DATA SCIENCE IN MODERN RESEARCH: A CASE STUDY - MAPPING EMPIRICAL DATA TO IDEAL MATHEMATICAL FUNCTIONS

A REPORT BY

JELSON LINO

MSc in Nuclear Physics

MSc Student in Data Science

BEng (Hons) in Nuclear & Mechanical Engineering

ABSTRACT

Data analysis is an important part of modern research. Researchers often analyze various pages of data to extract insights. The development of powerful computers and storage capacity resulted in the generation and storage of large amounts of data. However, analyzing a large amount of data is challenging if the right tools and techniques are not employed. Data science techniques can be employed to study the large amount of data generated in research and other real-world scenarios. In this study, different types of data analyses are discussed and a case study illustrating the importance of data science techniques in research is presented. Python is used to map an empirical dataset containing 400 entries and 5 columns to ideal mathematical functions and then tested using 200 entries of x-y pair of empirical data points. Using the empirical dataset, the program successfully selected four out of fifty ideal mathematical functions and mapped the 200 x-y pairs of empirical data points to the selected four ideal functions. It has been demonstrated through this case study that data science can help researchers save a significant amount of time that is usually employed studying data using conventional techniques and technology.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

SECTION 1 - INTRODUCTION

Data analysis is an important activity in research. Researchers often analyze various pages of data to extract insights. The development of powerful computers and storage capacity resulted in the generation and storage of large amounts of data. The massive amount of data available can significantly improve the quality and quantity of the information extracted from the data. However, analyzing a large amount of data is challenging if the right tools and techniques are not employed (Muni & Manjula, 2014, p.7172-7178). The development of digital technology resulted in the development of tools and techniques that can be employed to study massive amounts of data in a short time. These techniques can be employed to study data generated in various real-world scenarios (Sengupta, 2013, p.1206–1211), such as research, healthcare (Ren et al., 2010, p.59-65), and the manufacturing industry (Bollen et al., 2010, p.1-10; Kuehn, 2013, p.787).

Data science techniques are important in modern research because with them the study of the large amount of data generated is more accurate and significantly simpler (IBM, 2013). The techniques allow for a straightforward interpretation of the entire data, allowing researchers to grasp every possible story or detail that the data carries (Gubbi et al., 2013, p. 1645-1660). Hence, researchers who are knowledgeable in data science techniques can be more productive and efficient in their respective fields.

In this paper, the different types of data analyses are discussed and a case study illustrating the importance of data science techniques in research is presented.

The rest of the paper is organized as the following: Section 2 discusses the types of data analyses with a focus on the type of analysis conducted in this study. Section 3 presents the case study, while Section 4 briefly discusses the tools used in the case study. The results are presented in Section 5 and discussed in Section 6. Lastly, Section 7 presents the conclusions of the study.

SECTION 2 - TYPES OF DATA ANALYSES

Data analysis involves the analysis of information that can be presented in a single or different format. The massive amounts of data available are generated and stored in different formats and various sources. Data science techniques allow the cleaning and transformation of the data into a format that is appropriate for the particular study. After the data is cleaned and transformed into the right format, it can be used to uncover important information. When a large amount of data is cleaned, transformed, and studied as a whole, even minor details and patterns can be discovered. Oftentimes these minor details and patterns are ignored if the data is not studied collectively. Research involves the study of trends and patterns, and making and proving/disproving hypotheses. These are all supported by data. The use of data science techniques significantly improves these processes.

Data analysis can be classified into two main types, qualitative and quantitative analysis. Quantitative data analysis involves numerical data, while qualitative analysis studies qualitative data, such as symbols and summaries. In this paper, we will focus on quantitative analysis.

Quantitative Analysis

Any sort of data analysis performed on numbers falls under the category of quantitative data analysis. There are various techniques in quantitative data analysis. They can be employed in descriptive statistics or inferential analysis.

Descriptive statistics draws information about the data: it checks for patterns and details about the data. It is in descriptive statistics analysis that specific aspects of the data are seen. They include the mean, median, mode, and standard deviation.

The inferential analysis involves the different methods used in the prediction of future behaviors of the data. These techniques are used to understand relationships between different features of the

data. Techniques used in the inferential analysis include correlation and regression. Correlation expresses the relationship between two numerical features. Regression finds the mathematical relationship between a set of features.

