Seasons of Southern California Weather

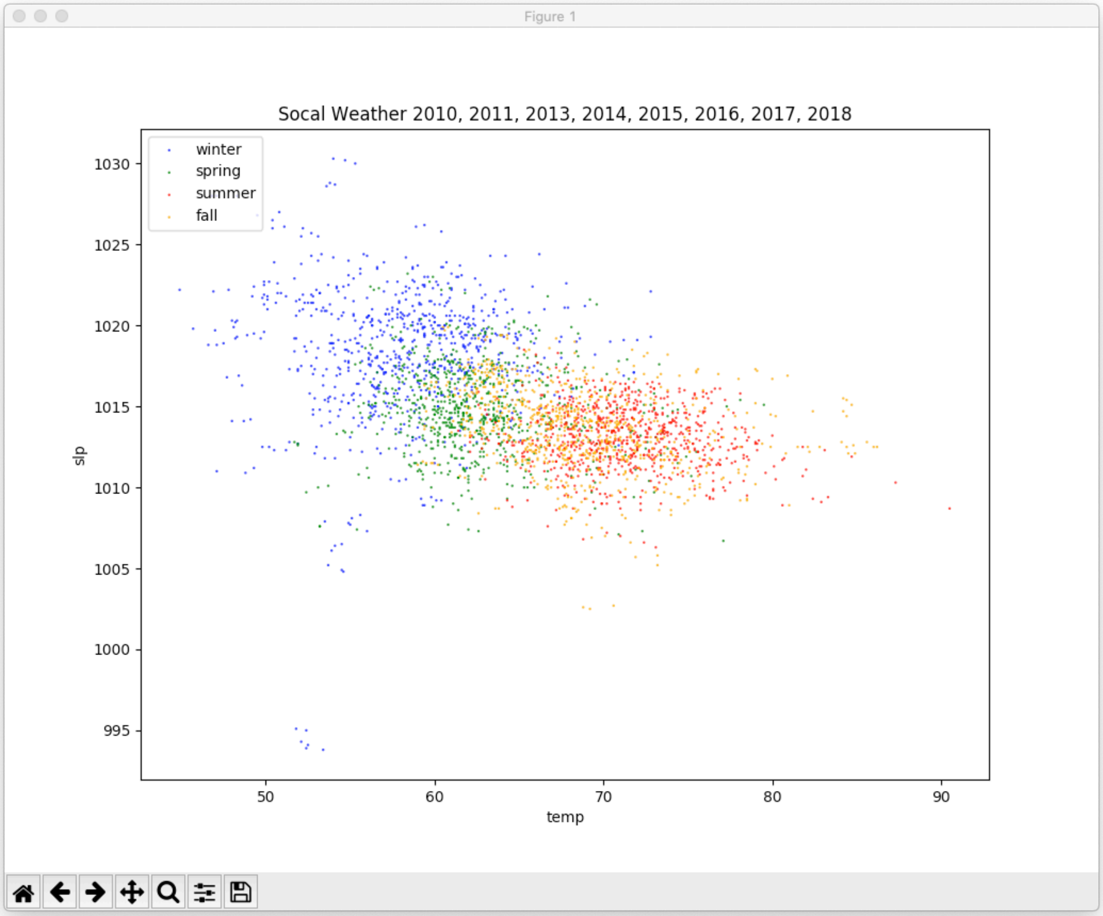
By Allen Sarmiento, Ryan Shim, Andrew Lopez

Goal:
Be able to classify the season based on the temperature and sea level pressure.

How to compile:
Make sure the makefile is available and execute the following:

```
make
./main
```

Analysis:
This is a scatterplot for the following years: 2010, 2011, 2013, 2014, 2015, 2016, 2017, 2018. 2012 is not included because the data made the plot difficult to read. Below is a scatterplot of our data:



From looking at this scatterplot, we could either cluster based on winter and summer only, or implement hierarchical clustering to try to identify all 4 seasons. Due to the overlapping data points, we decided to just compare both winter and summer.

Pseudocode for KNN:

1. Allocate memory for a months array and a distance array
2. Generate data from csv fiels. Partition training and test data of 72 and 25 percent, respectively
3. For each testing data, compare each point with each training data and calculate the Euclidean distance in parallel
4. Store the calculated Euclidean distance in the distance array along with the corresponding month
5. Sort the distances array alongside the months array to maintain the order in parallel
6. Get the smallest knn distances from the distances array as well as the corresponding month
7. Calculate the most occurring months and set that as the prediction
8. If there is a tie, the first month is selected
9. Continue until every testing data point creates a prediction
10. Print out the number of correct and incorrect predictions

Shortcomings:

Some of the shortcomings we encountered in this project are the following:

- Excluding the year 2012 due to many outliers
- Memory management could have been better; our program is allocating a lot of memory to hold the distances for the testing data, so a better method may be to store values in a smaller array and when the array is full, we perform an operation to remove unnecessary values
- Sequential run time was better due to a better algorithm. The sequential algorithm allocates an array the size of knn, and sorts that array while calculating Euclidean distance. We tried to implement this method in parallel, however we ran into issues with communicating the current status of the array. Instead, for parallel, we decided to store every Euclidean distance and sort the entire array, which we assume is taking up majority of the run time.

Conclusion:

Based on the given temperature and sea level pressure, we were able to identify the season as either summer or winter with over a 90% accuracy. Due to the data only being in Southern California, we removed the Spring and Fall seasons due to overlapping data. If we had data for a different location with a wider variety of data, we could possibly be able to predict all 4 seasons.