USER MANUAL

SYSC 3110 – Simple Machine Learning Framework

Joe Samuel and Eric Bedard

AI Dev | Carleton University

Version 3.0

Designed and authorized for use on Milestone 3 of AI Dev project.

AI Dev team members:

Dhrubomoy Das Gupta (100917323), Eric Bedard (101009607), Jobin Joe Mathew (101006153), Joe Frederick Samuel (100998314)

Table of Contents

1. Opening the executable project file

The project can be run without having to access the source code by using the executable project file attached in the project folder. To open the executable project file, you must first download the knn.jar (Java ARchive) file attached in the project folder. Opening up a JAR file depends on system type. Instructions for operation on Windows, macOS and most Linux distributions are listed as follows:

- **Download Java for your computer if it isn't already installed**. If you don't already have Java installed, you can go to Java's website at https://www.java.com/en/download/ and click the **Free Java Download** button below the latest version of Java. Install Java once download is complete.
- **Extract** the knn.jar file using a file unarchiver (e.g WinRAR, Unarchiver) or command line
- You will now be able to view the executable file view-exe.jar and knn-exe.jar.
- In order to see the result for the new ML problem for Milestone 3 (**tabulated_data.xlsx**), open **Terminal (Mac/Linux)** or **Command Prompt (Windows)** and **change directory (cd)** to the extracted folder and run the following command:
-

```
java -jar knn-exe.jar
```

-

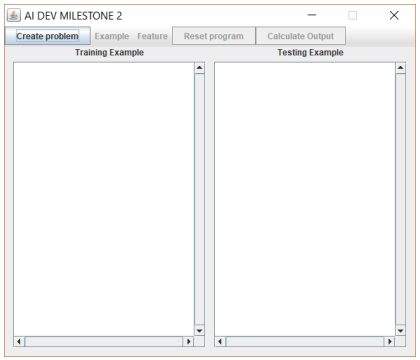
You should see the following output:

```
Prediction: (knn = 7)
Prediction for Testing Example 'Testing Example 1' is: Kick
Prediction for Testing Example 'Testing Example 2' is: Kick
Prediction for Testing Example 'Testing Example 3' is: Kick
Prediction for Testing Example 'Testing Example 4' is: Dash
Prediction for Testing Example 'Testing Example 5' is: Turn
Prediction for Testing Example 'Testing Example 6' is: Turn
Prediction for Testing Example 'Testing Example 7' is: Turn
Prediction for Testing Example 'Testing Example 8' is: Dash
Prediction for Testing Example 'Testing Example 9' is: Dash
Prediction for Testing Example 'Testing Example 10' is: Turn
Prediction for Testing Example 'Testing Example 11' is: Turn
Prediction for Testing Example 'Testing Example 12' is: Dash
Prediction for Testing Example 'Testing Example 13' is: Dash
Correct label prediction: 66.66666666666666%
```

- In order to run the GUI, double click **view-exe.jar** or open **Terminal (Mac/Linux)** or **Command Prompt (Windows)** and **change directory (cd)** to the extracted folder and run the following command:

```
java -jar view-exe.jar
```

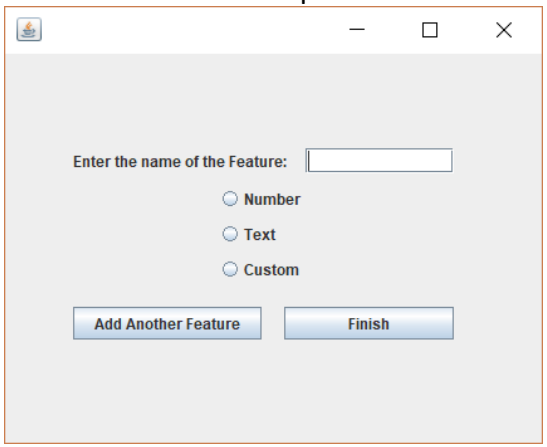
- The program should look like this:



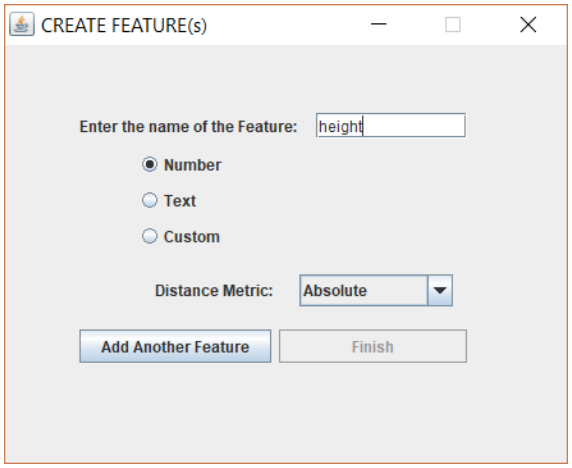
2. Using the program

Creating Features

- First, click the “Create Problem” Button at the top left of the window. This window should appear:



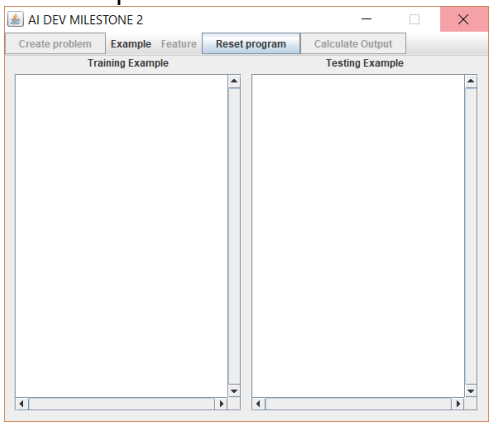
- The “enter the name of the Feature” field will be the name of the feature, and the radio button chosen will be the type of value in the feature. The drop-down menu is used to select the Distance Metric to be used. For example:



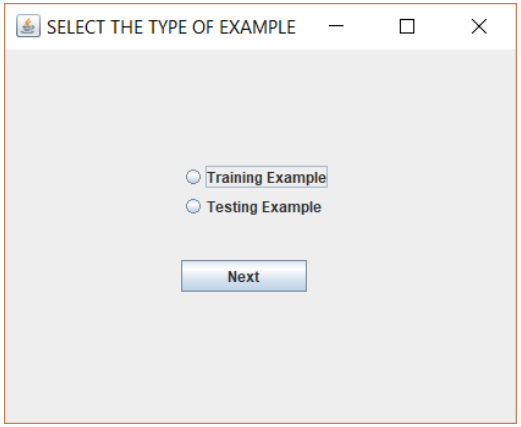
- Choosing the “Add Another Feature” button will save the current selection for the feature and then ask for information on another Feature that you would like to use. When you have created as many features as you like, click “Finish” after putting in information for the last feature.

Creating Examples

- After creating features, the “create example” menu will become available as seen below. Select this menu Select the available sub-menu to create an example.

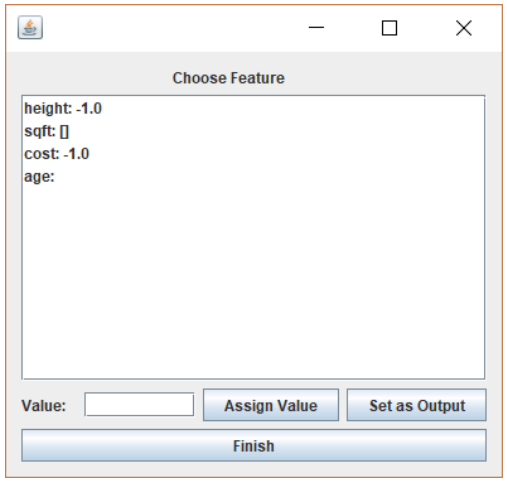


- The following popup will become visible. Select the type of example you would like to create. A TestingExample will be an example contain no value for “output” as it will be predicted by the KNN program. TrainingExample’s will be used to predict the output of the TestExample.



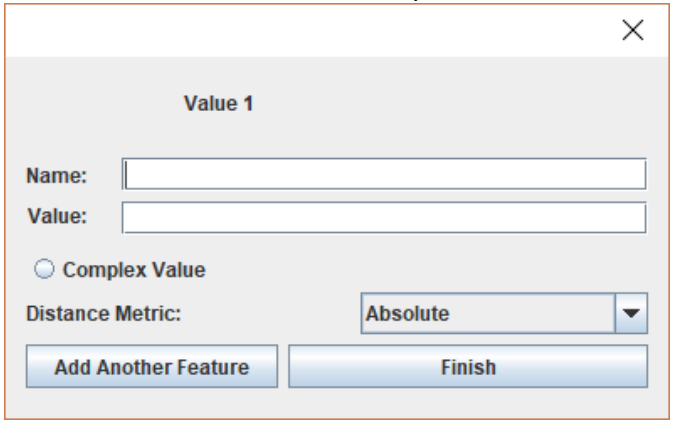
- Another popup will follow, which will determine the name of the Example

- The next window will determine which features will be in the example. To assign a feature select it from the list, add a value in the “Value” field and press assign value. Please do not select a feature more than once and make sure every example has the same features.



