Fall COMP 551 Assignment 1

October 2022
Group 3: Steve Wen, Chloe Si, Navid Hassan Zadeh

Abstract

In this assignment we constructed and compared K-Nearest-Neighbours (referred as KNN) and Decision Tree (referred as DT) models for downstream classification tasks on the hepatitis and diabetics datasets. Based on our findings, KNN outcompetes DT, but it is more vulnerable to random noise. Also, given the properties of the datasets, where some attributes did not show distinct patterns for different classes, we implemented feature selection on KNN, which could diminish the influence of those confounding attributes. On the other hand, we did not apply that to DT since it always greedy selects best feature and threshold.

Introduction

KNN and DT are simple and basic machine learning models that have been widely deployed yet generate good results. Based on the knowledge from the lectures, we implemented these models from scratch and did the hyperparameter tuning to compare their performance. Two datasets are then used to examine the models, the Hepatitis dataset and Diabetics dataset. These datasets have significantly distinct data properties so the performance on them varies. In general, KNN performs better than DT on the given datasets and we observed both the models perform well on the Hepatitis dataset. Due to the randomness nature of KNN, shuffling and averaging over a training batch are utilized to get an unbiased view of its performance. Moreover, we also applied feature selection in KNN. The detailed results will be discussed in the following sections.

Dataset

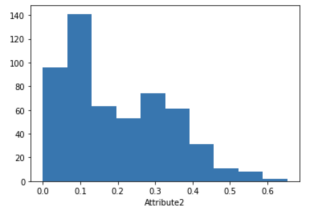
We performed a thorough visualization and exploration analysis on both datasets to have a better understanding of them. We first looked at all the features of both datasets and made several graphs to demonstrate the distribution of the data and provide some statistical analysis in general as well as separately for each class of each dataset. We also manipulated both datasets through Panda's dataframe and removed instances with missing or incomplete values. Then we applied normalization on both datasets to put all the features on the same scale.
(The left hand side figures show the most distinguishable feature in the two datasets, we could observe that the shape varies the most and so does the mean)

Hepatits Dataset

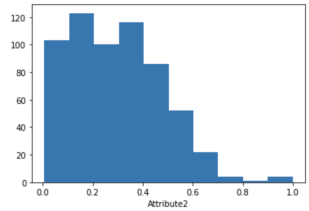
This dataset contains 155 instances of individuals with 19 recorded features with two classes DIE(1) or LIVE(2). Some of the instances had missing features denoted by "?" and they were removed from the dataset. After the cleaning, there remained 80 instances, 13 belong to class 1, 67 class 2.

Diabetic Retinopathy Debrecen Dataset

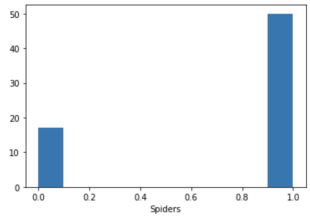
This dataset contains 1151 instances of individuals with 19 features, two classes 0(no sign of DR) or 1(contains signs of DR). There was no missing data in this dataset, but attributes 8-15 contained the same information as 2-7, with the only difference that they were normalized. To avoid duplicate normalization, we removed attributes 8-15. We ended up with 540 instances of class 0 and 611 instances of class 1, each with 11 features. The class does not have significantly different feature patterns for the 2 classes, which means that we cannot distinguish the data easily, and thus we expected to see a lower accuracy. Compared to the first dataset, this dataset is more balanced and has a higher number of instances, which means that it is less prone to give biased results.



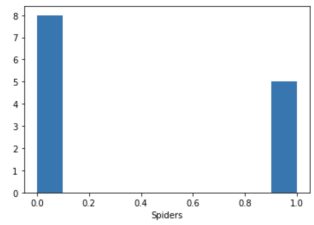
(a) Attribute 2 for class 0 in diabetics



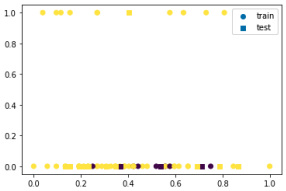
(b) Attribute 2 for class 1 in diabetics



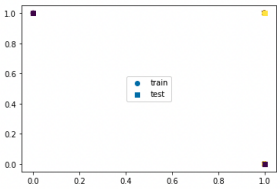
(c) Spiders for class 2 in hepatitis



(d) Spiders for class 1 in hepatitis



(e) Normalized Hepatitis distribution of first two dimensions



(f) Normalized Diabetics distribution of first two dimensions

Figure 1: Plots of the datasets

Method

The K-Nearest-Neighbour algorithm and the decision tree algorithm are both supervised classification learning models that can predict the class of a test data. The KNN algorithm is a lazy-training model so we only need to fit in all the training data and predict based on the k-nearest-neighbours, which are determined based on a distance function. The hyper-parameters in KNN are the k value and the distance function. In our model, we used the values 1 to 20 for k, and for the distance function we implemented the euclidean, manhattan and cosine functions. Moreover, since the model is prone to randomness and has varying precision under different data fed into it, we applied multiple shuffles with a random seed and averaged over the results, which would result in a more stable and generalized accuracy.

