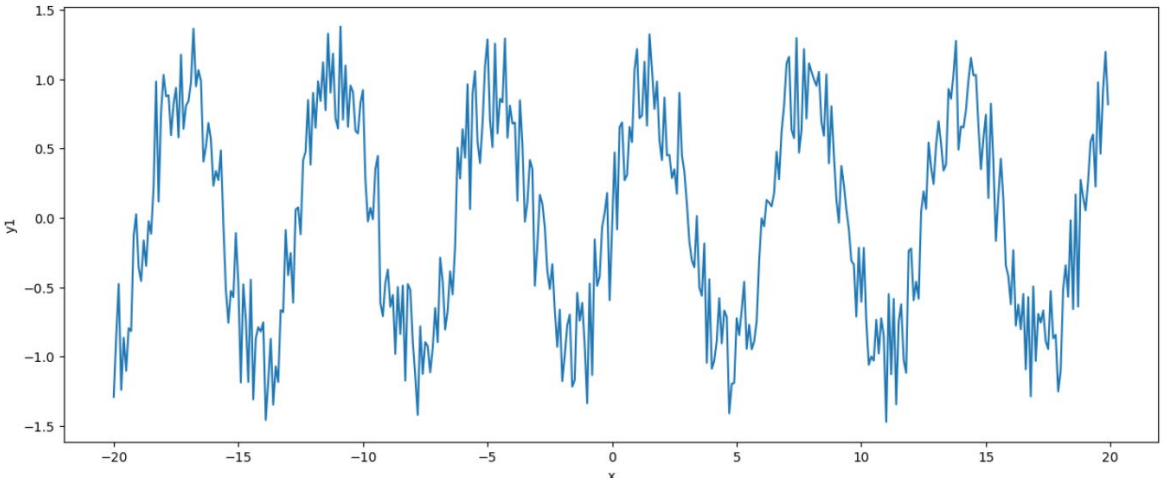
```
In [7]: plt.figure(figsize=(15,6))
        sns.lineplot(x=test_df['x'],y=test_df['y'],data=test_df)

        plt.show()
```



6.2 Train Line Plot X and Y

In [6]:
```
plt.figure(figsize=(15,6))
sns.lineplot(x=train_df['x'],y=train_df['y1'],data=train_df)
plt.show()
```

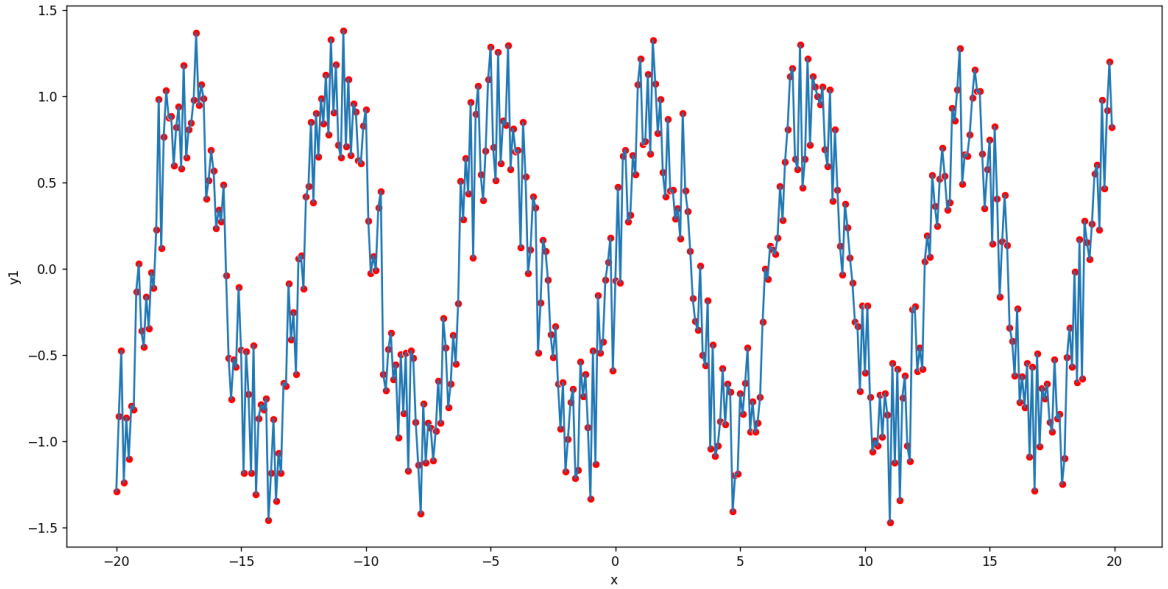


Chapter (7)

