

C964 Computer Science Capstone

Jason Whitby

Western Governors University

Signature:

Jason Whitby

Date:

[04/07/2023]

Springfieldington Greenhouses**Table of Contents**

A.1 Letter of Transmittal - Springfieldington Greenhouses	4
A.2 Project Recommendations	
A.2.1 Problem Summary	6
A.2.2 Application Benefits	6
A.2.3 Outline of the Data Product	6
A.2.4 Data Used Project	7
A.2.5 Objective and Hypothesis	7
A.2.6 Methodology	7
A.2.7 Funding Requirements	8
A.2.8 Impact of the Solution on Stakeholders	8
A.2.9 Ethical and Legal Considerations	8
A.2.10 Developer's Expertise	9
 B. Project Proposal for IT Professionals	 10
B.1 Problem Statement	10
B.2 Customer Description and Benefits	10
B.3 Existing Systems Integration	11
B.4 Data Needed	11
B.5 Project Methodology	11
B.6 Project Deliverables	12
B.7 Implementation Plan	13

Springfieldington Greenhouses

B.8 Evaluation Plan	13
B.9 Programming Environments and Related Costs	13
B.10 Timeline and Milestones	14
D. Developed Product Documentation	16
D.1 Business Requirements and Project Purpose	16
D.2 Raw Data and Cleaned Data	16
D.3 Code Analysis	19
D.4 Hypothesis Verification	23
D.5 Visualization and Effective Storytelling	23
D.6 Analysis of Accuracy	24
D.7 Testing of Application	24
D.8 Files of Application	25
D.9 User's Guide	26

Springfieldington Greenhouses

A.1 Letter of Transmittal

April 8, 2023

Josephine Swallow, Owner & Operator

Bloomingtonfielderston Labs

6342 Sequential Ave.

Salt Lake City, UT 84107

Dear Ms. Swallow,

You reached out to us here at Bloomingtonfielderston Labs to assist you in a way to help increase your profits by minimizing unnecessary expenditures by correcting errors with your flower transactions. It was noticed a few months ago that your profits seemed to be slipping due to the discrepancies with your iris seed intake and sales. Paying too much for the incorrect seeds from your distributor that cannot be determined until the flower is blooming has been causing your profit losses. This has been a common occurrence among the botany industry, especially among greenhouses. Sellers believe that they can scam newer members to the industry with the wrong seeds and scam them out of their money. More long-term measures can be taken in the future to assist with producing your own seeds for distribution between your stores. In the meantime though, you have asked us to assist you with a way to tell different iris species apart after blooming to help with sales and seed storage.

The solution that we are presenting to you is one that uses the Support Vector Machine algorithm to predict with high accuracy a result of the dimensions of the dataset it is trained on. The training uses the dataset that your employees have on hand from the measurements they have taken of the irises. It was built to support initially four dimensions being petal length, petal width, sepal length, and sepal width. This can be increased at a later point in time to add more dimensions for measurement if deemed necessary. Using the measurements from the dataset, the SVM then takes roughly 33% of the data and uses that to train itself for the most common dimensions for the three different species that you have in your greenhouses. That being the Iris Setosa, Iris Virginica, and the Iris Versicolor. The algorithm is in a graphical user interface format, so there is no need to worry about misunderstanding what needs to be taken and where to put it.

The objective of our application is to simply predict with high accuracy the particular species that an employee has in front of them by giving it the measurements they have taken. Once these measurements are put into the application and the employee clicks the button, the training will begin on the algorithm. After the training is completed, the algorithm will show the predicted

Springfieldington Greenhouses

species and the accuracy score of its prediction. This is something that the employee can show to the customer for them to verify that they are purchasing the correct species. This will in turn reinforce your reputation with your customers assuring them that they have exactly what they are looking for.

There is a caveat to the SVM algorithm though in that it does take substantial time to process as it gets larger. In time complexity it is an n -cubed algorithm meaning the larger it gets the slower it goes. Certain studies have shown that the upper bound for a dataset for the SVM algorithm is somewhere between 5 to 15 million rows.

The financing for this particular project is estimated at \$320,000 broken into two separate phases. This is due to scaling from the small 150 row dataset that we will be using for this demonstration and testing to find the maximum upper bound for the dataset. The two phases are as follows: deployment and data updates. The first phase will cost roughly \$140,000 for the hardware that our engineers will need, and should take roughly 120 engineering hours. The second phase will cost a little bit more at \$180,000 and take closer to 160 engineering hours to complete. This is due to the amount of data that will need to be collected to scale the application. But, due to the high accuracy we believe that you will see a very substantial increase in your profits in the first quarter after full deployment.

If you have any further questions, please do not hesitate to reach out to me at janos.audron@example.com or at (666)867-5309.

Looking forward to hearing back from you!

All the best,

A handwritten signature in black ink that reads "Jason Whitby". The script is fluid and cursive, with the first letters of each word being capitalized and larger than the rest of the letters.

Jason Whitby, CEO, CTO, CISO, COO Bloomingtonfielderston Labs

Springfieldington Greenhouses

A.2 Project Recommendations Summary

A.2.1 Problem Summary

Springfieldington Greenhouses is a premiere greenhouse specializing in the brokering and distribution of flowers and plants around the country. Recently they have noticed that they seem to be losing profits due to customers saying that they are not receiving or purchasing the correct species of iris. This has been traced to two problems in the business setup. The first problem is with the newer employees not being able to determine the species of the flower by sight alone. The second problem is with the sellers that are supplying the seeds. This is a problem due to the fact that it is not easy to determine the specific species of the flower until it blooms. The Iris Prediction Engine (IPE) takes care of these problems by first allowing newer employees to work directly in front of customers taking measurements of key properties of the flower to put into the prediction engine to determine the correct species. It then becomes more accurate with the increase in data that it receives to then allow the Springfieldington Greenhouses to start supplying their own seeds to themselves. This removes the third-party aspect altogether.

A.2.2 Application Benefits

There are a multitude of benefits that will arise from using the IPE. But, we will focus mainly on the ability to recoup the lost profits and gain new ones due to gaining back the confidence of the customer base. Having the ability to show customers in real-time that they are purchasing the correct species of the iris that they set out to plant and nurture at home will go a very long way in increasing the profits of the Springfieldington Greenhouses. It does this by using a predetermined dataset in the beginning and building off of it over time to increase the accuracy of the predictions. This then protects the integrity of the greenhouse and allows for seed harvesting that will then be used to supply the rest of the greenhouses in the network with the correct seeds to sell or plant.

A.2.3 Outline of the Data Product

The Iris Prediction Engine is being built using the Python 3.9 language for its wide spread usage in the data analysis and machine learning fields of computer science. There are a few libraries that are installed with this version of the language to make it easier to do things such as the analysis. These libraries are Scikit-Learn for machine learning, Pandas for data analysis, Seaborn and Matplotlib for data visualization. There is also one more library, PyQt5 which is being used to create a graphical user interface to allow for ease of use for any user. It would also make it easier to show customers the output in an easy to read way. The engine is hosted on a local computer, but can be packaged up and hosted wherever it is needed very quickly along with the necessary spreadsheets of data that it uses for predictions.