In the decision tree model, we fit the tree recursively from the root through greedily selecting the best feature and best threshold in each node. Then the input test data will move along the branches in the trained decision tree and the class of it will be determined by the leaf's class which it belongs to. The hyper-parameters in DT are the max-tree-depth, min-leaf-instances and cost function. In our model, we used 1-20 for max-tree-depth, misclassification rate, entropy and gini-index for the cost and we decided to keep the min-leaf-instances=1 as a constant.

Feature Selection

KNN: Key features are extracted before feeding as the model is sensitive to random noises. The first method we tried is to use a forest of trees to evaluate the importance of features on an classification task and selected features with highest importances. However, the performance suffers when we tested on the diabetics dataset. The test accuracy on the best K found by validation was only 53.75%. The one we finally decided to use is SelectKBest where it tries to select K features that have the highest scores computed by the score function. The score function we used is χ^2 as chi-square test measures dependence between variables. In this way, we are able to select features that are most relevant to classification. The features for hepatitis are sex, malaise, ascites, histology. For diabetics are results of MA detection, column 2 to 5 (No actual name given).

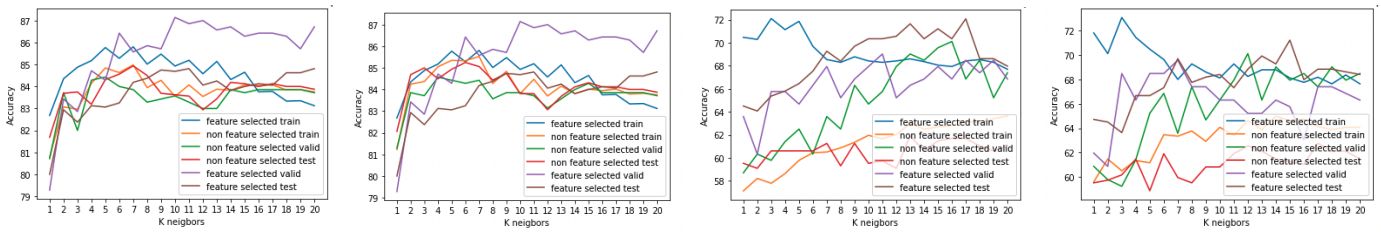
Decision Tree: The key features are selected by the model using greedy selection. That is we greedily split the data in a way that minimizes the cost on every node of DT. The feature we got based on the model with the best max_depth, for Hepatitis are Protime and age and bilirubin. For Diabetics, are results of pre-screening, MA detection($\alpha = 0.5, 0.6, 0.7, 0.8, 1.0$), the euclidean distance of the center of the macula and the center of the optic disc, and the diameter of the optic disc.

Results

To achieve better results for both algorithms (KNN and DT), we shuffle the data, split it into training data(train+validation) and testing data and find the most accurate hyper-parameter optimizations. Our obtained results are shown in Table 1. One thing to notice is that all the results for KNN are averaged to get the true accuracy of the model.

Dataset	Model	Accuracy	Hyperparameters
Hepatitis Dataset	KNN - Euclidean	**84.69%**	selected, k=10
	KNN - Manhattan	**84.69%**	selected, k=10
	DT - Misclassification	79.17%	max-depth=3, min-leaf=1
	DT - Entropy	83.33%	max-depth=1, min-leaf=1
	DT - Gini index	83.33%	max-depth=1, min-leaf=1
Diabetic Dataset	KNN - Euclidean	**70.35%**	selected, k=11
	KNN - Manhattan	69.70%	selected, k=7
	DT - Misclassification	64.16%	max-depth=8, min-leaf=1
	DT - Entropy	63.58%	max-depth=9, min-leaf=1
	DT - Gini index	63.01%	max-depth=9, min-leaf=1

