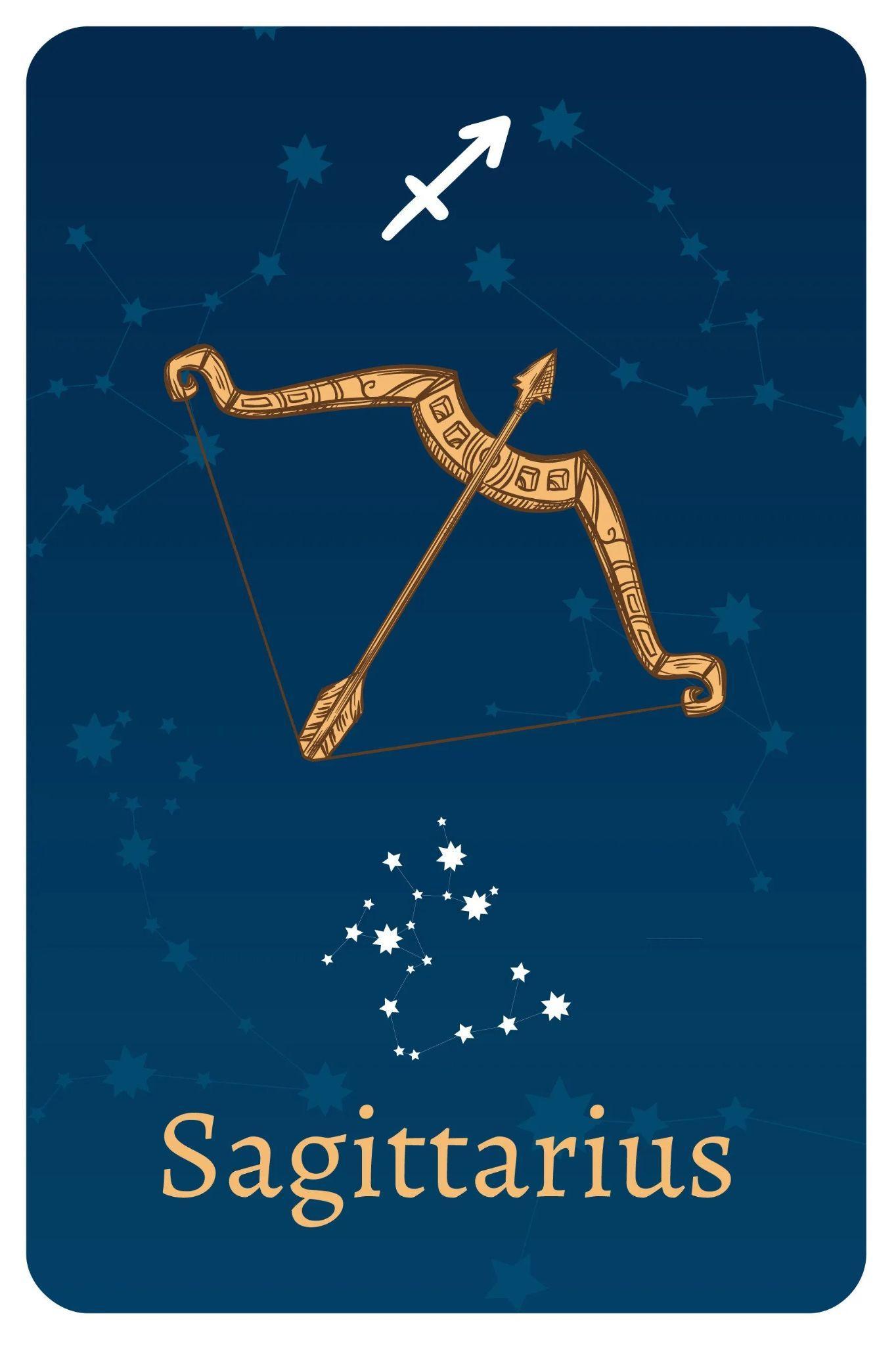
****

TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

Contents

[PURPOSES AND GOALS 3](#_heading=h.gjdgxs)

[HOW TO INSTALL THE PROGRAMS 4](#_heading=h.30j0zll)