Springfieldington Greenhouses

A.2.4 Data Used in Project

The dataset that is being used for the demonstration is in two different comma separated value files (.csv). The first one is a standardized file that has headers at the top that is used to display the data to the user. It has at the moment 150 rows divided into 5 columns for petal length, petal width, sepal length, sepal width, and iris species. The second file is almost identical to the first except that the headers have been removed due to the nature of the SVM algorithm and how it needs to parse the data. These files can and should be altered later by increasing the amount of rows that are in them to increase the accuracy of the engine. In the current testing environment there has been no less than ~90% accuracy on any one test. The engine randomly selects 33% of the data in the dataset to use for its training every time the button is pressed to begin the training. Increasing the data in the dataset would lead to a higher accuracy due to having more data to learn from.

A.2.5 Objective and Hypothesis

The main objective of this engine is to provide peace of mind to the customer base that has lost faith the Springfieldington Greenhouses due to errors with species classification. This will be done by showing the customers that they are indeed purchasing what the employees say that they are. The hypothesis for this engine is that over time with added data it will come closer and closer to 100% accuracy. But for the moment, it will remain at best around 98% due to the small amount of data that it has. But, it is hypothesized to remain above 90% for 99% of the time. The showing of the accuracy to the customers will increase profits by an estimated 35% over the first year.

A.2.6 Methodology

The methodology that will be used for this project is the Agile framework for CI/CD. The Agile framework specializes in Continuous Integration and Continuous Deployment which makes it very easy to make changes and get them rolled out in a very short period of time. Breaking the project into two phases that will then be broken into “sprints” for a more granular approach. The first phase consists of getting the base application distributed to all of the green houses to begin the employee training. The second phase will be adding the functionality for data collection and addition for the dataset to increase the accuracy of the prediction engine. During the first phase we will collect user sentiment to get a good idea if anything needs to be changed in the presentation. Things such as button placement, coloring, and ease of use. For the first sprint, we will focus solely on getting the engine packaged up and distributed to the greenhouses to begin the training. Second sprint will focus on beginning to use the engine in real-world setups to ensure that it is working properly. The third sprint will focus on linking all of the distributions

Springfieldington Greenhouses

to a central dataset that has gathered all of the data from the users using the engine. A master .csv file if you will that will then be downloaded periodically from the main file to the remote engines to use for training. This part will be a continuous thing thereafter.

A.2.7 Funding Requirements

The letter of transmittal gave the specifications for the funding requirements of this project, but I will reiterate them here. All of the software that is being used for the project is open source and is therefore free. Python and all of the libraries that are being used are free to use by anyone that desires to use them. The data analysis and visualization libraries are all part of the Python open source setup. There are no special tools that are required from a software perspective so that does save money there. The costs that are being incurred are through the engineers' time and salaries initially. As stated in the letter, there is the initial \$140,000 that is needed for the first phase for building the application and getting it deployed. The \$160,000 for the second phase is being incurred from setting up the ability to increase the training dataset that will be used to increase the accuracy of the engine. If you so choose for us to host the dataset in our servers it will cost an extra \$50,000 per year for the leasing and management of the dataset. Budgeting for this will be necessary for submission to your financial department.

A.2.8 Impact of the Solution on Stakeholders

The impact of the application solution on the stakeholders is to be substantial in terms of profits and revenue once it is fully rolled out. All of the information that the Springfieldington Greenhouses collect can be used to cut out the third-party vendors that they are buying incorrect seeds from to start using their own crops. This would save thousands year over year in incorrect seeds purchasing. Currently the estimates are roughly \$200,000 annually that Springfieldington Greenhouses is losing due to the incorrect seeds. Secondly, due to the nature of the engine being used to collect data on irises for increasing its accuracy it can be further utilized for other species. This can then be slightly retooled to then be sold to others to use in their particular setups for accuracy prediction generating another source of income for Springfieldington Greenhouses. Lastly, due to regaining the customer trust it is predicted that nationwide Springfieldington Greenhouses will see an estimated \$3 million increase in profit by this time next year.

A.2.9 Ethical and Legal Considerations

The data that is being collected by the IPE is not proprietary or sensitive in any way. This means that there are no ethical or legal issues that need to be addressed at the current time. If in the

Springfieldington Greenhouses

future Springfieldington Greenhouses would like to start hybridizing flowers and plants to create new species there would be a proprietary concern that would require more stringent storage measures. This would mean encrypting the data in transit and at rest in a secured database. Also, this would require the use of role based access controls for use of certain branches of the IPE. Any breach of the data could lead to loss of profits for Springfieldington Greenhouses.

A.2.10 Developer's Expertise

The engineers and developers that will be working on this project will be comprised of one data/machine learning engineer and one front-end application developer. Each one junior engineer/developer and one intern. This is get the juniors more exposure to large projects as well as to teach them some leadership functions with the interns. The seniors will act mostly as scrum masters to set up where the work needs to be focused on for the sprints. Over the time of the project the juniors and interns will get a greater understanding of how projects function and will be able to contribute more in the future.

Springfieldington Greenhouses

B. Project Proposal for IT Professionals

B.1 Problem Statement

In the realm of plant production there has been a recent trend emerging with supposed trusted vendors selling seeds that they say are from a specific species but are not. These sellers take the seeds of a cheaper plant and sell them at a much higher price to buyers that are expecting another species of plant. This type of scamming is very simple to implement due to the slow nature of plant growth. The victim does not know that they have been scammed until the plant starts to grow and mature. By that time the scammer has disappeared with the money and becomes almost impossible to find. A way to fight this kind of scamming is to have a very expensive DNA analysis system to analyze the seeds when they are purchased. This is not a feasible solution for most greenhouses.

The best way to combat this situation is to harvest and procure the seeds based from the species that is known to be of the species that the customer knows to be true. A problem with getting the species correct is that of a new employee that might not know specifically what the dimensions a specific flower might be. For this, we have the IPE. By using this tool, the employee can submit measurements that they take and put them into the tool. The IPE will then give a prediction based off of the dataset it has collected from other known measurements to predict with high accuracy what the particular species is that the employee is observing. This will then lead to having the ability to better classify the plants in the greenhouses and harvest and distribute the seeds from within the company cutting out a need for a third-party vendor. Thereby stopping this scam altogether and gaining back the lost profits.

B.2 Customer Description and Benefits

The customer(s) for this project is the Springfieldington Greenhouses and all of its branch locations in its network. Anything outside of that can be negotiated later if there is a desire for us to build a different tooling or implementation of the IPE for a subsidiary of the company or one of its partners. The benefits of the IPE have been described above but we will now describe the particular needs that the IPE will assist with:

1. The customer is trying to recoup lost profits due to repeatedly either being scammed or accidentally misled by third-party seed vendors with regard to their iris crops. The IPE will help Springfieldington Greenhouses with having accurate descriptions of their flowers that will allow for their own seed harvesting.
2. Springfieldington Greenhouses needs an easy to use tool for newer employees that are not very familiar with the particular species of flower they are dealing with. The IPE will

Springfieldington Greenhouses

be that simple tool that allows the newer employees to put the measurements in and see the result of the species of flower.

3. Larger profits could be made by using the tool to classify more species over time and become a reputable vendor of seeds to other companies.

B.3 Existing Systems Integration

All aspects of the IPE are sourced from open source software which is publicly available. The language that the IPE is written in is Python along with existing free libraries. There is not an existing setup for this particular implementation at the moment. But, there might need to be a data storage system that needs to be built to integrate with the data collection that happens with the engine. That will need to be built as a separate project if the Springfieldington Greenhouses wants us to house their data. The IPE as of now is a stand-alone tool that can be utilized from any one of the Springfieldington Greenhouses locations.