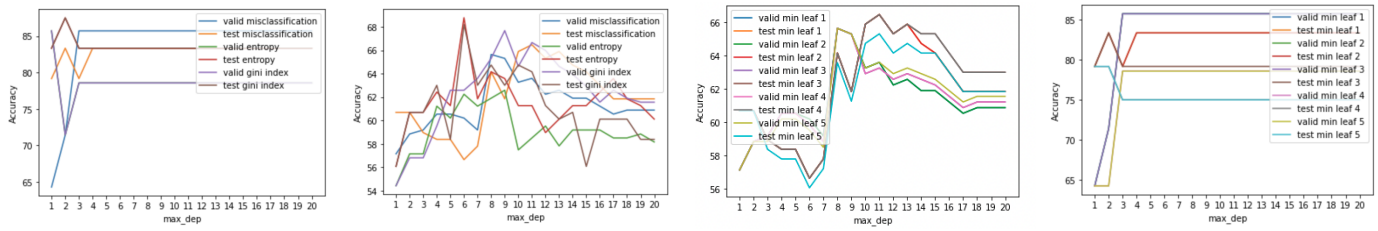
Table 1: Accuracy and Hyperparameter Obtimization Results (Feature selection applied on KNN models)



(a) hepatitis euclidean (b) hepatitis manhattan (c) diabetics euclidean (d) diabetics manhattan

Figure 2: Accuracy of predictions of KNN with different hyper-parameters and distance functions

As we can observe from the plots for the hepatitis dataset, for the model KNN, when K increases, The training accuracy first peaks and then decreases. This can be explained because when first K goes up, more examples are taken into consideration and would have a more accurate prediction. However, when K gets too big, it would start to consider those points are actually pretty far. Thus, the accuracy would drop as we start to consider unnecessary data points. Also, feature selected KNN shows better performance as we could see from the table and the plots. One thing worth talking about is the model performance on different sets of hyper-parameters in DT. To get a deeper understanding how the model works, we did experiments on various combinations of max-depth and min-leaf-instances. The performance on the hepatitis dataset is quite surprising as we did not observe any significant accuracy changes along with the changes of min-leaf-instances. One reason might be that hepatitis dataset only has a small number of instances. When fitting into DT and especially the max depth value is high, it is then highly possible that the leaf is made up of only one class, which is pure, so the change of min-leaf-instances won't affect the results(as a pure leaf won't further split). Further we want to investigate into this and to find out the exact reason why. And another thing to mention is the performance of certain cost function on the hepatitis dataset is unchanging given different max depth of the tree. More experiments need to carry to study the underlying mechanisms behind it. Lastly, we also implemented weighted-KNN, but we decided to keep the original one since the results become worse in weighted-KNN.



(a) DT on hepatitis with various cost functions

(b) DT on diabetics with various cost functions

(c) DT Misclassification diabetics various min-leaf-instances

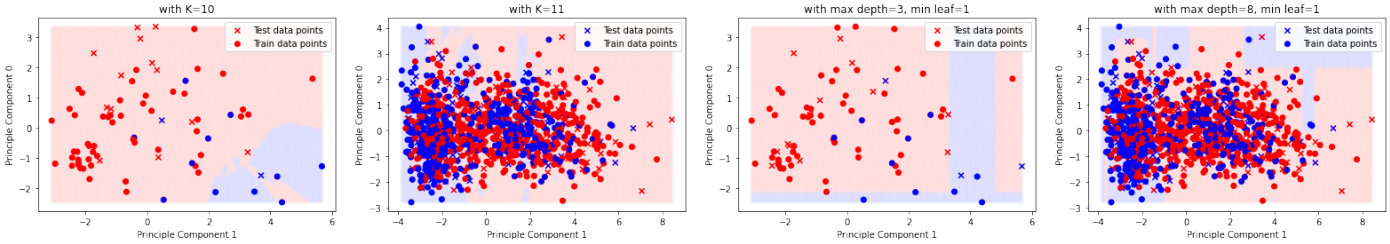
(d) DT Misclassification hepatitis various min-leaf-instances

Figure 3: Accuracy of predictions of DT with different hyper-parameters and cost functions

Decision Boundary

To draw the decision boundary (figure 4), we extracted the principal components and used the "mesh-grid" method in numpy to be able to classify all the points into a 2-d grid. Then, by running the model on a generated set of thousands of evenly spaced points throughout the rectangular grid, we generated an approximate decision boundary diagram for both models and both datasets. Note that for the training data we used the small filled-in circles, and for the test data points we used "x" symbols. Both are coloured according to their original classes. In the hepatitis dataset, red is class 2 and blue is class 1. In the diabetic dataset, red is class 1 and blue is class 0.

