Project Title: **Curve Fitting Algorithm for Engineering Applications.**

Team Members: **Michael Da Silva, Edith Gómez, Gabriel Torres**

ABSTRACT

This work focuses on generating a tight curve fitting interphase for users of different backgrounds interested in performing mathematical predictions based on experimental data. The main structure of this software is given by a python algorithm able to perform multiple regression models. The performance of each curve fitting model is studied through statistical metrics, such as the root mean square error (RMSE) and accuracy score (R2). The performance of either model would depend on the characteristics and complexity of the studied data set, so this routine provides the user with a menu of different models to choose from depending on the user's data preprocessing. However, this algorithm aids beginners in statistics by performing five different curve-fitting models in the background and recommends the user the most suitable model available for the given data set based on performance metrics. The predicted results will be graphically visualized by the user from comparative charts of the actual testing data sets and predicted models.

Keywords

- Fitting model
- Regression
- Coefficients
- Error measurement
- Graphical representation
- Model Equation
- Dependent variable
- Independent variable
- R-squared
- Mean Square Error

OVERVIEW

Curve fitting is a mathematical tool commonly used to predict the behavior of variables from observed data. In fact, curve fitting set the foundations of several modern applications, such as machine learning, big data management and artificial intelligence. In this work, principles of linear algebra and differential calculus are applied to generate adequate predictive models that can allow the user to not only predict the behavior of variables, but also to come up with different kinds of decisions based on previously test experimental data.

This project's nature consists of developing an application that utilizes 5 different regression models to find the prediction coefficients while minimizing the error (it is assumed that the dataset does not have missing values and "Not A Number" values). With this application the final user will be able to use a graphical user interface (GUI) to navigate through a created menu and upload a CSV file with the dataset willing to analyze, select the model of their choice and finally obtain the desired coefficients, equation, and error measurement of the selected model. Additionally, the program will display a recommendation of other model with better performance.

This application is important to the team because is a problem that is transversal to many areas. As a group formed by an Aerospace, Data science, and Robotics student this predictive application would be of use to several problems presented in these fields. Our objective is to develop an application that in the future could be used for projects and research in any of the areas of interest. This project represents a foundation in the

team members growth as professionals in STEM areas, deepening the understanding of basic concepts in statistics, mathematics, and their engineering applications.

MATERIALS

- An Introduction to Statistical Learning with Applications in R - Gareth James Daniela Witten Trevor Hastie Robert Tibshirani
    1. Chapter 2: Statistical Learning
       This chapter explores the fundamental concepts of statistical learning like regression model, estimation, predictor variable, response variable, mean square error, etc. The topics are useful to the implementation of the project since we are developing an application that creates a statistical model for a given dataset.
    2. Chapter 3: Linear Regression

       This chapter discusses the basic concepts about Linear Regression and how to implement a Linear Regression Model with R. Additionally, it discusses possible problems that one could encounter while fitting this model to a dataset. It is useful to the project because it gives a statistical and mathematical foundation needed to carry out the project.

- Scikit-learn library: https://scikit-learn.org/stable/

  Within this library there are predefined Classes, tailor made for handling regression. Some concepts found here are classification models and Regression models, both would give the team a guidance to implement the statistics and mathematics involved in the fitting process.

- Kaggle: https://www.kaggle.com/datasets

  Kaggle is a repository for a wide variety of datasets. The team will rely on this website to procure datasets to test the application for robustness.

- Towards Data Science - https://towardsdatascience.com/a-quick-way-to-build-applications-in-python-51d5ef477d88

  This website illustrates users on the steps needed to create applications in python. Some important concepts included in this page are graphical user interface (GUI) and Application Programming Interface (API), these two concepts will aid to the development of the project's code and make it easy to interact with.

- Listen Data - https://www.listendata.com/2018/03/regression-analysis.html#Support-Vector-Regression

  This webpage gives an overview of the different Regression Models that exist. In here users can find some of the models that are going to be implemented in this project. This information will give the team a better understanding of how the Linear Regression, Polynomial Regression and Support Vector Regression work; aiding the team to develop the logic behind the curve fitting models into the python environment.

- Statistics by Jim - https://statisticsbyjim.com/regression/curve-fitting-linear-nonlinear-regression/

  In this website it is explained the differences between linear models and nonlinear models, as well as how one can interpret the results obtained while utilizing any of these models. For this project it is crucial to understand the criteria for a fitting model to be considered adequate for a data set, therefore this website is considered relevant for background research. In addition, within this

website there are multiple examples of graphical outputs of models that can serve as inspiration on the development of the Graphical User Interface.

- Origin Lab – Curve and Surface Fitting

  https://www.originlab.com/index.aspx?go=Products/Origin/DataAnalysis/CurveFitting

  This webpage demonstrates the capabilities that Origin can perform in the matter of curve and surface fitting. There are explanations and examples of how Origin shows graphical outputs to the user. The importance of this website relies on giving the team ideas of how the GUI of this project could be created in terms of aesthetic and function.

- From Curve Fitting to Machine Learning - Achim Zielesny
  1. Chapter 2: Curve Fitting

     This chapter explains the basic concepts of curve fitting and its implementation methods. It also illustrates the problems that could be encountered while fitting the incorrect model to a dataset. With this information the team can plan strategies for facing possible obstacles while developing the curve fitting algorithms and developing the calculation of the error measurements.

