Final Report for Team Team CHR’s

The Exploration of Classification Algorithms

**I) Team Members**

Jonathan Houston

Ryan Chen

Jasmine Ramirez

**II) Design, Implementation & Testing**

For this project, our team built an application that allows for the classification of data sets using three different algorithms: K-Nearest Neighbor, Random Forest, Multinominal Logistic Regression. This work was be based on published works, either derived from textbooks or conference/journal paper, as well as existing techniques and algorithms for determining the classification of data sets. The application includes a simple user interface, created with the python extension tkinter, which allows the user to select the desired classification algorithm they would like to use as well as the data files they would like to use the algorithm on. The algorithms themselves perform quality evaluations and report statistical performance measures that are compared to their more well-known counterparts (SKLearn implementations) in order to capture their accuracy and completeness. Simple testing was carried out making sure the proper outputs based on the given inputs and selected algorithm. Testing was performed by everyone in the group to ensure not only accuracy but portability as well.

**III) User’s Manual**

Jupyter Notebook:

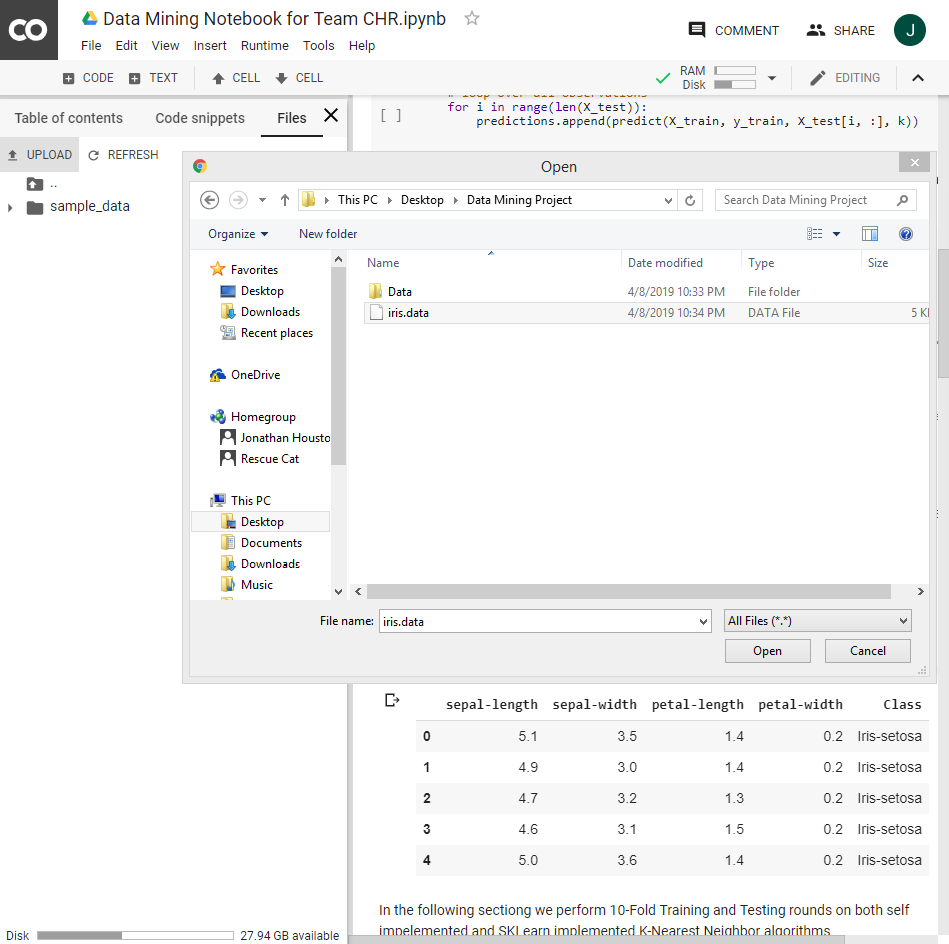
This notebook is included in the zip and is titled “Data Mining Notebook for Team CHR.ipynb”. It can also be accessed via the following link:

<https://colab.research.google.com/drive/1kjGfokONFLqqcED4xE7QTHgUCXcuzjzl#scrollTo=I6DLKHabzHE9>

If you use the above link each cell can be ran, but every cell must be ran sequentially to ensure packages and data are loaded correctly.

If you are using the above link, files need to be uploaded via the upload function:

1. Expand the left most pane
2. Choose Files Option inside the top of the expanded pane
3. Click Upload – A file browser will be opened and you will have the opportunity to find the data file (ie: iris.data and car.data). The files are located in the Data folder inside the submitted zip file.



