Slides:
https://docs.google.com/presentation/d/1GZxSMGs_TULa9doEAbu-Sq5qet45zzBIgvT0vNf7sW4/edit#slide=id.g2bc64b36b07_0_60
Repo: https://github.com/Logsterx7/cs5830_project4

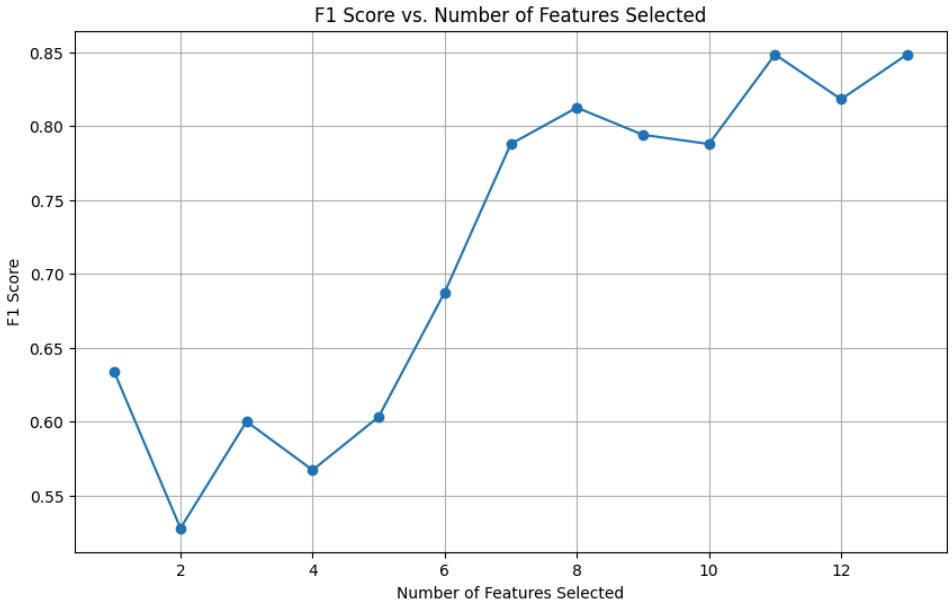
Part 1:

Introduction:

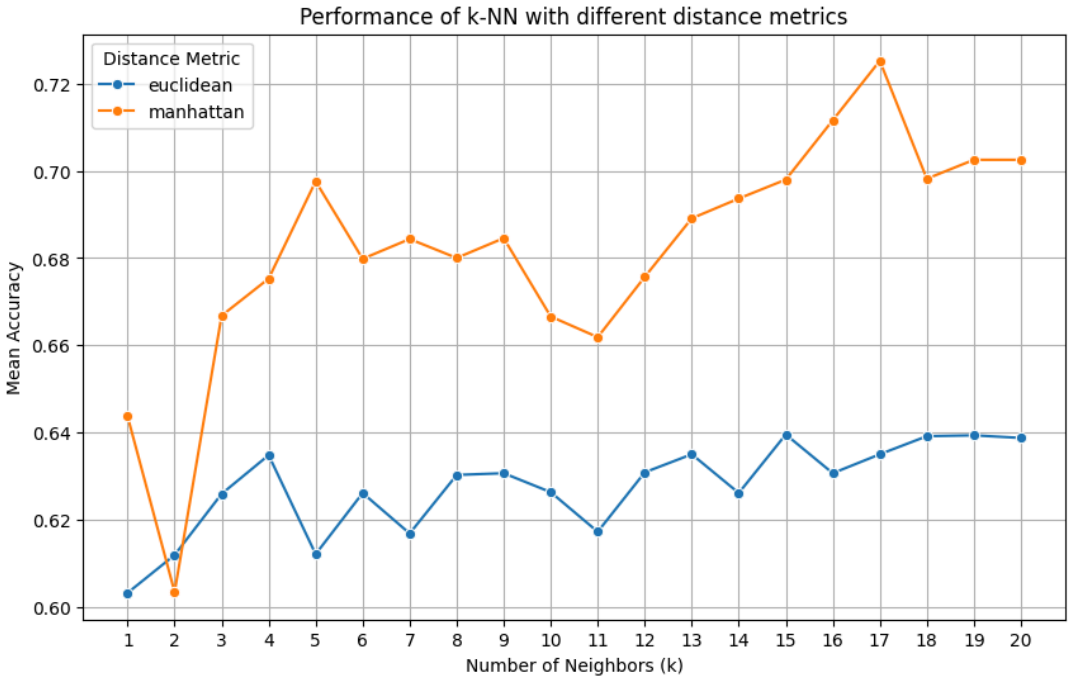
Part one of our project aims to help identify heart disease in patients to help catch diseases early and keep people healthy. To help do this we used a dataset from UC Irvine which gives us information about a patient's attributes such as age, sex, cholesterol levels, amount of exercise, and many others. One vital attribute included in the dataset is the presence or lack of heart disease within a patient. This data point will allow us to use K-Nearest Neighbors to help try and identify whether another patient with similar health circumstances also has heart disease.