B.4 Data Needed

The initial data that is being used for the IPE was sourced from <https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv> and was then replicated into its own .csv file. This was to allow for the cleaning of the .csv file to account for the SVM model not being able to read the headers of the file. The data that is being displayed in the table on the main page is located in the file Iris.csv whereas the data the the model is training and making its predictions on is at Iris2.csv.

Reading the .csv files is done using the Pandas library in Python to convert it to a dataframe that can be read from or written to depending on the need. This dataframe is then used by the SVM to train and then deliver its prediction based on the dimensions and species specifications that are in the data. It does this using a random selection of 33% of the data each time the training is run so it is constantly learning.

B.5 Project Methodology

For this project, we will be using the Agile methodology with a Scrum approach. This was discussed above earlier but will be stated more clearly in this section. This method was chosen specifically for its adaptability to a changing environment and quick process change turnaround. It is also a method that we at Bloomingtonfielderston Labs have used in the past to meet deadlines. An outlined process is provided just below this:

Springfieldington Greenhouses

B.5.1 – Requirements

- Our engine must provide predictions of iris flowers from three different species with above an accuracy of 90%.
- We will take in feedback from users of the engine to assess what improvements can be made and add those as feature requests.
- The Springfieldington Greenhouses should have the utmost confidence that have in their possession the correct flowers that they can sell to their customers.

B.5.2 – Development

- Two phases of the project are forecast to take roughly 7 weeks at 320 engineering hours. There is an extra week taken into consideration for possible bugs and extra training for employees.
- Phase one will focus on distributing the IPE to all of the greenhouses in the network and beginning the employee training.
- Phase two will be feature requests and updates for any found bugs in the IPE.

B.5.3 – Testing

- The first tests will be done at the primary greenhouse in the network to test the implementation in a real-world environment with employees.
- The unit and integration testing for the IPE will be done in-house at Bloomingtonfielderston Labs. White box testing will then be done with employees at the greenhouses.
- Collected data and feedback will be utilized from the phase one testing to improve the IPE in phase two.

B.5.4 – Distribution

- After phase one introduction and testing has been completed, the final product will be distributed to the stakeholders in all of the greenhouses in the network in phase two.

B.5.5 – Feedback

- There will be feedback collection efforts during phase one that can be utilized to integrate into phase two.
- The sources of feedback will be the main users as employees, and the managerial level of Springfieldington Greenhouses.

B.6 Project Deliverables

The IPE is the only deliverable that is necessary for this project.

Springfieldington Greenhouses

B.7 Implementation Plan

From an implementation perspective there are the two phases as mentioned above. The first phase will be the initial rollout with training for employees with 24/7 assistance from our team. The training materials will include a set of videos that walk through how to use the engine as well as the pre-loaded dataset for training purposes.

As far as access to the code is concerned, only the senior engineer and developer will be allowed to push changes. There will be manual checks put in the pipeline to ensure that nothing that is not cleared by the higher ups can make it into the production stage. It will be encouraged for the more junior developers and engineers to experiment in their own branches of the code base for feature requests and bug fixes. The access to the codebase though will remain strictly to the development team itself with role-based access controls in place. This is to prevent any outside sources from influencing the project.

The Springfieldington Greenhouses will have to house their own data unless negotiated after the project rollout. They will need to either store their data in a .csv format so they IPE can parse it, or they will need some conversion software that can reformat their database scheme into a .csv. The tool that they choose for this will need to be able to reconvert the data back to the original format for ease of use for displays that they may want to have later.

B.8 Evaluation Plan

To evaluate the reception of the IPE, surveys will be taken from the super users down to the daily regular users. As well as monitor how the accuracy changes over time as the Springfieldington Greenhouses introduce new data into the system. This accuracy curve will show us how the IPE is shaping up compared to initial testing. If it starts to fall in accuracy as time progresses, investigations will be made to figure out why this is occurring. This could be to employees putting in incorrect measurements on accident or as a joke that could skew the data. If that is the case, there will be a need to implement some sort of data check to ensure that the correct data is going in. This should be well under control by phase two after all of the employees are trained and using the IPE regularly. We expect erroneous behavior to take place during phase one.

Overall, for the final evaluation, the IPE has to be consistently at or above a 95% accuracy rate for a period of 500 inputs for the post training data input. As well as meet a 80% acceptance rate by the employees based on the survey results not related to feature requests.

B.9 Programming Environments and Related Costs

Programming environment:

Springfieldington Greenhouses

- Python 3.9 with Qt Designer, PyQt5, Sci-kit Learn, Seaborn, Pandas, and Matplotlib libraries.
- File conversion software for converting database files to .csv format.
- Software for tracking usage and accuracy statistics about the IPE during first phase.

No costs are to be had with the programming environment as all of the tools, languages, and libraries are open source software and free for public usage.

Environment Costs:

- It will be necessary to obtain hardware for workstations and storage for deployment phases. The workstations will need networking capabilities to link to a central network for the development, testing, and data collection and analysis.

There is an initial budget of \$35,000 for the costs associated with the physical hardware environment that is accounted for in the first payment.

People Resource Requirements:

- There is a requirement of three engineers and three developers with different levels of experience for a total of \$56,000 for the two month project. They will be paid according to the following:
 - o Seniors will be paid \$14,000 per month
 - o Juniors will be paid \$10,000 per month
 - o Interns will be paid \$4000 per month
- The first phase (Deployment): 120 engineering hours at a cost of \$140,000 which includes the cost of the first months salaries and equipment.
- The second phase (Features and bugs): 160 engineering hours with an extra week for backlog items. The total for this phase will be \$180,000 which will account for the following extras:
 - o Extra engineering hours
 - o Full scale deployment to all greenhouses
 - o Third-party testers
 - o Increased hardware for scaling
 - o Unseen cost reserves

B.10 Timeline and Milestones

Event	Start Date	End Date	Required engineering hours	Dependencies	Resources Assigned
Start of project	May 1st, 2023	May 1st, 2023	0	None	Primary stakeholders, Project Manager

Springfieldington Greenhouses

Phase: One	May 1st, 2023	May 19th, 2023	120	Start of project	Primary stakeholders, Project Manager
Project engineering and scaling	May 1st, 2023	May 5th, 2023	40	Start of project	Project Manager, Engineers, Developers
IPE deployment to primary greenhouse for training and integration tests	May 5th, 2023	May 19th, 2023	75	Project engineering and scaling	Developers, Engineers
IPE deployment to remaining greenhouses through network	May 19th, 2023	May 19th, 2023	5	IPE deployment to primary greenhouse for training and integration tests	Project Manager, Engineers, Developers
Phase: Two	May 22nd, 2023	June 16th, 2023	160	Phase: One	Project Manager, All Stakeholders, Developers, Engineers
1 st Major requested feature updates	May 29th, 2023	June 2nd, 2023	40	Phase: Two	Project Manager, Engineers, Developers
2 nd Major requested feature updates	June 5th, 2023	June 16th, 2023	80	1 st Major requested feature update	Project Manager, Engineers, Developers
Finalization for bug remediation and stakeholder signoff	June 19th, 2023	June 23rd, 2023	40	2 nd Major requested feature update	Project Manager, All Stakeholders, Engineers, Developers

Springfieldington Greenhouses

D. Developed Product Documentation

D.1 Business Requirements and Project Purpose

Throughout this document the business requirements have been outlined. But, they will be recapped in this section. The application for the Springfieldington Greenhouses will be built in the Python programming language using free and open-source libraries that are publicly available. In addition, there will be a total of \$320,000 required for the budget for the project unless there is a need to negotiate long-term data storage and management moving forward.

