KNNs and Decision Trees for Hepatitis and Diabetic Retinopathy Data

Cecilia Jiang, David Kronish and Matt Ludwig

1 Abstract

In this assignment we constructed and investigated the performance of two machine learning models on two biomedical datasets. We found that the Decision Tree (DT) achieved worse accuracy than the K - Nearest Neighbor (kNN) model for the Hepatitis data and that the DT achieved better accuracy than the kNN model for the Diabetic Retinopathy data. When fitting the kNN model on the data sets we found that selecting a $k = 5$ for the Hepatitis data set gave the best accuracy and selecting $k = 11$ for the Diabetic Retinopathy data set gave the best accuracy. Similarly, we found that when fitting the DT on the Hepatitis data, the best accuracy was observed with a tree depth of 6. We also found that when fitting the DT on the Diabetic Retinopathy data that the best accuracy was observed with a tree depth of 13.When comparing different distance and cost functions we found most of the cost functions that we compared had similar accuracy although some were slightly better than others. We found that the Manhatten distance and Euclidean distances have similar performance for the kNN and misclassification worked best for the decision tree. After noticing that our models had low accuracy on the Diabetic Retinopathy data set and that many of the features plots were noisy near the decision boundaries, we implemented an edited k - Nearest Neighbors to create a less noisy data set. After training our model on this new edited data set, we found that accuracy of our k - Nearest Neighbors and Decision Tree models improved from 64.5% to 71.3%.

2 Introduction

In this assignment we were tasked with processing and conducting an exploratory analysis of two different biomedical datasets, constructing and implementing K - Nearest Neighbors and Decision Tree models, evaluating these models under different features and parameters and with comparing them with one another. The first data set consists of Hepatitis data collected from 155 Hepatitis patients. For each patient, 19 different predicting attributes were recorded as well as patient survival. Many of the predicting attributes included observed physical characteristics of the patient such as the presence of Ascites (the build of fluid in the abdomen) or Varices (enlarged or swollen veins). Other predicting attributes included various synthetic liver function test such as the levels of Albumin and the Prothrombin Time or Protime which are used to measure potential liver damage or failure. For instance, low levels of the protein Albumin are associated with liver failure. Also since the liver produces the majority of blood clotting proteins, a longer Prothrombin time or the time it takes for a blood clot to form could suggest damage to the liver or Hepatitis.[oVAVHC] The data consisted of 69 male patients and 11 female patients, 32 or (16.25%) of patients died and 123 or (83.75%) of patients survived. The mean patient age was 40.7 years old with a standard deviation of 11.28 years, the youngest patient was 20 years old and the oldest patient was 72 years old. Figure 1 below shows the different distributions counts for the categorical features steroids, antivirals, fatigue, malaise, anorexia, liver big, liver firm, spleen palpable, spiders, ascites, varices and histology.

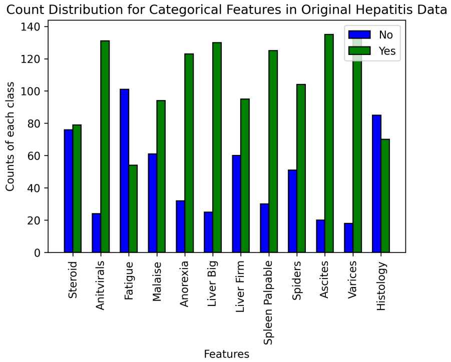

Figure 1: Count Distribution for Categorical Features in Hepatitis Data

The second data set consists of 1151 instances of 20 attributes that are used to predict whether or not an image contains signs of Diabetic Retinopathy which is complication of diabetes that affects the eyes and can lead to vision impairment or blindness.[Cli] These 20 features have been extracted from the larger Messidor (or ”Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology” in French) data set. Each of the 20 features is the result of several image processing algorithms and contains information about the image (quality assessment, pre-screening, AM/FM), the lesion (Exudates, MA or microaneurysms) or anatomical components (diameter of the optic disc, distance between center of the macula and center of the optic disc).[Ant] In this data set, 99.6% of the images are of high quality while .34% of the images are of low quality, 91.83% of images show severe retinal abnormality whereas 8.1% did not, 53.08% of images showed signs of diabetic retinopathy whereas 46.91% did not. Figure 2 below provides the distributions of the four most important features in predicting the presence of Diabetic Retinopathy.

