| **The Edward S. Rogers Sr. Department of**  **Electrical and Computer Engineering**  **University of Toronto** | | |
| --- | --- | --- |
| **ECE496Y Design Project Course**  **Group Final Report** | | |
| **Title:** Physics-based machine learning models for indoor wireless localization | | |
| **Team #:** | 2021415 | |
| **Team members:** | Name: | Email: |
| Prerna Anand | prerna.anand@mail.utoronto.ca |
| John Adrian Ambrad | john.ambrad@mail.utoronto.ca |
|  | Hiranya Maharaja | h.maharaja@mail.utoronto.ca |
|  | Deeksha Tewari | deeksha.tewari@mail.utoronto.ca |
| **Supervisor(s):** | Costas Sarris | |
| **Section #:** |  | |
| **Administrator:** | Tome Kosteski | |
| **Submission Date:** | 30th March 2022 | |

**Group Final Report Attribution Table**

This table should be filled out to accurately reflect who contributed to each section of the report and what they contributed. Provide a **column** for each student, a **row** for each major section of the report, and the appropriate codes (e.g. ‘RD, MR’) in each of the necessary **cells** in the table. You may expand the table, inserting rows as needed, but you should not require more than two pages. The original completed and signed form must be included in the hardcopies of the final report. Please make a copy of it for your own reference.

| **Section** | **Student Names** | | | |
| --- | --- | --- | --- | --- |
| Deeksha Tewari | Prerna Anand | Hiranya Maharaja | John Ambrad |
| Group Highlights | RD, MR, ET | RD, MR, ET | ET | ET |
| Individual Contribution | RD, MR, ET | RD, MR, ET | RD, MR, ET | RD, MR, ET |
| Acknowledgment | ET | ET | RD, MR, ET | RD, MR, ET |
| Executive Summary | RD, ET | ET | ET | ET |
| Introduction | RD, MR, ET | ET | ET | ET |
| Final Design | ET | RD, MR, ET | ET | ET |
| Testing and Verification | ET | ET | RD, MR, ET | ET |
| Summary and Conclusion | ET | ET | ET | RD, MR, ET |
| Project Management | ET | ET | ET | RD, MR, ET |
| Appendices | ET | RD, MR, ET | ET | ET |
| All | FP, CM | FP, CM | FP | FP |

**Abbreviation Codes:**

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The “**All**” row refers to the complete report and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information

RD – wrote the first draft

MR – responsible for major revision

ET – edited for grammar, spelling, and expression

OR – other

“All” row abbreviations:

FP – final read through of complete document for flow and consistency

CM – responsible for compiling the elements into the complete document

OR - other

If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:

OR1: enter brief description here

OR2: enter brief description here

**Signatures**

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

| **Name** | Prerna Anand | **Signature** |  | **Date:** | 03/30/2022 |
| --- | --- | --- | --- | --- | --- |
| **Name** | Hiranya Maharaja | **Signature** |  | **Date:** | 03/30/2022 |
| **Name** | John Ambrad | **Signature** |  | **Date:** | 03/30/2022 |
| **Name** | Deeksha Tewari | **Signature** |  | **Date:** | 03/30/2022 |

**Voluntary Document Release Consent Form**

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will not be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form.

**Consent Statement**

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

**Team #:** 2021415 **Project Title:** Physics-based machine learning models for indoor wireless localization **Supervisor(s):** Costas Sarris **Administrator:** Tome Kosteski

| **Name** | Prerna Anand | **Signature** |  | **Date:** | 03/30/2022 |
| --- | --- | --- | --- | --- | --- |
| **Name** | Hiranya Maharaja | **Signature** |  | **Date:** | 03/30/2022 |
| **Name** | John Ambrad | **Signature** |  | **Date:** | 03/30/2022 |
| **Name** | Deeksha Tewari | **Signature** |  | **Date:** | 03/30/2022 |

**Group Highlights (Author: Deeksha Tewari and Prerna Anand)**

The goal of our project is to develop a machine learning algorithm that uses RSSI data to predict with a high accuracy the indoor location of an object using signals over a WiFi network. After the progress report in January, the team has focused on testing and evaluating the most optimal machine learning model for this problem using parameter tuning.

We tested three models in parallel - the KNN, the ANN, and the Random forest. Prerna and Hiranya worked on the KNN custom model. Initially the model had an error of 4.99. After consulting with the TA, the team decided to add two more access points as data parameters. This helped improve the accuracy for all our models. After using the weight based algorithm the final hypertuned error of the algorithm was 4.08m, which was still above our threshold of 4m.

Deeksha worked on the ANN, using a sci-kit library model and hypertuning the number of epochs and hidden layers in the model to try and reduce the error. After using 5 access points, she was able to achieve an error of 5.17 m, which was way over our threshold. Normalizing the training and testing data did not significantly change the results obtained.

John worked on the Random Forest Regressor. Using the 5 access points data, he got an error of 2.77m with hypertuning. We evaluated the three models that were presented, and concluded that the random forest outperformed the other two models, and was selected by the team to be studied and hypertuned further. To this regard, the team has worked with normalizing the input data and studying outliers in the model’s predictions to better optimize it. While the normalization did not yield any significant improvements, studying the outliers has helped ascertain areas where the model is having difficulty

With respect to the data augmentation of our simulation data with collected data, we hit several roadblocks due to scheduling issues and, among other things, challenges posed by Covid-19. We already had a preliminary dataset that had been collected in November. The team noted that there were a few tweaks left to make to the raspberry pi power output. In February, after returning to campus, our supervisor recommended we focus on the machine learning model over in-person live data collection. In March, after selecting the model we were poised to begin data collection, but were seriously hampered by 3 out of 4 members falling sick due to Covid-19 and thus unable to come to campus.

In this final phase of the project, the team is currently studying the implemented random forest regressor, trying to optimize it by studying the predictions it is making and the outliers.

**Individual Contribution**

**Prerna Anand**

**Overview**

Post the progress report, I continued working on the K-Nearest Neighbor Model (KNN) and worked on new tasks which involved collecting data for 2 more access points and selection of the final design. After all models were implemented, they were evaluated on the same basis to select the ideal model.

**Tasks**

* **Data Collection:** Selected two more access points and simulated data for those points using the RayTracer. Final data contains RSSI values for 5 access points for training data set and testing data set. Adding the access points was favorable and brought down the mean absolute error to 4.18 m from 4.99 m. Post data collection, I worked on improving the prediction algorithm
* **KNN:** Implemented a weight based algorithm for the prediction function. The weight based algorithm sorts the 10 x-y coordinates based on their proximity to the actual x-y coordinates. The position which is closest to the actual position is given highest priority and furthest is given lowest priority. This algorithm was favorable and brought down the mean absolute error to 4.08 m from 4.18 m. Post the hypertuning, I worked with the team in selecting the final design.
  + An alpha value, a is selected between the range of 0 to 1 as a smoothing factor. I used an alpha of 0.18, which was selected based on hypertuning
* **Final Design:** The 3 models implemented were evaluated on the basis of their mean absolute error and scope of improvement. All 3 models were trained on the same training data set and were tested using the testing dataset to maintain consistency. Based on performance, the Random Forest was selected as the final design for the project

**John Ambrad**

**Overview**

After the progress report I was made to locate methods to improve the accuracy of the random forest regressor in 2 key ways. I worked with my colleague Deeksha Tewari to engineer the data that was generated by the simulator. Along with that I was also in charge of hypertuning different parameters of the random forest regressor to generate more accurate results.

**Tasks**

* **Data Engineering:** Working with Deeksha Tewari we sought different methods to modify or clean the data generated by the ray tracer simulator. We settled on the normalization method to modify our data
* **Hypertuning:** I wrote a script that modified numerous parameters of the random forest regressor in order to test out whether they would improve the mean absolute error. Such as the number of forests, depth, node split, and estimators. The script altered each hyperparameter one at a time and then tried different combinations of alterations to produce better accuracy results. In all we were able to reduce the error to 2.77 m

**Hiranya Maharaja**

**Overview**

After the progress report, I assisted John and Prerna in hypertuning the KNN and Random Forest Regressor respectively, and in selecting the final design. This involved introducing a weight based algorithm into the KNN instead of using a neighbor averaging function, as well as tuning parameters such as the total number of trees/forests, number of access points, and normalizing the input data in the Random Forest Regressor.

**Tasks**

* **KNN:** As mentioned in Prerna’s individual section, tuning the KNN involved the use of a weight based equation to calculate the most likely location of the static object, given the prediction of its possible neighbors. Previously, we had been calculating the average of the x and y coordinates of all the possible neighbors and using that as the final output prediction of the static object, which had a mean absolute error of 4.99 m
* **Random Forest Regressor Tuning:** This involved varying the number of forests from 2000 to 20,000, and increasing the number of simulated WiFi access points from 3 to 5. When combined with the other hyperparameters tuned by John, a value of 2000 trees and 5 access points yielded the lowest mean absolute error, at 2.769 m
* **Final Design:** As mentioned in Prerna’s individual section, the team evaluated the KNN, ANN, and Random Forest Regressor together based on their mean absolute error, and the strengths and limitations of each model. The Random Forest was chosen as the final design because it has the lowest error of the 3 models, and it can be improved upon in further explorations of this project

**Deeksha Tewari**

**Overview**

After the progress report, I assisted the rest of the team with hypertuning the ANN and random forest models to improve their accuracy, and live data collection using raspberry pis.

**Tasks**

* **ANN Implementation and Final Model Selection:** To reduce the mean squared error observed in our trials (5.13 ft), I tried to modify the values for the number of hidden layers, epochs, and number of neurons per layer. These results were illustrated in our Final Demo, and vastly overshot our limit of 4ft error between the predicted location and the physical location. Therefore, together with the team, we decided to focus our efforts on hypertuning the random forest regressor which provided more promising results.
* **Hyperparameter Tuning for Random Forest:** I worked with John and Hiranya to develop normalization scripts for the model’s testing and training data. Regression models benefit from normalization to prevent huge anomalies or deviations in the data, however because our problem was not a classification problem where the. Prerna and I consulted with the TA and supervisor, and realized that we should add more parameters to the machine models by way of increasing the number of Access Points to 5. This helped reduce the mean squared error of the random forest algorithm. I am currently working on understanding the points in the dataset where the predictions of the algorithm are incorrect above a certain threshold, and trying to ascertain the cause for this to further optimize our algorithm
* **Data collection and Augmentation using Raspberry Pis:** This aspect of our project was severely hindered by lockdown as well as loss of manpower due to Covid. Since all the team-members fell ill, we were unable to complete data collection during March. And after consulting professor Sarris, we decided that data augmentation using live data is not critical to the model. Instead we are currently focusing our energy into understanding our model.