From the aspect of the application, the main requirement is to build a user friendly prediction engine that is able to predict consistently above 95%. The engine will not be using any proprietary or sensitive information in the beginning. There might be a use case for this later if the Springfieldington Greenhouses want to begin growing hybrid plants. Though that is out of scope for this project. The purpose of this prediction engine is to recoup lost profits and increase new profits. This is due to being misled during the purchasing of seeds from third-party vendors and accidental wrong classification of flowers by newer employees.

D.2 Raw Data and Cleaned Data

All of the initial data for this project is formatted in a .csv format and was obtained from a free to use dataset that is hosted on Github at the moment (<https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv>).

There are two different files that are included in the source code for the engine: Iris.csv and Iris2.csv. These two files are used for two different purposes by different parts of the engine. The modified file Iris.csv is used to show the user the data that is being used to train the data. It has had a header row placed at the top to show the user what they are looking at in the data. The second file Iris2.csv is unmodified and so it can be parsed by the SVM algorithm properly for training and prediction. **Figure D.2.1** is the modified Iris.csv file that shows the header row that is displayed in the table view on the main page of the IPE. **Figure D.2.2** is the raw data that was extracted from the Github datasource.

Springfieldington Greenhouses

1	Sepal Length(cm),Sepal Width(cm),Petal Length(cm),Petal Width(cm),Species
2	5.1,3.5,1.4,0.2,Iris-setosa
3	4.9,3.0,1.4,0.2,Iris-setosa
4	4.7,3.2,1.3,0.2,Iris-setosa
5	4.6,3.1,1.5,0.2,Iris-setosa
6	5.0,3.6,1.4,0.2,Iris-setosa
7	5.4,3.9,1.7,0.4,Iris-setosa
8	4.6,3.4,1.4,0.3,Iris-setosa
9	5.0,3.4,1.5,0.2,Iris-setosa
10	4.4,2.9,1.4,0.2,Iris-setosa
11	4.9,3.1,1.5,0.1,Iris-setosa
12	5.4,3.7,1.5,0.2,Iris-setosa
13	4.8,3.4,1.6,0.2,Iris-setosa
14	4.8,3.0,1.4,0.1,Iris-setosa
15	4.3,3.0,1.1,0.1,Iris-setosa
16	5.8,4.0,1.2,0.2,Iris-setosa
17	5.7,4.4,1.5,0.4,Iris-setosa
18	5.4,3.9,1.3,0.4,Iris-setosa
19	5.1,3.5,1.4,0.3,Iris-setosa
20	5.7,3.8,1.7,0.3,Iris-setosa

Figure D.2.1 – Modified Iris.csv file

1	5.1,3.5,1.4,0.2,Iris-setosa
2	4.9,3.0,1.4,0.2,Iris-setosa
3	4.7,3.2,1.3,0.2,Iris-setosa
4	4.6,3.1,1.5,0.2,Iris-setosa
5	5.0,3.6,1.4,0.2,Iris-setosa
6	5.4,3.9,1.7,0.4,Iris-setosa
7	4.6,3.4,1.4,0.3,Iris-setosa
8	5.0,3.4,1.5,0.2,Iris-setosa
9	4.4,2.9,1.4,0.2,Iris-setosa
10	4.9,3.1,1.5,0.1,Iris-setosa
11	5.4,3.7,1.5,0.2,Iris-setosa
12	4.8,3.4,1.6,0.2,Iris-setosa
13	4.8,3.0,1.4,0.1,Iris-setosa
14	4.3,3.0,1.1,0.1,Iris-setosa
15	5.8,4.0,1.2,0.2,Iris-setosa
16	5.7,4.4,1.5,0.4,Iris-setosa
17	5.4,3.9,1.3,0.4,Iris-setosa
18	5.1,3.5,1.4,0.3,Iris-setosa
19	5.7,3.8,1.7,0.3,Iris-setosa
20	5.1,3.8,1.5,0.3,Iris-setosa

Figure D.2.2 – Clean Iris2.csv file

Springfieldington Greenhouses

The IPE functions by utilizing the Support Vector Machine algorithm (see **figure D.2.3**) to train, fit, and predict the species of iris that the user specifies through the dimensions. In **Figure D.2.4** you will see the portion of code that shows the confusion matrix and prints the accuracy percentage to the user interface. The two following figures can be seen in then Train_inference.py file.

```

92     def trainer(self):
93         iris = "Iris2.csv"
94         names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'type']
95         df = pd.read_csv(iris, names=names)
96         print(df.head())
97         mysvm_model = svm.SVC(max_iter=1000)
98         y = df.values[:, 4]
99         X = df.values[:, 0:4]
100        X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.33)
101        mysvm_model.fit(X_train, y_train)
102        y_pred_svm = mysvm_model.predict(X_test)
103        acc = 100*(metrics.accuracy_score(y_test, y_pred_svm))
104        accStr = str(acc)
105        self.label3.setText("Accuracy Score: " + accStr + "%")
106        print(acc)

```

Figure D.2.3 – First Data Filter Algorithm (two-coin flips)

```

108        xlabel = ["Iris-Setosa", "Iris-Versicolor", "Iris-Virginica"]
109        ylabel = ["Iris-Setosa", "Iris-Versicolor", "Iris-Virginica"]
110        cm = confusion_matrix(y_test, y_pred_svm, labels=mysvm_model.classes_)
111        sns.heatmap(cm, cmap="Greys", annot=True, xticklabels=xlabel, yticklabels=ylabel)
112        plt.title("Species Predictions Confusion Matrix")
113        plt.show()
114        num1 = self.lineEdit1.text()
115        num2 = self.lineEdit2.text()
116        num3 = self.lineEdit3.text()
117        num4 = self.lineEdit4.text()
118        try:
119            value = (num1 + ' ' + num2 + ' ' + num3 + ' ' + num4)
120            valueList = list(value.split(" "))
121            print(mysvm_model.predict([valueList]))
122        except:
123            msg = QMessageBox()
124            msg.setIcon(QMessageBox.Critical)
125            msg.setText("Error! Please close confusion matrix window \nand fill out all values with numbers.")
126            msg.setInformativeText("The model cannot predict accurately otherwise.")
127            msg.setWindowTitle("Error")
128            msg.exec_()
129            pass
130        self.label4.setText("Predicted Species: " + str(mysvm_model.predict([valueList])))

```

Figure D.2.4 – Second Data Filter Algorithm (three-coin flips)

Springfieldington Greenhouses

D.3 Code Analysis

In order to help the user understand how the predictions are taking place from the dataset, we have built a GUI that provides three different methods of understanding the data that is used. These specific are shown below:

Predictive Tool:

1.) Overview of input requirement for predictions

In order for the IPE to actually make a prediction, the user will need to submit measurement dimensions for the SVM algorithm measure its training against a single instance. This then shows the user a confusion matrix as well as the predicted species and an accuracy score. This example (**Figure D.3.1**) is derived from the Train_inference.py file.