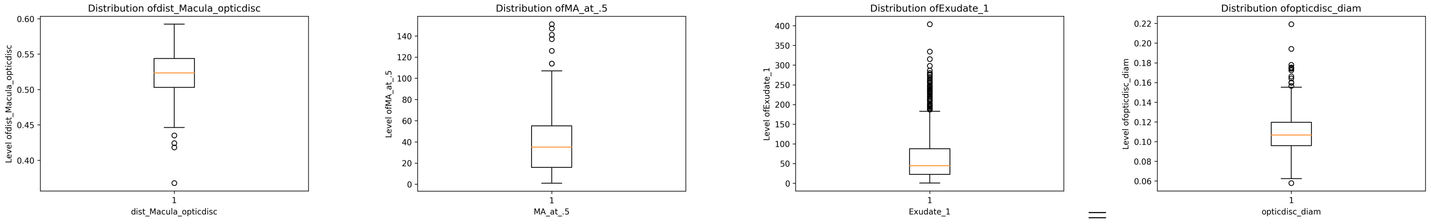

Figure 2: Distributions for the four most important features for predicting presence of Diabetic Retinopathy in the original data set

3 Methods

In this assignment we implemented two machine learning models: K - Nearest Neighbors or (kNN) and Decision Trees (DT). The kNN model is used to classify a new input point based on the classes of the k closet data points in the training data. The main parameters for the model are the number of nearest neighbors k and a distance function which tells the model how to measure the distance between points. Using these parameters, the model calculates the probability of each possible class that the new point could be classified as based on the classes of the k nearest points. Consider the following example from [Mur22]. In the image below, the kNN classifier compares the class of the new point, X, to the 5 nearest points. Since 3 of the 5 points are class 1 and 2 of the 5 points are class 2, the model will classify X as class 1.[Mur22] In general, kNN is an intuitively simple nonparametric model that works well for binary and multiclass clasification. On the other hand, kNN is does not perform well on imbalanced data or high dimensional data and is sensitive to both the scale of features and outliers.

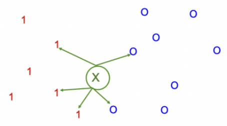

Figure 3: Example of K - nearest neighbors classifier on pg. 544 of [Mur22]

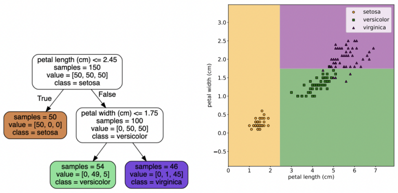

Figure 4: Example of Decision Tree classifier on pg. 5 of [Mur22]

The second machine learning model that we implemented was the Decision Tree (DT). Like kNN, the DT is used to classify a new input point into one of several possible classes. The model works by recursively splitting the data in to two smaller groups based on a specific feature and some value of that feature. As a result of recursively splitting the data into two smaller groups, the model takes on the shape of a tree as shown in the example below.[Mur22] The main parameters of the model are the number of classes, the maximum depth of the tree, the cost function and the minimum number of data points that are allowed in a leaf of the tree. The maximum depth of the tree and the minimum data points allowed in the leaf of the tree are used to control the size of the tree and affect the accuracy of the model. The cost function of the model determines the cost associated with splitting the data with a certain feature and feature value and is based on how uniform the classes are in the two groups. In the example below, the decision tree classifies a new point first into three possible classes based on the value of its feature petal length, then based on the value of its feature petal width. When the algorithm is done running, the new point will fall into the orange, green or purple node or region as shown as shown above.[Mur22] In general, the decision tree is a relatively simple non-parametric model that is not affected by the normalization or scaling of features and is robust to outliers. On the other hand, the DT often overfit data which can lead to wrong prediction are very affected by noise.