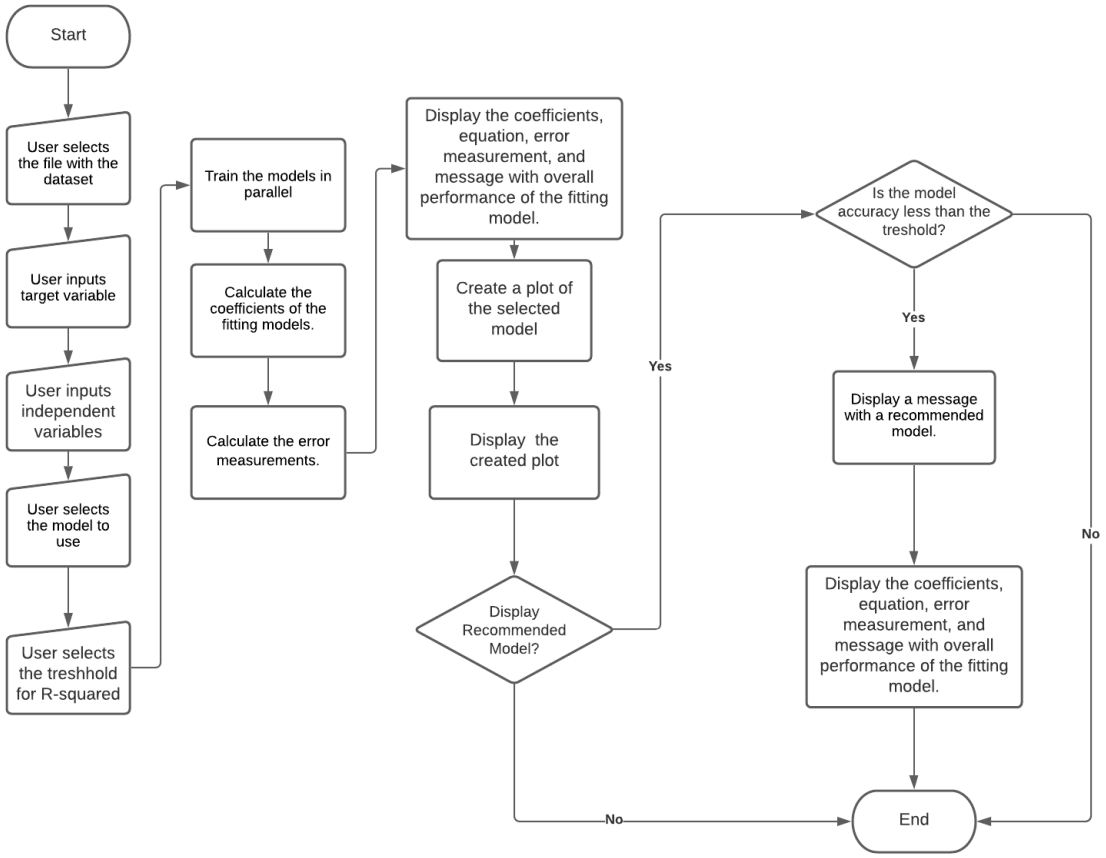
APPROACH

To achieve the curve fitting application, the team will construct a python program that uses multiple functions and one main driver. Each of these defined functions will perform distinct fitting models and return to the main driver the outputs obtained after performing the correspondent regression task.

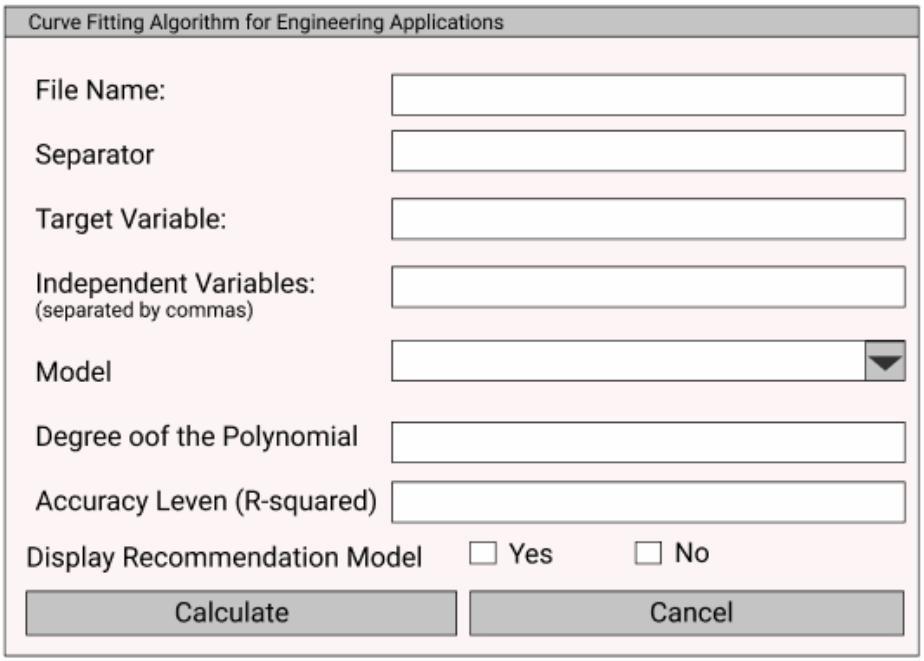
The inputs necessary from the user are a dataset file in CSV format containing the different variables of interest, the preferred fitting model for the program to display, name of the target variable, name of the independent variables (in case the dataset file contain extra variables that the user does not want to use in the curve fitting), desired accuracy level, choose to deploy the recommended model.

For the outputs, the program will return to the user the following: coefficients of the fitting model with their respective variable names, fitting model equation for the linear models, error measurement, R-square of the model, an indicative message of the general performance of the model and a recommendation of a more suitable model, a plot showing the created model. The intended application will perform 2-D graphs only, meaning that when the number of independent variables entered by the user is greater than one, the program will display pairwise graphs to observe each independent variable on the x-axis and the target variable on the y-axis.