**Acknowledgements (Author: Hiranya Maharaja and John Ambrad)**

We would like to acknowledge our supervisor, professor Costas Sarris, for offering us the opportunity to participate in this project and explore various ML concepts throughout the year. As well, we thank him for offering us valuable guidance and feedback.

Next, we would like to thank Aristeidis Seretis for teaching us how to use the ray tracer, and how to run and interpret the results of the convergence analysis. As well, we thank him for his helpful suggestions, such as adding access points, and working with supervised learning models. These suggestions were greatly useful in improving the accuracy of our model, and without him, we would not have been successful in reaching our goal for this project.

We would like to thank professor Tome Kosteski for his guidance. He added valuable comments to our presentation style and to our presentation slides that helped us improve.

We would like to thank communication instructors for reviewing over our documents and helping us in our engineering communication. They helped us word documents in a more concise and professional manner, fixed our vocabulary and spelling, and made sure that our terminology was consistent.

**Executive Summary**

Indoor localisation tools and software provide the x and y coordinate of the target within a space. Like outdoor localisation (which uses GPS), indoor localisation has great potential for facilitating indoor navigation and tracking. Since outdoor localisation strategies fail indoors (due to non line of sight issues in buildings), literature points to the usage of communication signals like WiFi to perform robust indoor localisation using existing architecture. The goal of this project is therefore to develop a machine learning model that uses Received Signal Strength data (RSS) from WiFi access points to predict the position of an object indoors. We constrain ourselves to achieving localisation on the 8th floor of the Bahen Center. Originally, we were also planning to augment the simulation data with real life RSSI data collected from Raspberry pi APs by a handheld device, however due to illness of team-members and the professor’s guidance to focus on the model this was not done.

The project is divided into two parts: data collection and localisation. Data collection was done by programming the provided raytracer simulator with receivers at uniform intervals and access point transmitters. Scripts were created to clean the data and split the dataset into a testing and training dataset. We then create a data structure with the RSSI strengths from each access point at each location on the map. Multiple model architectures were explored for performing localisation - K-Nearest Neighbors, Artificial Neural Networks, and Random Forest. After hyper-parameter tuning and adding more access points, we found the Random Forest yielded the best localisation accuracy with 2.77m.

The team chose to perform hyperparameter tuning for the Random Forest model by performing normalization and by investigating outlier points and their impact on the overall accuracy of the system. While normalization has not resulted in any significant changes in accuracy, the investigation into the outliers is still ongoing.

**Table of Contents**

[**Introduction (Author: Deeksha Tewari)**](#_ayhmzlxh1ai0) **14**

[1.1 Background and Motivation (Author: Deeksha Tewari)](#_siwok3ow07pd) 14

[1.2 Project Goals and Requirement (Author: Deeksha Tewari)](#_h87v4c9g2233) 15

[1.2.1 Goal](#_p0lbjsk9whdc) 15

[1.2.2 Requirements](#_hnffvgymm9fa) 15

[**Table 1: Project Requirements**](#_pp0nvv2of1ej) **16**

[**2. Final Design (Author: Prerna Anand)**](#_co5ith6zxvbe) **16**

[2.1 System-level overview](#_lpn9p4isme6) 16

[2.2 System Block Diagram](#_hsczm9sdzozg) 17

[2.3 Data Collection](#_bsg3fa5jmeeg) 18

[2.4 Data Cleanup](#_r72o712wp2yn) 19

[2.5 Localization](#_cu9uwy6lc3aq) 20

[2.6 Model Comparison](#_e45e5obe3zek) 22

[**3. Testing and Verification (Author: Hiranya Maharaja)**](#_qnfqv9xqgmbx) **23**

[3.1. Location in x-y coordinates of a single target given a WiFi signal strength](#_6xtvi64j1f8a) 24

[3.2. Create RSSI WiFi fingerprint map using ray tracer simulations](#_ihox54me1yau) 24

[3.3. Map of Bahen 8th floor layout for producing WiFi RSSI fingerprinting](#_t0oz3n8mo2l) 25

[3.4. Raspberry Pi access points (APs) configured separately in Bahen](#_7yeezp840cwz) 26

[3.5. Augment the training data](#_h3hi81e4pmxk) 27

[3.6. Prediction error should be < 4m](#_jejwanbr4k8b) 27

[**4. Summary and Conclusion (Author: John Ambrad)**](#_23xtlr92x1l7) **29**

[**5. Project Management (Author: John Ambrad)**](#_7l17w6j7e671) **30**

[5.1 Gantt chart](#_ci9gzb3zxsmh) 30

[5.2 Financial Plan and Budget](#_g8kgeq3mvp83) 30

[**6. References**](#_zg2bfuyk2n7c) **31**

[**7. Appendices**](#_3egned4b6fnk) **32**

[Appendix A : Simulation of data points](#_r6jtcxompb4) 32

[Appendix B : Data clean up](#_us0ayf70xz4r) 33

[Appendix C : Convergent Analysis](#_ba6fxjnd1e7n) 35

[Appendix D : K Nearest Neighbor Model (KNN)](#_8rz1bzpvs4uq) 36

[Appendix E : Artificial Neural Network Model (ANN)](#_jrfc14dfzw00) 38

[Appendix F : Random Forest Model](#_azyuujhvr5fn) 39

# **Introduction (Author: Deeksha Tewari)**

This report summarizes the motivation, design, and testing of physics based models for WiFi based localisation as part of our final year design course.

## **1.1 Background and Motivation (Author: Deeksha Tewari)**

Indoor localisation systems help locating a person or object within a building or other enclosed spaces. Its applications range from helping firefighters conduct search and rescue in an emergency, to customizing discounts based on the location of a customer, to allowing indoor robot assistants to map their surroundings better [1]. While Global Positioning System (GPS) technologies that allow for wayfinding for cars and people in outdoor settings, these technologies fail in indoor environments (buildings and other enclosed spaces) due to non-line-of-sight issues [2].

Wi-Fi signals and access points (APs) are ubiquitous indoors and therefore make ideal candidates for performing indoor localisation with [3] , and therefore localisation can be performed without architectural modifications. A common technique used in WiFi localisation is the use of fingerprinting, where the **Received Signal Strength Indicator (RSSI)** from an AP is measured at a specific point in the space, and the model is trained on WiFi Signal strengths in dB versus the location it was measured it. Physics based simulations can provide WiFi strengths that are very close to reality, however these simulations are computationally expensive (take a long time and a lot of compute resources). Therefore less accurate data can help us strike a balance between accuracy as well compute resources by training a machine learning model on this data.

Two teams before our own have tried to use ML models to these problems - however the models produced are either lacking in accuracy or require a high cost to implement

## **1.2 Project Goals and Requirement (Author: Deeksha Tewari)**

### **1.2.1 Goal**

To use machine learning and low-fidelity RSSI data from physics-based simulations, to create a model that can predict with high accuracy the indoor location of an object/person within a fixed indoor environment.

### **1.2.2 Requirements**

The following table highlights the functional requirements, objectives, and constraints in our design. Details about how each of these requirements was satisfied is covered in more detail in Section 3 - Testing and Verification.

| **ID** | **Project Requirement** | **Description** |
| --- | --- | --- |
| 1 | Location in x-y coordinates of a static target given a WiFi signal strength (RSSI) | **Functional Requirement:** this is the output specification of the project. |
| 2 | Create RSSI WiFi fingerprint map using ray tracer simulations | **Functional Requirement:** Using the Bahen .stl map create a low fidelity RSSI map over a set of defined receiver points from defined access points. |
| 3 | Map of Bahen 8th floor layout for producing WiFi RSSI fingerprinting. | **Constraint:** Other buildings have thicker walls which will interfere with the collection of accurate data. |
| 4 | Raspberry Pi access points (APs) configured as WiFi APs separately from APs in Bahen. | **Constraint:** We cannot run the risk of disrupting UofT routers and need to test using our own. |
| 5 | Augment training data | **Objective:** Live RSSI values collection using raspberry pi transmitters and laptop/phone receivers to be done to augment the low fidelity data with live measurements |
| 6 | Predictions should be less than 4 meters from the actual target | **Objective:** this would be an improvement on the previous teams’ margin of error. |

# **Table 1:** Project Requirements

# 

# 

# 

# 

# 

# 

# 

# **2. Final Design (Author: Prerna Anand)**

## **2.1 System-level overview**

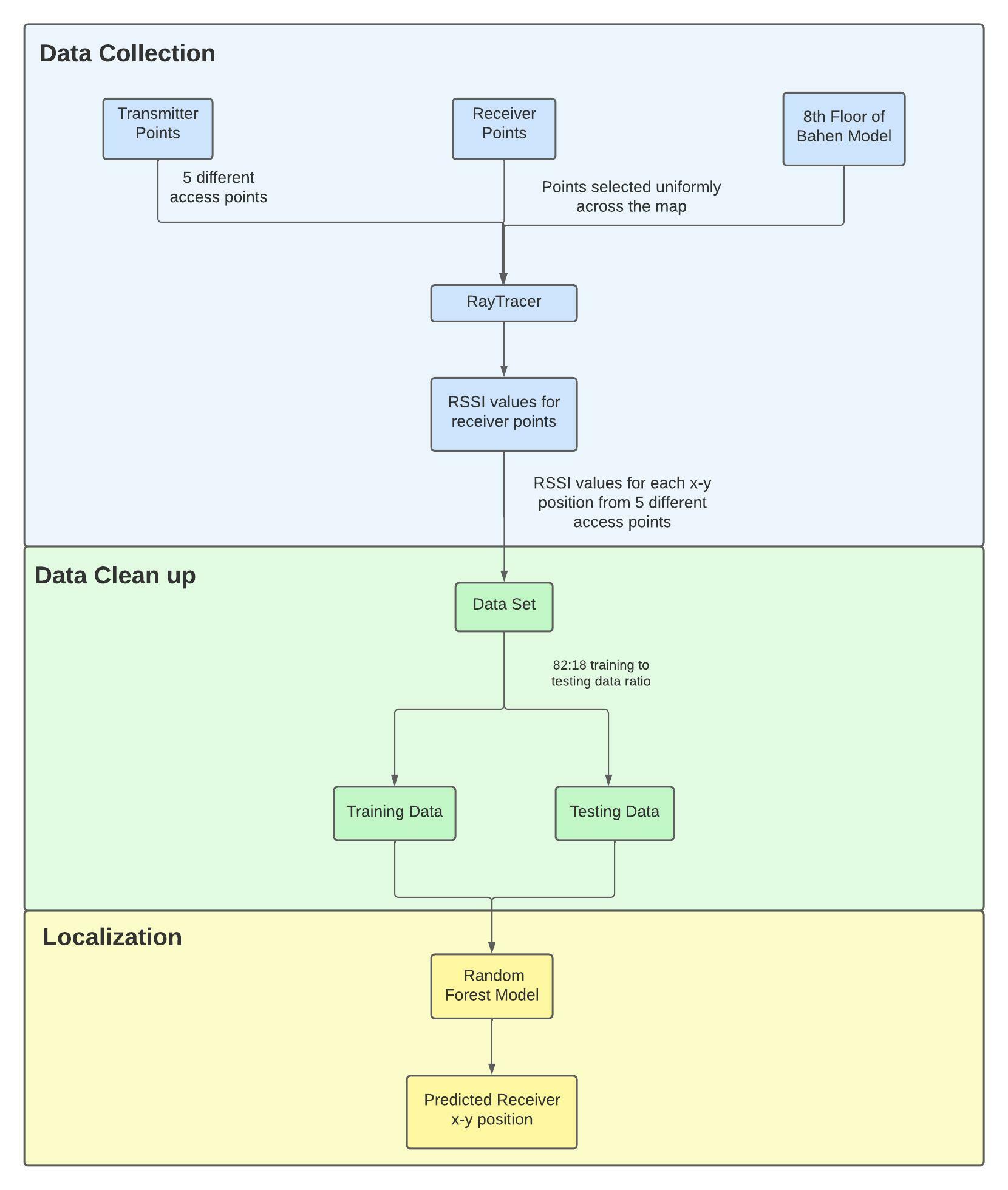
The project is divided into two parts: data collection and localization using machine learning.

