COMP551 Assignment 1

Alif Naufal Farrashady (261121584) Calvin Chan (261121702)
Katherine Meyers (260987652)

October 3, 2022

Abstract
For this project, we implemented our own version of the K-Nearest Neighbour (KNN) and Decision Tree (DT) classifiers. We then applied them both over a Hepatitis and a Diabetic Retinopathy Debrecen dataset provided by UCI Machine Learning Repository. We trained our models by using 10-fold cross validation and evaluated them based on the average accuracy.

When evaluating with KNN, we used various distance functions and values of k ranging from 1 to 15. With these methods, we were able to achieve an accuracy of 85% using the Hamming distance function and $k = 12$ for the hepatitis dataset. For the diabetes dataset, we were able to achieve an accuracy of 66% using the Manhattan distance function and $k = 9$.

When evaluating with Decision Trees, we used the greedy splitting at the nodes with various cost functions and maximum depths ranging from 1-10. For the hepatitis dataset, the best accuracy of 83% was achieved using the misclassification cost and a max depth of 2. For the diabetes dataset, the best accuracy of 64% was achieved using the Gini Index and a maximum depth of 5.

Introduction
In this paper, we explore two classification techniques, K-Nearest Neighbours (KNN) and Decision Trees (DTs). These two techniques are widely used but they take 2 very different approaches to solving a classification problem. In order to better understand how these 2 techniques might vary, we applied both KNN and Decision Trees to 2 different datasets in order to compare their performance. We will also be plotting the decision boundaries of both these techniques to better understand how they made their predictions.

We also noted that the model performance for the diabetes dataset was consistently worse than for the hepatitis dataset, which is likely due to the nature of the dataset - with diabetes dataset having a more even split of the positive and negative observations as compared to the hepatitis dataset.

When applying the KNN and Decision Tree classifiers to both datasets, we achieved a maximum accuracy of 85.2% when using KNN and a maximum accuracy of 83.7% when using Decision Trees. On the diabetes dataset, we achieved a maximum accuracy of 66.1% when using KNN and a maximum accuracy of 64.5% when using Decision Trees. We noted that the performance of KNN and Decision Trees were similar on both datasets, indicating that these models have similar abilities and limitations when classifying data.

When re-running KNN and DT on only the key features of the hepatitis dataset (BIG LIVER, FIRM LIVER and ASCITES), we obtained a maximum accuracy of 87.5% with KNN and a maximum accuracy of 87% with Decision Trees. When re-running KNN and DT on only the key features of the diabetes dataset (MA 1, MA 2, MA 6, Exudates 1 and Exudates 7), we obtained a maximum accuracy of 69.5% with KNN and a maximum accuracy of 64% with Decision Trees.

Methods

We have developed our own custom implementation of both the KNN and the Decision Tree Classifier algorithms. Similar to the established implementations from Scikit-Learn, our custom implementations utilise Object-Oriented Programming (OOP) concepts, where the algorithms are implemented through a class, and also implement the fit and predict methods. Throughout all parts of our analysis, we used 10-fold cross validation to train and test our models.

Our KNN implementation makes use of 3 distance functions, which are the Euclidean, Manhattan and Hamming distances. We will experiment with these distance functions and the choice of k in order to find the best performing model for each dataset.

Our Decision Tree implementation has 3 cost functions, which are the Misclassification, Entropy and Gini Index cost functions. Again, we will try out these cost functions, as well as varying choices of maximum tree depth in order to find the best performing model for each dataset.

We will also plot the decision boundaries for some of the best models to better understand how they would make predictions. In order to allow for clearer visualisation, we will choose continuous variables as the axes when plotting the decision boundaries, as opposed to binary features.

Finally, we will use permutation on a Random Forest model of each dataset in order to determine the key features. Once these features have been determined, we will simplify the datasets by removing all non-key features. We will then re-evaluate both datasets with KNN and Decision Trees to see if there is an improvement in performance.