High Level Processing Chart



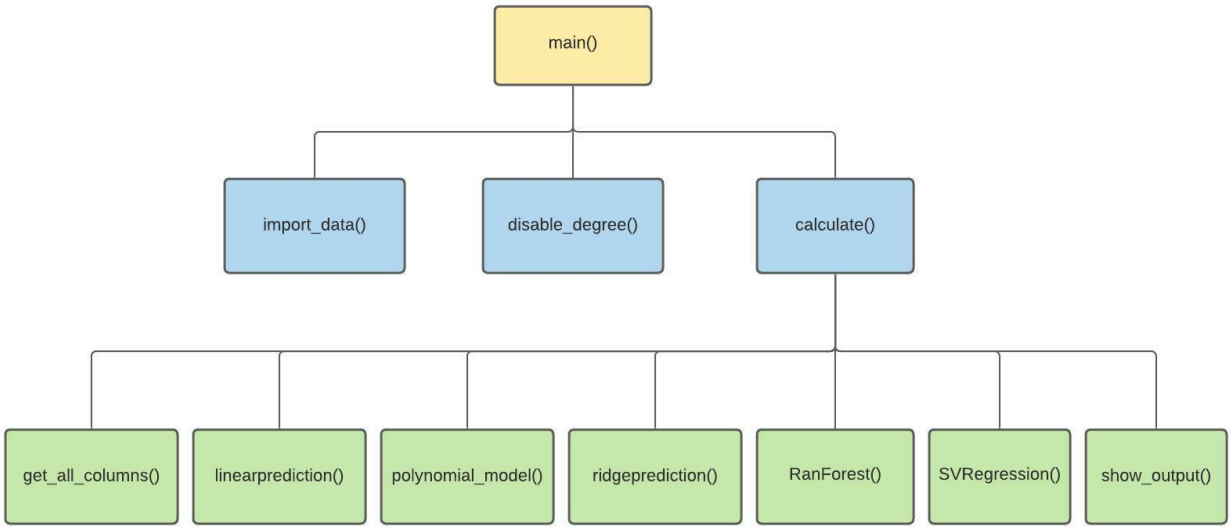
UI diagram



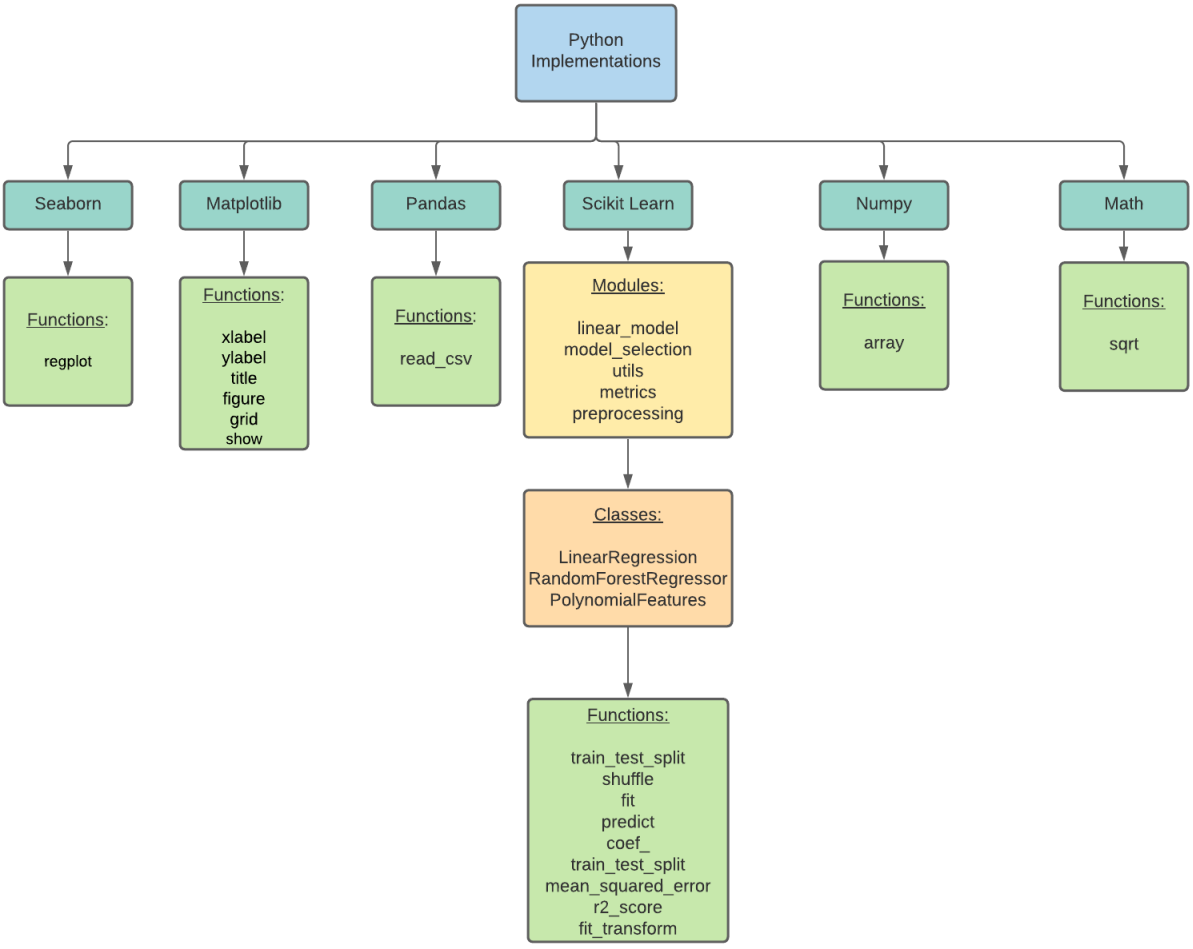
We have designed a Graphical User Interface that is easy to use for the final user. The File Name, Separator, Target Variable, Degree of Polynomial, Accuracy Level and Independent Variables are text boxes where the user should type the required names. The user can select which model from the catalog to display by using the Model dropdown. On the text box for the independent variables, it is indicated that the variable names should be separated with commas in order for the program to be able to process them. Next, there are two checkboxes where there can be indicated if the coefficients, model equation and error measurements of the recommended model want to be displayed after the calculation.