The goal of the data collection is to collect RSSI values at uniform points across the 8th floor of Bahen, which are simulated using the RayTracer. The RSSI values are collected for multiple access (transmitter) points located at different positions. This helps in accounting for changes caused by walls, corners or any other obstacles by measuring the RSSI value from different directions.

After collection, the RSSI values and their x-y positions are used as part of the machine learning model (ML). The ML model learns the pattern of the data using the training data set which includes the RSSI values and x-y position. Post training, it takes a RSSI value from the testing data set as the input and accurately predicts its location on the map.

The next sections will describe the final design, focusing on why it was chosen based on its strength and limitations.

## **2.2 System Block Diagram**

**Figure 1:** System block diagram

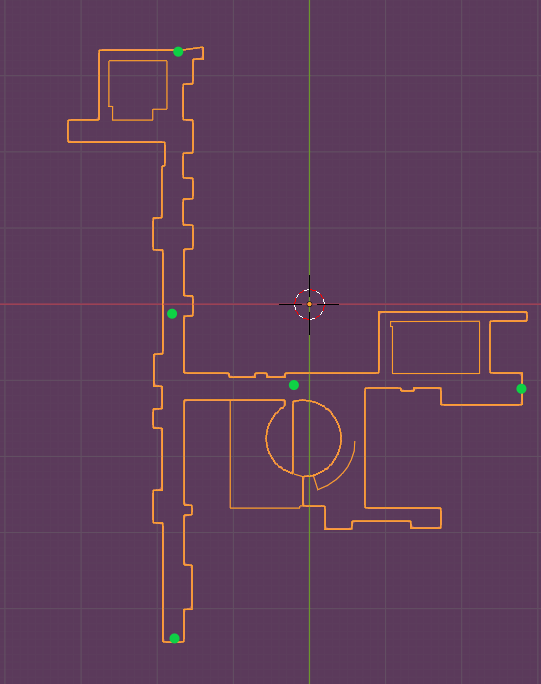
## **2.3 Data Collection**

The RSSI values needed for the localization were obtained using the RayTracer simulator provided by the supervisor. The RayTracer takes in 3 inputs, a 3D model of the 8th floor of Bahen, list of receiver x-y position on the map and specific configuration. It outputs an excel file containing the RSSI value for each receiver x-y position.



**Figure 2:** RayTracer simulator inputs and outputs when run 5 times

* The 3D model was provided by the supervisor as a stl file of the 8th floor
* To calculate the receiver points a python script was created. The script lists down x-y positions across the map which are uniformly spaced. Two different configurations were used to create receiver points for training and testing data set (Appendix A)



**Figure 3:** Bahen stl file with 5 access points

* The specific configuration includes the access positions and number of reflections
  + - Access x-y positions were selected based on geometry
    - Convergence analysis was performed by creating a script to find the point number of reflections. RSSI values for different numbers of reflections were analyzed to identify the point of convergence. (Appendix C)

The RayTracer was run 5 times using the same configurations and receiver points for training dataset and again run 5 times with different receiver points for testing data set.

## **2.4 Data Cleanup**

The next step is to combine the excel sheets obtained from the RayTracer. This was achieved by creating a python script.

Each excel sheet for training would differ slightly as with specific access points, certain receiver points are not considered due to geometry or obstacles. To account for this the 5 excel files for training data are combined based on the x-y position and ensure that positions are considered only if they have 5 RSSI values. Similarly the testing data is also cleaned up.

After cleanup, the training data set has 2657 points and the testing data set has 593 points with 5 RSSI values for each point corresponding to its access point (Appendix B)

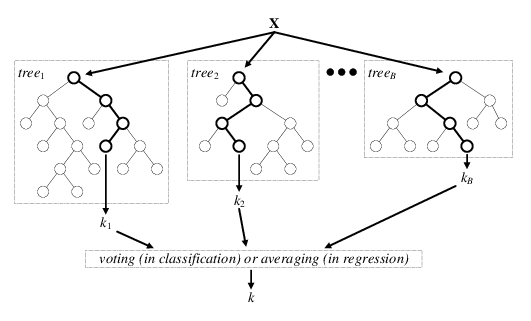
## **2.5 Localization**

The goal of the localization model is to determine the target’s x -y coordinates in the specified

environment using the RSSI values input. Multiple model architectures were explored and the Random Forest Model was selected as the final design (Appendix F)

The Random Forest Model uses the technique of bagging and features randomness to find a correlation between the data which are made into decision trees. Our model was implemented using the scikit library’s Random Forest Regressor model. It was hypertuned to have 2000 decision trees with a depth of 100. The Random Forest model performs localization in two steps:

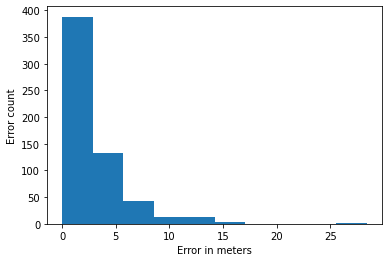
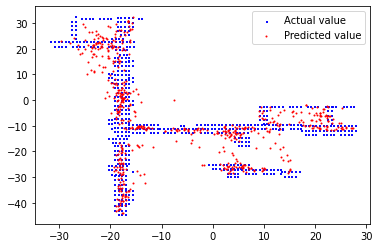
* **Training :** The model creates 2000 decision trees with a maximum depth of 100. Once the trees are created it fits the training data into the trees based on pattern recognition. The training data includes the 5 RSSI values and x-y coordinates for each receiver point



**Figure 4 :** Random Forest [4]

* **Prediction :** After the model finishes training, the testing data is used to check the accuracy of the prediction using scikit library’s prediction and mean absolute error functions. The prediction function analyzes the RSSI values and finds the closest x-y coordinate from relevant trees and then returns the average of the result as the predicted x-y coordinates. After the prediction, the mean absolute error finds the margin of error between the predicted and actual value

The team decided to choose this model as it has the lowest mean absolute error of 2.77 m and it can be improved with a custom model in the future.

**Figure 5:** Scatter plot of actual & predicted values **Figure 6:** Histogram of mean absolute error

The scatter plot describes the accuracy of the model on a visual scale. The actual values (blue) form the shape of the 8th floor of Bahen and the predicted values (red) can be seen on and around it. On the other hand, the histogram describes the error density lies between 0 to 4 m with few outliers which makes it the ideal choice.

At the same time, this model comes with a limitation. The error can only be improved using a custom model which is difficult to implement from scratch and also requires high computation power.

## **2.6 Model Comparison**

The Random Forest model was the preferred model for indoor localization. Two other models were also considered the Artificial Neural Network and K - Nearest Neighbor Model. The table below compares the different models that were considered.

| **Model Architecture** | **Description** | **Mean Absolute Error** |
| --- | --- | --- |
| Artificial Neural Network Model (Appendix E) | ANN with 2 layers. The 1st layer has 150 nodes and the 2nd layer has 10 nodes. Trains on the 2657 training points with 5 access points | 5.13 m |
| K Nearest Neighbor Model (Appendix D) | KNN with K = 10 (closest neighbors), 5 access points and 2657 training points | 4.08 m |
| Random Forest Model | Random Forest with 2000 decision trees with maximum depth 100 trained on 2657 training points (x-y coordinates and 5 RSSI values) and 5 access points | 2.77 m |

**Table 2:** Model Comparison

Based on comparison the KNN and ANN model both have a high localization error and therefore were not considered as the final model. The easy availability and simplicity of the scikit library’s Random forest regressor made it an ideal choice. The training time of the model is comparatively longer than the other models but the low mean absolute error outweighs that.

# **3. Testing and Verification (Author: Hiranya Maharaja)**

Each component of our design is evaluated based on the requirements defined in Section 1b. An overview of each of these components, as well as the testing methods and results, is provided in Table X, with further details about each row in the table being presented below.

| **ID** | **Project Requirement** | **Description** | **Proof/Results** |
| --- | --- | --- | --- |
| 1 | Location in x-y coordinates of a static target given a WiFi signal strength (RSSI) | **Functional Requirement:** this is the output specification of the project. | Pass - see section 3.1 |
| 2 | Create RSSI WiFi fingerprint map using ray tracer simulations | **Functional Requirement:** Using the Bahen .stl map create a low fidelity RSSI map over a set of defined receiver points from defined access points. | Pass - see section 3.2 |
| 3 | Map of Bahen 8th floor layout for producing WiFi RSSI fingerprinting. | **Constraint:** Other buildings have thicker walls which will interfere with the collection of accurate data. | Pass - see section 3.3 |
| 4 | Raspberry Pi access points (APs) configured as WiFi APs separately from APs in Bahen. | **Constraint:** We cannot run the risk of disrupting UofT routers and need to test using our own. | Pass - see section 3.4 |
| 5 | Augment training data | **Objective:** Live RSSI values collection using raspberry pi transmitters and laptop/phone receivers to be done to augment the low fidelity data with live measurements | Pass - see section 3.5 |
| 6 | Predictions should be less than 4 meters from the actual target | **Objective:** this would be an improvement on the previous teams’ margin of error. | Pass - see section 3.6 |

**Table 3:** Project Requirements

## **3.1. Location in x-y coordinates of a single target given a WiFi signal strength**

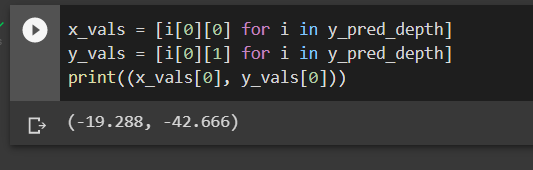
**Functional Requirement**

This is the output specification of the project. Given the RSSI values from the WiFi access points, the final model should output a set of x-y coordinates as a prediction for where the static object is located.