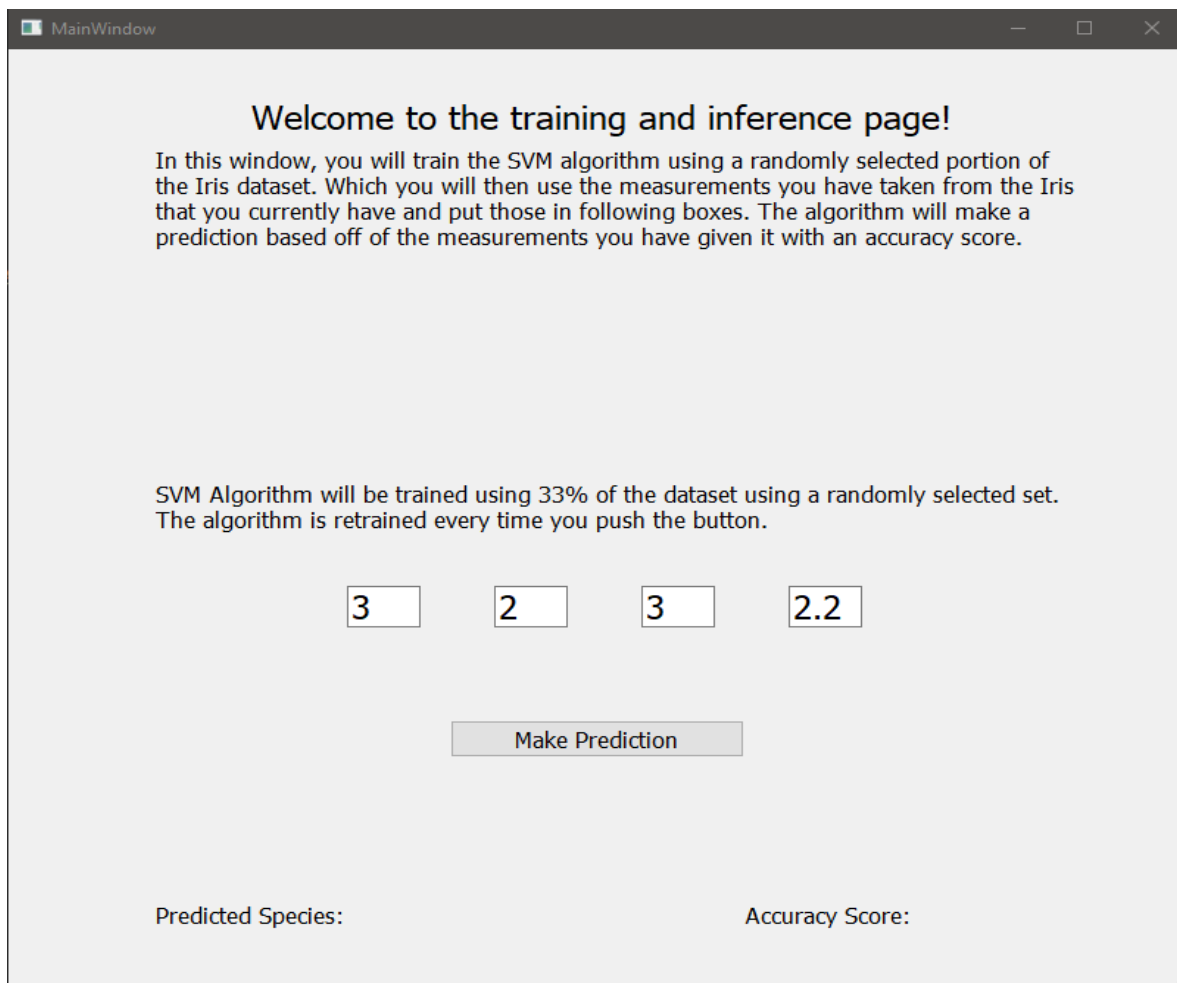


Figure D.3.1 – Percentage of privacy filtering applied to a dataset

Springfielding Greenhouses

- 2.) A confusion matrix can be used to help a user understand how the percentage for the prediction was made.

The confusion matrix is a representation that shows the measurements of what is predicted versus what was actually correct. It measures the true positive from the false negative predictions and shows that in a setup where the correct predictions are where the two axis meet. **Figure D.3.2** shows the confusion matrix as it pops out from the Training and Inference page but still shows the predictions on the page under it.

