COMP 551 Assignment 1 Report

Authors: Yiran Shi, Alina Tan, Ge Gao
Due Date: October 6th, 2022

1. Abstract

In this assignment, our group coded the k-Nearest Neighbor and the Decision Tree learning models for data classification tasks, and applied these algorithms on two representative datasets previously researched in the area of life sciences. We identified many apparent characteristics of these datasets and tested for best parameter settings for the learning models. We also tried to modify both the datasets and the learning algorithms. Eventually, we found out that the KNN learning model was often able to outperform the DT leaning model on relatively small datasets that have been standardized and feature-selected, while the DT learning model often had minor accuracy advantages but much more time consumption over KNN on large datasets without any pre-processing.

2. Introduction

The basic tasks of this assignment is to acquire and analyze two datasets, to implement k-Nearest Neighbor and Decision Tree data classification models that would effectively classify these two datasets, and eventually to find out the best-performing parameter settings on these models relative to classification accuracy on the datasets.

Both datasets come from life science experiments. The first dataset has to do with characteristics of hepatitis patients that are either still alive or have already died. More than half of the characteristics of this dataset are binary, and many samples are missing the data for one or more features. After we deleted all data points that had missing or questionable values, only a rather small dataset was left. It is notable that this hepatitis dataset seems to be widely used among many machine learning model research. For example, in Ensembles of Similarity-Based Models, researchers proposed multiple methods that combine multiple ordinary or weighted KNN models for better performance compared to a single KNN model. They used Manhattan distances and went through feature selection. Eventually, they found that basic KNN applied to this dataset is already outperforming many more complex models such as neural networks, and their "ensembled KNN" gives even greater performances (Wlodzisfaw Duch et al., 83-93). The second dataset is generated from many Messidor images to predict whether the patient has diabetic retinopathy or not. We noticed that many features of this dataset essentially came from the same test, but are just divided using different confidence levels. It is also notable that some of these features already contain standardized data.

Our KNN and DT models applied to these data sets gave multiple findings. Between the datasets, the Messidor dataset always gave much lower prediction accuracy than the hepatitis dataset. If we compare KNN with DT, at respective best parameters, KNN had major advantages with the hepatitis dataset, but DT could end up with a draw or some tiny advantage when analyzing the Messidor data.

One interesting finding is that, we tried applying some correlation-based feature selection to both datasets before feeding it into DT, and DT surprisingly gave better results compared to no preprocessing. Since DT is supposed to be selecting important features on its own, whether this phenomenon is coincidental or not would require further research.

3. Methods

The KNN prediction function was implemented as the following: we first calculate the distance (Euclidean or Manhattan) between each data point in the test dataset and each data point in the training dataset, then for each testing data point, we identify the indices of its K training neighbors with the smallest distance from it, count class labels of these neighbors, and return class probabilities and the most probable class of this data point.

The DT model was more complex: firstly a *greedy_test* function is responsible for selecting a reasonable separation criteria for data in a node. For each feature, given a set of possible threshold values, we can separate all data in a node into a smaller pile and a larger pile using that feature/value pair, then calculate the summed "impurity" (cost) of the two piles using a given cost function. Repeat for every possible threshold value of every feature, and return the feature/value pair with the lowest cost. Note that when choosing possible threshold values for large datasets, testing every average of two consecutive data points in a sorted column of the dataset will be extremely time consuming, so we modified the DT for Messidor dataset into choosing x evenly distributed values between maximum and minimum of the given feature, where x is customizable. When we want to fit a training set, starting at the leaf node containing all data points, we use *greedy_test* to give a separation criteria at the node, then divide the smaller pile and the larger pile into its children, then we recursively call *greedy_test* on its children, until any stopping criteria (max tree depth, too little instances in a node or complete purity in a node) is met at a node. In order to predict a test set, for each data point, let it descend the decision tree based on the separation criteria at each node, until it ends up in a leaf node. Class labels of training data in the leaf node are used to return class probabilities and the most probable class of this data point.