[DEPENDENCIES 4](#_heading=h.1fob9te)

[HOW TO RUN THE PROGRAMS 4](#_heading=h.3znysh7)

[DESIGN ARCHITECTURE 5](#_heading=h.2et92p0)

[PROCESS & WORKFLOW 5](#_heading=h.tyjcwt)

[TEST DATA 5](#_heading=h.3dy6vkm)

[VIDEO RECORDINGS 6](#_heading=h.1t3h5sf)

[CONCLUSIONS 6](#_heading=h.4d34og8)

[REFERENCES 6](#_heading=h.2s8eyo1)

# PURPOSES AND GOALS

The purpose of this lab is to introduce students to various curve fitting methods in Python and improve their comprehension of these approaches. Students will receive hands-on experience in applying several approaches to fit curves to diverse types of data in this lab. The lab's aim is to provide hands-on experience in curve fitting and related applications, which can be useful in fields like data analysis, signal processing, and scientific research.

Goals:

1. Curve Fit Using Polynomial Regression:
   1. Generate a polynomial curve using a chosen order and an online graphing tool.
   2. Develop a Python program to plot the same curve and verify its accuracy.
   3. Implement polynomial regression to fit the generated curve in Python.
2. Curve Fit Using Ridge or Linear Regression:
   1. Generate another polynomial curve using an online graphing tool.
   2. Create a Python program to plot this curve, ensuring it matches the generated one.
   3. Apply Ridge or Linear Regression methods to fit the curve in Python.
3. Damped Sine Wave Curve Fit:
   1. Define an arbitrary damped sine wave function and visualize it using an online graphing tool.
   2. Develop a Python program to plot this damped sine wave curve.
   3. Apply curve fitting to the generated damped sine wave curve.
4. Curve Fit with Noise:
   1. Generate or create tones with noise and prepare data in CSV format.
   2. Write a Python program to add noise to the signal data.
   3. Use curve fitting methods to fit a curve to the noisy signal.
5. Curve Fit with Multiple Variables:
   1. Write a Python program to generate waveforms with two independent variables.
   2. Test the curve fitting for five different scenarios, providing comments and observations for each.
6. Game Development: Coin Collector:
   1. Leverage existing base code for a "Coin Collector" game.
   2. Customize and enhance the game by adding your own hacks and tweaks.

# HOW TO INSTALL THE PROGRAMS

## DEPENDENCIES

Coin Collector: pip install pygame, pgzero

Pipeline Linear Regression: pip install numpy, sklearn, and matplotlib

Damped Sine Wave: pip install numpy, scipy, and matplotlib

Curve Fit with Noise: pip install numpy, scipy, and matplotlib

Multi-Variable: pip install numpy, scipy, and matplotlib

# HOW TO RUN THE PROGRAMS

Curve Fit Using Polynomial Regression: Run Code and see plots

Pipeline Linear Regression: To run the code for polynomial curve fitting, first install Python and required libraries (NumPy, scikit-learn, Matplotlib) using pip. Then, create a Python script (e.g., curve\_fit.py) with the code and run it using python curve\_fit.py. A plot will show actual data and the fitted curve.

Damped Sine Wave: To run the code, ensure you have Python installed, install required libraries using pip, create a Python script file with the code, and execute it with the python command in a terminal or command prompt. The results, a plot displaying actual data points and the fitted curve, will appear in a window.

Coin Collector: Run program through IDE, use arrow keys to move Fox.

Curve Fit with Noise: Install required libraries, then run python script to see plots.

Multi-Variable: Install required libraries, then run python script to see the plot.

# DESIGN ARCHITECTURE

Curve Fit Using Polynomial Regression: The computer is the principal piece of hardware on which the code executes.Python is the primary language used to execute the code, coupled with numerical libraries such as NumPy and SciPy. These libraries are used within this environment to perform mathematical calculations, polynomial fitting, and graph display.

Pipeline Linear Regression: This code demonstrates the collaboration between software and hardware blocks. The hardware (CPU, memory) provides the computational foundation, while software components (Python, libraries) leverage this hardware for data analysis and modeling. This interaction applies both locally and in the cloud, where software interfaces with remote hardware through network connections and APIs.