`

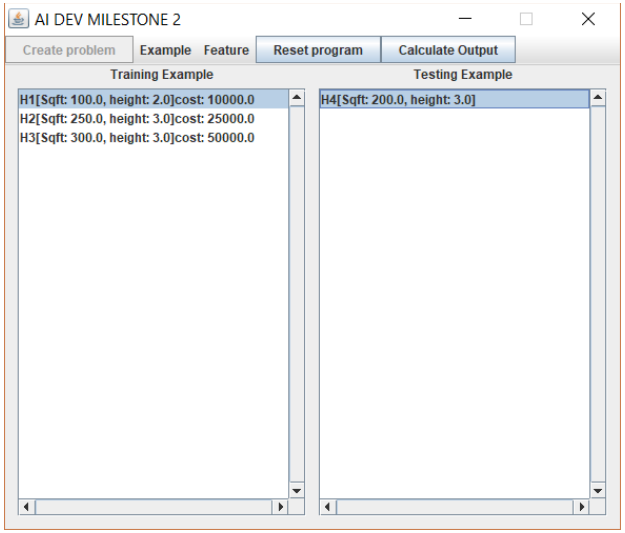
In the case of a custom feature, some extra steps must be taken. When “assign value” is pressed with a complex feature selected, a window will appear. Input a name and value to add a child feature to the custom feature. If you wish to add a custom feature as a child feature, select the “complex Value” radio button. The same window will open to add child values.



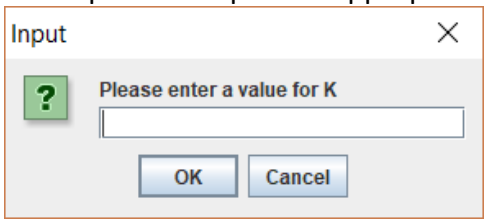
For the training example, an output feature will need to be chosen. The process is the same except instead of pressing “assign value” you should press “set output”. When making the TestExample, the output feature should not be added.

Calculating/making prediction

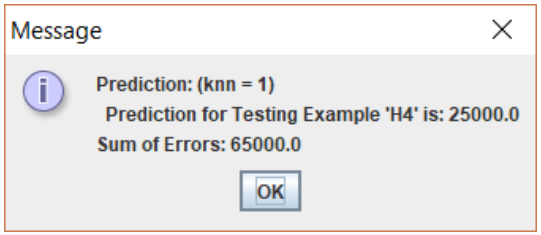
- Once you have selected your desired number of examples you may move on too predicting the output for the testExample. An example of Examples that are ready to calculate:



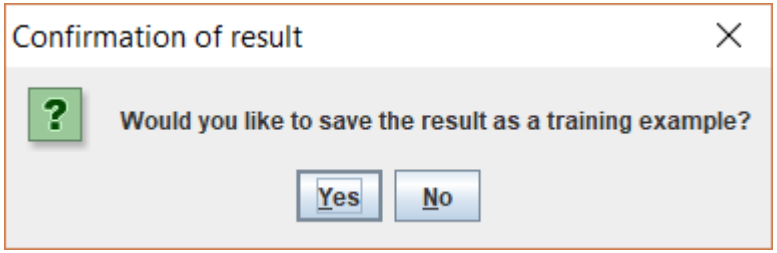
- A popup will appear, and you will be required to input the appropriate K for your problem.



- The next window will show the predicted output and sum of error as seen here:



- Once you press ok, the program will ask if the prediction should be put with the TrainingExamples with the predicted output set as the output.



Editing/deleting examples

- If you would like to edit or delete an example, you must first select it from the List. You can then select edit or delete from the "example" menu. Selecting "Delete Example" will remove the example from the List. Selecting "edit example" will allow you to change the title of the example.

Resetting

- To reset the program, press the "reset program" button on the main window. This will clear both list of examples and remove any previously created features.

