COMP 551 - Assignment 1 - Group 11

Yian Bian
yian.bian@mail.mcgill.ca

Yijia Jing
yijia.jing@mail.mcgill.ca

Zeyu Li
zeyu.li2@mail.mcgill.ca

Abstract

For this project we are given two datasets, ‘hepatitis’ and ‘messidor features’, and we used K-Nearest Neighbor and Decision Tree algorithm to analyze the data. Through KNN and Decision Tree, we can have a more integrated view of the data set, while at the same time we can compare the outputs and decision boundaries of the two models to decide which appears to be the most suitable one for the two datasets, and which one appears to be the one with less accuracy. Trying to obtain a better model with these two algorithms, we manipulated a lot of features and methods and drew conclusions about the two algorithms. In the general case, we found that K-NN model performs better than the Decision Tree model when the data sets are large, with higher accuracy in the ‘messidor features’ dataset, whereas the Decision Tree Model performs better when the data set is small, in other words, it works better on the ‘hepatitis’ data set. The construction of models and the whole analyzing process will be fully explained in the following paragraphs.

1 Introduction

In this project, we used K-NN and Decision Tree to analyze the two datasets. From previous studies [1] on the datasets of the UCI Machine Learning Repository [2], we know that both K-NN and Decision Tree are non-parametric methods, which means that they do not have an assumption on the distribution of the data, and both can be used for regression and classification problems. K-NN is usually known to have a better performance on larger data sets and Decision Tree behaves oppositely - it works better when the data set is small. They are suitable for analyzing both datasets and classifying the data points for further investigation. Our subsequent tests gave a brief illustration of how these two models perform on the two datasets under different distance functions and cost functions. For the dataset ‘hepatitis’, we found that the K-NN model performs better using Manhattan Distance, with an accuracy of 0.9375 under K=8, while Euclidean distance and Cosine Similarity have much lower accuracies. The Decision Tree model performs the best using misclassification and Gini Index when the depth is equal to 2, and they have the same accuracy of 0.8125. Decision Tree ended up having a larger area under both the ROC curve and PRC curve with the values of 0.89 and 0.99. On the second data set, KNN achieved an accuracy of 0.6435 using the Euclidean Distance when K=9, and Decision Tree performs well when depth = 6 with almost the same accuracy of 0.6435 using entropy. On this larger data set, KNN performs better in terms of both the ROC curve and PRC curve as expected. We also found that different distance functions and cost functions will have a huge impact on the performance of the model, with their effects being different for different datasets. Generally speaking, ROC curves should be used when there are roughly equal numbers of observations for each class like ‘messidor features’ and Precision-Recall curves should be used when there is a moderate to large class imbalance such as ‘hepatitis’.

2 Methods

2.1 Experiment Design

We are given two datasets to experiment on, which are Hepatitis and Messidor Features. We split each dataset into 60% for training, 20% for validation and 20% for testing. To understand the behaviors of both the K-NN and the Decision Tree models, we implemented the two models which we fit the two datasets into, predicted the class of a given data point and performed hyperparameter tuning with the data in the validation set. We also investigated the impact of different factors on a machine learning model. For example, the distance function for K-NN models and cost function for Decision Trees. We evaluated the performance with various types of metrics, including direct comparisons of accuracies and area under curves.

2.2 Model Implementation

We first implemented the K-NN model, which is a lazy learner that only saves the training samples we have upon fitting. When given an array of data to predict, it calculates the distances between the training and test samples using a distance function of our choice, gets the K nearest neighbors based on the sorted distances and returns the probability distribution by counting the appearance of the classes in the K nearest training samples. For a given sample, we pick the class that is higher in probability as the prediction.

The Decision Tree model is implemented with a greedy and recursive algorithm. We iterate through features and thresholds to compute the cost of having such a test using a cost function of our choice, and find the best feature and threshold that minimize the cost at each tree node. Again, we return the counted class distribution and make a prediction using the class with the highest probability.

Finally, we evaluate the accuracy by computing the percentage of predictions that are made correctly.

3 Datasets

3.1 Description

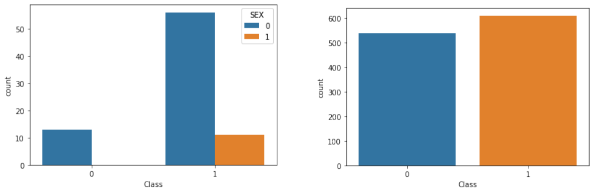
We performed analysis on two datasets. The first one, Hepatitis, is a 155*20 data set which is considered small, and the second one, messidor features, converted from an arff. file to csv, has a shape of 1151*20, a much larger data set than the previous one. 'hepatitis' contains missing values marked as '?' so they have to be cleaned later.