In this study, these methods were used to map datasets to ideal mathematical functions using python.

SECTION 3 - CASE-STUDY PROBLEM STATEMENT

In this study, four training datasets, one test dataset, and fifty ideal mathematical functions were used. All data consisted of x-y-pairs of values.

The study aims to write a python program that uses the empirical data, referred to as training data in the analysis, to choose four ideal functions that are the best fit out of the fifty ideal mathematical functions provided. Afterward, the program must use the test data provided to determine for every x-y pair of values whether or not they can be assigned to one of the four chosen ideal functions. In addition, the program also needs to execute the mapping and save it together with the deviation at hand.

The method for choosing the ideal functions for the training function is how they minimize the sum of all y-deviations squared (Least-Square). The method for mapping the individual test case to the four ideal functions is that the existing maximum deviation of the calculated regression does not exceed the largest deviation between the training dataset and the ideal function chosen for it by more than factor sqrt(2).

SECTION 4 - METHODS

Python was used as a tool to conduct the analysis. The main python package used was pandas. Other packages used include pathlib, NumPy, seaborn, matplotlib, plotly express, scipy, math, and sci-kit learn.

SQLite was used to create a database, save the comma-separated values (CSV) files, and save the results obtained from the analysis into the database. Once the data was loaded from the database, pandas was used to explore the data and perform calculation columnwise. Sklearn was used to build regression models, such as linear and polynomial regression models, that were used in the study of the training and ideal datasets. Visualization packages such as matplotlib, seaborn, and plotly express, were used to visualize the data throughout the analysis. Numpy, scipy, and math were used to perform calculations during the analysis.

SECTION 5 - RESULTS

The data was initially kept in CSV files. First, a database was created and three tables were created in the database as shown in Figure 1. Then, the data was loaded into the database as shown in Figure 2. Once the data was saved in the database, the data was subsequently loaded from the database into data frames to begin the analysis as shown in Figure 3.

```
In [2]: ## CREATING A SQLITE DATABASE
        Path('PWPA.db').touch()

In [3]: ## CREATE DATABASE CONNECTION AND CURSOR
        conn = sqlite3.connect('PWPA.db')
        c = conn.cursor()

In [4]: ## create three SQLITE TABLES

        c.execute('''CREATE TABLE train (x, y1, y2, y3, y4)''')

        c.execute('''CREATE TABLE test (x, y)''')

        c.execute('''CREATE TABLE ideal (x, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15, y16, y17, y18, y19, y20,
                                         y21, y22, y23, y24, y25, y26, y27, y28, y29, y30, y31, y32, y33, y34, y35, y36, y37, y38, y39,
                                         y40, y41, y42, y43, y44, y45, y46, y47, y48, y49, y50)''')
Out[4]: <sqlite3.Cursor at 0x1be3ab7b5e0>
```

Figure 1: Instructing the program to create and connect to a database

```
In [5]: ## LOAD THE CSV FILES INTO THE SQLITE DATABASE

        # load the data into Pandas DataFrames
        user_1 = pd.read_csv("train.csv")
        user_2 = pd.read_csv("test.csv")
        user_3 = pd.read_csv("ideal.csv")

        # write the data into sqlite tables
        user_1.to_sql('train', conn, if_exists='append', index = False)
        user_2.to_sql('test', conn, if_exists='append', index = False)
        user_3.to_sql('ideal', conn, if_exists='append', index = False)
Out[5]: 400
```

Figure 2: Saving the data into the database created

```
In [6]: # LOADING THE DATA FROM THE DATABASE INTO THREE DATAFRAME

        con = sqlite3.connect("PWPA.db")
        df_train = pd.read_sql_query("SELECT * from train", conn)
        df_test = pd.read_sql_query("SELECT * from test", conn)
        df_ideal = pd.read_sql_query("SELECT * from ideal", conn)

        con.close()
```

Figure 3: Loading the data from the database into data frames.