At the bottom of the window, we have positioned two buttons. The left one indicates the program to start running the calculations, and the right one serves as a stop for the program in case the user wants to do so.

Hierarchy Chart



Python Implementations



The graph above indicates all the libraries, classes and functions that have been implemented on the project product. We now describe the functionalities of the libraries presented, from left to right:

Matplotlib:

Used for graphical representations of data. This library was implemented in the creation of graphs with the data points and regression obtained. Nonetheless, the graphs are only going to be executed when the number of variables is less than 3. Since each independent variable represents a dimension on a plot, it could not be possible to graphically represent more than three dimensions.

Seaborn:

Commonly used in complementation with Matplotlib to create graphs and charts. In this project, the seaborn library was implemented to create scatter plots and lines to show the end users the behavior of the data with their corresponding curve fitting.

Pandas:

Has been implemented to manipulate the dataset. Pandas is a library that converts files, such as csv files in this case, into dataframe format. This way it is easier to extract columns and values from the dataset.

Scikit Learn:

This library is commonly used on machine learning tasks and predictive analysis. For this project, Scikit learn has been utilized for the fitting of the different models, splitting the dataset into training set and test set and obtaining the coefficients of the fitted models.

Numpy:

With Numpy it is possible to manipulate large multidimensional arrays and apply to them complex mathematical operations. For the purposes of this project, it has been employed to convert the data sets in arrays and utilized them in the machine learning algorithms

Math:

The Math library includes multiple mathematical functions that have been previously defined in other computer languages. It has been applied to calculate the error measurements of the different models.

Validation

At the moment it is estimated that the project is completed at its entirety. All the functions proposed on previous steps have been implemented and proved to function properly. In addition, the main program functions continuously providing the expected outputs for multiple scenarios without the necessity of running the program multiple times.

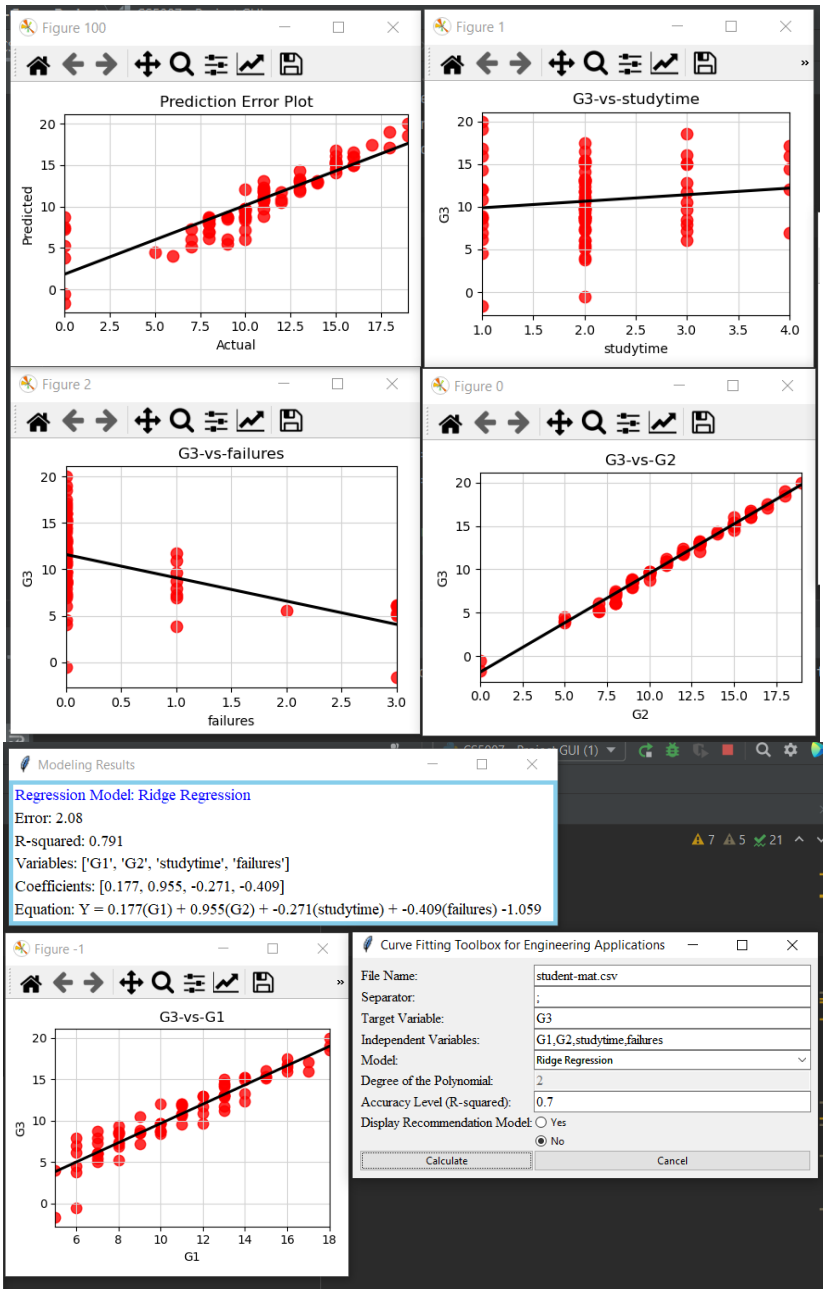
The team created the code necessary to implement graphical representations of the models in the catalog and create a functioning Graphical User Interface (GUI) as the one designed in this report. This way, all the functionalities of the program are complete, and the final product can be used by third parties.

