

# Assignment 1: K-Nearest Neighbors & Decision Trees

Group 5: Adam Motaouakkil (260956145), Frédéric Mheir (260636214), Yann Bonzom (260969653)

## Abstract

We are given datasets of a sample of diabetic patients and another of hepatitis patients. Our task was to implement K-Nearest-Neighbor (KNN) and Decision Tree (DT) models to predict whether in the former sample a patient has diabetes and in the latter if they die from hepatitis.

In the hepatitis dataset, baseline KNN and DT predict the target output with an accuracy of 56.2% and 58 percent respectively. Finding the optimal K improves the model's performance in training data but performs around baseline levels for testing. Data standardization improves KNN and negligibly improves DT performance. The Manhattan distance function performs similarly to the Euclidean distance function in KNN. Additionally, weighting KNN follows closely with the standardization performance. Our adjustments resulted in KNN accuracy of 100% for the hepatitis dataset and 68.4% for the diabetic dataset.

The Entropy and Gini Index functions provide the best performance depending on the dataset. For the diabetes set, the error functions do not have a significant impact on performance. Optimization for depth improves DT performance across the board. Using only features that are strongly correlated with the class improves model performance, for a DT accuracy of 87.5% for the hepatitis dataset and 64.5% for the diabetic dataset.

While KNN and DT perform similarly when unoptimized, KNN achieves better accuracy when refined. Note that both models vary in performance when testing size is varied, which is important as the datasets vary dramatically in size.

## Introduction

The assignment asks to use the multivariate dataset to make a prediction on a testing subset for a target output. The hepatitis data set contains categorical, integer, and real values while the diabetes set is only made up of integer and real values. The point of the exercise is to predict if a patient survives a bout of hepatitis and if a patient can be screened automatically for diabetes. Using baseline data without any manipulation or model optimization, the KNN and DT models generally operate around 50% to 60% accuracy depending on the dataset, but significantly improve after data and model optimization.

Our improved accuracy after optimization is similar to other researchers' papers. For instance, for the hepatitis dataset, *Z. Fern and Brodley's Boosted LazyDT* gave an accuracy of 85.42%, which is similar to our standardized DT accuracy of 87.5% (Xiaoli Z. Fern and Carla Brodley, *Boosting Lazy Decision Trees.*, ICML. 2003, p.4).

Finally, we have found that the testing accuracies vary depending on testing sample size, which is especially significant for the hepatitis dataset which has many missing values that bias the data's distribution, affecting KNN and DT performances. The diabetes set, which is significantly larger, leaves less room for error, and is especially perceptible in the results between validation and testing data. Their differences are less significant than in the hepatitis set. The effects of missing data and the effects of sample size is important to observe in the context of the results presented in the paper.

## Methods

The K-Nearest Neighbors approach is a powerful machine learning model that requires no initial training. It works as follows: we create an instance of the KNN class and provide it with training data consisting of inputs (denoted  $X$ , where each datapoint contains  $n$ ) and the corresponding outputs ( $y$ , the

class that each data point belongs to). To predict the class of a new data point  $x^*$ , we first calculate the distances between  $x^*$  and all the training points; we then find the  $k$  nearest neighbors using these distances; and lastly find the proportion of neighbors that belong to each class. The predicted class  $y^*$  for  $x^*$  is thus the class that has the highest probability.

## Datasets

**Figure 1: Diabetic Correlations**

Clearly, ASCITES and ALBUMIN are most strongly correlated with dying from hepatitis, and MA-1 and MA-2 are most strongly correlated with being diabetic. This information will be used later on in determining which features most impact model performance.

