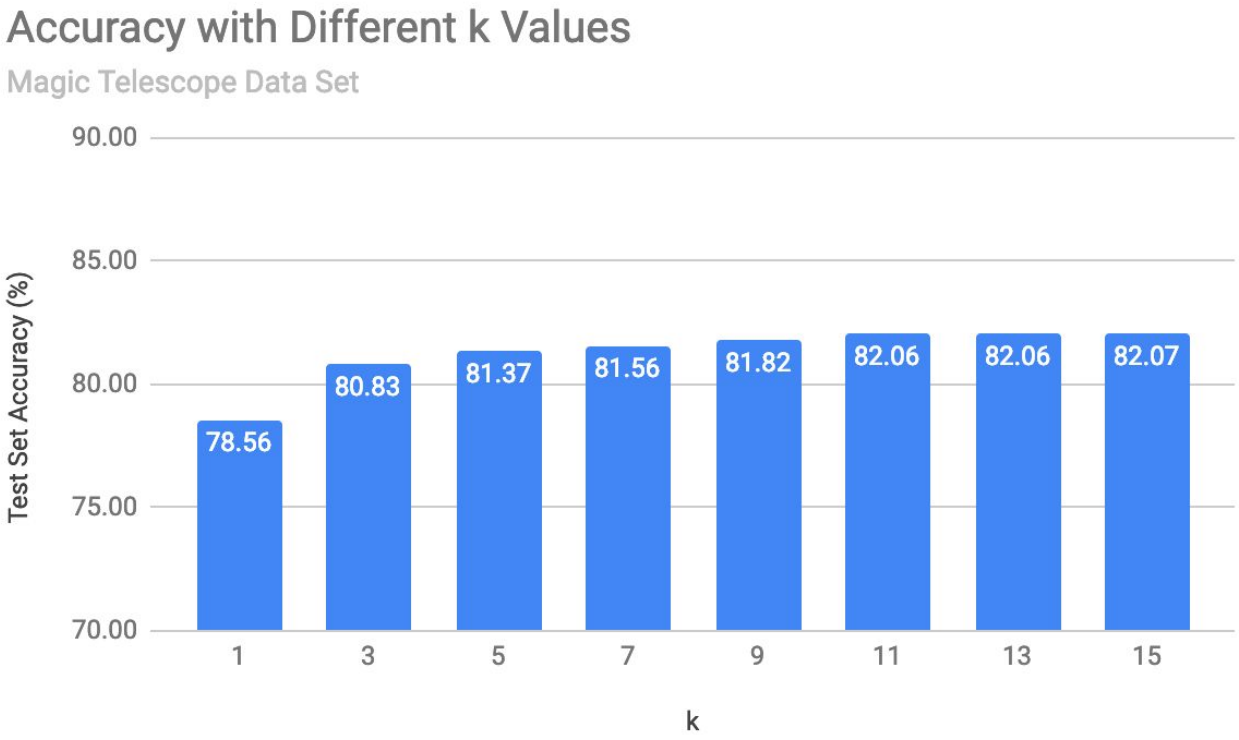
K Nearest Neighbor

1. I correctly implemented the K Nearest Neighbor Algorithm for both classification and regression and included the option to use distance weighting for both classification and regression. The filename is knn.py

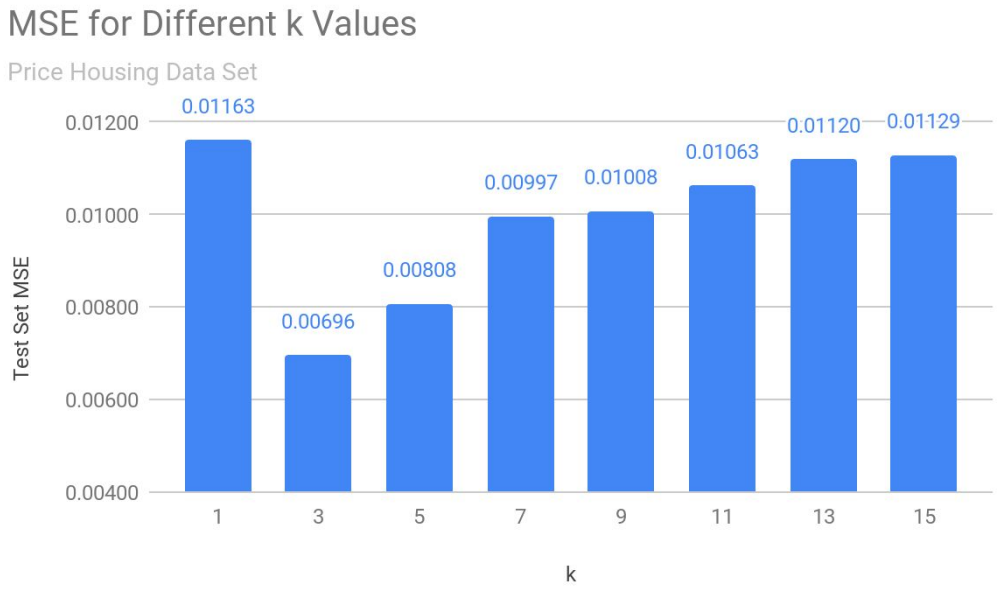
2. On the magic telescope dataset, I used k=3 and tried both normalized and without normalized data. The resulting accuracies of these normalized vs not normalized show that normalizing the data for k=1 were .79 and .75 and k=3 .808 and .815 showing an increase of accuracy of about .02 on average for normalized data.