3.2 Data Cleaning

After eliminating the '?' values for Hepatitis, we got a 80*20 data frame. Then, we confirmed that there were no remaining missing values and verified that none of the values is null. Through dtypes function we examined the data types of each feature for subsequent type conversion.

3.3 Basic Statistics

The describe function displayed the count, mean, and standard deviation of each field. We grouped the data by class and sex to find the mean values for each class and sex. The corr. function helped to plot the map of each data set as a heatmap. Through the heatmap we can see the correlations between features, uncovering how they might influence each other. For the first data set, histology is negatively correlated to ascites and spiders. For the second data set, the MA tests are positively correlated to each other. The count data helped to find the distribution of binary fields. In Figure 1, the first graph shows that for Class1, there are more positives than negatives, indicating that the data is imbalanced. In contrast, the second data set is relatively balanced, indicating that there are similar amounts of data for positive and negative samples. In the histograms presented in the notebook, we can see the distribution of each field.



a) Hepatitis b) Messidor Features

Figure 1: Class Distributions of the Two Datasets

4 Results

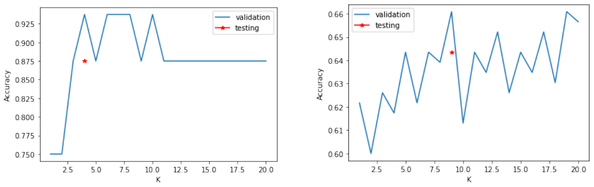
4.1 K-NN

4.1.1 Hyperparameter Tuning

We applied the K-NN model to the two datasets to predict the class given the input features, and ran experiments to find the highest accuracy after hyperparameter tuning on the value K. We first used Euclidean distance in the K-NN model. The validation set that we split from the dataset is used to evaluate the performance of a K value.

For the first dataset, Hepatitis, we found from the first graph of Figure 2 that when K=2, the model gives the highest accuracy on the validation set. Therefore, we choose K=2 and it gives a resulting accuracy of 87.5% on the test set. For the second dataset, Messidor Features, we can see from the second graph of Figure 2 that the peak is on K=9. Therefore, we choose K=9 and it gives a resulting accuracy of 64.35% on the test set.

The figures suggest that when K first increases, there is a general trend of increasing accuracy. However, the accuracy fluctuates or stays the same for subsequent K values. Therefore, there is not necessarily a linear association between K and accuracy. We need to choose K on a case-by-case basis by performing experiments on a validation set.



a) Hepatitis

b) Messidor Features

Figure 2: Accuracies with different K values

4.1.2 Distance Functions

To investigate the impact of distance functions on the performance of the K-NN model, we ran experiments on three types of distance functions: Euclidean, Manhattan and cosine similarity. For each type of distance function applied to the two datasets, we repeated the same experiments to choose the best K value that results in the highest accuracy. The results are shown in Table 1. For hepatitis, the Manhattan distance outperforms the other two distance functions. For Messidor Features, the Euclidean distance is slightly better than the other two. However, the accuracies are very similar and the difference can presumably be accounted for by the randomness in dataset splitting.

Table 1: Accuracies with different distance functions

	Euclidean	Manhattan	Cosine
Hepatitis	87.5%	93.75%	75%
Messidor	64.35%	60%	57.83%

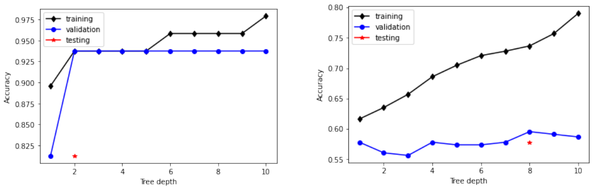
We also noticed that regardless of the distance function chosen, the resulting accuracy on the Hepatitis dataset is always higher than the Messidor Features dataset, which is much bigger in size. Since the K-NN model is sensitive to noise and outliers, more data might cause a drop in accuracy.

4.2 Decision Tree

4.2.1 Hyperparameter Tuning

We fit the Decision Tree models with the two datasets to predict the class given the input features. For this model, we tuned the hyperparameter Tree Depth by measuring the accuracies on the validation set. We first used the Misclassification Rate as the default cost function when we choose a feature as a test.