Regarding the test cases, the team has used multiple free source datasets from the website Kaggle to validate the adequate function of the program. These datasets are in csv format and are already depurated, contain no missing values. A trial-and-error process has been used with different combinations of the set of variables contained in the datasets, with the purpose of encountering a variety of outputs and unexpected malfunction. The used datasets have the following names: student-mat.csv, Concrete_Data.csv and CarPrice_Assignment.csv

At the end, the correct operation of the final product will be verified with the deployment of the different graphs and error measurements for multiple datasets. Below, various test cases are presented to show the correct functioning of the application:

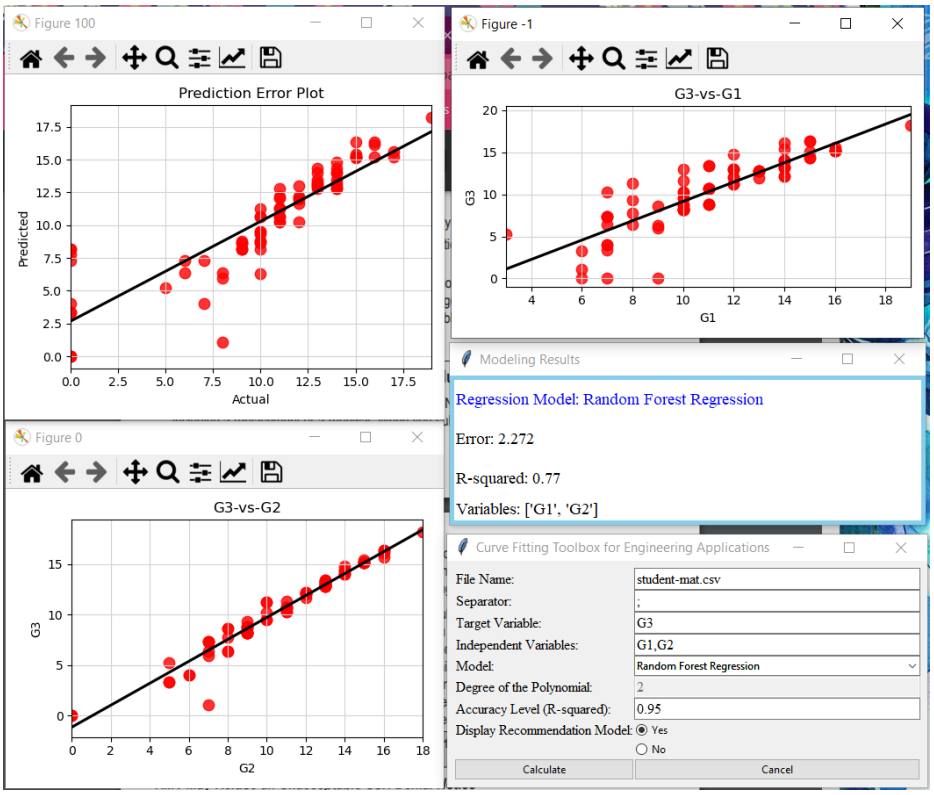
Test Cases
1) Dataset including more than 3 independent variables
2) The user enters a high threshold for the accuracy level
3) Dataset with only one independent variable
4) The user inputs a low threshold for the accuracy level
5) The user selects a high degree for the Polynomial Regression

Case 1: Dataset including more than 3 independent variables

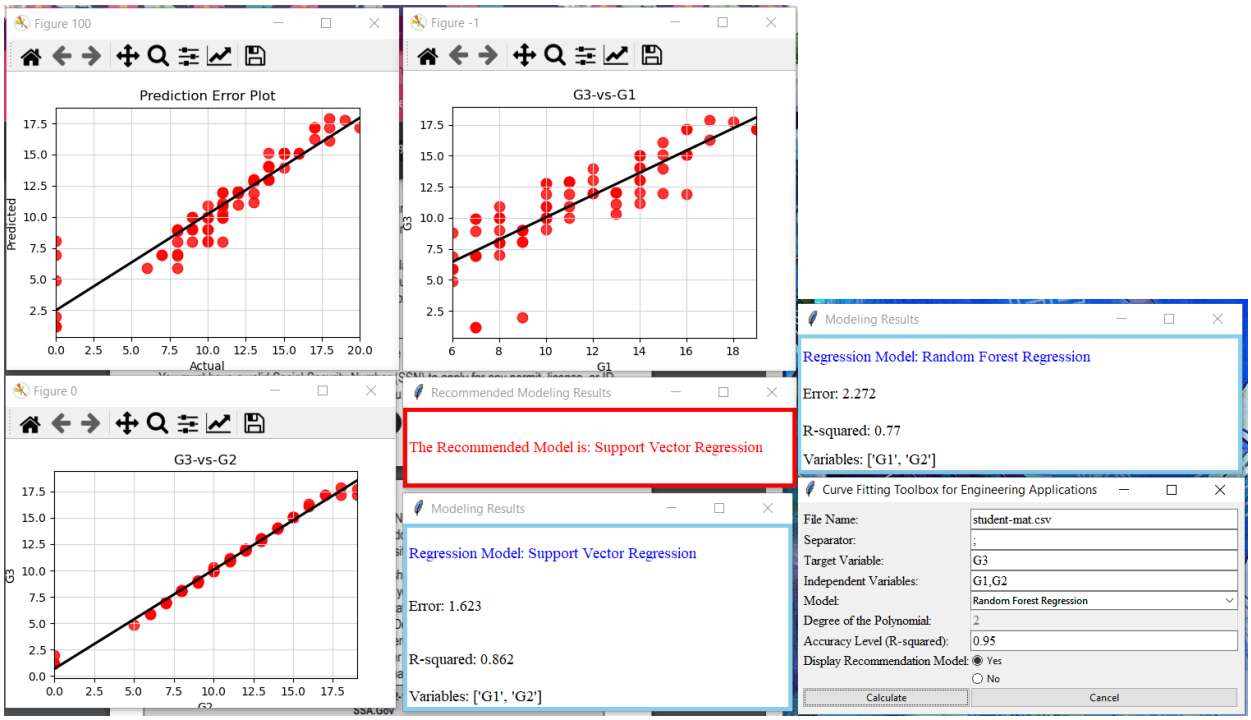


Case 2: The user enters a high threshold for the accuracy level

Displaying the selected model:

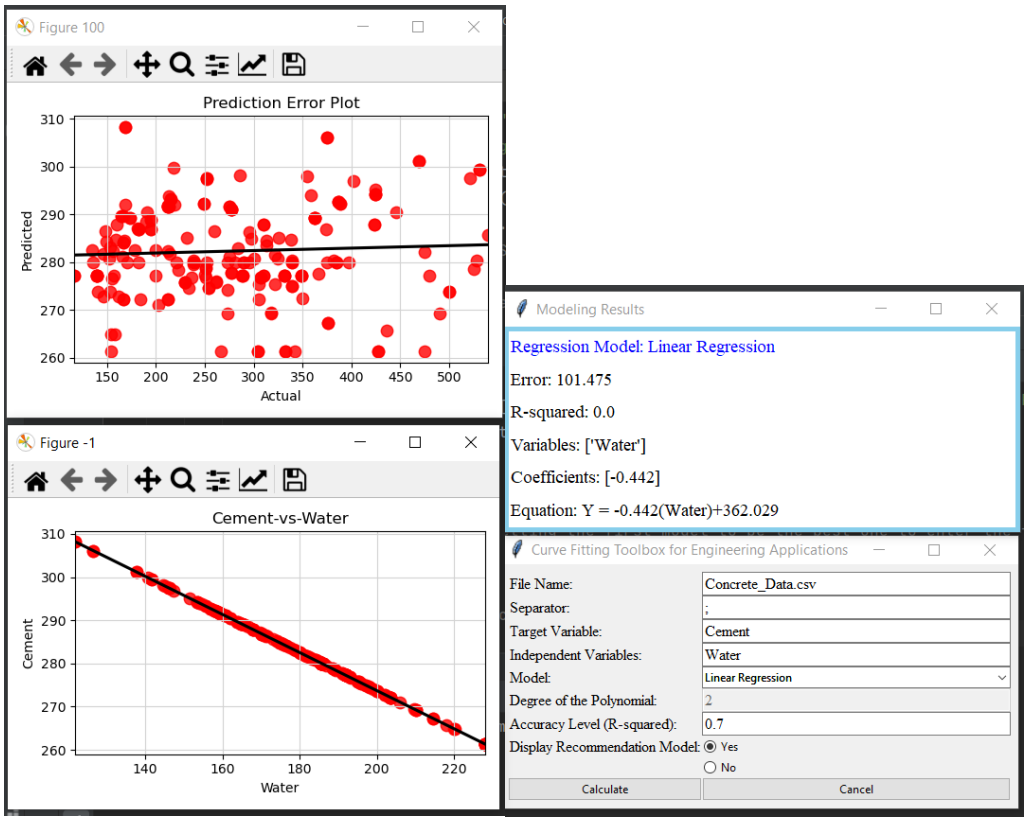


Displaying the recommendation model:

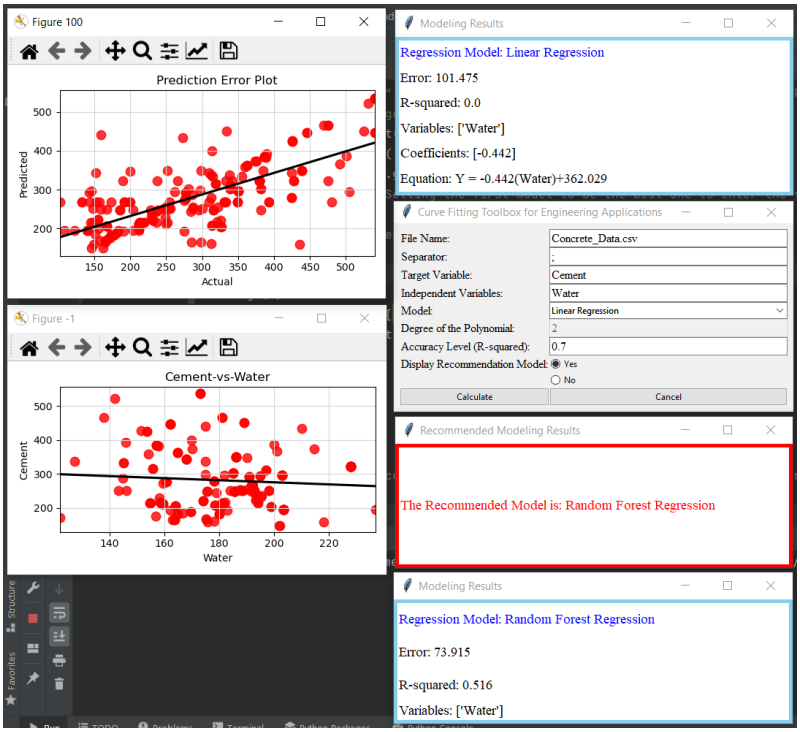


Case 3: Dataset with only one independent variable

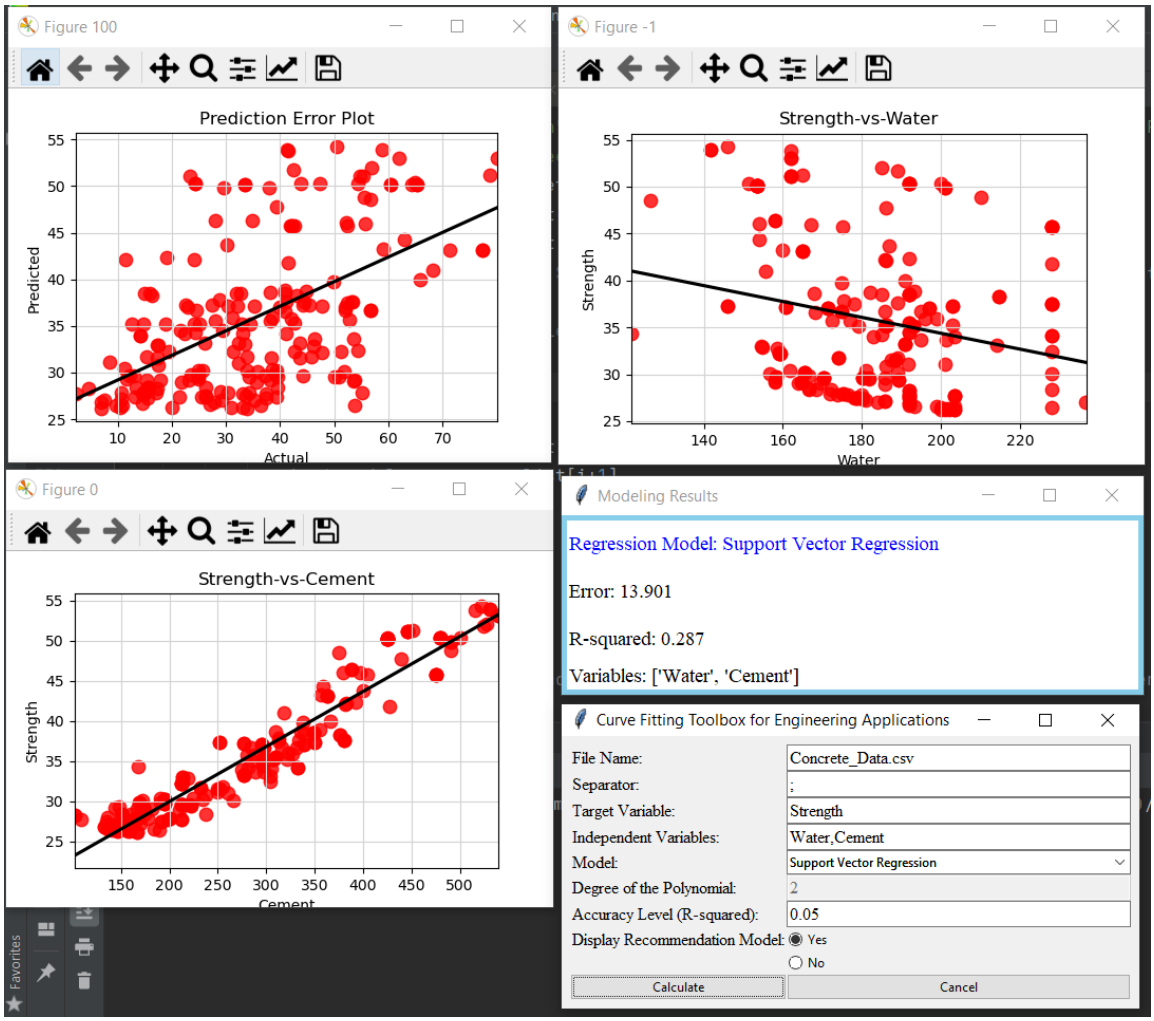
Displaying the selected model:



Displaying the recommended model:

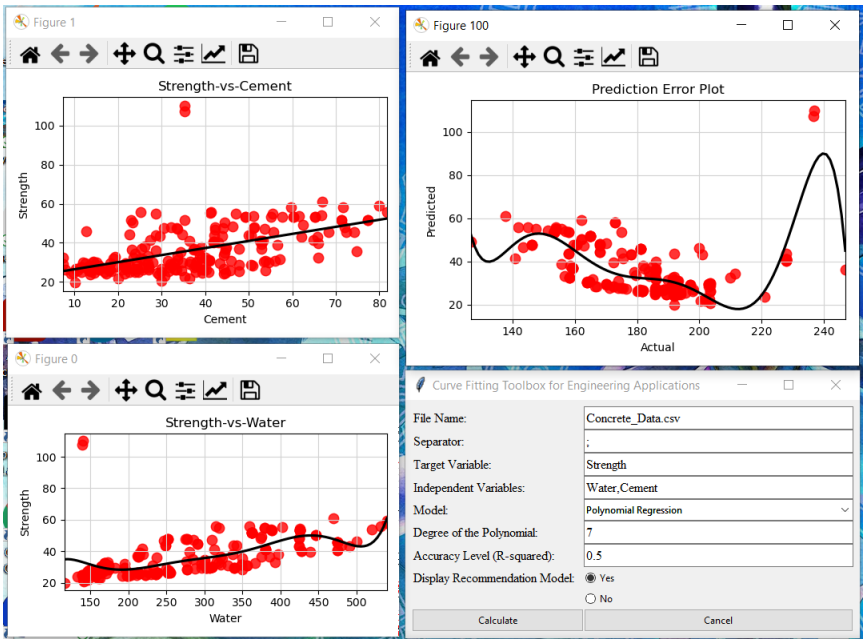


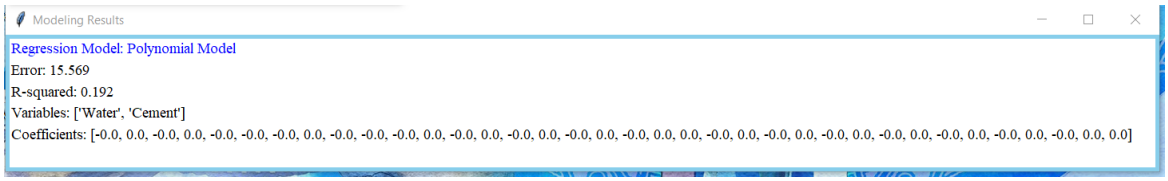
Case 4: The user inputs a low threshold for the accuracy level