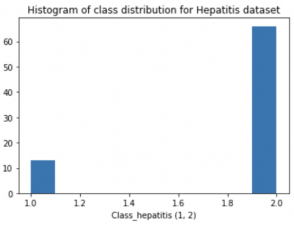
4. Datasets

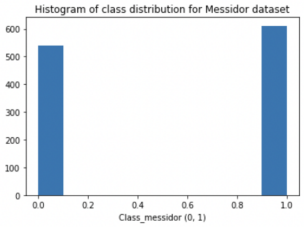
Hepatitis originally had 155 data points with 20 features including the Class feature. We first removed all the data points that has missing values. There were 80 data points left after this step. Then to exclude any malformed feature, we limited all the categorical features between 1 and 2, and we also set some limits for the numerical features according to the data range given by the website. The limits for all numerical features are shown in Table 1. There are 79 samples left. The Class distribution is also shown below, there are 13 samples in Class 1 and 66 samples in Class 2.

We then computed the related mean and standard deviation for each numerical feature. After finding the correlation between Class feature and other features, we observed that the correlation between Class and $LIVER\_FORM$ and the correlation between Class and $SGOT$ are very close to 0, and so they are uncorrelated to the Class. We could exclude these two features when training the data. $ASCITES$ ($Corr(ASCITES, Class) = 0.48$)

Feature (i)	Range
AGE	10 <= i <= 80
BILIRUBIN	0.25 <= i <= 5.0
ALK_PHOSPHATE	25 <= i <= 300
SGOT	13 <= i <= 500
ALBUMIN	2.1 <= i <= 6.0
PROTIME	5 <= i <=100

Table 1





and ALBUMIN ($Corr(ALBUMIN, Class) = 0.48$) both have the highest absolute value of correlation with Class, which indicates that they are strongly positively correlated to the Class. In the further step, we will also use these two features to demonstrate the decision boundary.

The original Messidor dataset has 1151 samples without any missing value. Then to exclude any malformed feature, we limit the categorical features between 0 and 1. Since the website does not give the range for the numerical features, we keep all the numerical features without setting any limit. We change the original Class feature b'0' and b'1' to 0 and 1 which will make the further programming step more efficient. There are 1151 samples left after cleaning the data. The Class distribution is shown below, we have 611 samples in Class 0 and 540 samples in Class 1.

We computed the correlation and observed that the feature $Exudate1$, $Exudate2$, $Exudate3$, $Distance$, $Optic\_Disk$, and $BiClass$ are uncorrelated with the Class feature since their correlations with the Class feature are close to 0. Comparatively, the feature MA1 and MA2 are strongly positively correlated to Class.

5. Results

We run several experiments to test the performance of our algorithms. First, we shuffled the two datasets and splitted the datasets into training set, validation set and test set with the ratio of 1:1:1. We then performed hyperparameter tuning for KNN to find the best k and distance function. For the Hepatitis validation set, the best k was 7 with the accuracy of 1.0 via Manhattan distance calculation (Table 2 A). The best k was 8 with the accuracy of 0.68684 via Manhattan distance calculation for the validation set of Messidor (Table 2 B). This result revealed that standardizing data and/or dropping low correlation features could improve the accuracy of the KNN. The distance calculation with the Manhattan method worked better than the Euclidean method. Next, we did hyperparameter tuning for the Decision Tree method to find the best tree depth and the best cost function. Specifically, different cost functions led to different accuracies. For the validation dataset of hepatitis, setting max_tree_depth to 1 with misclassification cost function gave the highest accuracy of 0.88461(Table 2 C). Feature selection by dropping low correlation features did not improve the performance in the hepatitis dataset. For Messidor's validation set, max_tree_depth of 3 with entropy cost function gave the highest accuracy which was 0.66052 (Table 2 D). In this case, deleted low correlation features improved the accuracy from 0.63947 to 0.66052. The $min\_leaf\_instance$ did not affect the accuracy significantly. The decision boundaries for KNN and decision trees of both datasets are shown in Figure 1.

