Emma Lynn & Ethan Ford

Project 4 Report

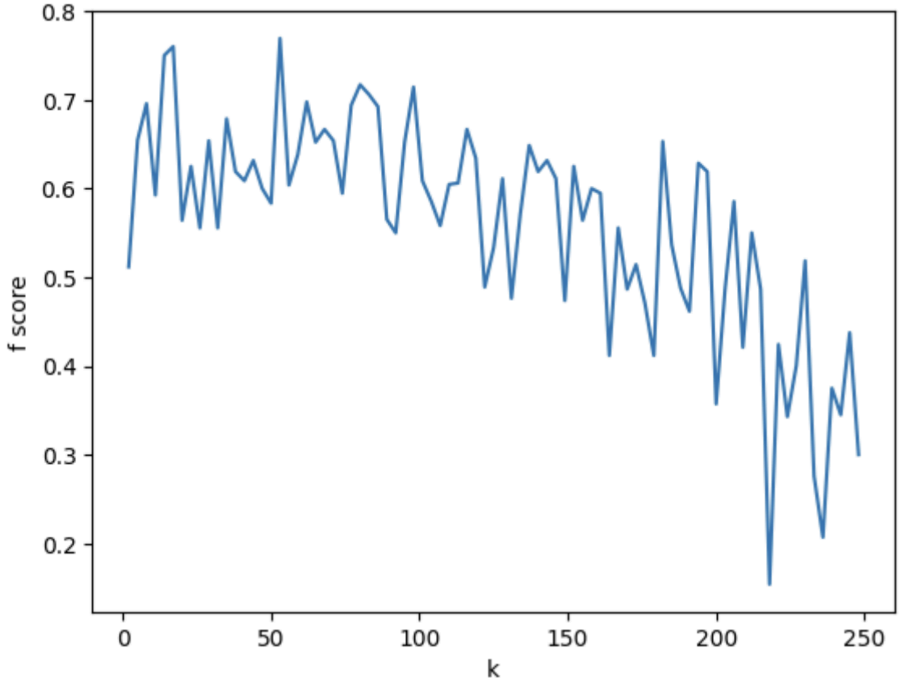
Our slides are linked here and GitHub repo is linked here.

Part I

Introduction - Heart disease remains one of the leading causes of mortality worldwide, forcing the need to develop accurate, data-driven models to predict its onset. Advances in machine learning offer promising approaches for enhancing tools and providing personalized care. We explore the application of the K-Nearest Neighbors method to predict the likelihood of heart disease, using key patient attributes from the Cleveland Heart Disease dataset.

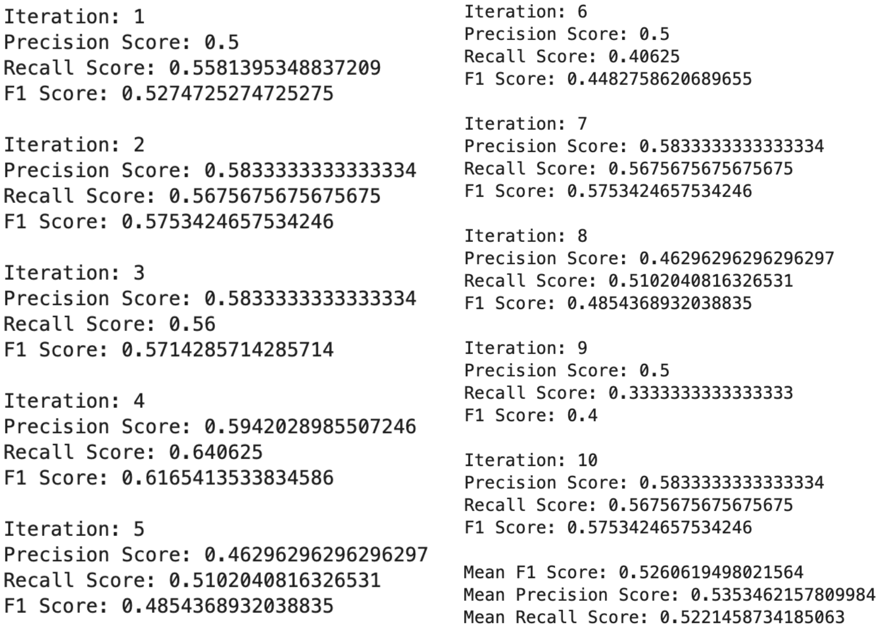
Methods - To determine k, we created a *create_scores* function to classify patients based on certain attributes using the KNN algorithm. The function returns precision, recall, and F1 scores on a sample size of 50 patients. We then created a *determine_k* function, which tests values of k that are inputs into our *create_scores* function. It gathers the resulting recall, precision, and F1 scores for the different values of k, throws them into a list, and sorts them to find the best k-value out of all values. As shown in **Figure 1.,** we use the elbow method to determine which k-value is the optimal k. We concluded that our optimal k-value was around 10.