The three datasets were already cleaned and structured before the analysis. Thus, no time was spent cleaning or structuring the data during the analysis. Figures 4(a), (b), and (c) show the first five rows and all columns of the datasets, while Figures 5(a), (b), and (c) show the number of rows and type of data contained in the three datasets.

```
In [7]: # Explore
        df_train.head()
```

Out[7]:

	x	y1	y2	y3	y4
0	-20.0	19.682550	-8.643847	-0.792252	-8795.289
1	-19.9	20.401932	-8.724133	-1.132539	-8667.858
2	-19.8	19.618414	-8.848171	-0.610908	-8541.768
3	-19.7	19.495897	-9.648293	-1.236778	-8416.403
4	-19.6	20.155570	-9.133513	-0.416563	-8292.688

(a)

```
In [9]: df_test.head()
```

Out[9]:

	x	y
0	19.0	17.754017
1	-0.7	-3444.012500
2	-5.6	-232.979830
3	6.1	5950.886000
4	0.2	5.371201

(b)

```
In [8]: df_ideal.head()
```

Out[8]:

	x	y1	y2	y3	y4	y5	y6	y7	y8	y9	...	y41	y42	y43	y44	y45
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	12.995732
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	12.990720
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	12.985682
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	12.980619
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	12.975530

5 rows × 51 columns

(c)

Figure 4: First five rows of the (a) training, (b) testing, and (c) ideal datasets

```
In [8]: df_train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       400 non-null    float64
 1   y1      400 non-null    float64
 2   y2      400 non-null    float64
 3   y3      400 non-null    float64
 4   y4      400 non-null    float64
dtypes: float64(5)
memory usage: 15.8 KB
```

(a)

```
In [10]: df_ideal.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 51 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       800 non-null    float64
 1   y1      800 non-null    float64
 2   y2      800 non-null    float64
 3   y3      800 non-null    float64
 4   y4      800 non-null    float64
 5   y5      800 non-null    float64
 6   y6      800 non-null    float64
```

(b)

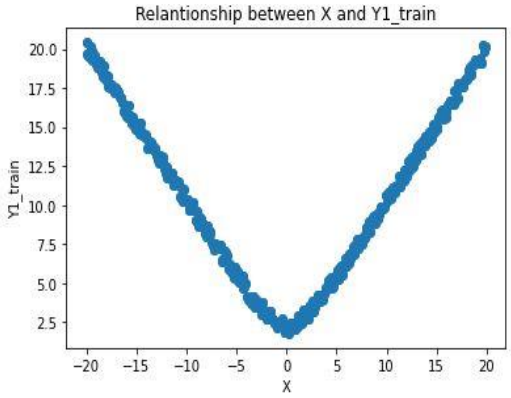
```
In [11]: df_test.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       200 non-null    float64
 1   y       200 non-null    float64
dtypes: float64(2)
memory usage: 3.2 KB
```

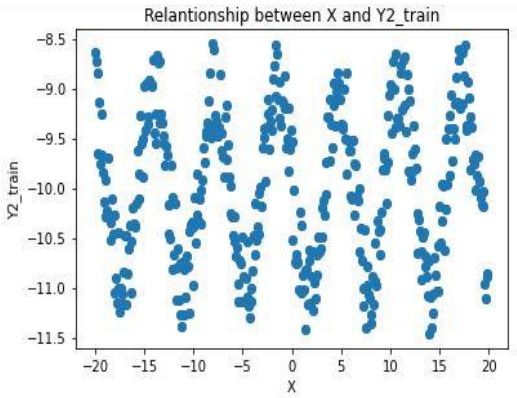
(c)

Figure 5: Data type and number of rows and columns of the (a) training, (b) ideal, and (c) testing datasets

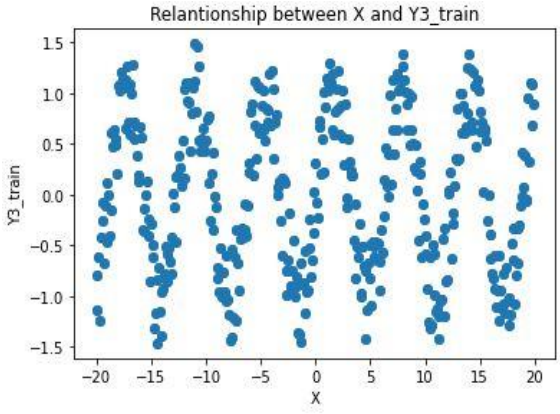
The starting point of the analysis was the data contained in the training dataset. The training dataset contained one independent variable and four functions, namely y1, y2, y3, and y4. In this study, we shall refer to them as Y1_train, Y2_train, Y3_train, and Y4_train, respectively. Each of these four training functions was analyzed and their relationship with the independent variable was studied. Figures 6(a), (b), (c), and (d) show the relationship between the independent variable and each training function.