1. Click Open in the file browser

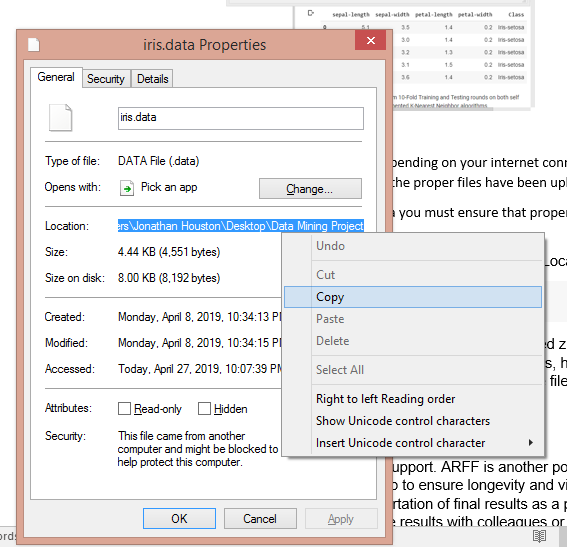
Note: Upload speed will vary depending on your internet connection. You cannot begin running cells until the proper files have been uploaded

If you are running this notebook in anaconda you must ensure that proper paths are set for the data files.

1. Note here “iris.data” must be changed to “Some/File/Location”:



1. Simply navigate to the Data folder inside the submitted zip file, right click on the corresponding data file, choose Properties, highlight the entire “Location” of the file, paste the location in front of “iris.data”



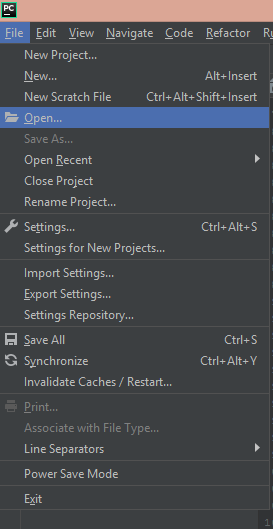
1. Note that the entire path name uses backslashes and each backslash must be escaped by another backslash in order for the path to be read:



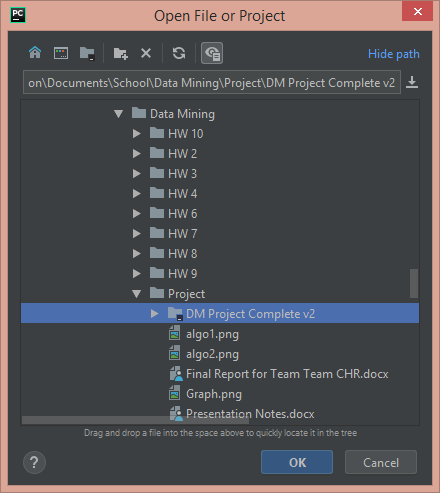
User Interface Program:

The program was ran and built in Pycharm but can be ran in any Python IDE.

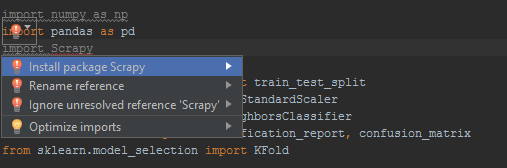
1. Find the zip folder titled “Program” and unzip it.
2. Open Pycharm
3. In the top left select File > Open



1. In the navigator highlight the unzipped folder and select Ok

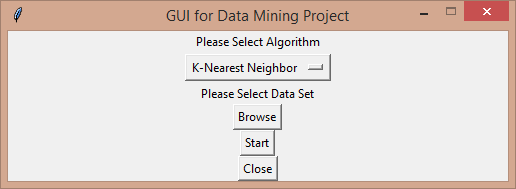


1. Note that some of the libraries may not be installed, this can be resolved by clicking on the line with the uninstalled library, a red light bulb will appear, click on the red light bulb, from the drop down select Install Package.

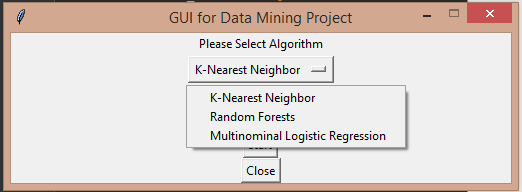
\

Note: installations may take some time, the console at the very bottom will notify you when libraries are finished installing.