**Verification:**

The RSSI values generated by running the convergence analysis on the ray tracer are included in the CSV fed into the model as the input, and the final training/testing output is a set of x-y coordinates that can be compared to the x-y labels generated by the ray tracer. The requirement is directly observable through these properties in our model.

**Proof:**

****

**Figure 7:** Prediction of a single point of location on the x-y plane, negative because it is below and to the left of the defined origin. Refer to Figure 5 above to see all the predicted values plotted on the 8th floor of Bahen.

## **3.2. Create RSSI WiFi fingerprint map using ray tracer simulations**

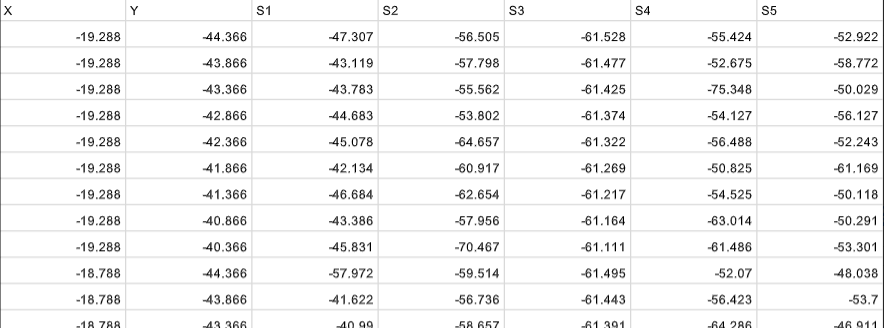
**Functional Requirement:**

The received signal strength can be calculated using the known WiFi access points on the interior of the desired building/floor, which would be the 8th floor of Bahen building in our case. As outlined in section 1.1, other methods of data collection are expensive, and have large training requirements, whereas WiFi data only needs to be collected and fed into the model itself.

**Verification:**

This constraint is directly measurable by analyzing the team’s data collection process and the values taken in by the model as input. The training, testing, and validation data were generated by running a convergence analysis on the Bahen .stl files that were provided to the team by the supervisor. These files provided a view of the 8th floor of Bahen, and the team placed up to 5 access points using them, as well as created a script to process the x-y coordinates and RSSI values into a CSV that could be fed into any of the three models our team considered.

**Proof:**

****

**Figure 8:** CSV of the final training data, with the last 5 columns corresponding to RSSI values in dB for the chosen 5 access points. Refer to appendix C for the convergence analysis and appendix C for the CSV generation code.

## **3.3. Map of Bahen 8th floor layout for producing WiFi RSSI fingerprinting**

**Constraint:**

Because other buildings have thicker walls and different reflection coefficients, collecting data from them and running accurate convergence analyses of their values is infeasible. Therefore, the team is limiting itself to collecting data from the 8th floor of Bahen to ensure accuracy.

**Verification:**

This constraint was verified through a review of the design. Throughout the process, the team used only the bahen .stl files provided by the supervisor to generate the simulated dataset, which is a map of the 8th floor of Bahen. Thus, the final model is trained and tested on WiFi RSSI values of the 8th floor of Bahen, ensuring its accuracy.

**Proof:**

See Figure 3 above.

## **3.4. Raspberry Pi access points (APs) configured separately in Bahen**

**Constraint:**

UofT has its own APs on the 8th floor of Bahen, which we cannot risk disrupting or jamming in the event of an error. However, we still need APs to collect and test real-world data. To do this, we would need to transmit a WiFi signal as the ‘source’, which can then be used to calculate RSSI values to feed into the model as input, and so the team built a transmitter with the raspberry pi to serve this purpose. The raspberry pi transmitter was used to gather 12 real-world data points from the 8th floor of Bahen.

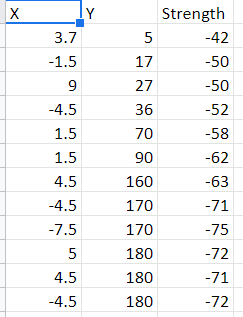
**Changes:**

Due to covid-19 restrictions, the team only collected 12 points with the raspberry pi transmitter, and was unable to collect anymore. These 12 WiFi signals were verified to yield sensible RSSI values that could be used as input to the model, but needed to be adjusted to be compatible with the RSSI values generated using the ray tracer. As a result, the supervisor advised the team to solely focus on using the ray tracer dataset, and only create a separate, real-world dataset if time permits. However, the team was unable to use the transmitter to finish the real-world dataset after contracting covid-19. More details on the verification and adjustment are presented below.

**Verification:**

This requirement was tested and verified with direct measurement. The raspberry pi transmitter was able to collect sensible WiFi signals and convert them to RSSI values. However, as mentioned in section 1.2, the team used low fidelity data (low number of reflections) to reduce the computational costs of running a convergence analysis. Due to this, the simpulated RSSI values required an adjustment on the dB scale to be compatible with the raspberry pi transmitter RSSI values, and so the two datasets could not be integrated. The transmitter can be used in the future to collect more real-world data and re-train the model based on the new dataset.

**Proof:**

****

**Figure 9:** Calculated RSSI values in dB for 12 real-world measured data points from the 8th floor of Bahen

## **3.5. Augment the training data**

**Objective:**

There are differences between simulated ray tracer data and the real world, mainly the reflections. Using live, real-world data collected with the raspberry pi transmitter as part of the training data fed into the model allows it to form connections in the data that can account for some of these differences, and generalize to other real-world scenarios better than a model trained on just simulated ray tracer data.

**Verification:**

This objective was verified with the same direct measurement as in section 3.4. The live/real-world data points collected were sensible, but not used in training the final model for the same reasons.

**Proof:**

See proof for section 3.4 above.

## **3.6. Prediction error should be < 4m**

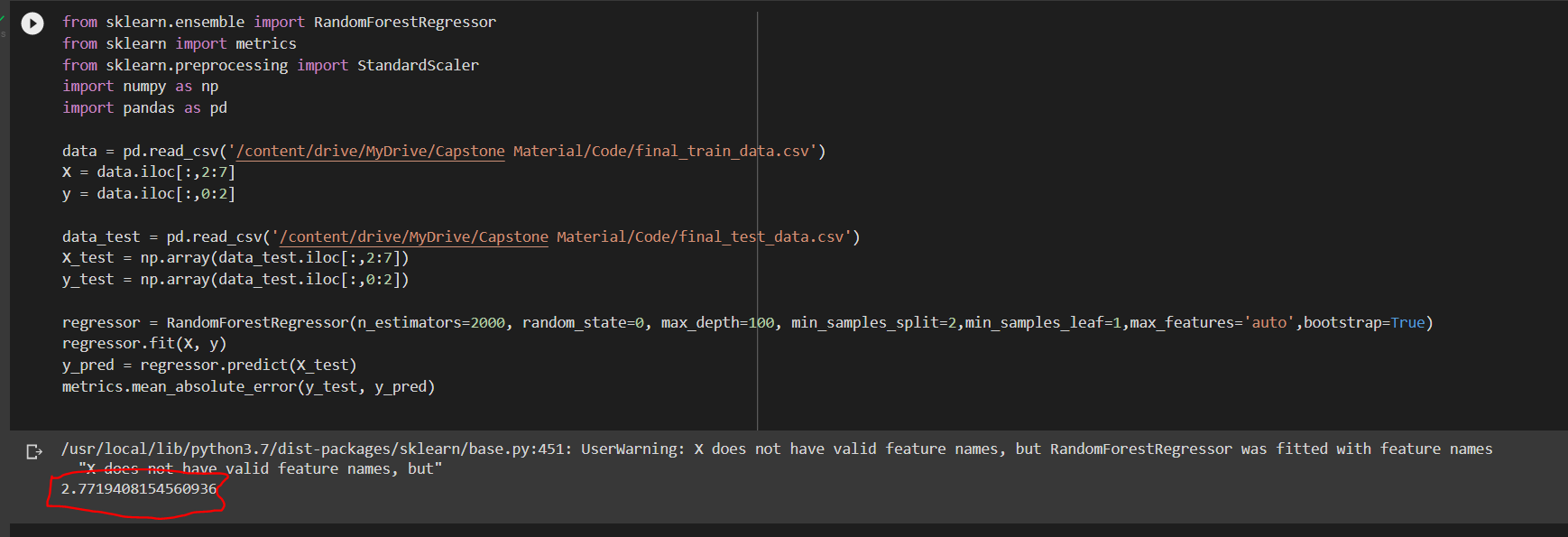
**Project objective:**

Mean Absolute Error (MAE) that is under 4 metres would be an improvement upon the margin of error obtained by the previous year’s team. The MAE refers to the average of the absolute errors between each (x-y) prediction tuple and the label tuple, meaning that our model’s margin of error in either the x or y direction is equal to the MAE.

**Verification:**

The three models that were tuned and evaluated are described in section 2.6, and the Random Forest Regressor that was chosen to be our final model has a MAE of 2.77 meters. This is directly measurable to be under 4 meters.