To illustrate one example of how KNN's choice of k works on the hepatitis dataset, we see that in its decision boundary map, since we had chosen k=10, isolated data points of class 1 (blue) that are surrounded by class 2 (red) points do not produce any boundary region around them, while the class 1 data points that have some neighbours of their own class around them produce a boundary region of their own.



(a) KNN (euclidean distance) on hepatitis dataset with k=10

(b) KNN (euclidean distance) on diabetics dataset with k=11

(c) DT (misclassification) on hepatitis dataset with max-depth=3, min-leaf=1

(d) DT (misclassification) on diabetics dataset with max-depth=8, min-leaf=1

Figure 4: Decision boundaries of KNN and DT on both datasets with the selected hyper-parameters

Confusion Matrix

Figure 5 shows the confusion matrices for the predictions of KNN and DT on both datasets. Note that in the Hepatitis dataset, we consider class 1 (DIE) as negative, and class 2(LIVE) as positive. However in the diabetic dataset, we consider class 0(contains signs of DR) as negative and class 1(No signs of DR) as positive.

Discussion and Conclusion

We implemented and applied the KNN and the decision tree (DT) models on two different datasets with different number of properties and instances. We also performed a feature selection procedure in KNN

Confusion Matrix of KNN (Euclidean,k=10) for Hepatitis data set

	Predicted Negative	Predicted Positive
Actual Negaive	1	1
Actual Positive	2	12

Confusion Matrix of KNN (Euclidean, k=11) for Diabetic data set

	Predicted Negative	Predicted Positive
Actual Negaive	71	56
Actual Positive	33	71

(a) KNN Confusion Matrix on both datasets.

Confusion Matrix of Decision Tree (Misclassification,max-depth=3, min_leaf=1) for Hepatitis data set

	Predicted Negative	Predicted Positive
Actual Negaive	1	0
Actual Positive	2	13

Confusion Matrix of Decision Tree (Misclassification,max-depth=8, min_leaf=1) for Diabetic data set

	Predicted Negative	Predicted Positive
Actual Negaive	43	44
Actual Positive	61	83

(b) DT Confusion Matrix on both datasets.

Figure 5: Confusion Matrices of both models (KNN and DT) and on both datasets(Hepatitis Diabetic)

and evaluated its efficiency by comparing the prediction accuracy results. Additionally, we analyzed the effects of the hyperparameters of each model on both datasets to arrive at optimal choices of hyperparameters. The decision boundaries and confusion matrices at the end provide a summary for comparing the general prediction performance of each model on each of the datasets.

We observed that the hepatitis dataset was easier to classify in both models because the fact that the provided features were effective in determining the class of each instance. However, since this dataset contained a rather small number of instances, the final accuracy often fluctuated in different shuffles, especially in KNN. This is why we decided to repeat the procedure on 100 random shuffles and report an average of the accuracy overall to arrive at more stable results. However, we only repeated the KNN process for 5 shuffles for the diabetics dataset since it contained a higher number of instances, and, thus, this problem didn't occur as much.

In comparing the datasets, it is easy to see from the decision boundary diagrams that the diabetics dataset doesn't strictly follow a good pattern in any particular form of distribution of its data points of different classes. In other words, the clusters of datapoints of the different classes are intertwined in each other. This is because none of the features of this dataset provide a good source of information for distinguishing the different classes of the instances. Therefore, both KNN and DT performed much better on the hepatitis dataset than the diabetics dataset.

Overall, KNN seems to perform slightly better than DT on both datasets, however the final accuracy of both models is close to each other. Furthermore, In KNN, both euclidean and Manhattan distance functions resulted in a very similar accuracy, with euclidean outperforming Manhattan by about 1% in either dataset. In DT, both misclassification and gini index methods led to a similar accuracy for both datasets, while the entropy method gave less accurate results.
For future investigations, we would like to do more experiments on understanding DT behaviors with different cost functions.

Statement of Contributions

Navid Hassen Zadeh: Decision Boundary, Confusion Matrices, Report Writing
Chloe Si: Data Preprocessing Visualization, Model Construction, Report Writing
Steve Wen: Model Construction, Model Debugging, Feature Selection, Report Writing

References

- How to Normalize Data Between 0 and 1
  `https://www.statology.org/normalize-data-between-0-and-1/`
- Sklearn SelectKBest
  `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html`

- Sklearn Chi2-Test
  https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html#sklearn.feature_selection.chi2
- Chi-Square Statistic: How to Calculate It / Distribution
  https://www.statisticshowto.com/probability-and-statistics/chi-square/