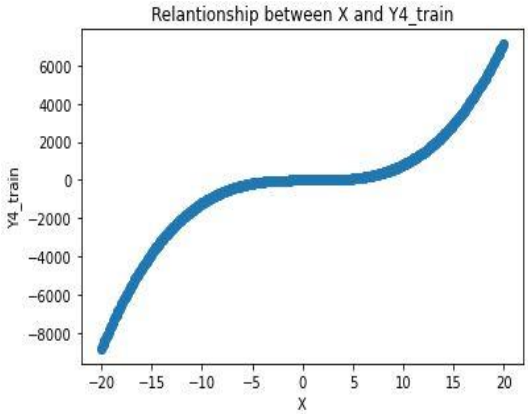
(a)



(b)



(b)



(d)

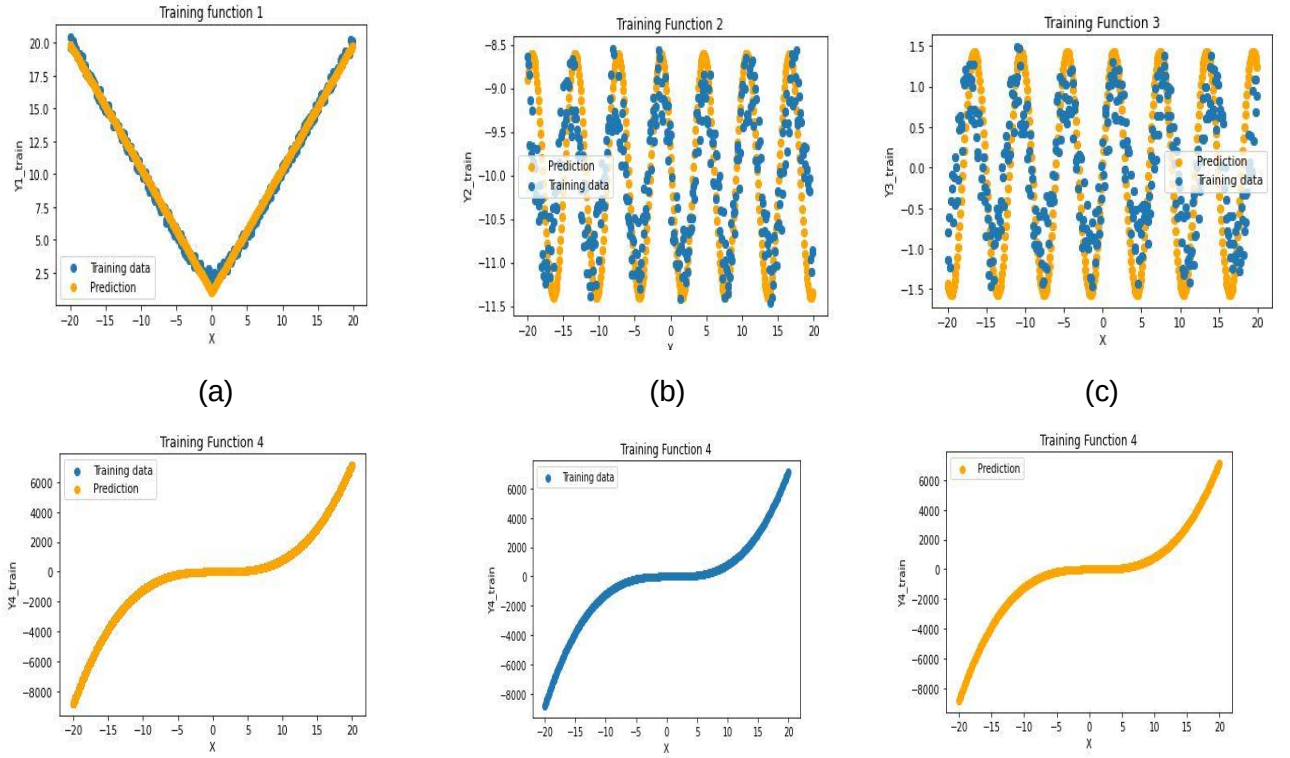
Figure 6: Plots of (a) Y1_train, (b) Y2_train, (c) Y3_train, and (d) Y4_train.