Datasets

The first dataset that we worked with is the Hepatitis dataset provided by the UCI Machine Learning Repository. We noted that the PROTIME and ALK PHOSPHATE columns had a high percentage of missing data, therefore we considered these features to be malformed and removed them from the dataset. We then removed all entries that had missing features values after removing the malformed features. There were also many columns using values 1 and 2 to represent binary variables, which we changed to 0 and 1 for clarity. It is worth noting that this dataset contains a significant majority of negatively labelled entries.

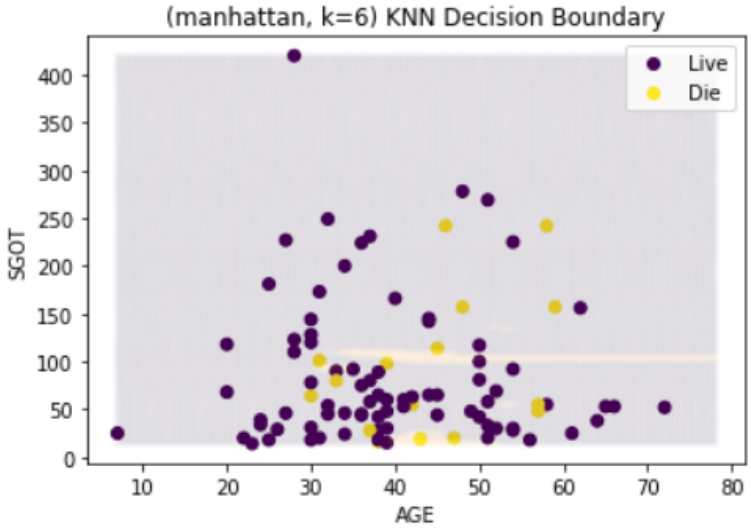
The second dataset that we worked with is the Diabetic Retinopathy Debrecen dataset provided by the UCI Machine Learning Repository. This dataset was much more complete, and we did not observe any malformed features or missing results. We then normalised the data in this set such that the values for all features ranged from 0 to 1. This dataset was well balanced between the number of positively and negatively labelled entries.

Results

We used 10-fold cross-validation to test out the different possible hyperparameters for both KNN and Decision Tree, as well as the different distance functions for KNN and cost functions for Decision Trees. From this, we can pick the best performing KNN and Decision Tree models for each dataset, by averaging the accuracy across the 10 folds for each hyperparameter combination. We can then compare whether KNN or Decision Tree performs best for a given dataset.

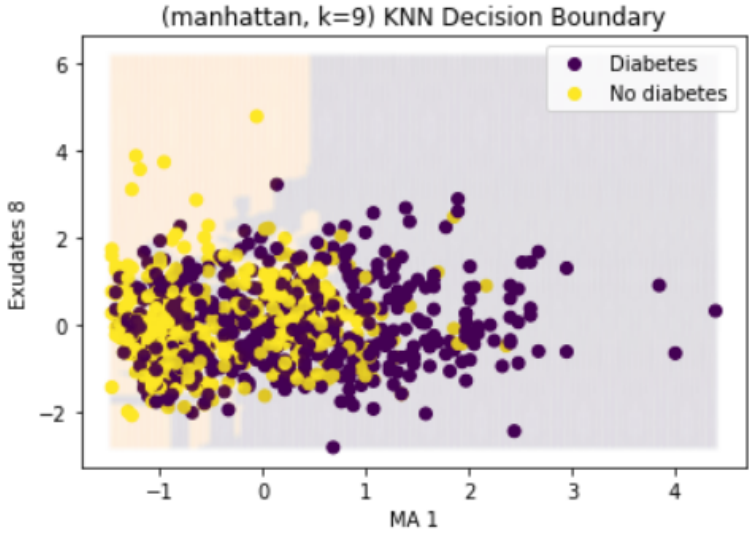
Firstly, using KNN on the hepatitis dataset, we found that the 3 distance functions behaved very differently. The Euclidean and Manhattan distances generally did not see any significant improvement in accuracy beyond $k = 10$. However, the accuracy fluctuates greatly for the hamming distance. Nevertheless, (Hamming, $k = 12$) achieves the highest accuracy, at 85%. However, high values of k will increase the prediction time of KNN. A possible alternative is (Manhattan, $k = 6$), which has an accuracy of 81%.