## Results

	KNN Results		DT Results <i>MD = Max Depth</i>	
	Hepatitis	Diabetic	Hepatitis	Diabetic
<b>Baseline</b>	Acc.: 56.2 / K: 3	Acc.: 58 / K:3	Acc.: 75 / MD: 20	Acc.: 56.3 / MD: 10
<b>K-optimization</b>	Acc.: 56.2 / K: 4	Acc.: 65.8 / K: 13	N/A	N/A
<b>Depth-optimization</b>	N/A	N/A	Acc.: 81.2 / MD: 1	Acc.: 62.8 / MD: 10
<b>Standardization</b>	Acc.: 100 / K: 3	Acc.: 68.4 / K: 23	Acc.: 87.5 / MD: 20	Acc.: 57.6 / MD: 20
<b>Best Cost Function</b>	Acc.: 100 / K: 3	Acc.: 68.4 / K: 23	Acc.: 87.5 / MD: 20	Acc.: 60.2 / MD:20
<b>Weighted KNN</b>	Acc.: 100 / K: 3	Acc.: 66.7 / K: 30	N/A	N/A
<b>High Corr. Features</b>	Acc.: 87.5 / K: 5	Acc.: 54.5 / K: 205	Acc.: 87.5 / MD: 1	Acc.: 64.5 / MD: 14

We begin with a discussion of our KNN model and the related model exploration. We have found that, though the basic KNN implementation is not particularly performant, standardizing the data in particular provided a significant performance improvement. *Note: for readability, accuracy percentages will be given in [X, Y] format where X is the accuracy for hepatitis and Y is the accuracy for diabetes.*

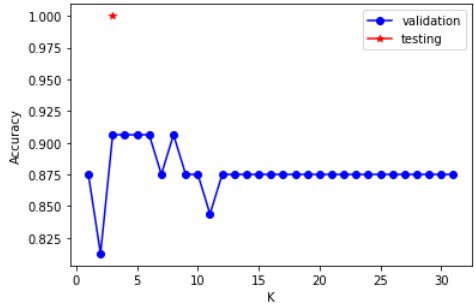
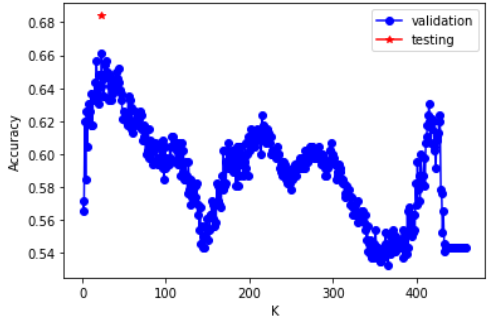
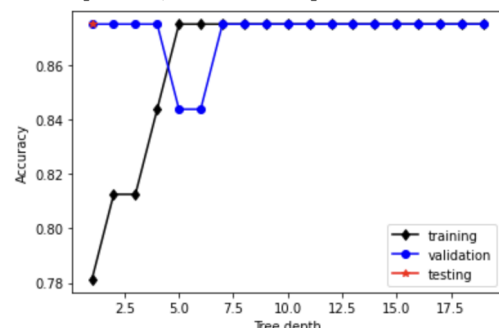
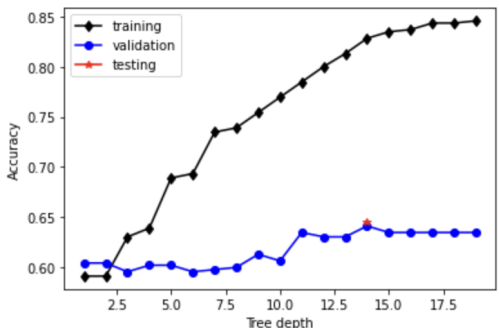
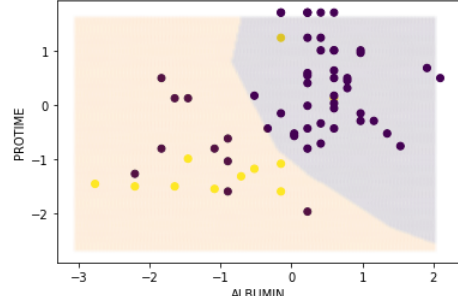
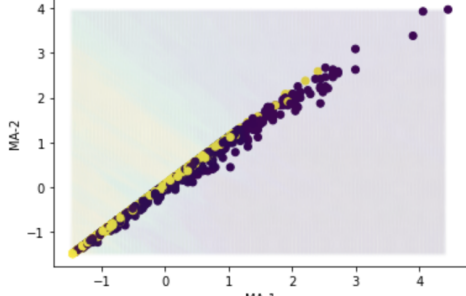
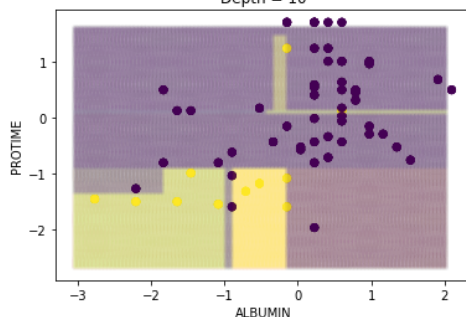
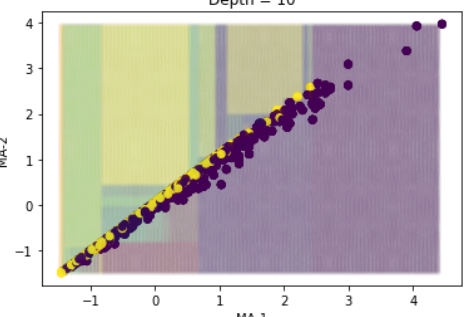
Standardizing the data led to accuracies of [100, 68.4], clearly improving performance for both datasets. The 100% accuracy seems extremely high, but we suspect this to be a result of the limited amount of data as well as its heavy bias towards with only 13/80 patients dying from hepatitis. These improvements are to be expected as the scale of features in KNN is important in determining the outcome.