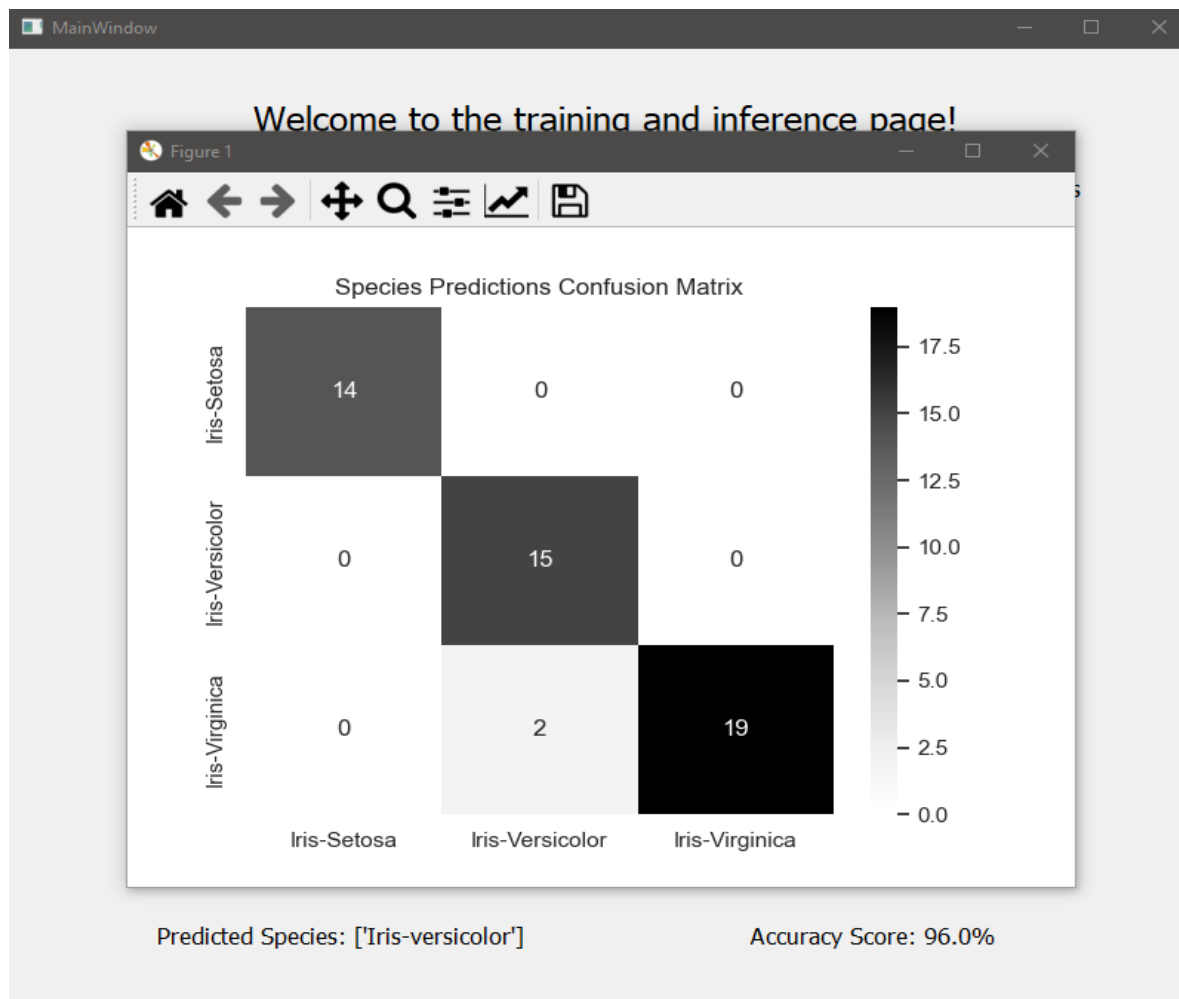


Figure D.3.2 – Confusion matrix displayed over prediction and accuracy

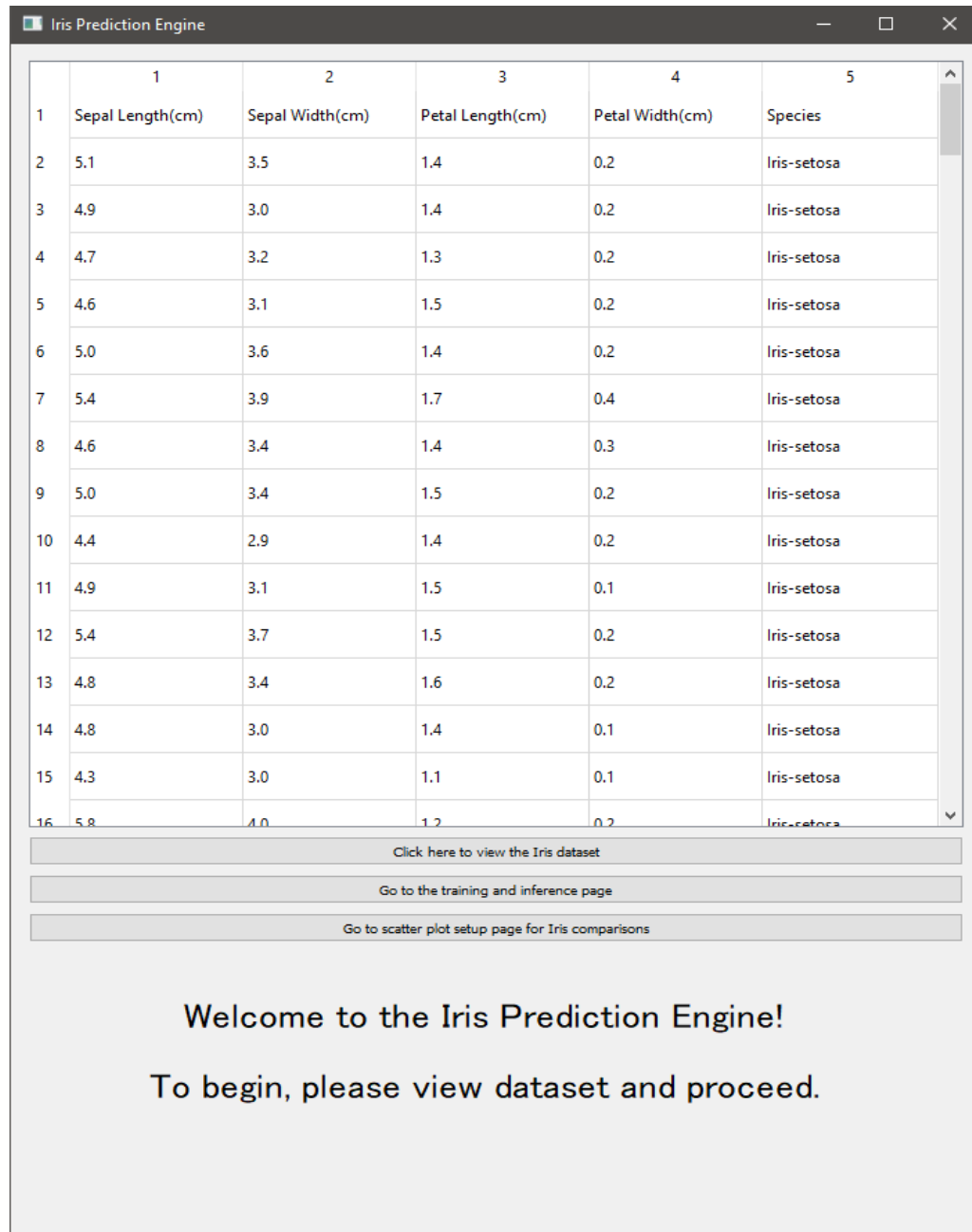
Descriptive Tools:

- 1.) A Database view of a specified Dataset

A database view is provided where the user can analyze the dataset with the classification headers. This allows the user to see what is really going into the algorithm when it is training.

Springfieldington Greenhouses

The IrisMainWindow is where this is visible as well as the main hub where all of the other pages are accessed from. See **Figure D.3.3** to see the main window example.



	1	2	3	4	5
	Sepal Length(cm)	Sepal Width(cm)	Petal Length(cm)	Petal Width(cm)	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16					

Click here to view the Iris dataset

Go to the training and inference page

Go to scatter plot setup page for Iris comparisons

Welcome to the Iris Prediction Engine!

To begin, please view dataset and proceed.

Figure D.3.3 – Main menu currently displaying the unaltered employee dataset

2.) Scatterplot of the dataset by dimension choices

Springfieldington Greenhouses

There is a button on the main screen that takes the user to a window with two dropdown boxes that are choices of dimensions that they can choose to act as the axis for a scatterplot (**Figure D.3.4**). It is a dependent setup so the user cannot choose the same dimension for both sides. Once the choice is made for the first box, that limits the choices for the second box. Once the second choice is made, a scatterplot window appears showing the different species by the dimensions selected. This shows the user the relations between the different species by their measurements in a more graphical way. See **Figure D.3.5** for the scatterplot.