A regression was calculated for each training function, the coefficients were extracted and are shown in Table 1.

Table 1: Calculated regressions

Function	Calculated regression	Type
Y1_train	$Y = 0.9429 * \|X\| + 1.0093$	Linear function
Y2_train	$Y = 1.4 * \sin(\pi X + 3) - 10$	Sinusoidal function
Y3_train	$Y = 1.5 * \sin(\pi X + 0.08) - 0.07$	Sinusoidal function
Y4_train	$Y = 2.675 * e^{-03} * X^3 + 2.000 * X^2 + 0.999 * X + 7.124 * e^{-07}$	Polynomial function

Based on the calculated regression, predictions were made and compared with their corresponding training functions. Figures 7 (a), (b), (c), (d), (e), and (f) show how each calculated regression compares with their respective training data.



(a) (b) (c)

(d)

Figure 7: Comparison between training data and calculated regression for (a) Y1_train, (b) Y2_train, (c) Y3_train, and (d) Y4_train.

Subsequently, the criteria described in Section 3 was applied to determine the corresponding ideal function out of the fifty functions available in the ideal dataset. Table 2 shows the ideal function selected for each training function. Figures 8 (a), (b), (c), and (d) show how each training function compares with its respective ideal function as selected by the program.

Table 2: Training functions with their corresponding ideal functions

Training function	No of the corresponding ideal function
Y1_train	33
Y2_train	5
Y3_train	1
Y4_train	30

