COMP 551 - Mini Project 1

Nicolas Fertout, Luna Dana, Anthony Kumar

February 2022

Abstract

This project involved implementing the traditional classifying machine learning models: K-Nearest Neighbour (KNN) & Decision Tree and using these two models to make predictions on data from the Hepatitis & Messidor datasets, and comparing the performance of both models with each other across both datasets. We found the KNN model achieved a slightly better average accuracy across both datasets compared to the decision tree model.

1 Introduction

In this report, we investigate and analyze the performance between the KNN and decision tree classifiers on the Hepatitis & Messidor datasets (described in the section on Datasets). We use a K-Nearest Neighbour and a Decision Tree approach, both of which were implemented in the Python programming language, for classifying new instances of data into one of their respective classes. Our focus was on identifying the most important parameters for the models, as well as the model that yield the highest average for both dataset.

Overall, we found that KNN gives a better fit (using K = 5 for Hepatitis and K = 20 for Messidor). We also discovered that, while varying parameters can change the accuracy, those changes are not substantial. Besides, it seems that both datasets present issues: Hepatitis have only very few instances with Y = 2, while Messidor had lots of unlabeled and correlated features.

We will start by explaining what our data consists of and how it was cleaned appropriately in order to be used in our models, then we will give some insights on the methods and design choices made during our project, to finish with a presentation of our results and a discussions of our findings.

2 Datasets

2.1 Data Description

Hepatitis

The Hepatitis (means inflammation of the liver) set consists of 155 instances of 20 features. Each instance describes a case dying or not from hepatitis. There are some categorical features such as the Sex of the patient and some continuous variables such as the Age. The response variable is the CLASS which is a binary variable implying that we will fit a classification model on this data.

Messidor

The Messidor (which stands for Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology) set consists of 1151 instances of 20 features.

There are some categorical features such as the binary result of quality assessment and some continuous variables such as the diameter of the optic disc. The response variable is the CLASS which states whether an image contains signs of diabetic retinopathy or no. It implies that we will fit a classification model on this data since the response is binary.

2.2 Data Cleaning

Since some instances were incomplete, we had to clean the data in order to have more accurate results. The hepatitis data set only has 80 instances after the cleaning since some contained incomplete values.

The messidor data set remains intact. Since the raw files had .data and .arrf as extension, we converted them into .csv files to be able to use them as panda dataframes afterward.

2.3 Data Analysis

Key Statistics

Class	#	OpticDisk	MA (0.5)	MA (1)	EuclidDist	Quality	Screen	AM-FM
DR	611	0.108	45.473	22.966	0.523	Good : 100%	+ : 89%	1 : 31%
No DR	540	0.109	30.457	19.098	0.523	Good : 99%	+ : 94%	1 : 35%

Table 1: Statisitcs (mean) about the Messidor data set

Class	#	Age	Sex	Protime	AlkPhosphate	Bilirubin
Die	13	46	Men : 100%	41	125	1.91
Live	67	39	Men : 83%	66	98	1.08

Table 2: Statisitcs (mean) about the Hepatitis data set

Concerns

- Size of dataset : Overall, the dataset is not very big so it can be hard to see patterns or draw some conclusions. For example, in the hepatitis data set, the age mean is higher in the dying population but there is only 80 instances so this result might not be significant.
- Quality of dataset : In the hepatitis dataset, only 11 out of the 80 instances are women and none of them die. Therefore, it makes the results biased involving that being a woman limits the chances to die which might not be necessarily true. Therefore, we might have to discard the Sex feature in further analysis. Besides, we also lack observations with Y=”DIE”. We would need more to improve our models, but this rises ethical challenges.

3 Methods and Design Choices

3.1 KNN

For this model we had to make several design choices and tuning of parameters.

First, we need to adjust the hyper-parameter K, number of neighbors, to get the best performance. We decided to run our models using all possible values of K and choose the one with best performance over a large number of randomly split train and test sets.

Then, we also have to choose an appropriate distance function. Since the data contains both categorical and continuous variables of different ranges, we decided to use two different approaches. First, we rescaled all the continuous variables to be between 1 and 2, and then we used Euclidean and Manhattan distance functions, treating all variables as continuous (since their range is between 1 and 2). Second, we built another dataset, changing all the continuous variables in categorical (by applying a mask setting all values greater than the mean/median to 2, and the rest to 1). We then used the Hamilton distance function on this dataset.

Finally, since distances in high dimension are often similar between most points, we decided to fit our models using all the features, and then only the ones that contribute the most to the model. Those latter features were chosen after fitting the model on all possible couples of features, and then use the ones with higher accuracy.

3.2 Decision Tree

In implementing the decision tree model, we have to tune three parameters to our data: the max tree depth, the minimum number of instances required to split a node, and the cost function.

The choosing of the optimal value of max depth is based on the datasets. The optimal value is found for both datasets by simply keeping the other parameters constant and computing the average accuracy of the model over a sufficiently large number of random training data splits. By letting the max depth vary while holding the other parameters constant, we should get the max depth that maximizes our model's performance.