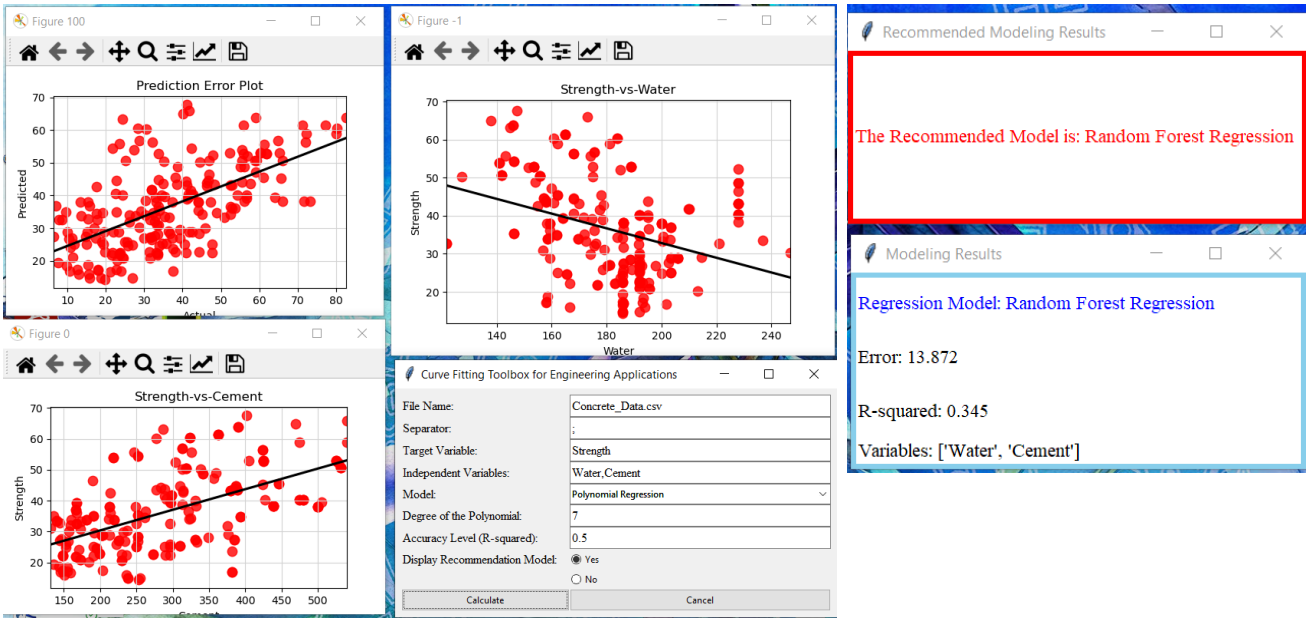
Case 5: The user selects a high degree for the Polynomial Regression

Displaying the selected model:





Displaying the recommended model:



LESSONS LEARNED

This project has been a great opportunity to improve the team members skills on programming, since it is the first experience they have with Python. In every step of the process, the team members managed to converge ideas and implement the data structures and algorithms learned during the class time. Some values like respect, responsibility and collaboration were put into practice along the way. It is important to say that there were also moments of frustration or uncertainty where the proposed approaches to the problem weren't resulting as expected. Nevertheless, this impulse the team's creativity and problem-solving skills, making the members think outside the box and perform deeper research on certain topics in order to come with an alternative solution.

On the developing of this project some new skills were gained. Firstly, logical thinking was one of the most used skills during the entire project since it is the base for the development of applications and algorithms. Next, new libraries like scikit-learn and seaborn were explored and implemented; as well as tkinter and matplotlib. Now the teammates have a better idea about the possible outcome that can be obtained with these. On the other hand, the team gained more expertise regarding machine learning basics and curve fitting since this was the theoretical background of the project. Finally, soft skills like communication, critical thinking, and adaptability were also put into practice.

In conclusion, with the creation of the final product all team members learned more about python programming, data structures and design of applications. Now they have a wider understanding about the requirements needed to develop computer programs. This has been a wonderful first step into the vast world of computer science.

CONCLUSIONS AND FUTURE WORK

Achieved tasks:

According to the proposed task table and timeline schedule, the performance of the team has been on time with the deadlines. All the five models proposed, and their algorithms have been developed and validated. Due to some encountered technical issues, two of the initial models where replaced. By suggestion of the instructor, the team added to the task table the implementation of a Graphical User Interface and graph deployments for the curve fitting models proposed. The following are the completed tasks:

- Algorithm generation for the Linear Regression Model and Polynomial Model.
- Algorithm debugging and Feedback comments and implementation.
- Algorithm generation for Ridge Regression, Random Forest Regression and Support Vector Regression
- Implementation of a Graphical User Interface as the final product
- Deployment of the corresponding graphs for the proposed models

Unachieved Tasks:

- Deployment of the different graphs in a single window

  This has not been completed since the implementation of this capability was provoking the graphs to decrease in size, obstructing the view of the datapoints

- Creating a combo box that displayed the datasets column names for the user to select the independent and target variables.

  The team could not implement this on the GUI due to technical issues encountered. This particular implementation resulted in an error indicator. The library used, tkinter, could not read the column names and immediately display them in a combo box since at the beginning the file name entry did not contain any value inside.

Future Work:

To resolve the encountered problems and implement the unachieved tasks the team's future plan is:

- Research on different python libraries similar to tkinter or other GUI creation programs, where there is an alternative manner to display the application and graphs created.
- Perform deeper research on the capabilities of tkinter, to investigate if there exists an alternative way to design the application

Additional ideas that the team has to improve this application:

- Giving the user the possibility to customize their own graphs with the use of buttons or other interactive elements
- Create a cellphone application with the same capabilities of the executed program
- Addition the option to display descriptive analytics and graphs of the input datasets
- Providing the program to run on a web environment, so it can be used by third parties without the necessity of running Python.