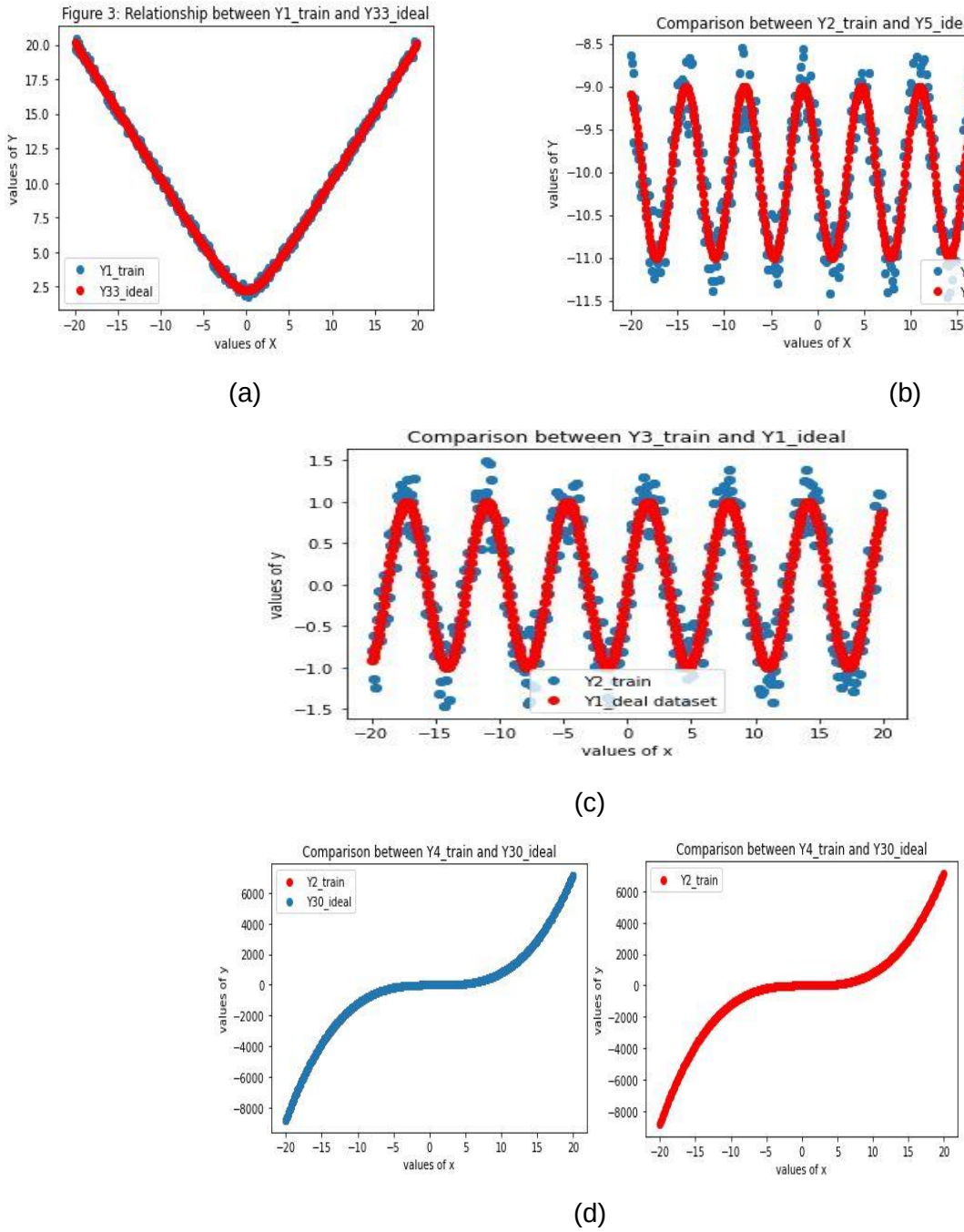


Figure 8: Comparison between training functions and their corresponding ideal functions.

Based on the criteria presented in the last paragraph of Section 3, the criteria for mapping the individual test case to the four ideal functions can be expressed mathematically by equation (1):

$$A - B \leq \sqrt{2} * B \qquad (1)$$

Where A denotes the maximum deviation of the calculated regression, and B denotes the largest deviation between the training dataset and ideal function.

The values of A and B for the test dataset and the chosen functions were calculated and stored in a data frame. Afterward, equation (1) was applied and the mapping was performed. Figure 9 shows the mapping relative to the delta and the test dataset.

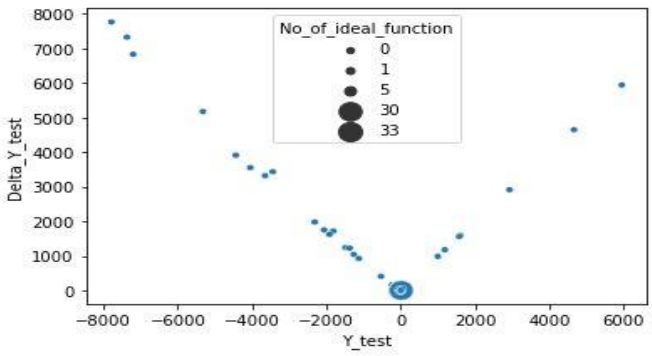


Figure 9: Relationship between Delta_Y_test, Y_test, and selected ideal functions

Table 3 shows 13 out of the 100 mapping results (for the full results please see appendix).

Table 3: Mapping results

X (test func.)	Y (test func.)	Delta Y (func.)	No of the selected ideal function
19.0	17.754017	1.170135	33
-0.7	-3444.012500	-3444.012500	0
-5.6	-232.979830	-232.979830	0
-13.9	-7.812611	1.001583	5
0.2	5.371201	0.200000	0
-9.5	9.409774	0.556965	33
6.8	-10.070290	0.827502	5
0.1	-10.980573	1.032183	5
15.5	1.352991	2.274405	0
-15.2	0.855583	0.731966	1
-5.9	0.044731	0.160776	1
-1.0	-1.871560	0.566612	30

Once satisfied with the mapping (as shown in Figure 9), the data frame containing the results was saved into the database as shown in Figure 10. For the complete and detailed python code, please see the Appendices.

```
In [52]: # SAVE THE RESULTS INTO THE SQLITE DATABASE

         # create a table in the dataframe to store the results
         c.execute('''CREATE TABLE results (X_test, Y_test, Delta_Y_test, No_of_ideal_function)''')

         # write the data into sqlite tables
         df_final_results.to_sql('results', conn, if_exists='append', index = False)

Out[52]: 100
```

Figure 10: Python code instructing the program to save the results in the database

SECTION 6 - DISCUSSION

Because the three datasets were already cleaned and structured, the analysis did not involve cleaning or structuring the data.