A

KNN					
Hepatitis dataset	without standardization (std)	with std	with feature selection	with feature selection and std	with feature selection and std
Best k	k = 3	k = 3	k = 4	k = 7	k = 7
Distance function	Euclidean distance	Euclidean distance	Euclidean distance	Euclidean distance	Manhattan distance
accuracy	0. 84615	0.96154	0.88461	0.96153	1.0

B

KNN					
Messidor dataset	without standardization (std)	with std	with feature selection	with feature selection and std	with feature selection and std
Best k	k = 3	k = 1	k = 7	k = 1	k = 8
Distance function	Euclidean distance	Euclidean distance	Euclidean distance	Euclidean distance	Manhattan distance
accuracy	0. 65789	0.53947	0.66579	0.53947	0.68684

C

Decision Tree						
Hepatitis dataset	without standardization (std)	without std	without std	with feature selection (drop low correlation)	with feature selection (drop low correlation)	with feature selection (drop low correlation)
Best tree depth	1	1	1	1	1	1
Cost function	misclassification	entropy	Gini index	misclassification	entropy	Gini index
Min leaf instance	1	1	1	1	1	1
accuracy	0. 88461	0.80769	0.80769	0.88461	0.80769	0.80769

D

Decision Tree						
Messidor dataset	without standardization (std)	without std	without std	with feature selection (drop low correlation)	with feature selection (drop low correlation)	with feature selection (drop low correlation)
Best tree depth	4	6	1	3	3	4
Cost function	misclassification	entropy	Gini index	misclassification	entropy	Gini index
Min leaf instance	1	1	1	1	1	1
accuracy	0. 61578	0.63947	0.61052	0.65789	0.66052	0.65789

Table2. The tables A-B showed the best k and corresponding accuracy with KNN algorithm in terms of feature selection, standardization, and distance calculation (Euclidean and Manhattan distance). The tables C-D showed the accuracy derived from different combinations of hyperparameters: max_tree_depth, cost function, and min_leaf_instance.

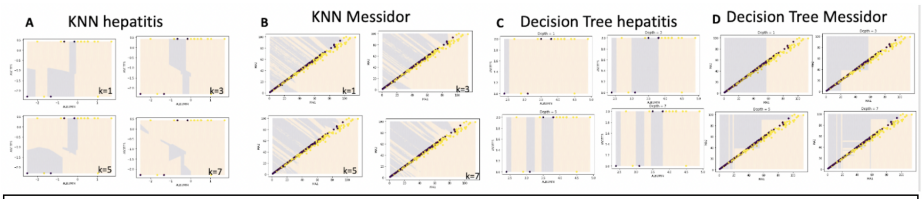


Figure1. The figures A-B consists of two decision boundaries for KNN algorithm with various hyperparameter k (1,3,5,7). The decision boundary of hepatitis dataset used two highest correlation features, albumin and ascites. The Messidor dataset used MA1 and MA2. C-D consists of two decision boundaries for Decision Tree algorithm with various hyperparameter max_tree_depth (1,3,5,7). The decision boundary of hepatitis dataset used two highest correlation features, albumin, and ascites. The Messidor dataset used MA1 and MA2.