4 Datasets

Upon loading the datasets into Python, we used Scikit-learn's function SimpleImputer in order to replace the missing values in the dataframes with the features mean or most frequent observation depending on whether the feature was continuous or categorical. When then found and removed outlier points, which we considered to be points in the $99th$ quantile of each continuous feature. This was an important step as kNN is not sensitive to outliers. Upon calculating the class distributions of both data sets, we found that the class distribution of survival in the Hepatitis data was very imbalanced as only 32 patients died and 123 patients survived. Since the kNN model does not perform well on imbalanced data, we implemented a synthetic minority oversampling technique (SMOTE) from the Imbalance-Learn Python library in order to generate more minority samples and balance the data set. After performing SMOTE on the Hepatitis data we had a data set with 123 observations in which the patient died and 123 observations in which the patient lived. Following this, we implemented Scikit-learn's RandomForestClassifier in order to determine which of the features were most important in predicting the presence of Diabetic Retinopathy or in determining whether a Hepatitis patient would survive. The RandomForestClassifier uses multiple decision trees on sub-samples on the data and determines feature importance based on how much the average cost associated with this feature is reduced. As a result, the features that decrease the cost most or increase the uniformity or purity of the two groups will be the most important feature. Therefore the most important features will have the highest mean decrease in impurity as shown in the graph below. For more information regarding the RandomForestClassifier see [sl]. Using this method we found that the features Age, SGOT, Bilirubin and Spiders were most influential in predicting the survival of a patient with Hepatitis and we found that the number of MAs found at confidence level $\alpha = 0.5$, the number of Exudates found at $\alpha = 0.5$, the distance from the center of the macula and the center of the disc as well as the diameter of the optic disc were most influential in predicting whether a patient had Diabetic Retinopathy.

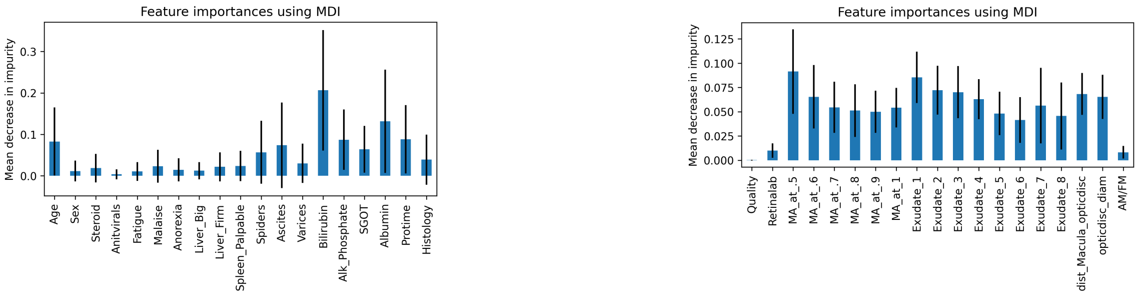

Figure 5: Feature Importance using Mean Decrease in Impurity from RandomForestClassifier for Hepatitis and Diabetic Retinopathy Data

5 Results

Using the 19 features in Hepatitis data set and the 20 features in the Diabetic Retinopathy data set and the best values for k and depth, we evaluated the accuracies of the models. In order to compare the accuracy of kNN and DT models on the two data sets we used Receiver operator and Precision Recall Curves (ROC and PRC) as well the area of the curve or (AUC) metric. We found that the accuracy of the kNN was higher on the Hepatitis data set while the accuracy of DT was higher on the Diabetic Retinopathy data set. This can be seen in Figures 6 - 9. In order to find the best value for k and best depth to use for our models we also compared the accuracy for different values of k and different depths as shown in Figures 10 - 13. After this we considered using different distance functions for kNN and different cost functions for DT and compared the accuracy of the models. When comparing different distance and cost functions we found most of the cost functions that we compared had similar accuracy although some were slightly better than others. We found that the Manhatten distance and Euclidean distances have similar performance for the kNN and misclassification worked best for the decision tree. This is shown in Figure 12 - 15. After this we generated decision boundaries for kNN and DT on both datasets. Since the decision boundaries are $2-$dimensional, we selected two of the most important features for predicting survival and presence of Diabetic Retinopathy based on the results of the RandomForestClassifier. This is shown in Figures 18 - 21.