Looking at the decision boundaries of these 2 models, (Hamming, $k = 12$) seems to always predict 'live' for any dataset. On the other hand, (Manhattan, $k = 6$) has some regions where it will predict 'die', for example, at low values of SGOT and ages of 40-50.



Secondly, using KNN on the diabetes dataset, all 3 distance functions fluctuated greatly as k was increased. However, the accuracy remained within a range of $60 - 66\%$, unlike the hepatitis dataset, where the accuracy ranged from 67% to 85%. For the diabetes dataset, the best performing hyperparameter choice is (Manhattan, $k = 9$), which achieved 66% accuracy. Again, high values of k may not be ideal due to the lengthened prediction time. (Manhattan, $k = 5$) was able to hit 65%, thus it may also be a reasonable choice.

Comparing the decision boundaries of these 2 models, they are largely similar. Both models will predict 'no diabetes' for cases with high levels of Exudates 8, or those with negative values of MA 1. Visually, it seems there is a central cluster which is closely grouped together, with a somewhat even mix of both target features. This may be where the model struggles to predict and differentiate cases, which is why the overall accuracy is generally less impressive, even when trying out different hyperparameters.

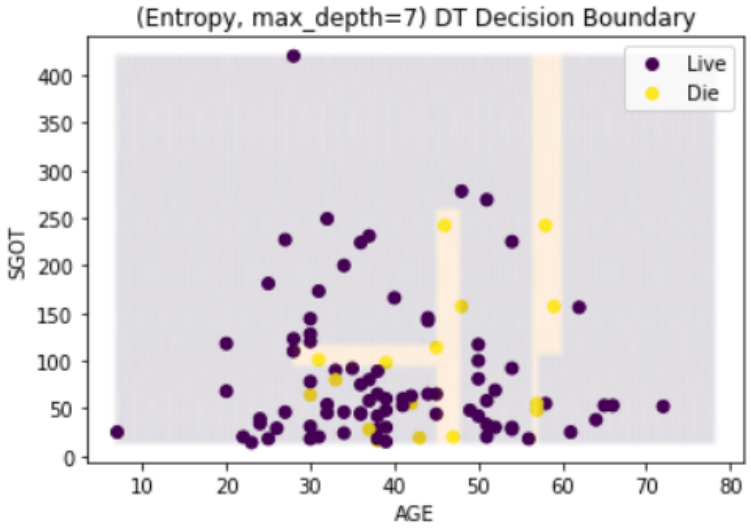


As such, comparing KNN across these two datasets, it has shown some potential cases where

KNN might struggle. As seen from the decision boundary on the hepatitis dataset, it is possible for KNN to only predict the majority class all the time. Although this optimised accuracy, it may be less meaningful to always predict the majority class. For example, in the hepatitis dataset, predicting that patients will live 100% of the time is not very useful, this means that we will not detect any potential deaths and prevent them.

When Decision Trees were used on the Hepatitis dataset, we found that the best result was obtained using misclassification cost and a maximum depth of 2, which resulted in an accuracy of roughly 83%. Interestingly, when using misclassification cost and Gini index, the accuracy decreased with an increasing tree depth, while with entropy the accuracy increased with an increasing tree depth. The high accuracy results with low tree depth are likely due to the imbalance between positive and negative data points in the data set. Therefore, while the accuracies for entropy with a higher tree depth have suboptimal accuracy, we believe that they will correctly classify the positive results more often, providing a more meaningful model.

For the best result with Decision Trees on the Hepatitis dataset (misclassification cost, max depth = 2), the decision boundary does not generate anything useful. This is because the split that was made is not along any of the continuous variables, so no decision boundaries will show up using any continuous axes for this case, and no results of note can be generated on a decision boundary with two binary features on the axes. However, looking at one of the next best results (entropy, max depth = 7), produces a much more interesting decision boundary.