The weighted KNN implementation (used to better consider distances and value further-away points less than closer ones), results in accuracies of: [100, 66.7]. Clearly, this did not help, and, with non-standardized data, it performed worse than just a standardized model with hyperparameter tuning.

For KNN, the Euclidean accuracies are [100, 68.4] and the Manhattan accuracies are [93.8, 64.9]. Clearly, Euclidean is better so we chose that. For DT, Misclassification Cost accuracies are [81.2, 60.2]; Entropy accuracies are [87.5, 60.2]; and Gini Index accuracies are [87.5, 57.6]. So, we opted for Entropy as it worked best in both datasets.

Finally, we looked into having our model only use the most important key features. Doing such changes helps improve model performance, since calculating distance between points now requires calculating the differences between fewer features. We decided to pick features with the highest correlations with the *class* (determined in Datasets section above). So, we picked *ascites* and *albumin* features for the hepatitis set, and *MA-1* and *MA-2* features for the diabetes dataset. This resulted in [87.5, 54.5] accuracies for KNN. As the hepatitis dataset's *ascites* and *albumin* features have much higher absolute correlations than the *MA-1* and *MA-2* features of the diabetic dataset (0.479 and 0.477, vs. just 0.293 and 0.266), this indicates why the former set's model performs better.

Please note that, for the hepatitis decision boundaries below, we have used *protime* instead of *ascites*, since *protime* is a continuous feature and only has a slightly lower correlation with the *class*. To make features comparable, we also standardized the data on the decision boundaries graphs.

	Hepatitis dataset	Diabetic dataset
<b>KNN</b> <b>Best accuracy</b>	<p>best K = 3, test accuracy = 100.0</p> 	<p>best K = 23, test accuracy = 68.4</p> 
<b>DT</b> <b>Best accuracy</b>	<p>best depth = 1, test accuracy = 87.5</p> 	<p>best depth = 14, test accuracy = 64.5</p> 
<b>KNN</b> <b>Decision Boundaries</b>		
<b>DT</b> <b>Decision Boundaries</b>	<p>Depth = 10</p> 	<p>Depth = 10</p> 

Next, we discuss our results for DT. The basic DT implementation with  $depth=20$  resulted in [75, 56.3] accuracies. This outperforms the basic KNN implementation. Upon standardizing our data (which was done to transform discrete feature values to continuous ones as well as reduce the size of gaps