Since we observed that the diabetic retinopathy dataset appeared messy and had low accuracy for kNN we decided to implement and run a undersampling technique called edited nearest neighbors in order to clean the database by removing samples that are dissimilar to the their nearest neighbors. Edited KNN works as follows: for each point in the dataset, look at the K = 3 nearest neighbors. Determine the majority class of the 3 nearest neighbors. If the point's class differs from the majority class of its 3 nearest neighbors, remove the point. Return the new smaller data set. After running this algorithm on the diabetic retinopathy data set we compared the accuracies of kNN. We found that kNN gave 64.5% accuracy before running the edited nearest neighbors algorithm and 71.3% accuracy after suggesting that performing the edited nearest neighbors algorithm improved the accuracy of kNN.

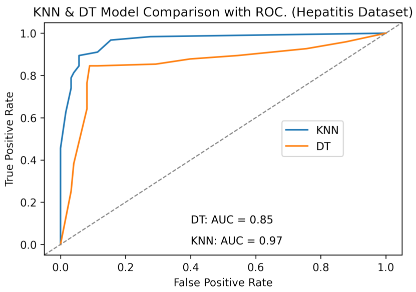

Figure 6: Accuracy for KNN and Decision Tree on Hepatitis Data - ROC Curves

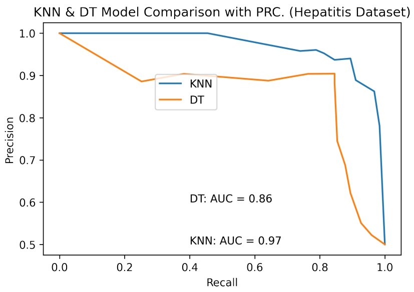

Figure 7: Accuracy for KNN and Decision Tree on Hepatitis Data - PRC Curves

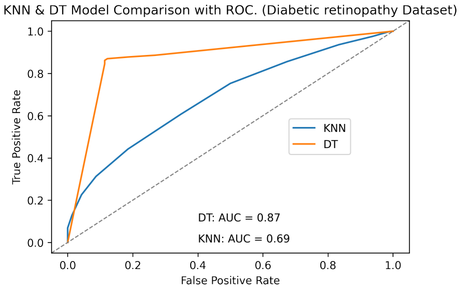

Figure 8: Accuracy for KNN and Decision Tree on Diabetic Retinopathy Data - ROC Curves

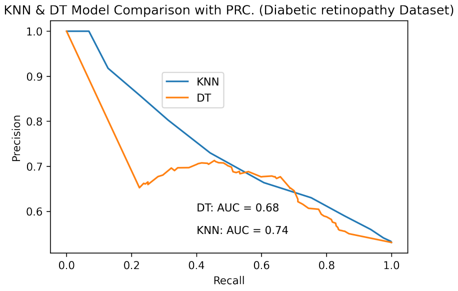

Figure 9: Accuracy for KNN and Decision Tree on Diabetic Retinopathy Data - PRC Curves