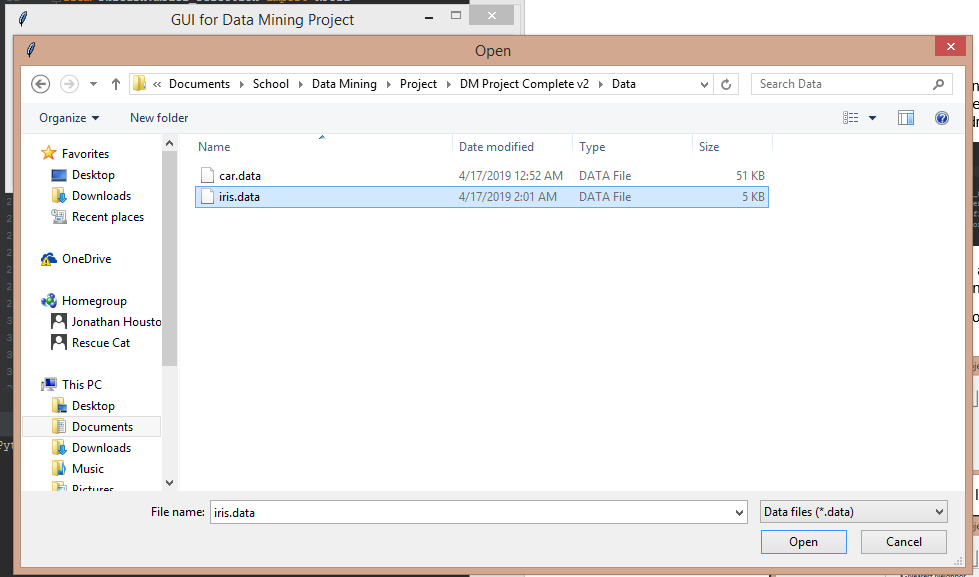
Once the proper libraries are installed you can run the program by pressing the green play button at the top and the UI will appear.



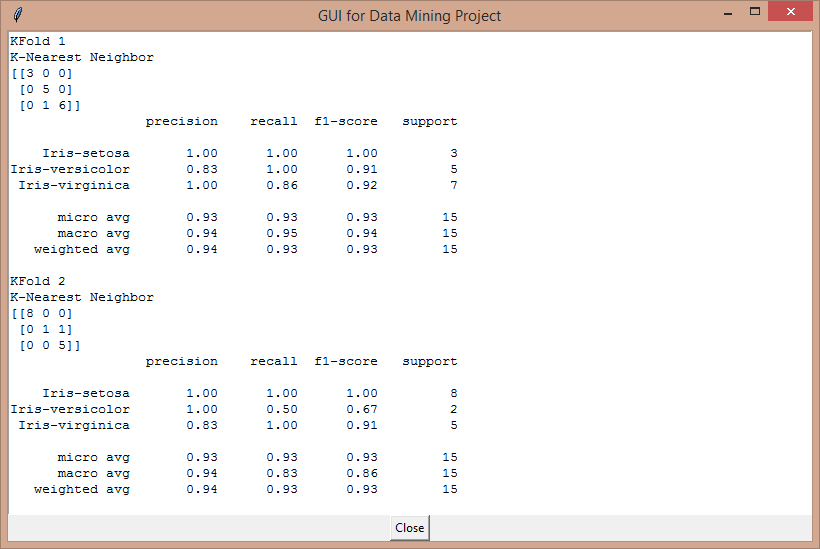
You can select the algorithm you would like to run from the drop down



Click the browse button and a file navigation window will appear, navigate to the data folder and find the desired data file. **NOTE**: KNN and Random Forest are only compatible with the “iris.data” file and MLR is only compatible with the “car.data” file at this time



You can run the algorithm by pressing the start button and another window will pop up. This window will contain all the useful data collected from each round or testing/train including confusion matrix, f-scores, recall, precision, as well as statistical data at the very end.



When completed you can close the window and click close on the main UI.

**IV.) Future Extensions**

In the future we would like to implement ARFF support. ARFF is another popular format for large data and implementing support for it would help to ensure longevity and viability. We would also like to implement a function that allows exportation of final results as a pdf. I believe being able to export results would greatly help share the results with colleagues or friends.

**V.) Experiences**

I had a pleasurable experience building this program with my team. Everyone was willing to do their part and meet expectations. I feel satisfied with the final results but I also know that more functions could be implemented to make it a well-rounded program and the UI could be changed to make it more visually appealing. – Jonathan Houston

**VI.) Work Completed by Team Member Name**

*Implementation of K-Nearest Neighbor*: Jonathan Houston

*Implementation of Random Forests*: Ryan Chen

*Implementation of Multinominal Logistic Regression*: Jasmine Ramirez

*UI Implementation*: Jonathan Houston, Ryan Chen, Jasmine Ramirez

*Jupyter Notebook*: Jonathan Houston, Ryan Chen, Jasmine Ramirez

*Proposal, Progress and Final Report*: Jonathan Houston, Ryan Chen, Jasmine Ramirez

*Presentation*: Jonathan Houston, Ryan Chen, Jasmine Ramirez

**VII.) References**

Fisher, R., & Marshall, M. (1988, July 01). Iris Data Set. Retrieved from http://archive.ics.uci.edu/ml/datasets/Iris

Han, J., Kamber, M., & Pei, J. (n.d.). *Data Mining: Concepts and Techniques*(3rd ed.). Morgan Kaufmann.

Polamuri, S. (2016, September 25). Classification and clustering algorithms. Retrieved from http://dataaspirant.com/2016/09/24/classification-clustering-alogrithms/