**Proof:**

****

**Figure 10:** The final MAE of the Random Forest Regressor is circled in red at the bottom.

# **4. Summary and Conclusion (Author: John Ambrad)**

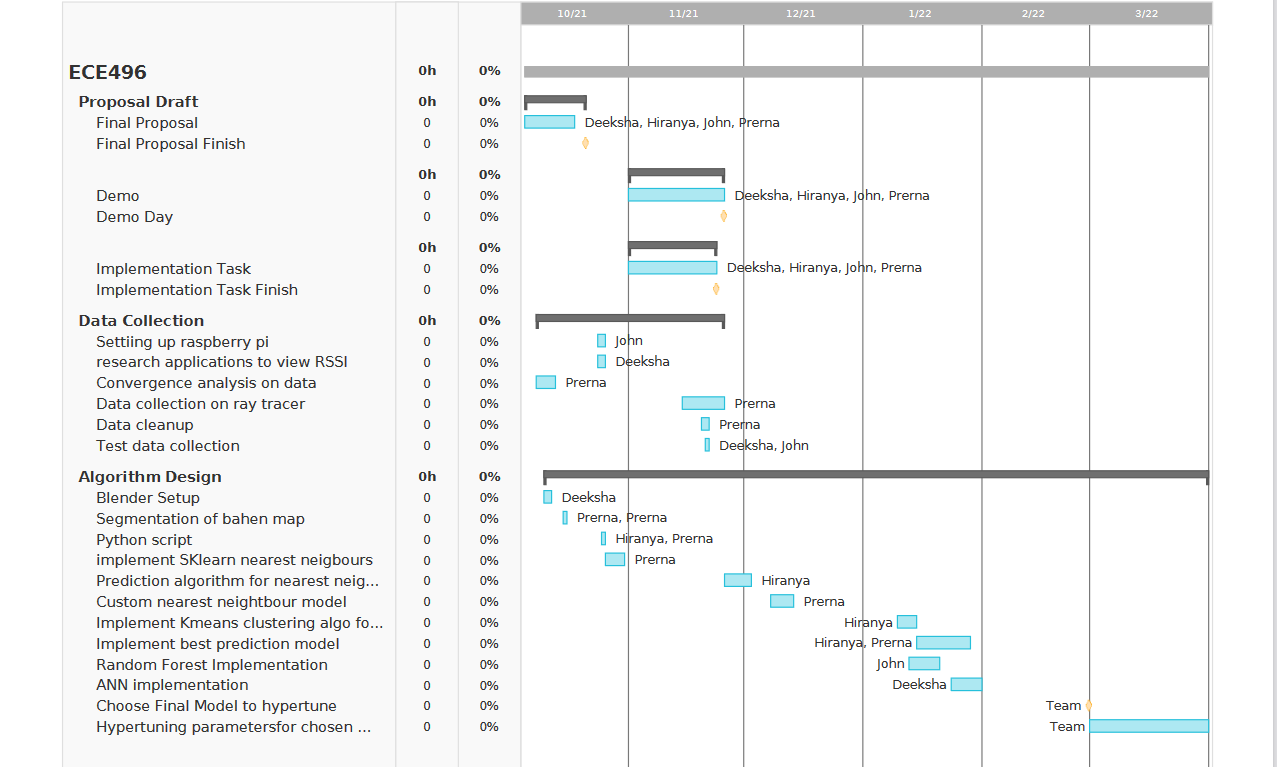
Our project has made headway in attempting to overcome the obstacles of indoor localization. Other solutions can range from being too costly or being too inaccurate to be effective. Our project attempts to bridge the gap between cost and accuracy by leveraging the near ubiquitous WiFi signals that surround us. The project has achieved an accuracy of 2.77 m which exceeds the target accuracy of within 4 m. As we observe from the results of our testing data we can conclude that the project has achieved all of the testing and verification parameters. The data obtained displays that we are below the expected threshold set by Professor Costas. Our testing and verification work proves the accuracy of our final design as well as the low cost as compared to alternatives.

Given that our work has a low cost we could see this design used in different industries.

We believe our work can be utilized in search and rescue operations, business analytics and marketing. We also believe our work is one of the highlights that could be analyzed in some academic work on the effectiveness of different machine learning methods with regards to interpolation/regression problems.

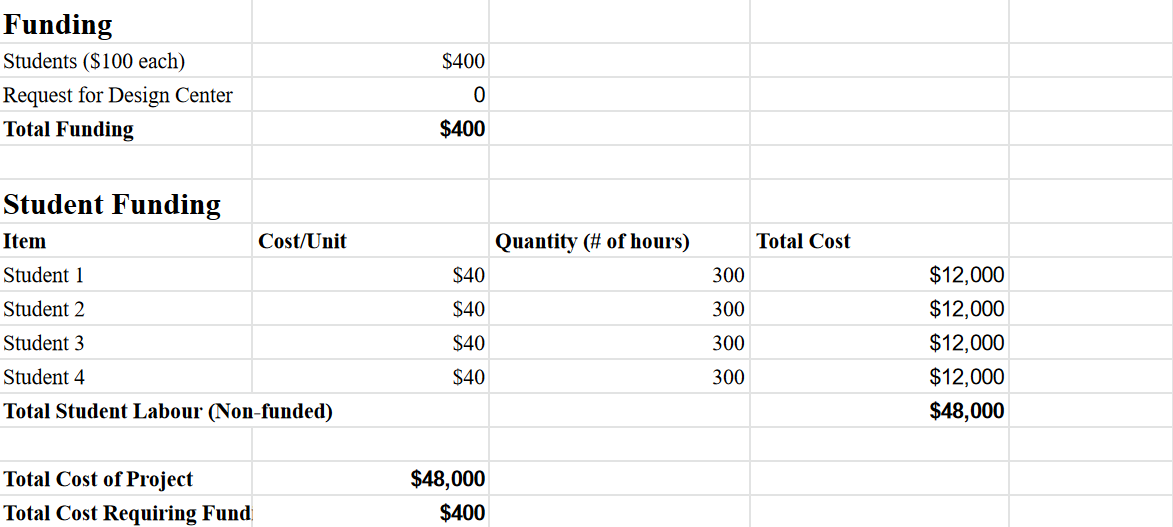
# **5. Project Management (Author: John Ambrad)**

## **5.1 Gantt chart**



**Figure 7:** Final Gantt chart

## **5.2 Financial Plan and Budget**



**Figure 8:** Budget table for the team

# **6. References**

[1] Roy, P., Chowdhury, C. A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems. J Intell Robot Syst 101, 63 (2021). <https://doi.org/10.1007/s10846-021-01327-z>

[2] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali, “Indoor positioning and wayfinding systems: a survey,” Human-centric computing and information sciences, vol. 10, no. 1, 2020, doi: 10.1186/s13673-020-00222-0

[3] S. Lee, J. Kim, and N. Moon, “Random forest and WiFi fingerprint-based indoor location recognition system using smart watches,” Human-centric computing and information sciences, vol. 9, no. 1, pp. 1–14, 2019, doi: 10.1186/s13673-019-0168-7

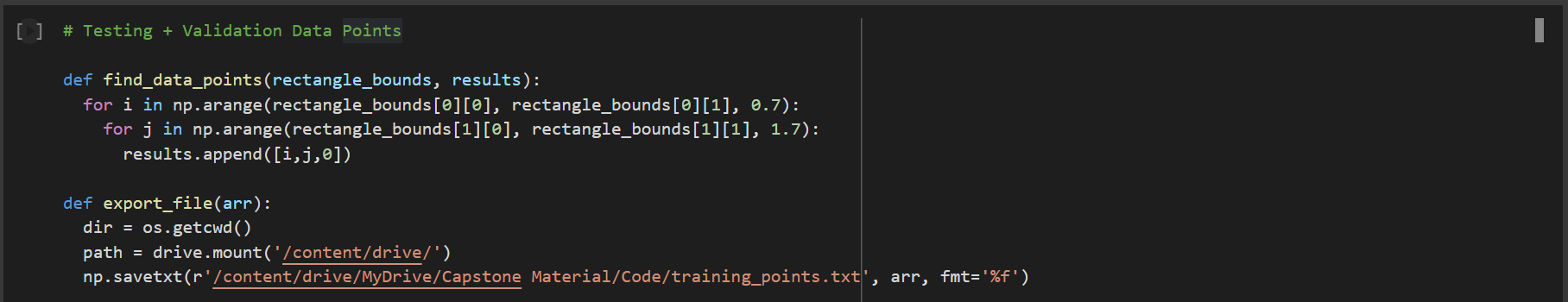
[4] A. Verikas, E. Vaiciukynas, A. Gelzinis, J. Parker and M. Olsson, "Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness", *Sensors*, vol. 16, no. 4, p. 592, 2016. Available: 10.3390/s16040592

# 

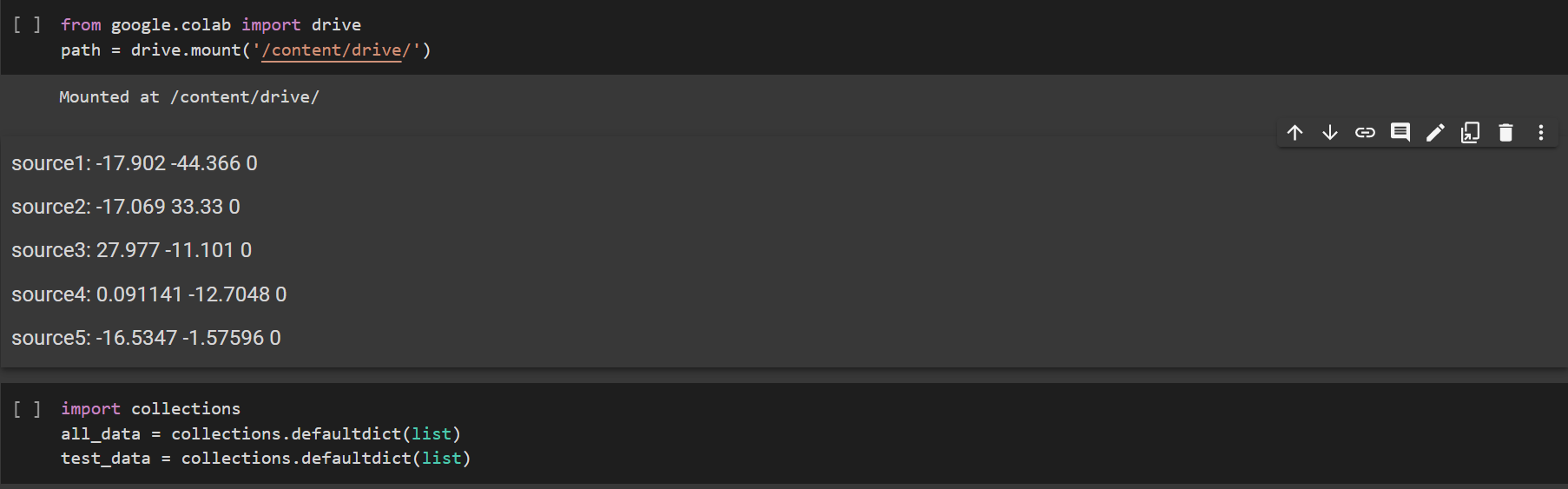
# 

# **7. Appendices**

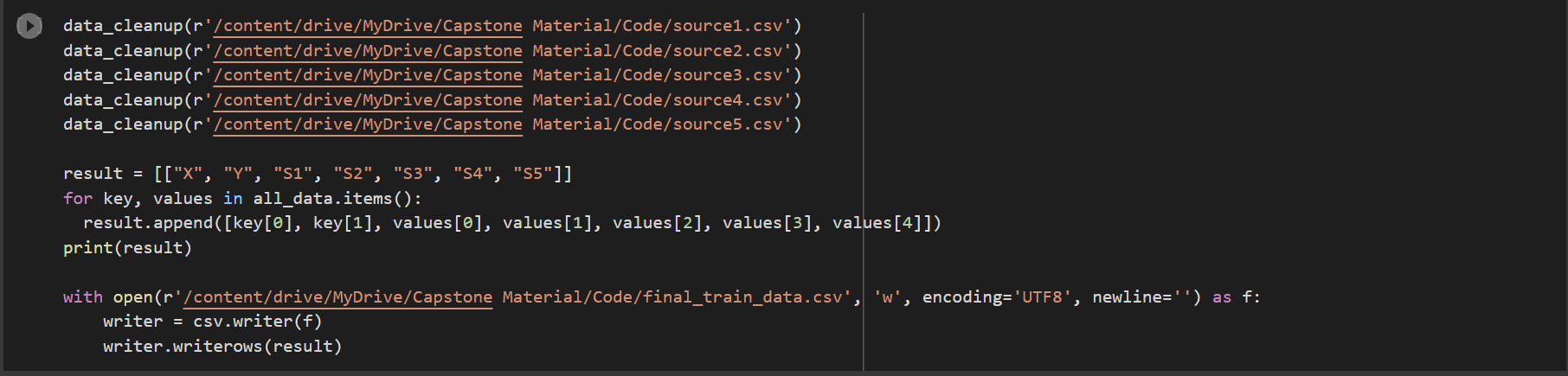
## **Appendix A : Simulation of data points**