Tree Depth (D) is a hyperparameter we looked to tune to achieve the best performance. For the first dataset, hepatitis, we found from Figure 3 that when D=2, the model yields the highest accuracy on the validation set, and the resulting accuracy on the test set is 81.25%. For the second dataset, Messidor features, the model gives the highest accuracy on the validation set when D=8, and the resulting accuracy on the test set is 57.83%.



a) Hepatitis

b) Messidor Features

Figure 3: Accuracies with different tree depths

We observed that the increase in Tree Depth does not necessarily result in increased accuracy. For Hepatitis, the accuracy stays the same for all depths greater than 2. For Messidor Features, while the accuracy on the training set keeps increasing as tree depth increases, the accuracy on the validation set does not visibly improve, which suggests the possibility of overfitting at high Tree Depth.

4.2.2 Cost Functions

When we determine the features to use for the next test of a Decision Tree, we want to minimize the cost and be more certain about our choice. We often apply one of the three types of cost functions: Misclassification Rate, Entropy and Gini Index. To find out the one that best fits our needs, for each of the cost functions, we ran experiments on the two datasets to choose the best depth that gives us the highest accuracy and recorded it in Table 2.

Table 2: Accuracies with different cost functions

	Miscl. Rate	Entropy	Gini Index
Hepatitis	81.25%	75%	81.25%
Messidor	57.83%	64.35%	58.7%

We found that for the hepatitis dataset, Misclassification Rate and Gini Index both give a high accuracy of 81.25%, while for the Messidor Features dataset, Entropy outperforms the other two with an accuracy of 64.35%. There is not a determining conclusion as to which cost function is certainly the best. It can be seen from the results that a cost function can suit a particular dataset well but yields poor results for another.

4.3 Decision Boundaries

4.3.1 Hepatitis

The decision boundaries of the Hepatitis Dataset for the KNN and the Decision Tree model are shown

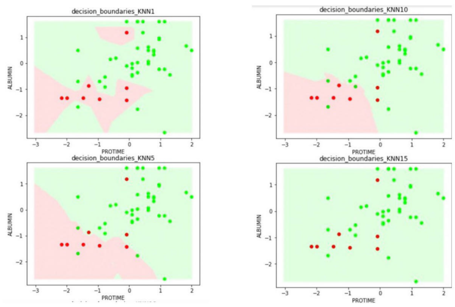


Figure 4: Hepatitis: Decision boundaries for KNN

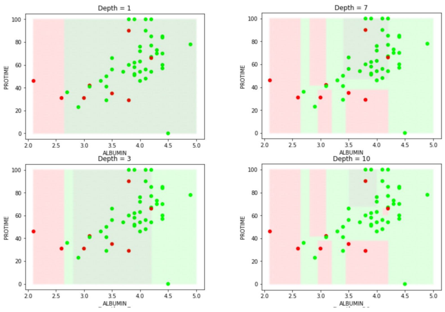


Figure 5: Hepatitis: Decision boundaries for Decision Tree

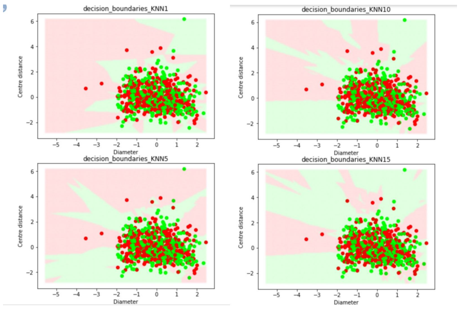


Figure 6: Messidor Features: Decision boundaries for KNN

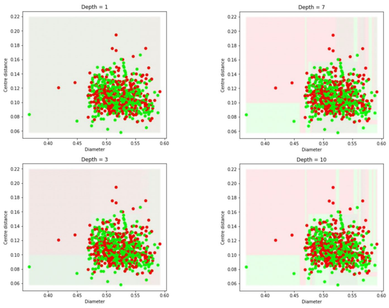


Figure 7: Messidor Features: Decision boundaries for Decision Tree

in Figure 4 and 5, respectively. Because Hepatitis Dataset has several features and we can only plot a 2-dimensional graph, we choose to plot 'Albumin' and 'Protime'. Although it is for sure that these two features can not represent the actual decision boundaries, except for the binary features, these two features have the highest correlation with the Class and they are not highly correlated. There is no doubt that from these coarse decision boundaries, some useful observations can be made.

It is clear from the graph that when K = 1, there is a large probability of over-fitting since there's a single red point that has a decision boundary abound it. However, the graph of K = 15 suggests a probability of under-fitting as it classifies all the points in one class. Comparatively, it seems quite reasonable to choose K = 5 or K = 10 according to the graphs.