The first step involved exploring the data. As shown in Figures 4(a), (b), and (c) and 5(a), (b), and (c) the training dataset is organized in 5 columns and 400 entries, the ideal dataset has 51 columns and 800 entries, and the test dataset has two columns and 200 entries.

In the training dataset, the first column was the values of the independent variable and each of the remaining columns was an independent variable referred to as Y1_train, Y2_train, Y3_train, and Y4_train in this paper.

Figure 6(a) shows that Y1_train is a linear function. The training data was used to build a linear regression model. The calculated regression is shown in Table 1 and below:

$$\mathrm{Y} = 0.9429 * |\mathrm{X}| + 1.0093 \qquad (2)$$

Figure 7(a) shows that equation (2) is truly a mathematical expression that can describe the data in Y1_train. Equation (2) (or calculated regression for Y1_train) was used by the program to determine the ideal function for Y1_train. As shown in Table 2, the ideal function for Y1-train is Y33_ideal. Figure 8(a) shows Y1_train plotted with Y33_ideal; it proves that the program made an accurate selection.

Figure 6(b) shows that Y2_train is a sinusoidal function. From this plot, the parameters for a sinusoidal function were estimated and the calculated regression was estimated as

$$\mathrm{Y} = 1.4 * \sin(\pi\mathrm{X} + 3) - 10 \qquad (3)$$

Figure 7(b) shows that equation (3) can be used to mathematically represent the data contained in Y2_train. Equation (3) was used to determine the ideal function for Y2_train. Table 2 shows that the selected ideal function for Y2_train is Y5_ideal. Although Y2_train does not match Y5_ideal for y values outside the range -1 to 1, it meets the criteria used to select the ideal function and this selection has been regarded as an appropriate calculated regression in this study.

Figure 6(c) shows that Y3_train is a sinusoidal function. Based on Figure 6(c) the parameters for a sinusoidal function were estimated and the calculated regression was estimated as

$$\mathrm{Y} = 1.5 * \sin(\pi\mathrm{X} + 0.08) - 0.07 \qquad (4)$$

Figure 7(c) shows that equation (4) is a mathematical expression that can represent the data contained in Y3_train. Equation (4) was used to determine the ideal function for Y3_train. Table 2 shows that the selected ideal function for Y3_train is Y1_ideal. Although Y3_train does not match Y1_ideal for y values outside the range -1 to 1, it meets the criteria used to select the ideal function and this selection has been accepted in this study.

Figure 6(d) shows that Y4_train has a polynomial nature. The data contained in Y4_train was used to build a polynomial regression model. As shown in Table 1, the calculated polynomial regression is

$$Y = 2.675 * e^{-03} * X^3 + 2.000 * X^2 + 0.999 * X + 7.124 * e^{-07} \qquad (5)$$

Figure 7(d) shows that equation (5) is a perfect mathematical expression that can represent the data contained in Y2_train. They are perfectly matched to each other to the point that plotting one above the other makes the first one completely covered by the other (as seen in the first plot in Figure 7 (d)). Thus, there are three plots in Figure 7(d). Equation (5) was used to determine the ideal function for Y2_train. Table 2 shows that the selected ideal function for Y4_train is Y30_ideal. Figure 8(d) shows Y1_train and Y33_ideal; it proves that the program made an accurate selection for the ideal function for Y1_train. Note that there are two plots in Figure 8(d) because Y4_train and Y30_ideal are perfectly matched to the point that if plotted together one completely covers the other. Thus, two plots are shown.

Figure 9 shows the data contained in the data frame with the final results excluding only the first column, which is the column with the independent variable. This plot shows that the program is performing the mapping appropriately. Those rows, or testing functions, whose ideal function number is 0 are those that could not be mapped to any of the four chosen ideal functions. There are 400 data entries, or x-y pairs in the test dataset, and not all of them should be expected to be represented

by one of the four selected ideal functions. Perhaps, they could all have been selected if all fifty ideal functions were used in the mapping.

Figure 9 shows that the test points that were mapped to ideal functions 1, 5, 30, or 33 have test deviations (Delta_Y_test) very close to zero. The method based on which the four ideal functions were selected is expressed by equation (1), which imposes a limit to the test deviation for a given test data point. Thus, it is a positive thing that the deviations of all mapped data points are around zero. This shows that the program is working appropriately because only data points satisfying equation (1) were selected.