After hypreparameter tuning, we run experiments on the test set. For KNN implementation, we explored the best k was 7 for hepatitis and the accuracy of the test dataset was 0.85185 for both Euclidean and Manhattan distances. The best k was 8 for Messidor and the accuracy was 0.63775 via Manhattan distance. For Decision Tree, the test accuracy for Hepatitis could reach 0.81481 when setting max_tree_depth to 1 and using the misclassification cost function. Other cost functions could reach their maximum accuracy when max_tree_depth is 1, and both led to an accuracy of 0.66667. Also, the test set of Messidor performed best when max_tree_depth was 4 with the Gini index cost function with an accuracy of 0.61479. The Entropy cost function showed the accuracy of 0.60969 with a depth of 6, and the misclassification function worked best when the depth was 3 with an accuracy of 0.57143. To further improve the accuracy, we select the three most correlated features of the Hepatitis dataset ($ALBUMIN$, $ASCITES$, and $HISTOLOGY$) and run the decision tree method on the selected features. The accuracy improved significantly. Specifically, misclassification with depth 1 had an accuracy of 0.88888; entropy with depth 2 had an accuracy of 0.92592 and Gini index with depth 2 had an accuracy of 0.92592. Similarly, we chose four highly correlated features for the Messidor dataset and the accuracy also increased. The misclassification with depth 10 had an accuracy of 0.64796; entropy with depth 6 had an accuracy of 0.66327 and the Gini index with depth 9 had an accuracy of 0.67092. Finally, we used the ROC curve to visualize the relationship between sensitivity and specificity (Figure2). The hyperparameters used were chosen from the ones that perform with the best accuracy. The ROC score (AUC) of hepatitis was higher the decision tree and the AUC of Messidor was higher in KNN. We also used weighted KNN to boost the performance for messidor dataset, but the result did not show significance. Finally, we decreased the run time when deal with a huge dataset by improving the decision tree algorithm.

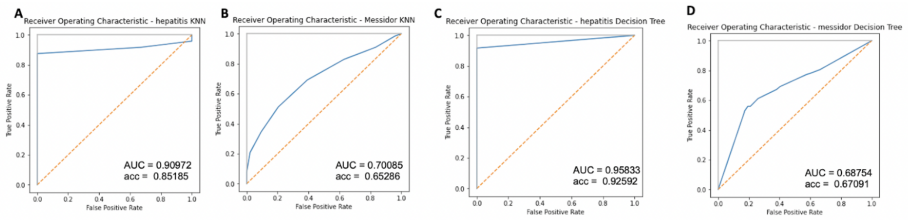


Figure2. This figure combined the ROC curve for hepatitis (A-B) and Messidor (C-D) dataset with KNN and decision tree. The grey line represented the ideal curve and the orange dotted line indicate the no predictive value. The blue line represented the actual test performance. AUC is the area under the curve and acc stands for accuracy.

6. Discussion and Conclusion

Our purpose is to test performance of classification for KNN and Decision Tree on Hepatitis and Messidor dataset and improve the accuracy. First we compared the unstandardized dataset and standardized dataset to fit for the KNN model, we found that the standardization improves the accuracy of Hepatitis, whereas it decreases the accuracy of Messidor. We think the reason of this decreased accuracy is that some features of Messidor should be standardized but the categorical features might not. And so, standardization did not improve the quality of Messidor dataset. Feature selection by dropping the uncorrelated features for both two datasets also improves the accuracy. We also compared the accuracy of using Euclidean distance and Manhattan distance. We found that Manhattan distance provides higher accuracy for both two datasets. The high ROC score shows that using KNN model with Manhattan distance to classify Hepatitis is good. However, KNN does not perform well on Messidor.

We used similar strategy to improve the accuracy of Decision Tree for both two datasets. Since scaling would not affect the performance, we directly started by dropping the uncorrelated features. However, the ROC score was very low. We then tried to only select highly correlated features. We selected three features for Hepatitis and four features for Messidor. The test accuracy and ROC score were both improved for Hepatitis. Although there is also improvement for Messidor dataset, the test accuracy and ROC score are still low which shows that Decision Tree classification does not perform well on Messidor.

We used our conclusions of data-processing to draw the decision boundary. The decision boundary shows the distribution of the data points by only selecting two features with highest correlation with Class. Prediction using KNN and Decision Tree shows similar decision boundaries for both two datasets.

To conclude, Decision Tree with hyperparameter of $depth = 6$, $min\_leaf\_instance = 1$, and entropy cost made the best classification for Hepatitis. Since neither KNN nor Decision Tree generated high accuracy for Messidor dataset, we could try out other classification models such as linear regression for further investigation.

7. Statement of Contributions

All three of us wrote the implementation of KNN and decision tree algorithm together. We did the experiments of validation datasets together. For the report, Yiran Shi is responsible for abstract, introduction and method; Ge Gao is responsible for dataset and discussion and conclusion; and Alina Tan is responsible for result and statement of contributions.