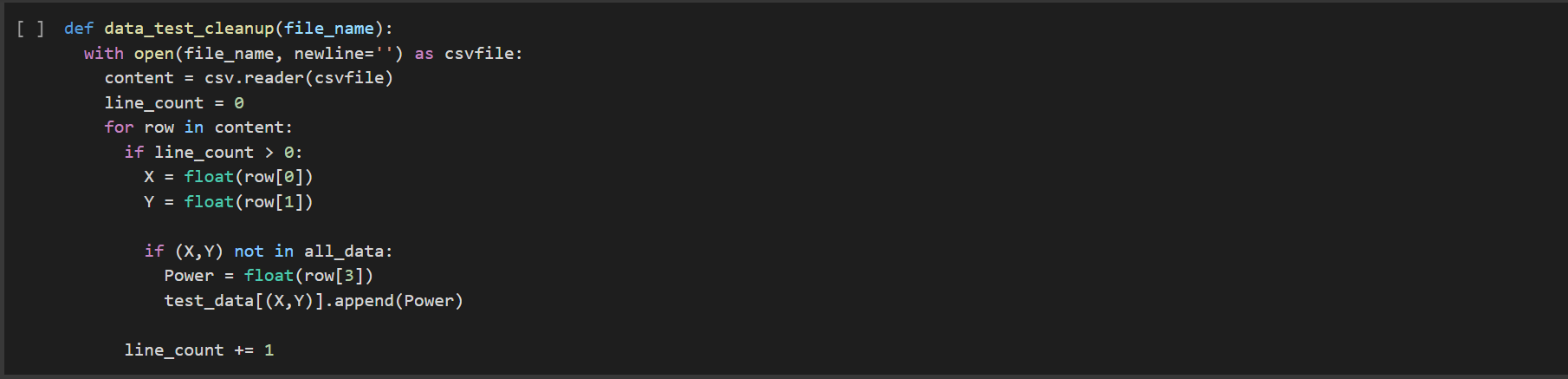


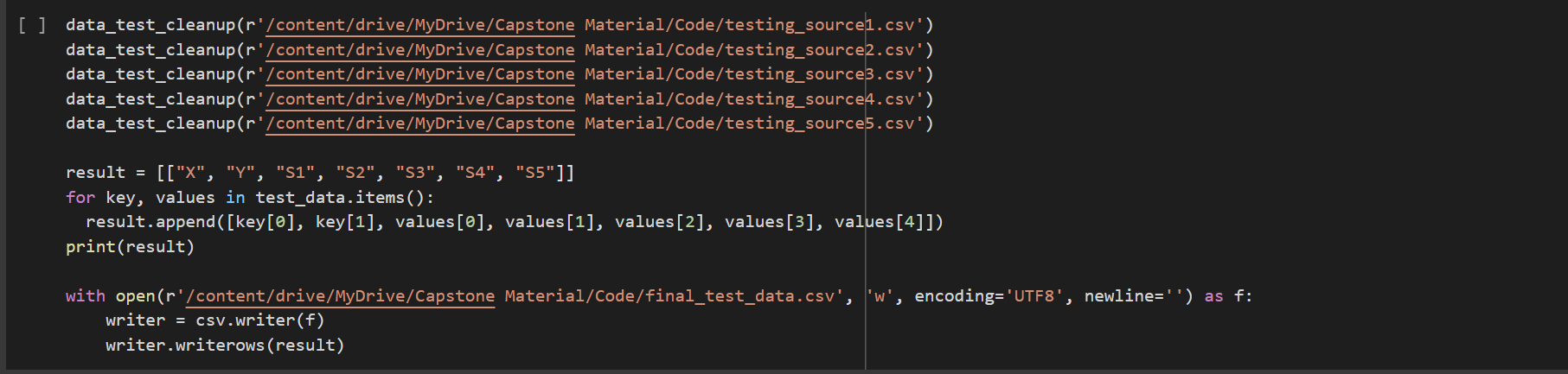
## **Appendix B : Data clean up**











## 

## 

## 

## 

## 

## 

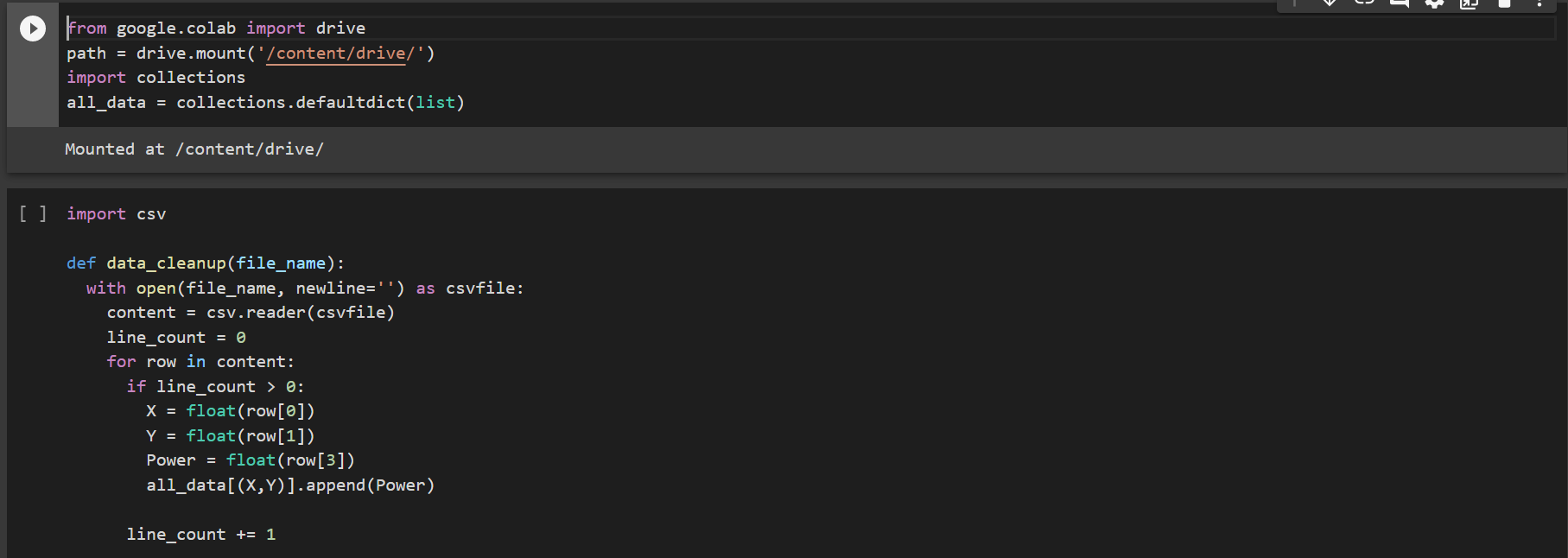
## 

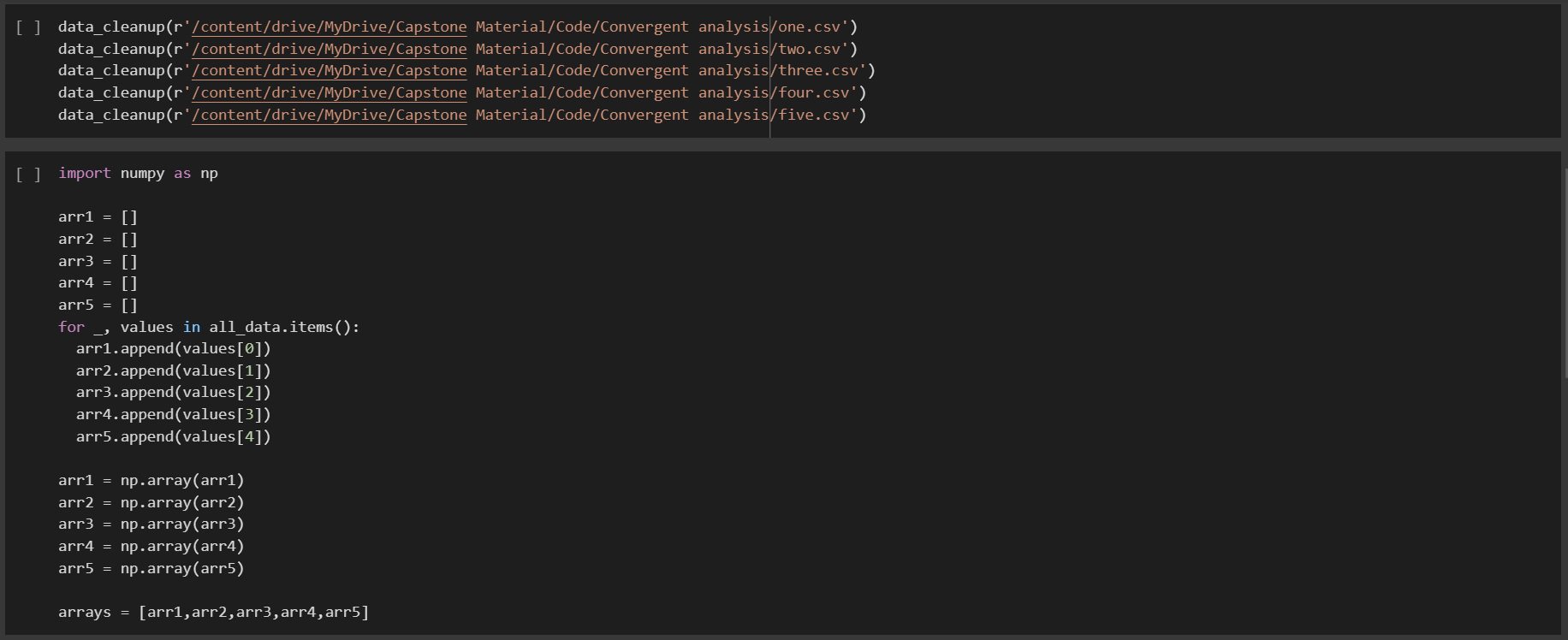