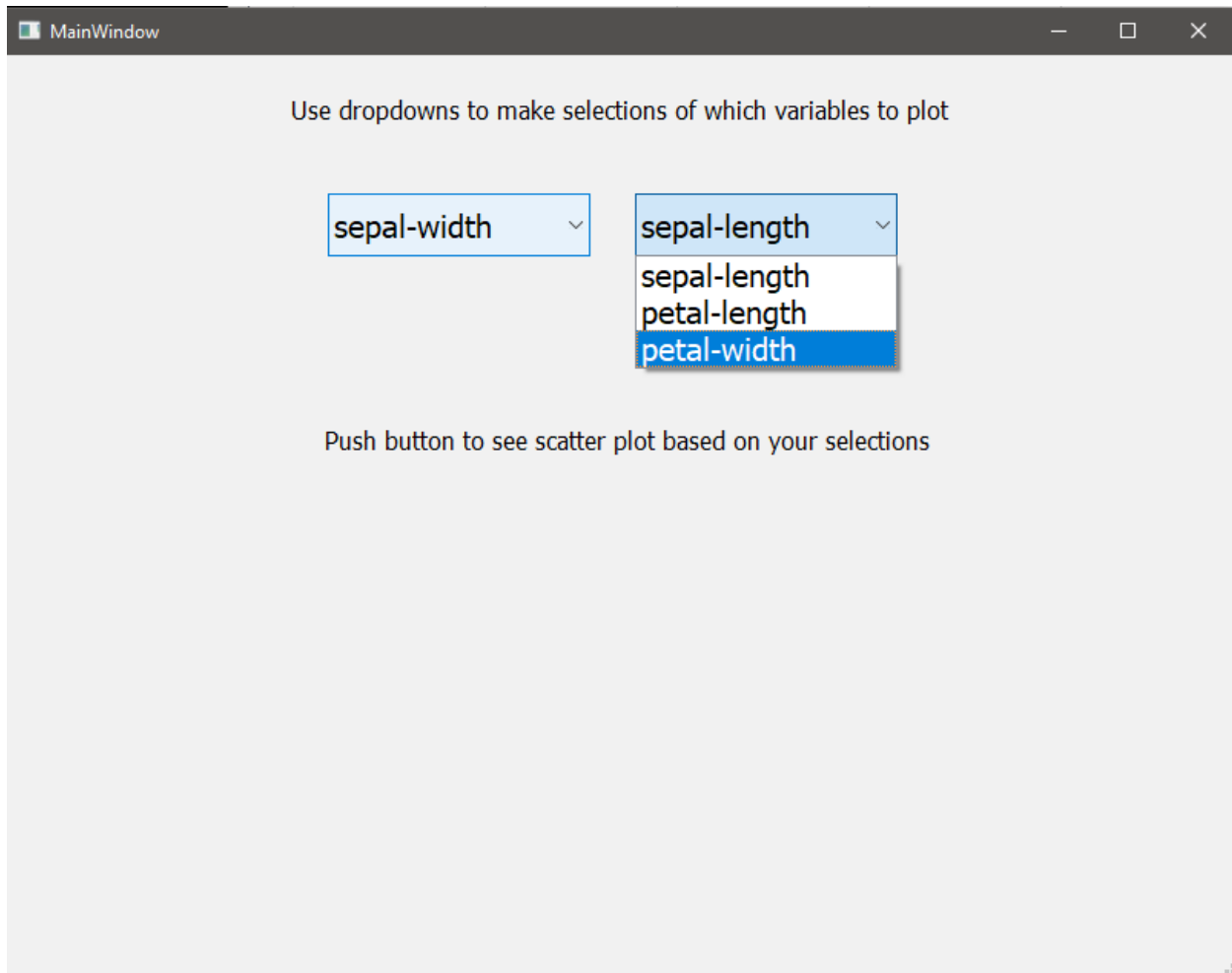


Figure D.3.4 – Scatterplot choices in scatterplot window

Springfielding Greenhouses

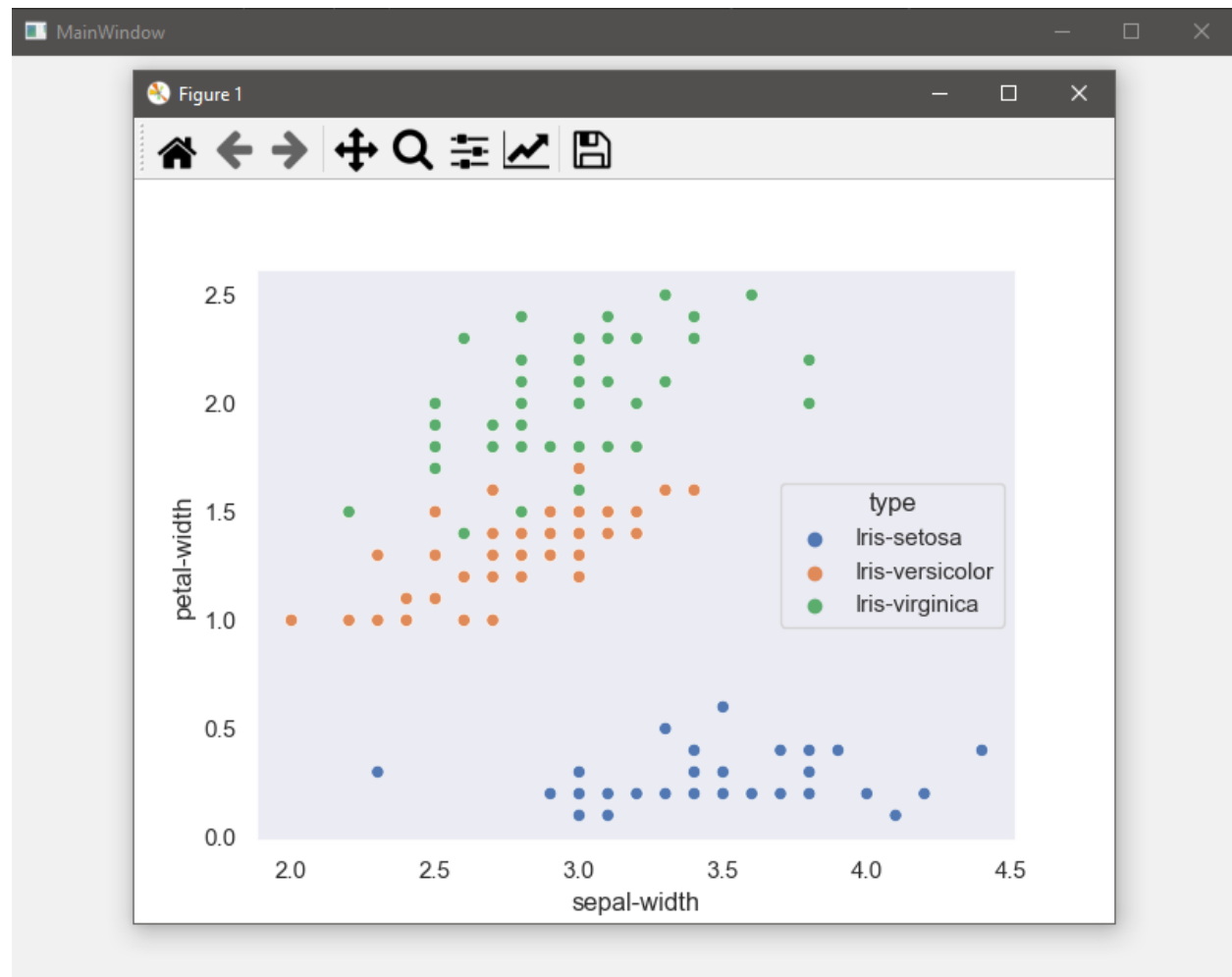


Figure D.3.5 – Interactive scatterplot after dimension selection

D.4 Hypothesis Verification

The initial hypothesis for the IPE was that the accuracy score would remain above 90% for 99% of the usage time. In our unit testing that was proven to be correct with the accuracy score never dropping below 92%. The SVM algorithm is a high accuracy model that runs at n-cubed time so it will take longer to get results as it approaches the theoretical data threshold. This should not happen for quite some time though as putting in dimension measurements by hand will take quite a while to fill the dataset.

D.5 Visualizations and Effective Storytelling

Section D.3 goes more in-depth on the visual elements of the IPE but they will be discussed a little bit here in this section. The particular visual aspects of the IPE allow the user to understand

Springfieldington Greenhouses

what it is that the IPE really does by showing them the dataset and then allowing them to see how it is used. The ability to compare different dimension measurements of the different species and see how closely related some of them can be shows how amazing the IPE predictions really are. After seeing the relationships between the dimensions the user then can see the confusion matrix along with the accuracy prediction that further reinforces how much better the IPE can be with more data.

The mention of the random selection of the training data allows the user to know that the IPE is always learning newer stuff depending on how stale the data in the dataset is. In order to become better at its function, the IPE needs more data to be able to become more accurate. The randomization of the training data is extremely important due to the nature of real life learning. Repetitive material will not allow for actual growth when it comes to learning. This can be said for sentient creatures as well in that the environments that they live in are constantly in flux. They need to learn and adapt to their environments. So, we should offer the same to the machine learning algorithms the same setup by keeping that randomization in their learning and training.