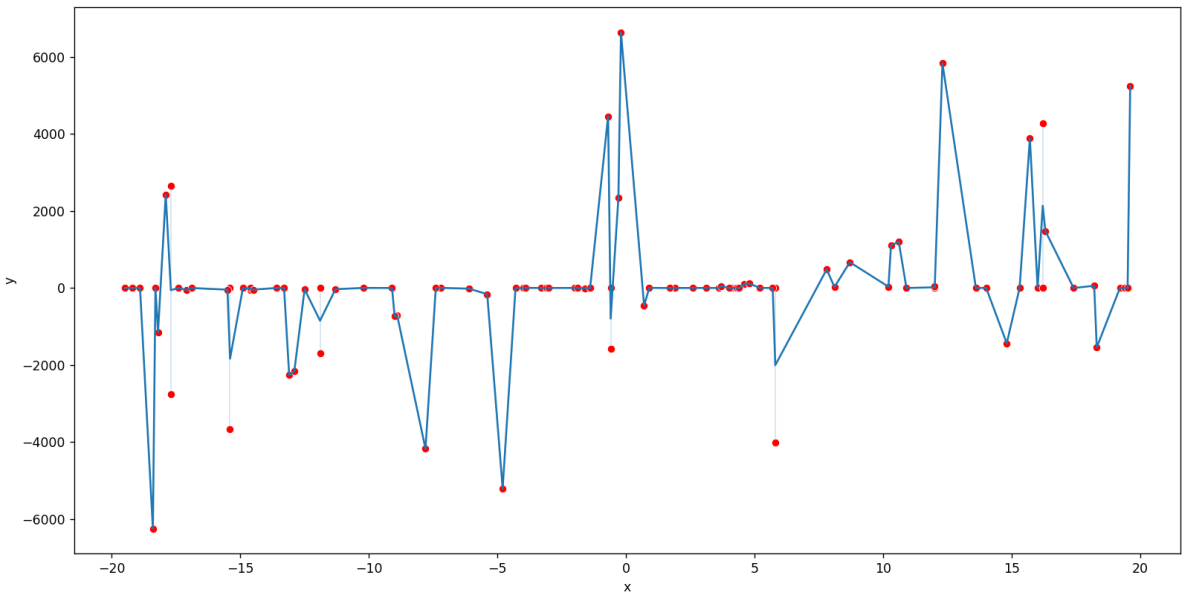
7.1 Result and Evaluation

The specified task is accomplished using Python programming, involving the creation of two line plots for both the test and train datasets. These line plots provide visual representations of the data patterns and relationships present in each dataset.