Figure 1.



For determining our attributes, we ran our *knearestneighbors* function, which splits our dataset into training and testing sets and computes the different scores 10 times, as well as computing a mean F1 score. We determined that the optimal attributes would be 'age’, 'trestbps', 'chol', and 'thalach', solely based on the fact that this combination of attributes is what gave us a higher mean F1 score.

Results - Our results showed off a mean F1 score, precision score, and recall score of around .50-.55 consistently (**Figure 2.**). After carefully computing optimal k-values and attributes, we might conclude that the dataset as it is probably doesn’t have enough column information to allow us to accurately predict whether or not a patient has heart disease or not. Key factors like diet, lifestyle, or other important causative factors, could’ve been implemented in the dataset to allow for more attributes.



```
Iteration: 1
Precision Score: 0.5
Recall Score: 0.5581395348837209
F1 Score: 0.5274725274725275

Iteration: 2
Precision Score: 0.5833333333333334
Recall Score: 0.5675675675675675
F1 Score: 0.5753424657534246

Iteration: 3
Precision Score: 0.5833333333333334
Recall Score: 0.56
F1 Score: 0.5714285714285714

Iteration: 4
Precision Score: 0.5942028985507246
Recall Score: 0.640625
F1 Score: 0.6165413533834586

Iteration: 5
Precision Score: 0.46296296296296297
Recall Score: 0.5102040816326531
F1 Score: 0.4854368932038835

Iteration: 6
Precision Score: 0.5
Recall Score: 0.40625
F1 Score: 0.4482758620689655

Iteration: 7
Precision Score: 0.5833333333333334
Recall Score: 0.5675675675675675
F1 Score: 0.5753424657534246

Iteration: 8
Precision Score: 0.46296296296296297
Recall Score: 0.5102040816326531
F1 Score: 0.4854368932038835

Iteration: 9
Precision Score: 0.5
Recall Score: 0.3333333333333333
F1 Score: 0.4

Iteration: 10
Precision Score: 0.5833333333333334
Recall Score: 0.5675675675675675
F1 Score: 0.5753424657534246

Mean F1 Score: 0.5260619498021564
Mean Precision Score: 0.5353462157809984
Mean Recall Score: 0.5221458734185063
```