## 

## 

## 

## **Appendix C : Convergent Analysis**



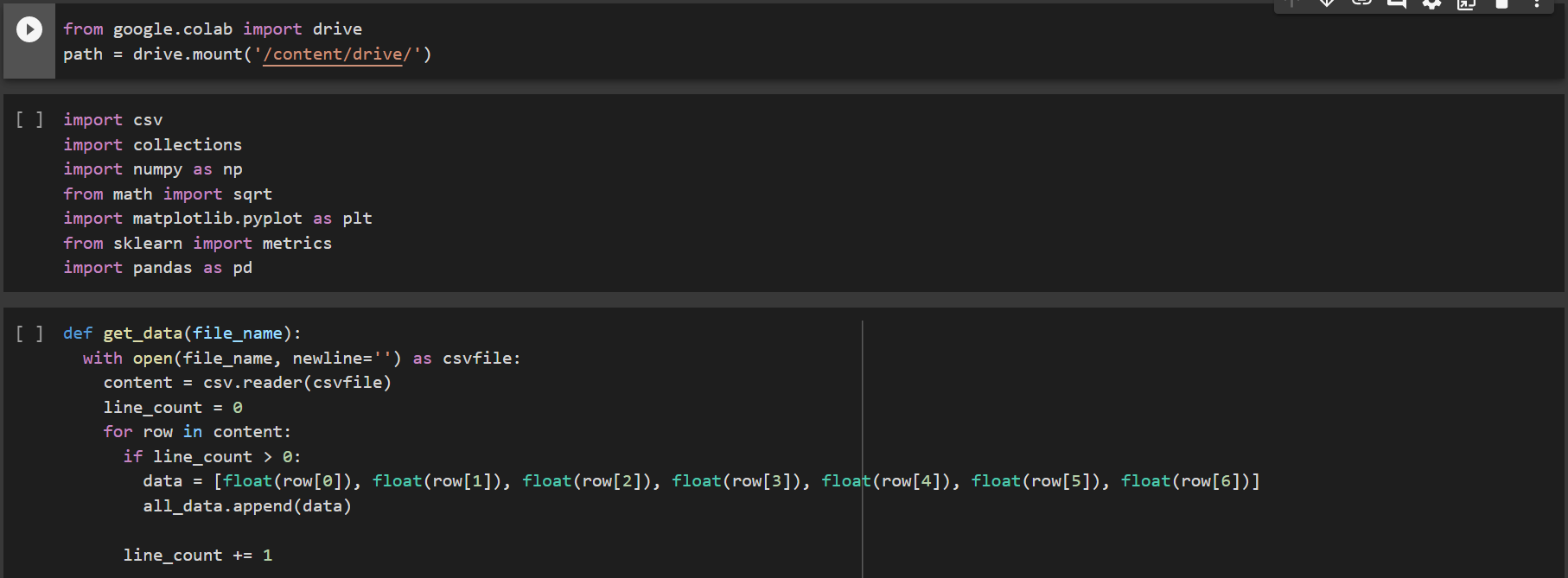


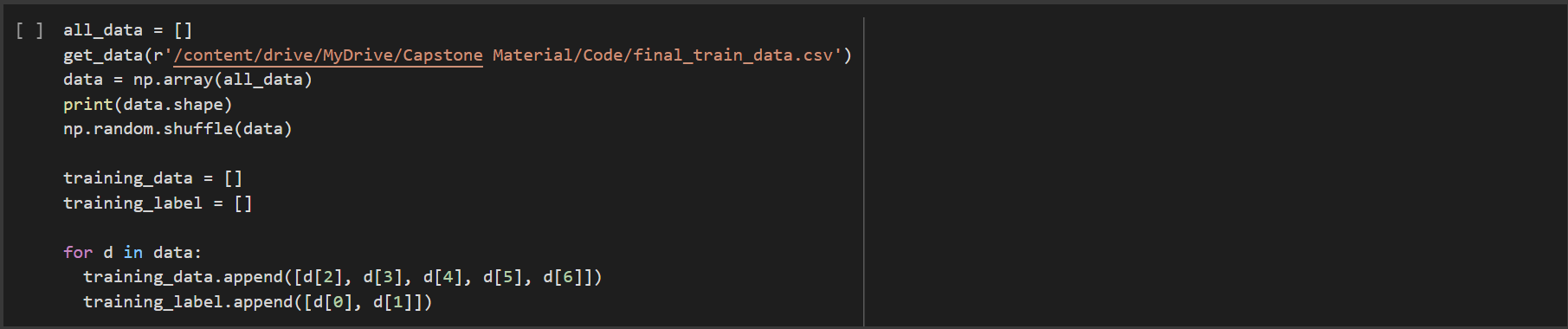


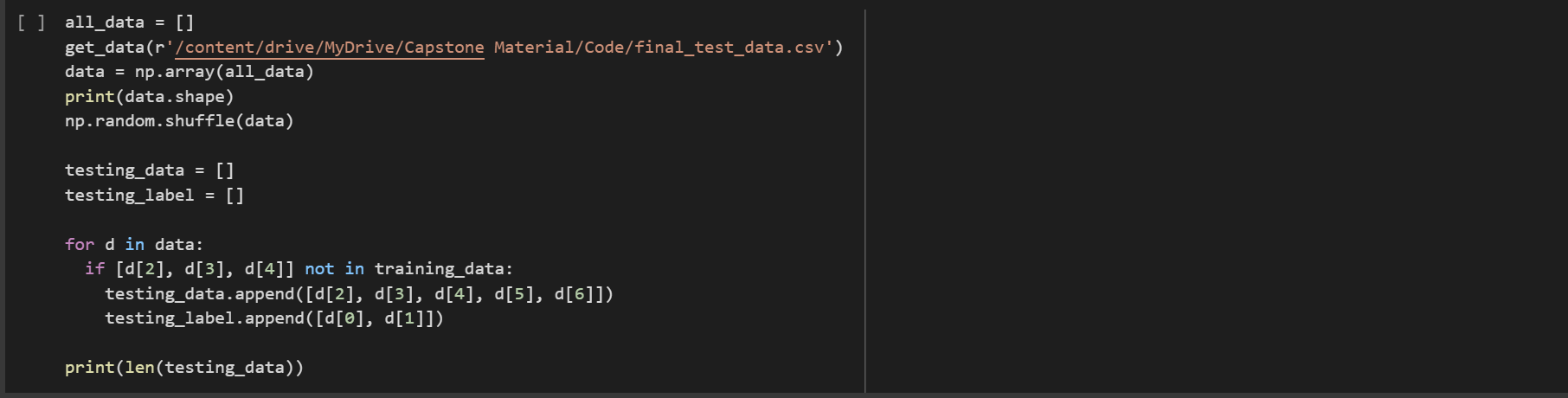
## 

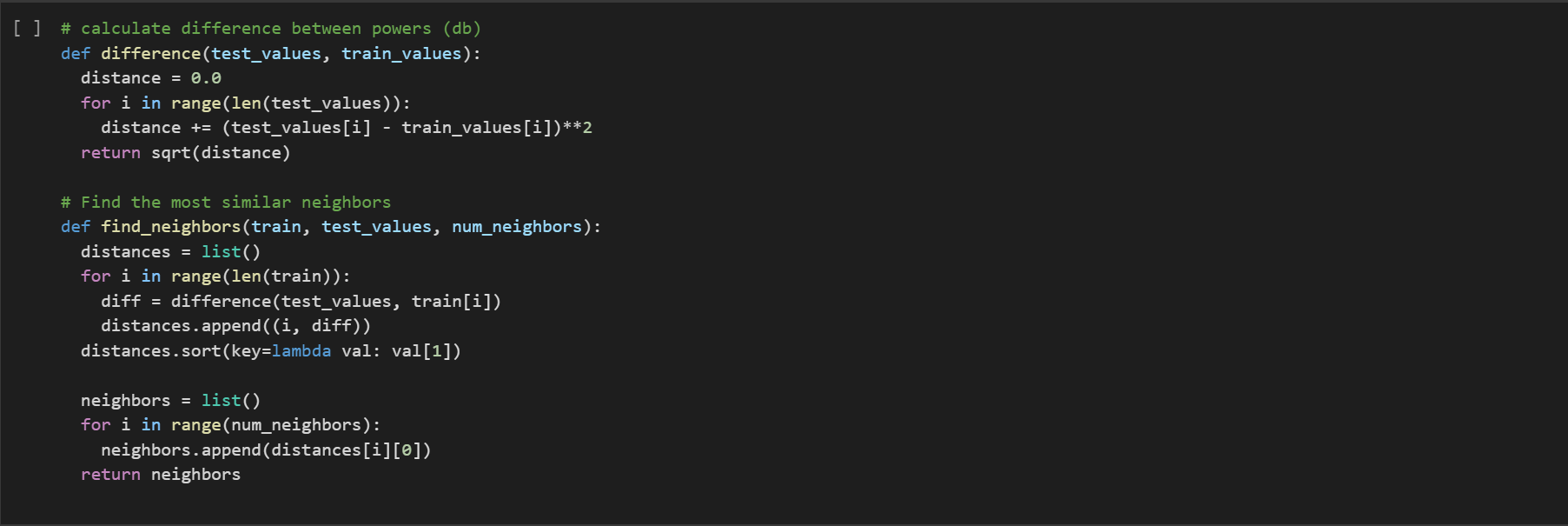
## 

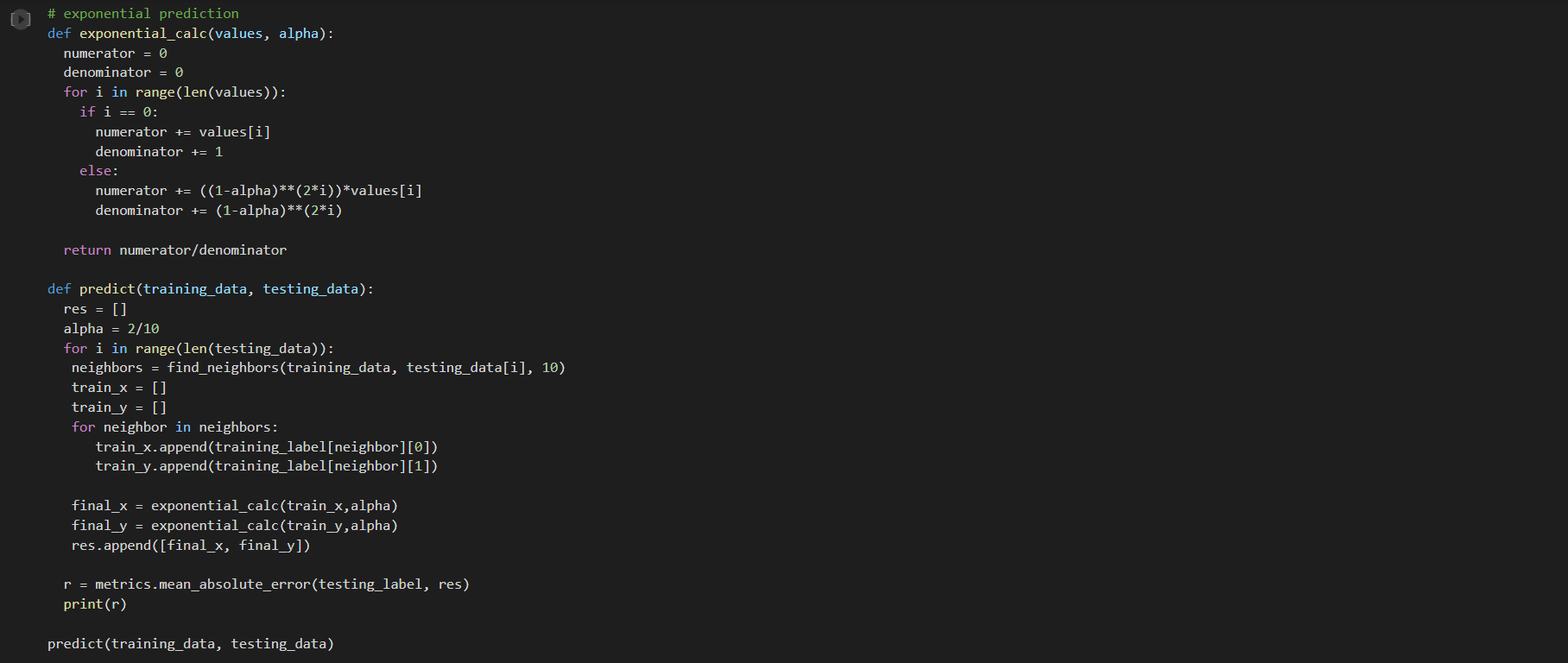