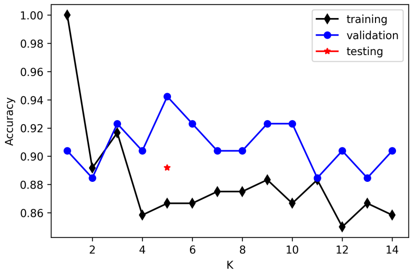

Figure 10: kNN Accuracy for Different K on Hepatitis Data

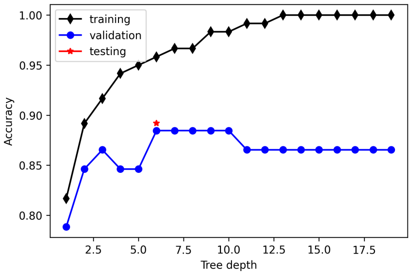

Figure 11: Accuracy for Decision Tree for Different Depths on Hepatitis Data

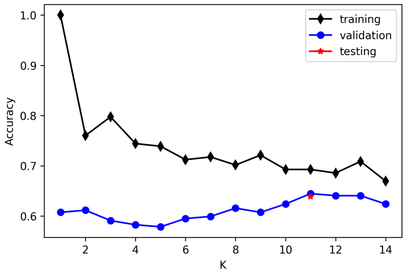

Figure 12: kNN Accuracy for Different K on Diabetic Retinopathy Data

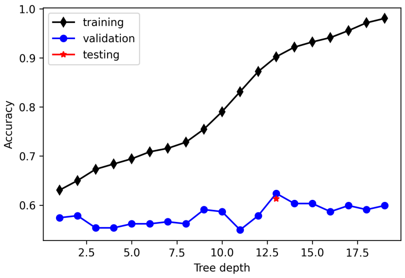

Figure 13: Accuracy for Decision Tree for Different Depths on Diabetic Retinopathy Data

6 Discussion and Conclusion

Overall we found that the Decision Tree (DT) achieved worse accuracy than the K - Nearest Neighbor (kNN) model for the Hepatitis data and that the DT achieved better accuracy than the kNN model for the Diabetic Retinopathy data. When fitting the kNN model on the data sets we found that selecting a $k = 5$ for the Hepatitis data set gave the best accuracy and selecting $k = 11$ for the Diabetic Retinopathy data set gave the best accuracy. Similarly, we found that when fitting the DT on the Hepatitis data, the best accuracy was observed with a tree depth of 6. We also found that when fitting the DT

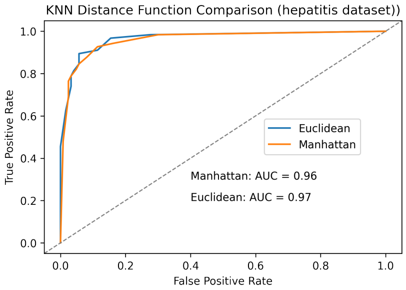

Figure 14: Accuracy of KNN for Different Distance Functions ROC for Hepatitis Data

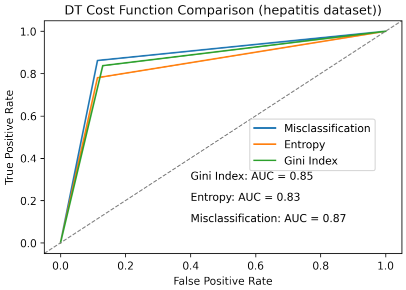

Figure 15: Accuracy of Decision Tree for Different Cost Functions ROC for Hepatitis Data

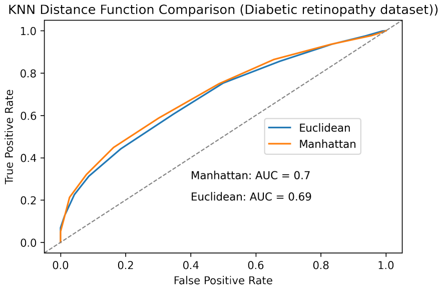

Figure 16: Accuracy of KNN for Different Distance Functions for Diabetic Retinopathy Data

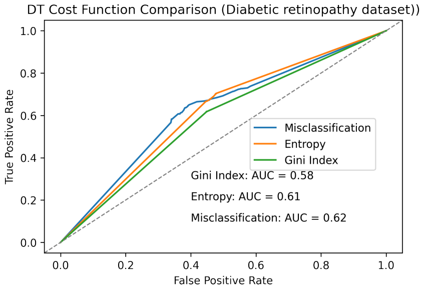

Figure 17: Accuracy of Decision Tree for Different Cost Functions for Diabetic Retinopathy Data