In the decision boundaries for the Decision Tree, we can find that there are overlapping areas, and as the depth increases, the overlapping area decrease. But since there are only two features in the graph, we need more features to get more accurate decision boundaries.

4.3.2 Messidor Features

Similarly, we choose 2 features based on correlation with the Class. In theory, we should choose features 3 and 4 which are results of MA detection at the confidence levels alpha = 0.5 and 0.6, however, these two features are highly correlated (a correlation of 0.996), same things happen with other features with comparatively high correlation with Class, so finally we choose to plot the features "Centre distance" and "Diameter", but these two features do not have a high correlation with the class. From both decision boundaries, we can find that there are so much data in the Messidor Dataset and they cluster in one corner instead of spreading out, and also may be because the comparatively low correlation which combines makes it hard to draw the decision boundaries and also hard to tell which graph is better.

4.4 Model Evaluation

From Tables 1 and 2, we can see that for the Hepatitis dataset, the highest accuracies from the K-NN and the Decision Tree models are 87.5% and 81.25%, respectively. For the Messidor dataset, the highest accuracies from the two models are both 64.35%. These results suggest that the performance of K-NN and Decision Tree are similar in terms of the two datasets.

To understand the difference between K-NN and Decision Tree better, we used two common metrics to make comparisons. The Receiver Operating Characteristic (ROC) Curve shown in Figure 8 is a graphical plot that is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The Precision-Recall Curve (PRC) shown in Figure 9, as the name suggests, is created by plotting the Precision against the Sensitivity, or Recall. The higher than area under those curves are, the better a model performs.

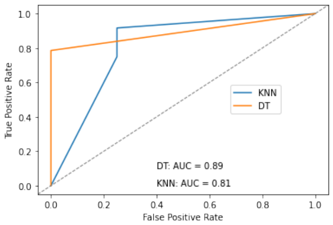


Figure 8: Hepatitis: ROC for the KNN and DT models

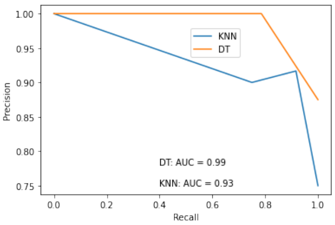


Figure 9: Hepatitis: PRC for the KNN and DT models

We plotted both ROC and PRC for the two models applied on the two datasets, and computed the values of Area Under Curve (AUC). We summarized the results in Table 3.

Table 3: Model Evaluations on KNN and DT

Metric	Hepatitis		Messidor	
	KNN	DT	KNN	DT
AUROC	0.81	0.89	0.71	0.56
AUPRC	0.93	0.99	0.75	0.57

As we have explored in the Basic Statistics Section, the Hepatitis dataset contains class-imbalanced data while the Messidor dataset contains class-balanced data. Therefore, we would generally prefer ROC for Messidor Features and PRC for Hepatitis, although in our experiments, the metrics showed that the two types of plots produce similar results. For Hepatitis, the Decision Tree results in a slightly higher AUC, while for Messidor Features, the K-NN model has a much higher value in AUC than the Decision Tree model. Interestingly, we found that the same model can perform very differently based on the nature of a dataset.

5 Discussion and Conclusion

5.1 Key Takeaways

The key takeaway from the project is that although we have empirical data on which model might fit a dataset better, there is no set rule for the perfect model or hyperparameter. We need to design experiments thoughtfully that test multiple aspects of the performance of a model. Examples of such experiments include the value of K and distance function for a K-NN model, the tree depth and cost function for a Decision Tree model, decision boundaries and multiple evaluation metrics.

5.2 Future Investigation

On account of dimension limitation of graph, we only pick two features for each model to graph the decision boundary, it is for sure that two feature can not epitomize the whole dataset, so to be more specific and accurate, we can first extract the features by using Principal Component Analysis, or some other methods, since both dataset has too many features, which may result the curse of dimensionality and impact the performance of models. After the feature extraction, it may be better and easier for us to draw the decision boundaries.

We could also investigate the complexity overhead of each algorithm, as when the experiments were carried out on a large dataset, there was a perceivable time delay especially for the Decision Tree model. We could possibly optimize the models algorithmically to decrease such overhead.

References

[1] Bálint Antal, András Hajdu. An ensemble-based system for automatic screening of diabetic retinopathy, 30 Oct 2014

[2] Dua, D. and Graff, C.. UCI Machine Learning Repository. University of California, School of Information and Computer Science, 2019