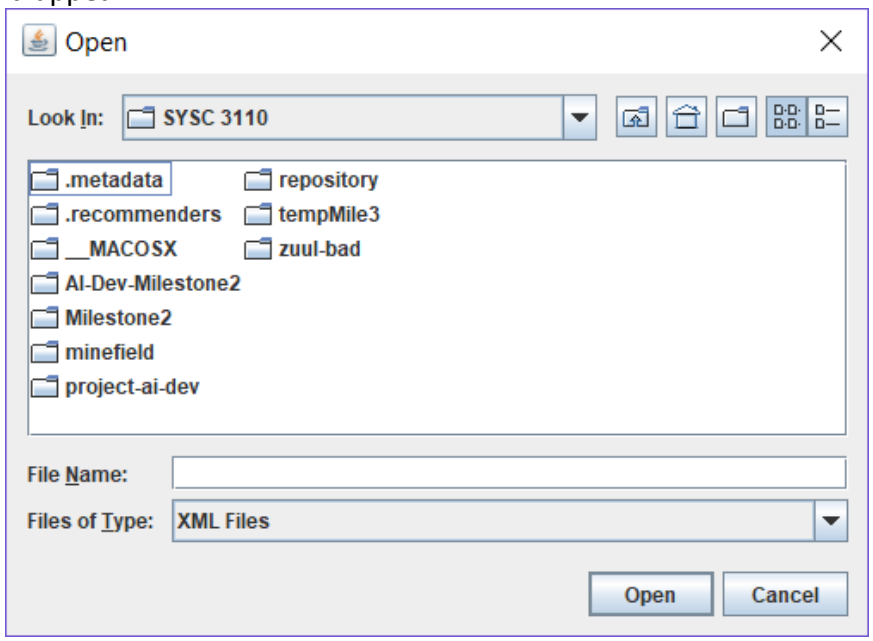
Exiting

- To exit the program, press the red "X" at the top right of the window.

3. Importing and Exporting

Importing

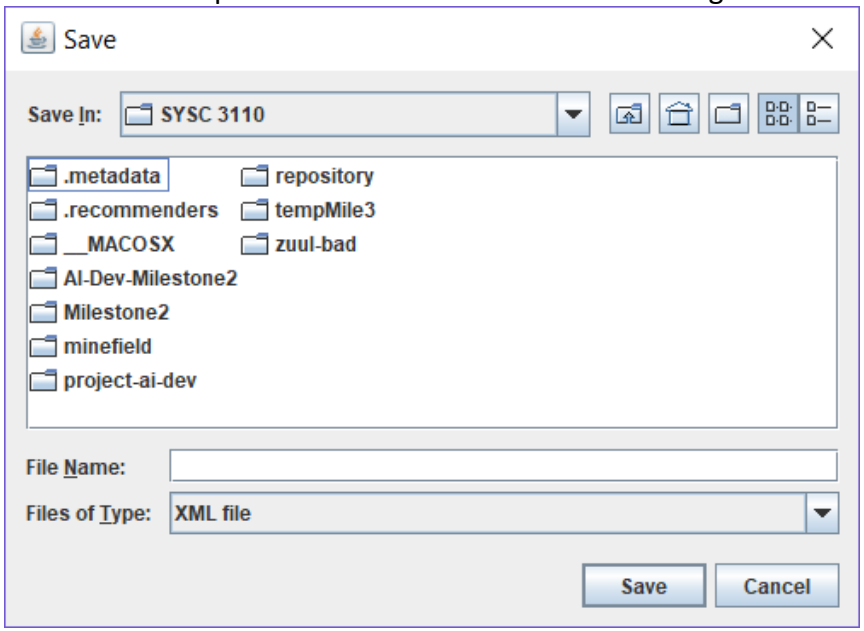
- Click on the “File” menu at the top right of the screen, next to the “Create problem” button. Select the “import” submenu. A new window should appear:



- Navigate the folders to find the file you would like to import. Select it and press “open” to import the Examples.

Exporting

- To Export already created Examples, press the “File” menu at the top right of the screen next too the “Create problem” button. Select the “save as XML” option from the submenu. The following window should appear:



- Navigate the files to select the location you would like to save the examples in. Give the file a name and press “save” to save the examples in XML format.

4. Opening the source code for the project

To open the source code of the project included as java classes in the project folder, you will need to use a Java IDE such as Eclipse Java (https://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/marsr) or BlueJ (https://www.bluej.org). Instructions for Eclipse IDE for Java has been listed below. Instructions for most IDE's should be similar in operation. For detailed instructions on your IDE import features, please refer to their user manual.

On Eclipse IDE:

- Launch Eclipse IDE and select **Import** from **File** menu
- In the **Import** dialog box, expand **General** folder and select **Exisiting Projects into Workspace** and click **Next**
- This will display the **Import Projects** dialog box. Choose **Select archive file** and click **Browse**
- Navigate to the folder of the exported file. Select the file and click **Open**
- In the **Import Projects** dialog, ensure that browsed path is displayed and click **Finish**
- Ensure that the imported project is displayed in the Eclipse IDE

To view the individual java files, **Expand** on the project > **Expand** on the **src** folder > **Expand** on **default package.**

URL Catalog

Java for PC: https://www.java.com/en/download/

Eclipse IDE for Java: https://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/marsr

BlueJ IDE: https://www.bluej.org