D.6 Analysis of Accuracy

As it has been mentioned throughout this document, the overall project was to design a prediction engine that would stay above 90% accuracy for 99% of the usage time. Throughout the testing and initial usage phases it has never been seen to drop below 92% on the small 150 rows of data that it has access to. The randomization of the 33% of the training data makes that accuracy score all the more impressive. With more data, the accuracy will do nothing but go up getting closer and closer to 100%. It is said statistically that humans have a 3% error rate at everything that they do. So, getting to a consistent 97% would put the IPE on the same lever as a human doing the same thing.

D.7 Testing of Application

The testing of the application has been discussed in previous sections but it will be reiterated here. The tests can be broken down into two distinct lifecycle phases.

Phase One: Demo testing and training phase

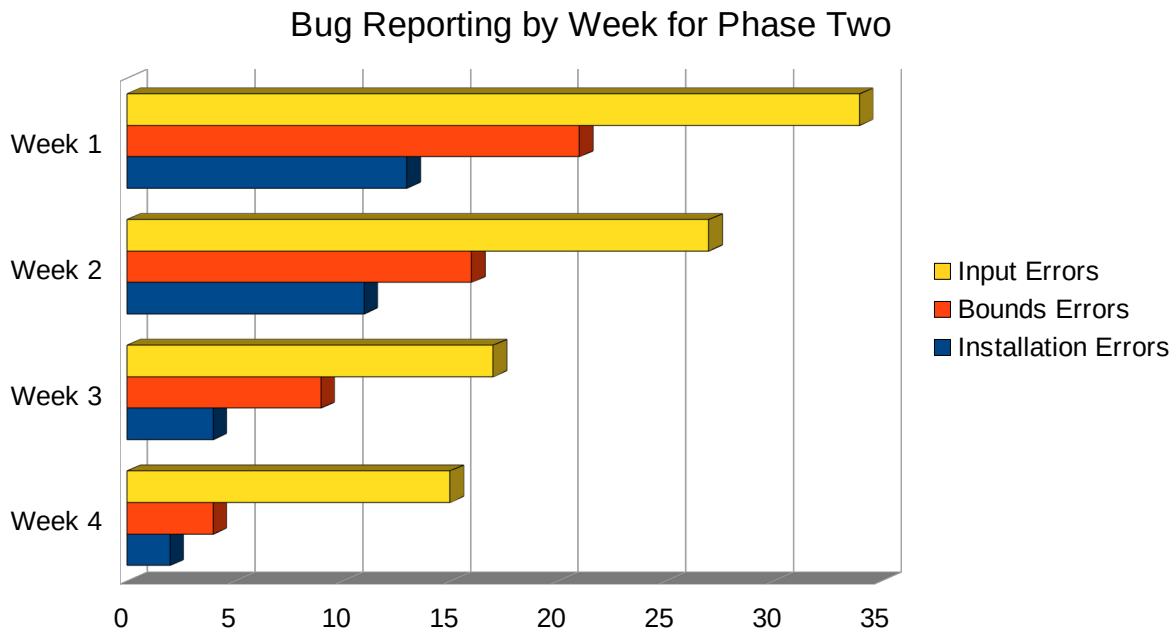
This phase was getting the IPE out to the main greenhouse to start having the employees get familiar with how it works and what they can expect. It was more of a “get to know the product” phase. We then collected sentiment about the ease of use and appearance of the IPE over this phase that was then be utilized in the second phase.

Phase Two: Updates, features, and bugs

Springfieldington Greenhouses

This phase was utilized to incorporate some the features that the users during phase marked in their sentiment discovery. These were then built into the application to be of better and easier use for the users. There were some bugs that were noticed over phase one that were taken into account and dealt with during phase two.

Aside from the testing that is described above, there were demonstrations that were done for the main stakeholders before signoff. Below is a bug tracker of over the period of phase two:



D.8 Files of Application

All of the required files to view and run the initial demonstration version of the application can be found at the following URL: https://github.com/735783D/C964_JasonWhitby_Capstone

The files that are in the github repository are also submitted with this documentation. The project is designed to be run in an IDE that supports Python 3.9. For the application to successfully run you must install the PyQt5 libraries. Plus, you will need to include the following Python libraries: scikit-learn, seaborn, pandas, matplotlib, and csv. Once everything is installed the application should run smoothly.

A brief description of the main files in the repository is provided for you below:

IrisMainWindow.py – Main application window and should be the file that is run to start the application. Navigation for the application through buttons and contains tableview capability for dataset viewing.

Springfieldington Greenhouses

Train_Inference.py – Window that allows for the user to input the measurements they have taken of the iris to be predicted.

Scatter.py – Allows the user to select the specific dimensions they would like to see a relational scatterplot for.

Iris.csv – Treated dataset file that has a header to define the columns for the tableview in the IrisMainWindow.py file in a .csv format.

Iris2.csv – Untreated raw dataset file that is consumed by the SVM algorithm on the Train_Inference.py page as well as the Scatter.py page in a .csv format.

For further assistance, please see User's Guide below

D.9 User's Guide

Evaluators and users trying to run the application, please follow the walkthrough below:

- 1.) Unzip the associated files and place them into a project folder.
- 2.) Make sure that you have at most python 3.9 installed on the desktop you are accessing the program from. This can be done by navigating to your local command prompt and entering "python --version". The version is very important due to PyQt5 not being supported past Python 3.9. If Python is not installed on your system please see the Advanced Trouble Shooting section below.
- 3.) Load that folder into an IDE that can support Python 3.9 (It is recommended to use [JetBrains Pycharm Community Edition 2023.1](https://www.jetbrains.com/pycharm/)).
- 4.) Download and install the PyQt5 library.
 - a. This can be done in PyCharm by navigating to file/settings/project/project interpreter and locate the "+" on the right-hand side of the menu. From here, type in the package names and install them into your environment. There are ore detailed instructions at the following link: <https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html>
- 5.) Open up the python file called IrisMainWindow.py and start the application by right clicking the file name and selecting "Run". Tip: if you cannot run the file this way, hold down your shift key and press F10, or you can left click on the green arrow on line 90 then click "Run 'IrisMainWindow.py'"
- 6.) Once the application is running you can use one of the three buttons to navigate other pages each of which have directions in them.
 - a. "Click here to view the Iris dataset" button pulls up the table of the data currently in the Iris.csv for you to view.

Springfieldington Greenhouses

- b. “Go the training and inference page” button takes you to a page where you put in your measured dimensions for the petals and sepal for the algorithm to make its prediction.
 - c. “Go to scatter plot page for Iris comparisons” button takes you to the page where you can select your dimensions to see comparisons and relationships of the different species and measurements in the dataset.
- 7.) On the Training and Inference page you will need to fill in all of the boxes with values to avoid getting an error. Once you fill in all of the boxes you push the button and you will see a confusion matrix pop up along with a change on the lower part of the window showing the updated prediction and accuracy score.
- 8.) On the Scatterplot Comparisons page you will find two dropdown boxes that need to be clicked on to select the dimensions. Though there is something in the left box already, you still need to click on it and select a dimension for it to register the selection. Then, you can select your second dimension in the box on the right. Once the selections are made, an interactive scatterplot will show up. You can click on it and drag it around to see all of the information.

Advanced Trouble Shooting:

- 1.) If you cannot locate Python on your computer, or the wrong version is installed, follow this link to properly install Python and set the correct path variable for Windows 10:
<https://phoenixnap.com/kb/how-to-install-python-3-windows>
- 2.) If you cannot locate PyQt5, you can manually install the packages by using this link:
<https://www.riverbankcomputing.com/software/pyqt/download5>