   Using the normalized data, I tested out k values between 1 and 15 only using the odd values and compared the accuracies. It seems that after k=11, all the values are equally the best. The graph below shows the results of this test.



Graph of the magic telescope data set with accuracies of the Test set on the y-axis at different k values on the x axis. Notice how the accuracy increases more from k=1 to k=3 than from any other change. These results conclude that the KNN algorithm does not benefit from a large number of points/k values after a about k=5.

3. Using K Nearest Neighbors with regression on the house pricing data set resulted in a decrease and then an increase in MSE as the k value increased. The optimal k value is k=3.



See graph below to see the MSE over odd values of k between 1-15.

Graph of the price housing dataset with the MSE of the Test set on the y-axis at different k values on the x axis. Notice how the MSE decreases the most from k=1 to k=5 but then begins to increase again as k reach 15. These results could mean that the KNN regression algorithm has more error as more points are included and this makes sense because there are more chances for outliers and noise. The optimal k value is k=3.

4. I repeated the this experiment for the classification with distance weighting and regression with distance weighting and found that weighting the distances gave me much better results for the MSE on the price housing dataset and slightly better accuracy for the magic telescope dataset

   Results for k=3 and k=15 are shown in the table below for classification and regression with and without distance weighting.

Type	Without Distance Weighting	With Distance Weighting
Classification (k=3)	0.8157	0.8156
Regression (k=3)	0.00696354	0.00602133
Classification (k=15)	0.8315	0.8304
Regression (k=15)	0.0112906	0.0062401

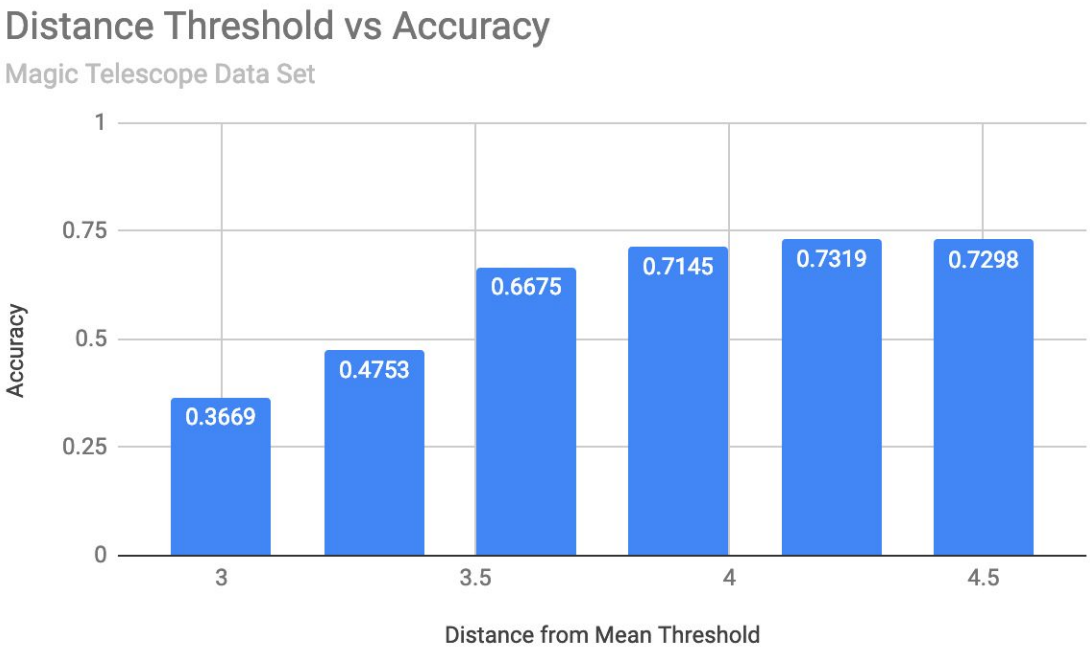
This table shows the difference in KNN classification and regression with and without weighting by distance for values k= 3 and k=15. Note that the value for classification is accuracy and regression is MSE. The accuracy and the MSE go down as weighted distances are used.

5. I used the KNN classification on the data set credit-approval which contains continuous and nominal, and don't know values. The distance metric I implement was the Heterogeneous Euclidean-Overlap Metric which basically takes checks if the value is a don't know and sets it to the distance 1 and then checks if the value is nominal and if the two points are the same value, it sets the distance to 0 otherwise the distance is set to 1. In the case that the value is continuous I took the Euclidean distance. I split the dataset 75/25 training data and testing data with a k value of 4. My resulting accuracy was 68% on the test set.

6. For my own experiment I used a threshold for each output based on it's mean value. For example if there are three output types, I found the mean value for each feature and iterated through each instances and if they difference between the mean and the instance attribute was larger than my threshold, I didn't use that feature which choosing a nearest neighbor later on. Of course I had to account for the fact that KNN uses the closest points to the new point and therefore I increased my k to values of 100 and I used this on KNN classification with distance weighting. I created a new file because I needed to edit my KNN algorithm to do this.

   Code for this experiment is found in the file knn_thresh.py

   The results are found in this graph below which shows the different thresholds and the accuracy at each threshold.



Graph of the the max distance from the mean and its effect on the accuracy. Accuracy is shown on the y-axis and the threshold for the distance from the mean is shown on the x-axis.

I had hoped that through this experiment I would find better accuracies. I thought I would find a sweat spot, but really I found the worst spot. Points in between the mean and the closest point to

the new point are the worst points to use, or at least that is what this data is showing. In the end, I learned a lot from doing this, but it didn't really help in finding better accuracies.