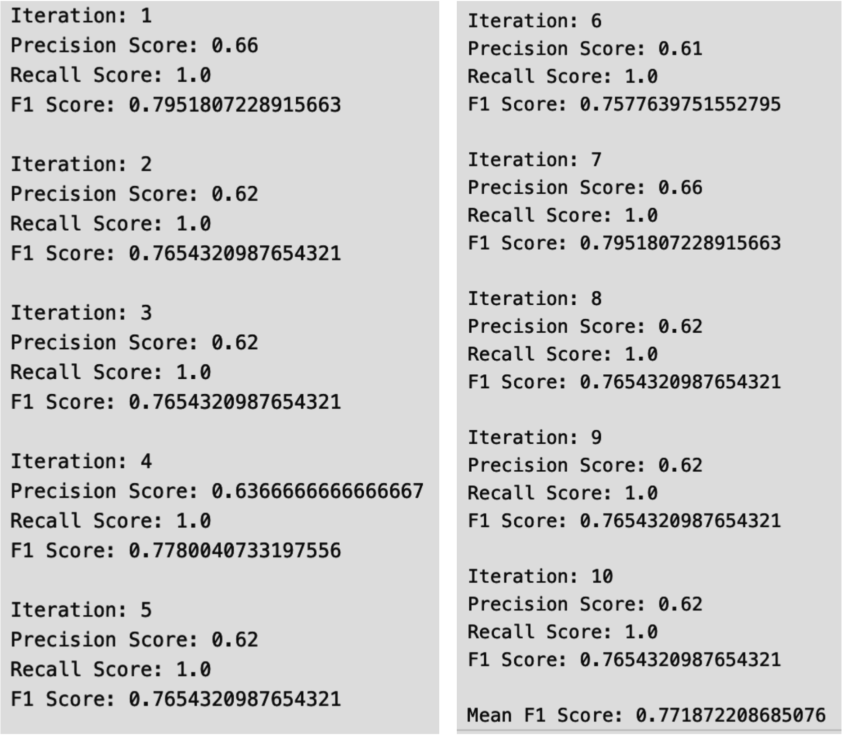
Figure 2.

Part II

Introduction - We decided to do Part II of our analysis on a dataset listing URLs, some of their attributes, and if they are legitimate or phishing. There are many ways being able to detect whether a URL is legitimate is very helpful, such as in the creation of a new web browser where the developer wants to be able to warn a user if a site they are attempting to access is likely to be dangerous. We performed a K-Nearest Neighbors analysis on this data, using the same methods as described in Part I of this report.

Dataset - This dataset is from the UC Irvine Machine Learning Repository and was put together by Arvind Prasad and Shalini Chandra. It is suitable for analysis because it is very large, gives a lot of helpful information about each URL, and is from a reputable source. The first thing we noticed about this dataset is that it is extremely large, with over 200,000 rows. Since this was significantly larger than the datasets we've worked with in class and would take a long time to run analyses on, we decided to only use a section of the data. We took the first 1000 rows of the dataset to use in our analysis (after checking that there is still a fairly evenly distributed amount of legitimate and phishing urls). We also made sure to standardize the column values we used in our subsequent analysis.

Results - After completing our analysis of the data, we were able to very consistently recognize 100% of phishing URLs (recall). The URLs we identified as phishing were actually phishing URLs about 60-65% of the time (precision). Overall, we had a mean F1 score (combination of precision and recall) of about .772 (**Figure 3.**). This method would be very successful if you care more about catching all phishing URLs than encountering false positives. Future work could still be done in making the model more precise in its identification of phishing URLs.



```
Iteration: 1
Precision Score: 0.66
Recall Score: 1.0
F1 Score: 0.7951807228915663

Iteration: 2
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321

Iteration: 3
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321

Iteration: 4
Precision Score: 0.6366666666666667
Recall Score: 1.0
F1 Score: 0.7780040733197556

Iteration: 5
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321
```

```
Iteration: 6
Precision Score: 0.61
Recall Score: 1.0
F1 Score: 0.7577639751552795

Iteration: 7
Precision Score: 0.66
Recall Score: 1.0
F1 Score: 0.7951807228915663

Iteration: 8
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321

Iteration: 9
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321

Iteration: 10
Precision Score: 0.62
Recall Score: 1.0
F1 Score: 0.7654320987654321

Mean F1 Score: 0.771872208685076
```

Figure 3.

References

Prasad, A., & Chandra, S. (2023). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 103545. doi: https://doi.org/10.1016/j.cose.2023.103545