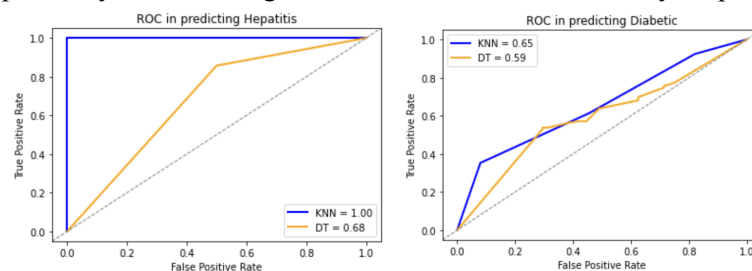
between feature values), we get: [87.5, 57.2] and thus further improve the model. This performance jump is not as significant as for the standardized KNN, which is expected as KNN depends on distances between points whereas DT does not. In terms of cost functions, we experimented with Misclassification Rate, Entropy, and the Gini Index. Entropy was best for the diabetes dataset, but GI was better for hepatitis, resulting in the following accuracies: [87.5, 60.2]. Performing hyperparameter optimization for the depth, we could further improve model performance with  $depth=1$  and  $depth=10$  for hepatitis and diabetes respectively, resulting in [81.25, 62.8]. By repeating the above process of choosing only the two features most strongly correlated with the class, we train DTs with accuracies [87.5, 64.5] which is slightly better. This might indicate that some features add noise and worsen performance when present.

## Discussion and Conclusion

Performing model explorations and tests allowed us to develop new models, leading to improved performances for both datasets, indicating that these performance boosts are somewhat dataset-agnostic.

For KNN, we found the optimal version to come from using standardized data along with hyperparameter optimization, leading to [100, 68.4] accuracies. Here, doing a weighted KNN did not improve the performance, likely due to most features' unstandardized values not being too far apart. For DT, the best set of accuracies ([87.5, 64.5]) came from a DT implementation that makes use of standardized data, uses hyperparameter optimization, and uses just the most highly correlated features instead of selecting all of them. Only the cost function varied, as we used different ones for each dataset.

Comparing the accuracies given by the two algorithms, we find that KNN outperforms DT. This conclusion is further supported by the following ROC curves, where KNN mostly outperforms DT.



Given that the two datasets contain similar data on disease and health information, this motivates the question of how different models are more suitable for different datasets. Exploring datasets from different areas, potentially also comparing datasets that are entirely binary vs. entirely discrete, could provide interesting insights into when to use either model to get the best results.

While KNN has zero training time, DT was relatively computationally expensive (especially when doing the hyperparameter optimization). Additionally, as we've found that using highly correlated features provides a good alternative to using all features, this reduces KNN's computational complexity when predicting classes for new inputs. These factors can be important, depending on the use case.

Further exploration can be done with the cost functions in DT. Exploring why one works better in a dataset than the other, especially with regards to the data's structure, could provide interesting insights into how to better pick cost functions. Lastly, looking into ways of better handling data points with missing values (75/155 rows of the hepatitis dataset) to increase the amount of information (such as using cross-validation or a DT implementation that handles incomplete data) could also be very interesting.

## Statement of Contributions

All 3 teammates were involved in every aspect of the project, but each section was owned by a specific member. Adam worked on the weighted KNN implementation as well as the first sections of the write up. Frédéric worked on the DT explorations, graphs, and organizing and structuring the Colab notebook. Yann worked on the KNN implementation and the remaining write-up sections.

## Bibliographical References

The only references used are the course lecture slides and the course Github/Colab resources, as permitted by the assignment guidelines. Our models, implementations, and graph plotting are based on those resources, while some modifications have been made to adapt them to our needs. In addition, we used *Boosting Lazy Decision Trees* paper (see reference below) as a background information to our hepatitis dataset and DT algorithm.

- *Course GitHub*: <https://github.com/yueliyl/comp551-notebooks>
- *KNN Google Colaboratory*:  
<https://colab.research.google.com/github/yueliyl/comp551-notebooks/blob/master/KNN.ipynb>
- *DT Google Colaboratory*:  
<https://colab.research.google.com/github/yueliyl/comp551-notebooks/blob/master/DecisionTree.ipynb>
- *Model Evaluation and Selection Google Colaboratory*:  
<https://colab.research.google.com/github/yueliyl/comp551-notebooks/blob/master/ModelEvaluationAndSelection.ipynb>
- Xiaoli Z. Fern and Carla Brodley., *Boosting Lazy Decision Trees.*, ICML. 2003