When Decision Trees were used on the Diabetes dataset, we found that the best result was obtained using Gini index and a maximum depth of 5, which resulted in an accuracy of roughly 64.5%. Using misclassification cost, the accuracy did not follow a strong trend as the maximum depth increased. However, both entropy and Gini index reached a peak after approximately a maximum depth of 3, and the accuracy remained roughly constant for higher depths. Therefore, our results show that using both entropy and Gini index with a maximum depth ranging from 3 to 10 will produce a relatively accurate model. However, since the highest accuracies were in the mid 60s, Decision Trees were not able to generate very accurate predictions on this dataset.

On the Hepatitis dataset, the maximum accuracy we obtained using KNN was 85.2% and the maximum accuracy we obtained using Decision Trees was 83.7%. On the diabetes dataset, the maximum accuracy we obtained using KNN was 66.1% and the maximum accuracy we obtained using Decision Trees was 64.5%. When comparing the performance of KNN to that

of Decision Trees on each dataset, we noted that their performances were roughly the same, with KNN performing slightly better than Decision Trees on both datasets. This indicates that these models have similar abilities and limitations when classifying data.

We used permutation on a Random Forest model of each dataset in order to determine the key feature. This method indicated that the key features for the Hepatitis dataset are BIG LIVER, FIRM LIVER and ASCITES. It also indicated that the key features for the Diabetes dataset are MA 1, MA 2, MA 6, Exudates 1 and Exudates 7. These key features apply to both the KNN and Decision Tree models since they are dependent on each dataset, rather than the models used to analyse the dataset.

We then simplified both datasets to contain only the key features and re-ran our experiments on the simplified datasets. On the hepatitis dataset, the maximum accuracy for KNN was 87.5% and the maximum accuracy for DT was 87%. One interesting thing to note is that on this dataset, when running KNN the three distance functions produced the same results, and when running DT the three cost functions produced the same results. This is due to the fact that the data was reduced to only 3 binary features, and was thus oversimplified. On the diabetes dataset, the maximum accuracy for KNN was 69.5% and the maximum accuracy for DT was 64%. We observed that the performance of Decision Trees remained similar for the simplified datasets. This is because Decision Trees are able to value certain features more than others, since it chooses which feature is the best to split at each node and thus only the most important features are used. Therefore, when we eliminated the less relevant features, the results were similar since the original Decision Trees had already isolated the most important features. However, we observed that higher accuracies were obtained when using KNN on the simplified datasets. This is because KNN is sensitive to class-irrelevant features, since it treats all features equally. Therefore, when the less relevant features were removed, KNN was better able to make focused predictions, resulting in higher accuracy.

Discussion and Conclusion

We learnt how to implement DT and KNN algorithms and achieved a better understanding of the algorithms behind the pre-built libraries of scikit-learn. We also familiarised ourselves with the process of data cleaning, exploration and evaluation, as well as visualisation of the model's performance.

In addition, we saw that when algorithms optimise accuracy naively, it may result in a model that is not useful. This is seen when using the KNN model on the hepatitis dataset, where it optimised accuracy by predicting the majority class 100% of the time. Further investigation could be done into methods to avoid the accuracy being predicted in this way.

When re-evaluating both datasets with only the key features included, we received similar results using the decision tree model and improved results using the KNN model. This reflects the fact that KNN is sensitive to class-irrelevant features, whereas decision trees have the ability to distinguish the most impactful features when splitting each node. Further investigation could be done into more advanced methods of feature selection.

Statement of Contributions

Calvin cleaned the datasets. Alif did the implementation for the KNN while Katherine did the implementation for the DT. The whole group helped with the experimentation and report writing.

References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Dr. Balint Antal and Dr. Andras Hajdu, Department of Computer Graphics and Image Processing Faculty of Informatics, University of Debrecen, 4010, Debrecen, POB 12, Hungary antal.balint '@' inf.unideb.hu

Scikit Learn. Feature importance with a forest of trees.
[https://scikit-learn.org/stable/auto$_e$*xamples/ensemble/plot*$_f$*orest*$_i$*mportances.html*].