## **Appendix D : K Nearest Neighbor Model (KNN)**





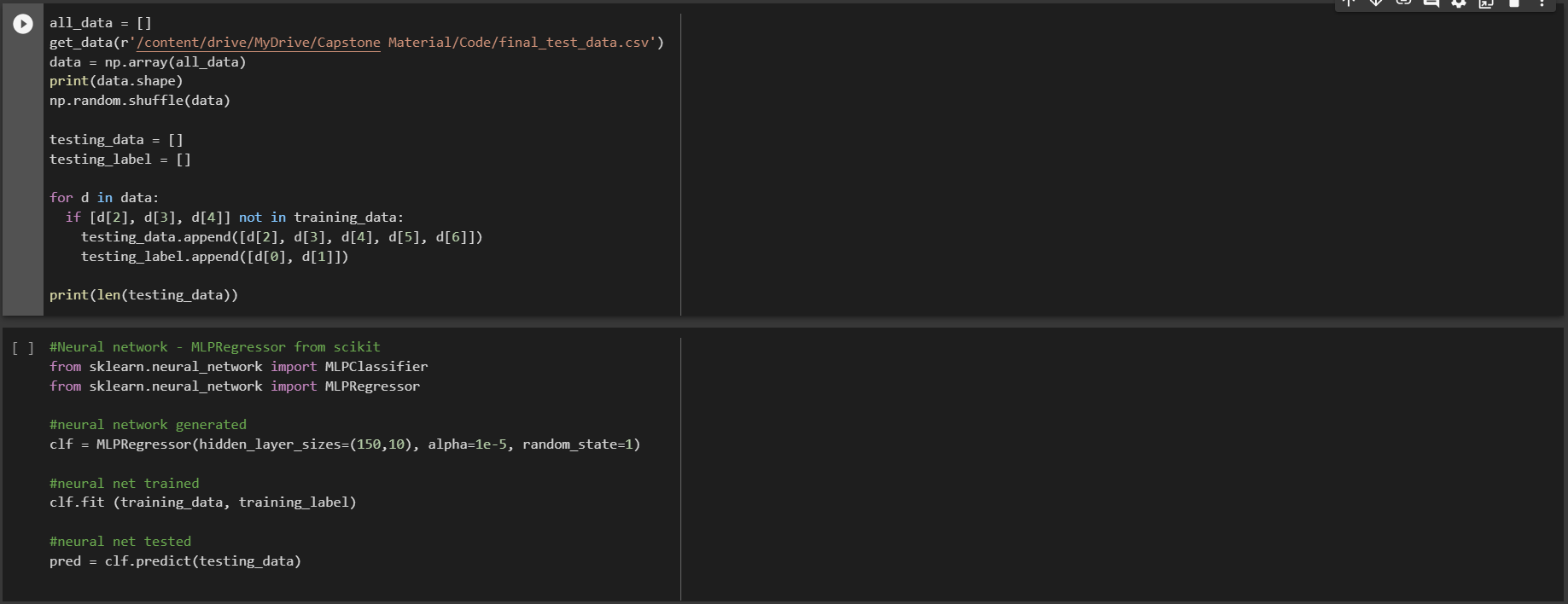


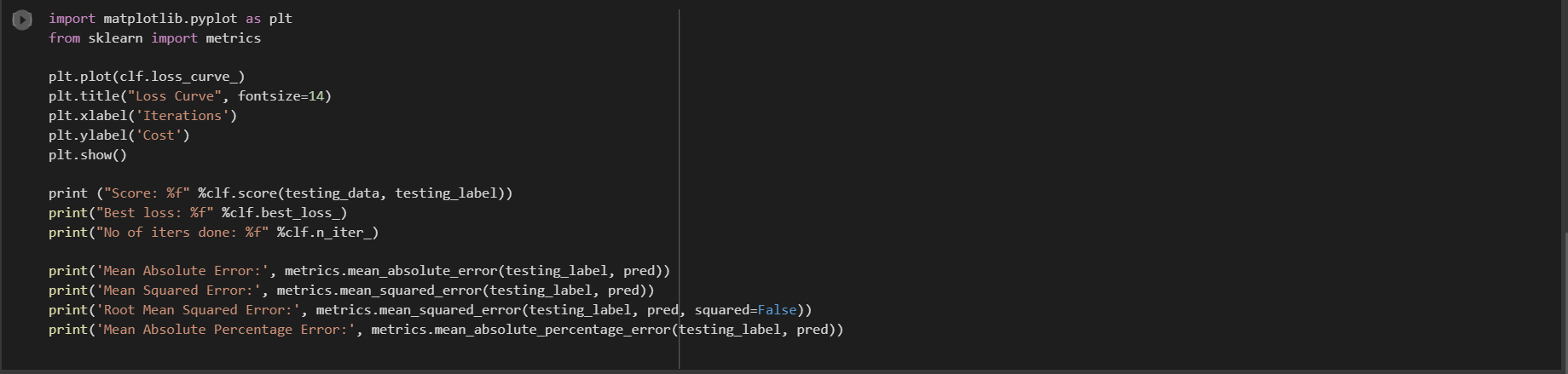




## **Appendix E : Artificial Neural Network Model (ANN)**

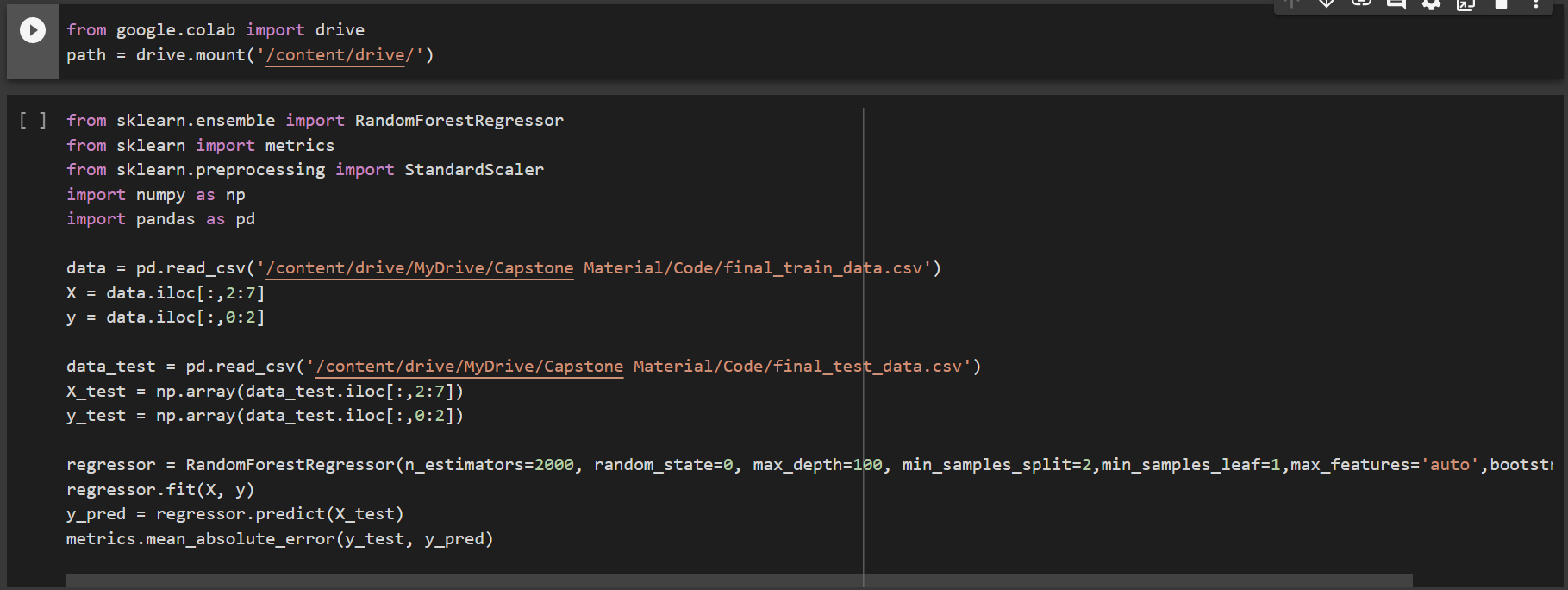






## 

## **Appendix F : Random Forest Model**



## 