Similarly, to find the value for the minimum number of instances which maximizes the model's accuracy, we let the minimum required number of instances vary while keeping all other parameters constant and select the value with the highest accuracy.

We keep in mind that higher values (at extremes) of max depth and/or lower values of min instances will overfit and vice versa will underfit the model.

Finally, in our selection of cost functions, we choose the cost function which gives our model the highest accuracy.

4 Results

4.1 KNN Model

The KNN model had better performance for the Hepatitis dataset than the Messidor. However, varying parameters did not seem to have a substantial impact on the former.

Hepatitis

Number of Features Used	Average Accuracy
All Features	82.9%
10 Features	83.6%
5 Features	**85.8%**

Table 3: Average accuracy on the KNN classifier for different number of features on the Hepatitis dataset (*Euclidean distance, K=5*).

Distance Function	K = 1	K = 3	K = 5	K = 10	K = 30
Euclidean	84.9%	86.3%	85.8%	86.4%	83.5%
Manhattan	85.3%	84.7%	85.1%	85.8%	82.9%
Hamilton (*threshold:* mean)	82.6%	84.0%	85.5%	**86.5%**	83.2%
Hamilton (*threshold:* median)	82.6%	**86.5%**	**86.5%**	85.0%	82.9%

Table 4: Average accuracy of the KNN classifier on 100 randomly split train and test sets for different values of K and distance functions on the Hepatitis dataset, using 5 scaled features.

The *threshold* specified is the value at which we split continuous variables into categorical variables.

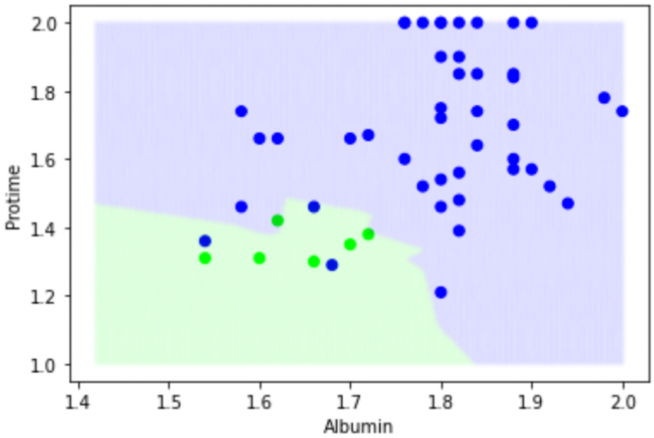

Figure 1: Decision boundaries given by the algorithm using 2 features (*Euclidean distance, K = 5*)

Messidor

Number of Features Used	Average Accuracy
All Features	65.4%
10 Features	**68.3%**
5 Features	61.2%

Table 5: Average accuracy on the KNN classifier over 100 randomly assigned train and test sets for different number of features on the Messidor dataset (*Euclidean distance, K=20*).

Distance Function	K = 1	K = 5	K = 20	K = 100	K = 300
Euclidean	65.4%	66.1%	**68.3%**	65.6%	61.3%
Manhattan	66.0%	66.6%	67.6%	65.0%	60.1%
Hamilton (*threshold:* mean)	55.6%	56.0%	58.1%	59.3%	58.3%
Hamilton (*threshold:* median)	54.5%	57.3%	59.3%	60.3%	58.6%

Table 6: Average accuracy on the KNN classifier for different values of K and distance functions on the Messidor dataset, using 10 features.

The *threshold* specified is the value at which we split continuous variables into categorical variables (1 or 2).

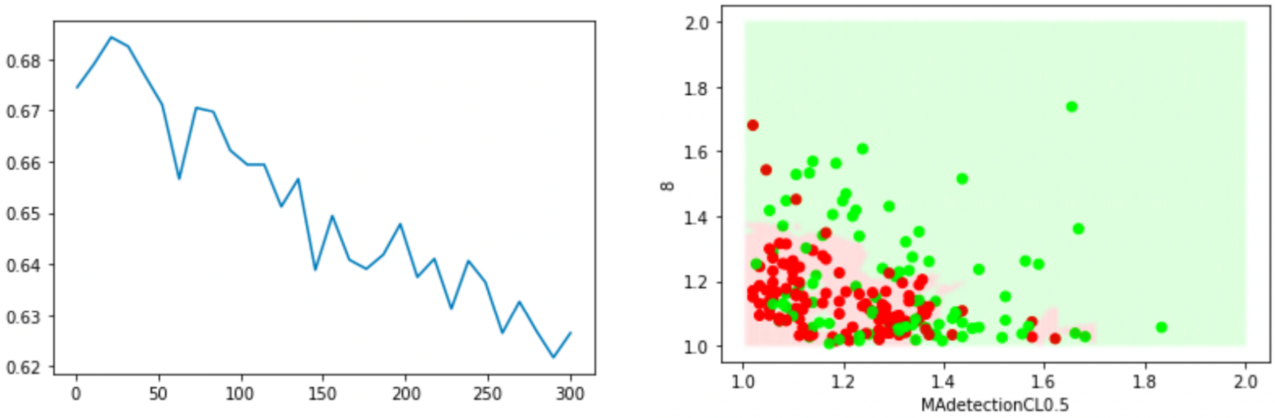

Figure 2: **LEFT**: Average accuracy on 50 train and test sets over values of K (*x-axis: K, y-axis: accuracy*), **RIGHT**: Decision boundaries given by KNN using 2 features (*Euclidean distance, K = 20*)

4.2 Decision Tree Model

The Decision Tree model had better performance for the Hepatitis dataset than the Messidor. However, varying parameters did not seem to have a substantial impact on the former.