Damped Sine Wave: The hardware forms the essential backbone for running the software. In this case, the software, which includes Python code and various libraries, harnesses the hardware's power to carry out tasks like data generation, curve fitting, and creating visualizations. Although this code is designed for local hardware, it can be modified to function in cloud environments, where it can tap into remote resources for increased scalability.

Coin Collector: Graphics rendering and keyboard input is handled by the display and input hardware. Graphics gear used to render game features. Game logic sets up game elements and manages game state. Pygame and Pygame Zero are used to represent actors and game objects. Screen drawing, user input, and game updates are all managed by event handling. Coin placement and collection are determined by random number generation. This relationship is orchestrated by the code, which creates a game in which the user controls a fox to collect coins. It facilitates this connection by utilizing Pygame and Pygame Zero within a gaming framework.

Curve Fit with Noise: This python code uses both of the computer’s software and hardware to communicate with each other to execute the program. It requires a set of libraries of pre-written code to perform specific tasks. With this, it is able to calculate and plot the x and y values for the given data.

Multi-Variable: The computer hardware uses the processor and memory to process the software with python language, which is translated to binary code for the hardware to execute certain tasks. It also uses a set of libraries of pre-written code to perform certain tasks.

# PROCESS & WORKFLOW

Pipeline Linear Regression: The code's process involves generating synthetic data, setting up a data processing pipeline, training the model, predicting values, and visualizing results. In the workflow, it begins with data generation, proceeds to pipeline setup, model training, prediction, and finally visualization. These steps collectively illustrate how data is processed and analyzed within the code, with the workflow encapsulating the sequential flow of these processes.

Damped Sine Wave: the process begins with library importation, where necessary libraries like NumPy, SciPy, and Matplotlib are imported. Next, a custom mathematical function 'eq' is defined for data generation and curve fitting. Synthetic data points for 'x' and 'y' are then generated. The code performs curve fitting using SciPy's optimization function and concludes by visualizing the results through a plot. This structured workflow ensures that each step builds upon the previous one, ultimately showcasing the fitted curve.

Coin Collector: The player starts the game by running the code.The player sees the game screen with the fox character, a coin, and the current score. The player uses the keyboard arrow keys to move the fox character left, right, up, or down. The player's goal is to move the fox to collect the coin.If the fox collides with the coin, the player's score increases.After a fixed time interval, the game may end. The screen turns pink, and the final score is displayed. The player can restart the game by running it again.

Curve Fit with Noise and Multi-Variables: This program has a set of libraries imported in order to plot the data. Matplotlib is for the visualization of the graph, scipy is used for curve fitting, and numpy is used for arrays.

# TEST DATA

| Test Name | Code Output | Successful |
| --- | --- | --- |
| Pipeline Linear Regression |  | Yes |
| Damped Sine Wave Curve |  | Yes |

# VIDEO RECORDINGS

| **Recording Title** | **URL** | **Notes** |
| --- | --- | --- |
| Coin Game | <https://youtu.be/o0Mks2PAGog> | Demo for Coin Game |
| Polynomial Regression | https://youtu.be/iGoLje9ssB4 | Demo for Polynomial Regression |
| Pipeline Linear Regression | <https://youtu.be/laxoks4FEjY> | Demo for Pipline Linear regression |
| Damped Sine Wave Curve | <https://youtu.be/djoR1fNWGAM> | Demo for Damped Sine wave curve |

# CONCLUSION

In this lab, our team learned how to perform plots using the library matplotlib to create graphs while calculating the x and y values. With this, we were able to create a polynomial curve with regression fit, use linear regression methods to fit the curve, plotting a damped sine wave curve, creating a curve fit with noise, generating waveforms with different scenarios, and developing a coin collector game.

We learned to use research to make our programs stronger, and the importance of code organization to find bugs and to easily find and edit the program code.

# REFERENCES

* What Is Agile Reporting? (Definition and How To Complete) <https://www.indeed.com/career-advice/career-development/agile-reporting>
* Test Report <https://strongqa.com/qa-portal/testing-docs-templates/test-report>
* Signal Generator <https://www.wavtones.com/functiongenerator.php>