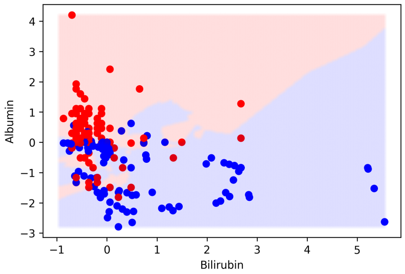

Figure 18: Decision Boundary for kNN for best K = 5 for Hepatitis Data

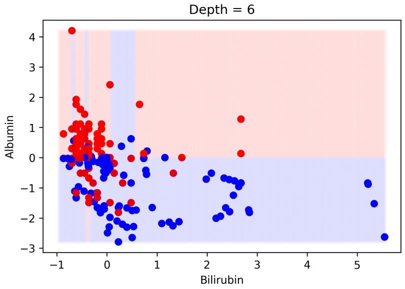

Figure 19: Decision Boundary for Decision Tree for best depth = 6 for Hepatitis Data

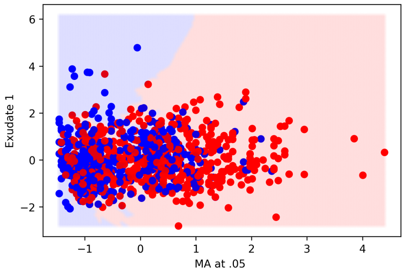

Figure 20: Decision Boundary for kNN for best K = 11 for Diabetic Retinopathy Data

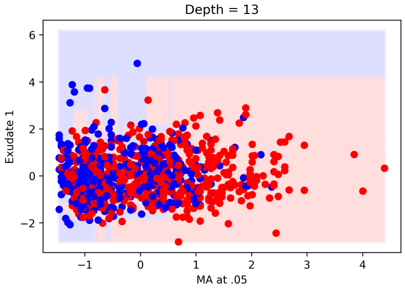

Figure 21: Decision Boundary for Decision Tree for best depth = 13 for Diabetic Retinopathy Data

on the Diabetic Retinopathy data that the best accuracy was observed with a tree depth of 13. When comparing different distance and cost functions we found most of the cost functions that we compared had similar accuracy although some were slightly better than others. We found that the Manhatten distance and Euclidean distances have similar performance for the kNN and misclassification worked best for the decision tree. One possible direction for future investigations would be to try and compare different pre-processing techniques that balance the Hepatitis dataset. For instance, one could implement a more generalized form of SMOTE called Adaptive Synthetic Sampling Approach which incorporates the distribution of the minority data samples while generating more minority samples. Another possible direction for future investigation could consider using Imbalanced Learn's RandomUnderSampler or TomekLinks to clean and balance both datasets before running the kNN and decision tree models. For more information on these techniques see [Via]. It would also be interesting to implement other variations of kNN such as weighted or fixed radius k - Nearest Neighbors on the cleaned and balanced datasets and compare the accuracies with our k - Nearest Neighbors model.

7 Statement of Contributions

Matt did Task 1, 2, most of write-up, decision boundaries and helped with task 3. Cecilia did task 3 numbers 2 and 3 and helped with write-up, task 2 and debugging and implimenting models. David did cost and distance functions in task 3 and helped with debugging and task 3.

References

[Ant] András Antal, Bálint; Hajdu. An ensemble-based system for automatic screening of diabetic retinopathy. `https://arxiv.org/pdf/1410.8576.pdf`.

[Cli] Mayo Clinic. Diabetic retinopathy. `https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611`.

[Mur22] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[oVAVHC] U.S. Department of Veterans Affairs: Viral Hepatitis and Liver Disease Website Course. Evaluating liver test abnormalities: Understand the pathophysiology of liver disease, synthetic liver function tests. `https://www.hepatitis.va.gov/HEPATITIS/course/index.asp?page=/provider/courses/livertests/livertests-11&`.

[sl] scikit learn. sklearn.ensemble.randomforestclassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`.

[Via] Raden Aurelius Andhika Viadinugroho. Imbalanced classification in python: Smote-enn method. `https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50#:~:text=The%20Concept%3A%20Edited%20Nearest%20Neighbor,the%20observation’s%20class%20or%20not.`.