Methods:

To start identifying heart disease in patients we first needed to figure out what attributes are most likely to point to signs of heart disease. To start selecting attributes we first had to clean our dataset and make sure it was all in an acceptable format. Once we had our data preprocessed we used a RandomForestClassiffier, a learning algorithm that uses a decision tree to make accurate predictions, to help find the best amount and combination of attributes.
To make sure we were getting the best attributes we compared the F1 scores for all the different combinations chosen by the RandomForestClassiffier using 10-Fold Cross-Validation. In the end, we found using 11 or 13 features was the best amount of features to use for a high F1 score.



Once we had our attributes selected we tested a range of different amounts of k neighbors using a GridSearchCV to help tune the hyperparameters. We found a higher k around 17 to 19 yields the best F1 scores with our selected attributes. We also found that using a Manhattan distance calculation worked the best to find the k-nearest neighbors. So rather than finding the closest neighbors as the bird flies, we found that using a gridlike fashion works best, as the taxicab drives. We believe this is the case because of the high dimensionality of our attributes.



Results:

Having done cross-validation on both our attributes and the number of neighbors we then tested the precision, recall, and F1 scores using 10-fold cross-validation. Using 11 attributes and setting our k number to 17 we were able to get a mean precision of 0.693 with a mean recall of 0.615 with an average F1 score of 0.637. So in the end our attributes and k are okay, it only predicts false positives about 30% of the time with about 62% of positives being actual positives. So overall we have a good balance in performance of correctly identifying individuals with heart diseases while minimizing false positives as indicated by the F1 score. There is definitely room for improvement within our choices, but overall we have a good start towards helping identify heart disease in patients. We feel our model can help get patients the help they need before they really need it.

Part 2:

Introduction:

Part one of our project aims to help identify forest fire risks to help prevent them. To help us achieve this we looked at different values in this data set including temperature, relative humidity, location, and others. An important attribute from this dataset to help us was the area burned, this allows us to use K-Nearest Neighbors to help try and identify the conditions of the area that were burned and predict when fires may occur in the future.

Dataset:

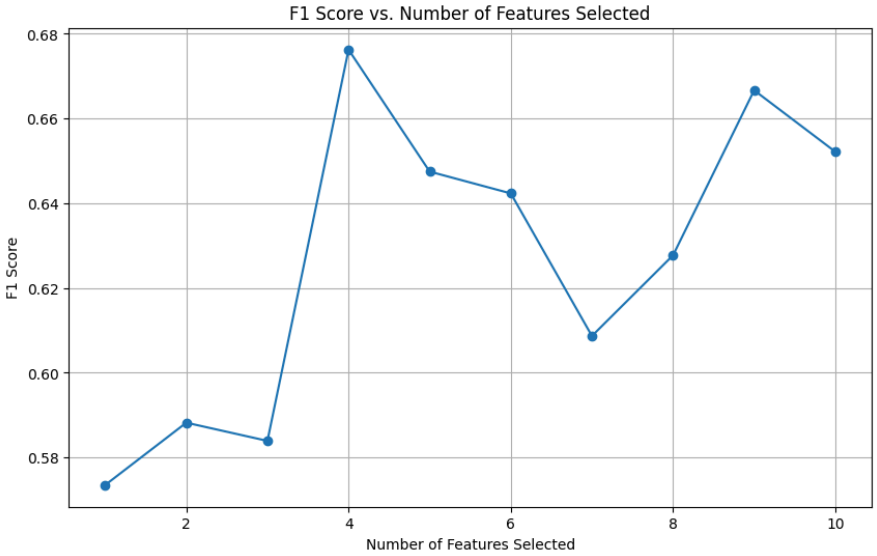
This data set was found in the UC Irvine Machine Learning Repository. This data set contains some meteorology data, location data, and time data. To clean up this data set we decided to take out the month and day section since we were more interested in the area conditions. We also adjusted our area to represent a true or false value on whether the area was untouched by fires or not rather than just how much an area burned alone.

Methods:

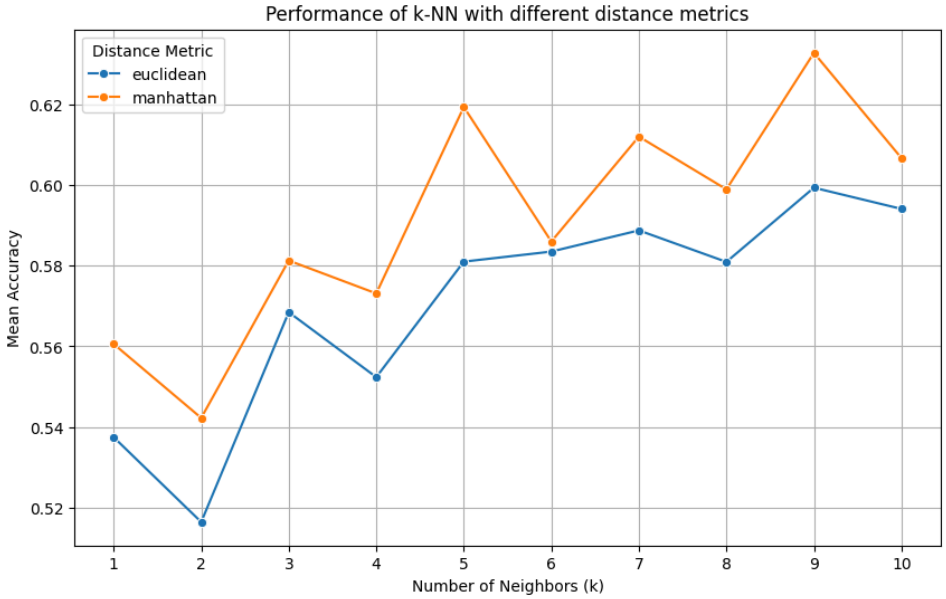
We didn't change anything with our methods to get the best combination of attributes and number of k neighbors.

Results:

Interestingly enough when running a similar method to before we found that only 4-6 values were providing us with higher F1 scores, the best combination we had here was relative humidity, temperature, Duff Moisture Code index from the Fire Weather Index system, and Drought Code index from the Fire Weather Index system.



After having our values selected and checking out a wide variety of different k values, we then found out that our best F1 scores with our selected attributes actually jumped at the 5 through 10 number of neighbors.



Finally, similarly to before we tested the precision, recall, and F1 scores using 10-fold cross-validation using a lower amount of our attributes than before we were able to get a mean precision of 0.650 with a mean recall of 0.671 with an average of F1 score of 0.651. In the end here we certainly have a good beginning of predicting these forest fires. While it isn't exactly the most accurate around a 35% error margin, it still gives us some good insight into the conditions that they may occur in which can always help out our old friend Smokey The Bear at identifying the weather conditions and patterns for finding when these forest fires are more likely to occur in the future.