Hepatitis

For the Hepatitis dataset, the decision tree model yielded its best performance with a max depth of 20 and using the misclassification cost function. In general, the model tends to perform slightly better using the misclassification cost function.

Cost Function	Depth = 1	Depth = 3	Depth = 5	Depth = 10	Depth = 20
Misclassification	81.0%	82.9%	82.9%	81.8%	**83.4%**
Entropy	77.9%	82.6%	82.4%	82.0%	82.1%
Gini Index	80.2%	81.6%	82.3%	81.8%	82.3%

Table 7: Average accuracy on the Decision Tree classifier for different values of max depth and cost functions on the Hepatitis dataset, using 5 features. Min instances = 5.

Messidor

Cost Function	Depth = 1	Depth = 3	Depth = 5	Depth = 10	Depth = 20
Misclassification	59.3%	59.5%	60.8%	62.6%	61.0%
Entropy	56.9%	63.1%	63.1%	62.5%	61.7%
Gini Index	57.0%	63.1%	**63.5%**	62.2%	61.0%

Table 8: Average accuracy on the Decision Tree classifier for different values of max depth and cost functions on the Messidor dataset, using 10 features. Min instances = 5.

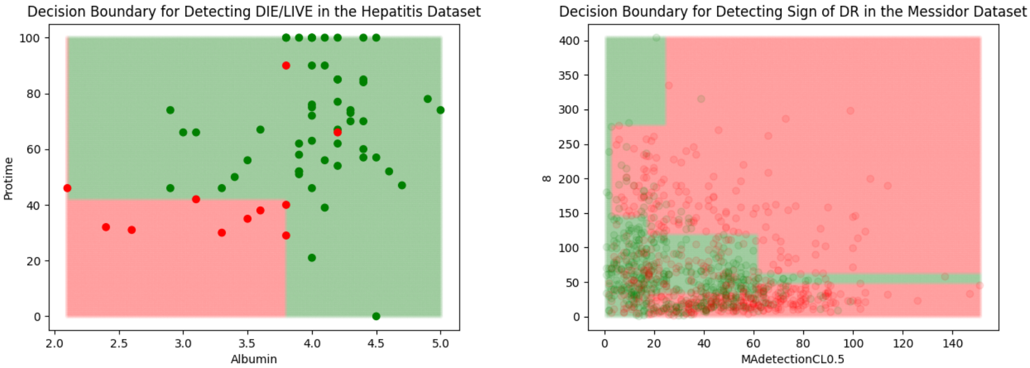

Figure 3: **LEFT:** Decision boundary for the Hepatitis dataset using 2 features: Albumin & Protime. **RIGHT:** Decision boundary for the Messidor dataset using 2 features: MAdetectionCL0.5 & 8.

5 Discussions and Conclusion

Overall, the best model appears to be the KNN classifier, even though both models give close accuracy.

From our results on the KNN classifier, we notice that the number of features included in training the model does cause a significant change in model accuracy. This aligns with the theory of the KNN algorithm since it is sensitive to noisy/less correlated features. However, for the Decision Tree classifier, the number of features in training the model does not cause a significant change in accuracy. This aligns with the theory of the decision tree algorithm since having a lower number of correlated features with the labels doesn't significantly effect the accuracy as those splits will, in general, have higher costs.

Besides, we observe that the best settings for KNN seem to be Hamilton distance and K = 5 for Hepatitis and Euclidean with K = 20 for Messidor. We can see this peak for Messidor, with the accuracy decreasing after this value of K (due to over-fitting). For Decision Tree, the best settings appear to be misclassification cost with Depth = 20 for Hepatitis and Gini index with Depth = 5 for Messidor.

However, it is worth noting that the Hepatitis data contains only 13/80 observations with Y=2 yielding volatile models and difficulty to conclude (since the accuracy obtained are often close to the proportion of Y = 1). Besides, the Messidor dataset contains a lot of unlabeled and highly correlated features, which is doing an "artificial" scaling of the correlated values, and makes it harder to interpret.

Both models gave a similar decision boundary for both datasets. Since we had to scale the features in the case of KNN, the values along the axis are not the same as the values in the graphs for decsion tree. Regardless, we observe the shape/region to be very similar. The boundaries differ in certain areas due to the models over/underfitting, as well as difference in the train sets used.

6 Group Member Contributions

- Everyone: Global planning of the project, discussion of design choices, writing of the report
- Luna Dana: Data cleaning script in Python, data analysis and statistics.
- Nicolas Fertout: KNN implementation in Python, tests using different settings.
- Anthony Kumar: Decision Tree implementation in Python, tests using different settings.