Further, the Y_test values of all mapped test data points appear to be around zero. This could be the result of two things. First, these values appear to be around zero because of the calibration (range) of the values shown in the plot. The plot starts at a value smaller than -8000 and goes up to a value higher than 6000. In such a range, all those values with y equal to 17, 6, and any number of this order of magnitude, will appear to be very close to zero.

Furthermore, Table 3 shows that there are some values in the order of thousands in the test dataset. Figure 8(d), 7(d), and 6(d) show that Y4_train and Y30_ideal range from a value lower than -8000 to a value higher than 6000. However, based on Figure 9, none of the large numbers were mapped, not even for Y30_ideal. This could suggest that perhaps the program only performs well for values that are below that order of magnitude. On the contrary, it could be that the large numbers could not be mapped simply because they do not match any of the four chosen idea functions. This is certainly an appropriate topic for further research.

Although the python code meets all the requirements set in the problem statement, during the study it was noted that some of the test functions could be mapped to more than one ideal function. For this study, we have only chosen one selected ideal function for each test data point. For further research, it would be worth investigating whether or not this has any effect on the final results.

The 400 columns multiplied by 5 columns in the training dataset, added to 800 entries multiplied by the 51 columns in the ideal dataset, added to 200 entries multiplied by 2 columns in the test dataset, make a total of 43000 values that needed to be analyzed to accomplish the tasks accomplished in this study. Using conventional techniques, it would take a significant amount of human effort, time, and energy to do what we have done in minutes. As demonstrated in this study case, with the application of data science techniques in research, a significant amount of time, energy, and human effort can be saved. The time, energy, and effort saved from applying data science in research can certainly be applied somewhere else. In addition, at an individual level, researchers skilled in data science techniques will certainly be more productive and efficient in their fields. The time and energy spent analyzing large sets of data can be devoted to studying another important thing.

SECTION 7 - CONCLUSION

In this study, we have used a case study involving the mapping of data to ideal mathematical functions to demonstrate how useful data science techniques can be in modern research. Using python, we analyzed a large amount of data and built one linear regression, one polynomial regression, and two sinusoidal regressions, which aided us in selecting four out of fifty ideal functions that best describe the training data. Further, the chosen four ideal functions were mapped to 400 data points.

The fact that we successfully selected four ideal functions out of fifty and then mapped the test dataset to the chosen ideal functions in minutes demonstrates how useful data science techniques can be. These techniques can help researchers save a significant amount of time and energy.

The scope for further research includes the following two points:

- Exploring whether or not the fact that some test data points could be mapped to more than one ideal function impacts the mapping ability of the program or it is simply anchored on the criteria used in the mapping process;
- Although there are large numbers in the range of Y4_train, Y30_ideal, and the test set, no large numbers were mapped to any of the four chosen ideal functions. The further study would explore the question of whether this is a limitation of the program created or truly no large test data point can be mapped to any of the four chosen ideal functions

REFERENCE LIST

Bollen, J. Mao, H., Zeng, X. (2010). Twitter mood predicts the stock market. *Journal of Computer Science*, 2 (1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29 (7),1645–1660. https://doi.org/10.1016/j.future.2013.01.010

IBM Corporation (2013). *Data-driven healthcare organizations use big data analytics for big gains.*https://silo.tips/download/ibm-software-white-paper-data-driven-healthcare-organizations-use-big-data-analy

Muni, N. & Manjula, R. (2014). Role of big data analysis in rural health care- a step towards svasth bharath. *International Journal of Computer Science and Information Technologies*, 5 (6), 7172-7178. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.903&rep=rep1&type=pdf

Ren, Y., Werner, R., Pazzi, N., Boukerche, A. (2010). Monitoring patients via a secure and mobile healthcare system. *IEEE Wireless Communications*, 17 (1), 59–65. Doi:10.1109/MWC.2010.5416351

Sengupta, P. P. (2013). Intelligent platforms for disease assessment: novel approaches in functional echocardiography. *JACC: Cardiovascular Imaging,* 6 (11), 1206–1211. https://doi.org/10.1016/j.jcmg.2013.09.003