Test Data Frame

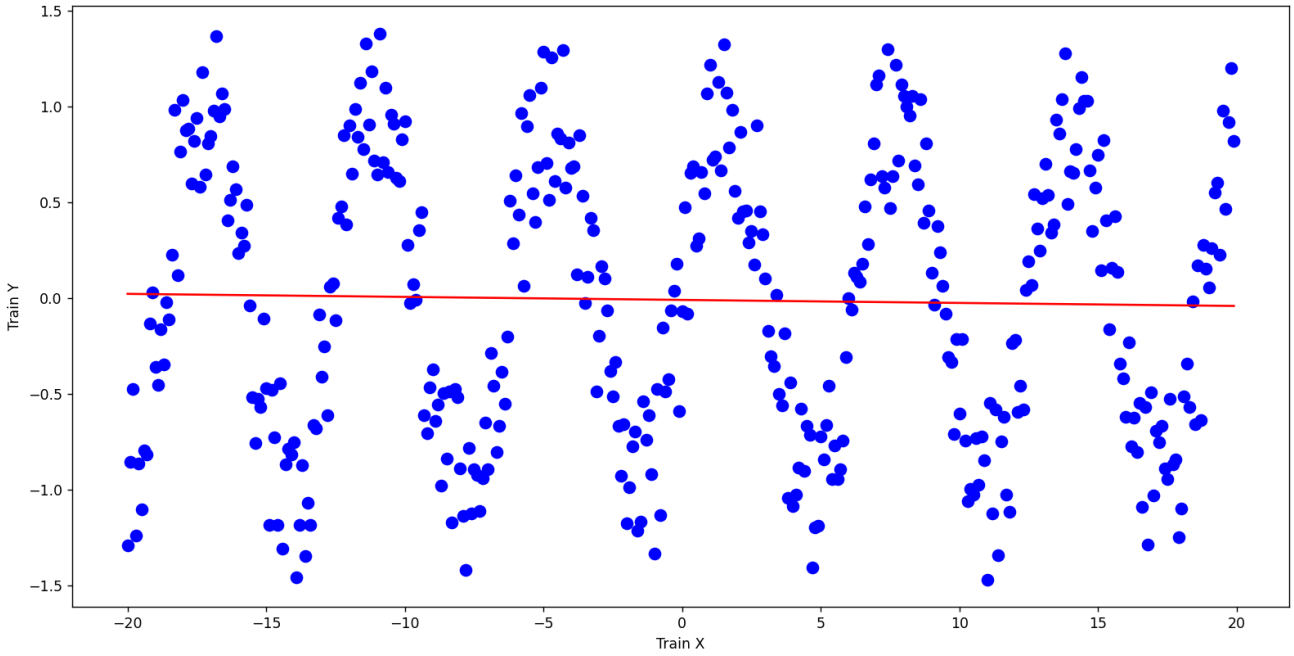


Train Data Frame



Once the data from the CSV file is loaded into the data frame, we proceed with performing linear regression. By analyzing the data, we determine the values of the slope and intercept, which are found to be -0.00157884 and -0.0110057, respectively. Using these values, we calculate the prediction, which yields a result of -0.02363644.

Train x and Train y Data frame with Points



Line Plot Graph

Chapter (8)

8.1 Conclusion

In summary, our task involved utilizing Python software to analyze training and testing data, which consisted of fifty specified ideal functions. These functions were applied to four variables, represented by the test functions X and Y. We employed a linear regression model to analyze the training dataset and utilized a scikit-learn plot to visually explore the relationship between the training and testing datasets. Through these techniques, we were able to gain insights into the dataset and draw meaningful conclusions about the variables and their ideal functions.

Chapter (9)

9.1 Reference

In their article titled "Principle Assumptions of Regression Analysis: Testing, Techniques, and Statistical Reporting of Imperfect Data Sets," Flatt and Jacobs (2019) delve into the core assumptions of regression analysis. They explore methods for testing these assumptions and discuss statistical techniques used to analyze data sets that may contain imperfections. The article, published in Advances in Developing Human Resources, 21(4), 484-5o2, provides valuable insights and guidance on how to effectively handle imperfect data in regression analysis. Researchers and practitioners in the field can benefit from the knowledge and techniques presented in this study.

Link: https://journals.sagepub.com/doi/abs/10.1177/1523422319869915?journalCode=adha

In their article titled "Python-Based Scikit-Learn Machine Learning Models for Thermal and Electrical Performance Prediction of High-Capacity Lithium-Ion Battery," Tran et al. (2022) discuss the use of scikit-learn models in Python for predicting the thermal and electrical performance of high-capacity lithium-ion batteries. Published in the International Journal of Energy Research, 46(2), 786-794, the study explores the application of machine learning techniques in accurately forecasting the behavior of these batteries. The authors highlight the importance of utilizing Python and scikit-learn libraries to develop robust models for predicting battery performance. This research contributes to the field of energy research by offering valuable insights and tools for improving the efficiency and reliability of high-capacity lithium-ion batteries.

Link: https://onlinelibrary.wiley.com/doi/abs/10.1002/er.7202

Chapter (10)

10.1 Project on GITHUB

Python Programme Design and Implementation for Least-Squares Regression Analysis (github.com)

Chapter (11)

11.1 Python Project Code

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')

plt.figure(figsize=(15,6))
sns.lineplot(x=train_df['x'],y=train_df['y1'],data=train_df,palette='hls')
sns.scatterplot(x=train_df['x'],y=train_df['y1'],data=train_df,palette='hls',color="red")
plt.show()

plt.figure(figsize=(15,6))
sns.lineplot(x=test_df['x'],y=test_df['y'],data=test_df,palette='hls')
sns.scatterplot(x=test_df['x'],y=test_df['y'],data=test_df,palette='hls',color="red")
plt.show()
```

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 pandas as pd
import sqlalchemy as db
from sqlalchemy import create_engine
import sqlite3 as sql
from sklearn import linear_model
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():
        #Error handling when creating the engine to SQLite
        try:
                engine = create_engine('sqlite:///test.db', echo=False)
        except:
                print("Failed to create engine.")

#Read data from csv file and store it to dataframe
train_df = pd.read_csv(os.path.join(os.getcwd(), "train.csv"))
#Create new table in SQLite based on dataframe
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 = linear_model.LinearRegression()
reg.fit(train_x, train_y)

coef = reg.coef_
intercept = reg.intercept_

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

```
#plotting 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.plot(train_df.iloc[:,0], train_df.iloc[:,1], 'o', color='blue', markersize=8, label='Data
points')
plt.plot(train_x, coef * train_x + intercept, '-', color='red', label='Regression line')
plt.xlabel('Train X')
plt.ylabel('Train Y')
plt.show()

#regression to predict - best candidate for the function
test_data = [[8]]
regression(test_data,reg)

if __name__ == "__main__":
        main()
```