Project1 Report

KNN and Decision Tree

Group 11

Chuqi Wang, Jiechen Zhang, Yubai Zhang

Feb 7, 2022

COMP 551

Abstract

In this project, we utilized K-Nearest Neighbors (KNN) and Decision Trees, two machine learning models used to classify samples randomly selected from two datasets. We also investigated the performance of two machine learning models on two benchmark datasets and we found that their outcomes are quite different. The model performance is directly impacted by our choice of feature selection. The choice of each model's hyperparameters will also be set to optimize its performance. In this project, we will explore these topics in detail. In general, the results of running the experiments show that the Decision Tree algorithm achieved relatively higher accuracy and has faster performance than the K-Nearest Neighbors (KNN) algorithm on the two datasets.

1 Introduction

We implemented two models, one using the K-Nearest Neighbors (KNN) and the other using Decision Trees. K-Nearest Neighbors (KNN) and Decision Trees are widely used algorithms in classification tasks. These two models were compared in their performance and accuracy on two datasets, namely the Hepatitis Dataset and Diabetic Retinopathy Debrecen Dataset. We trained these two models on these datasets, using a random selection of training data and testing data from the datasets. For the Hepatitis Dataset, we replaced missing data with the median[4] of each feature instead of dropping corresponding rows. For the Diabetic Retinopathy Dataset, we found that it is hard to make predictions since the correlations for each feature with the label are quite low and many of the features are highly correlated. For the feature selection step, after reading this paper[3], we chose features for both datasets through a correlation matrix. Under selecting the same features, the Decision Tree model was found to be performing slightly better than the K-Nearest Neighbors (KNN) model for both datasets. We designed many experiments about feature selection and found that it is not true that more features result in higher accuracy, which is counterintuitive. However, selecting two features that are highly correlated with classes is enough to improve the accuracy of the constructed model.

2 Datasets

2.1 Summary

For this project, there are two datasets which are the Hepatitis dataset[1] and the Diabetic Retinopathy dataset[2]. They are given from the UCI machine learning repository. The Hepatitis dataset is a dataset of 155 hepatitis patients' personal information and a list of their symptoms and the label of this dataset is whether these patients died or not. The Diabetic Retinopathy dataset contains features extracted from the Messidor image set and is used to predict whether an image contains signs of diabetic retinopathy or not. There are 3100 samples with 167 missing samples in the Hepatitis dataset and 23020 samples in the Diabetic Retinopathy dataset.

2.2 Hepatitis Dataset

The shape of this dataset is 155 rows $\times$ 20 columns but there are 167 missing samples. The majority of missing data is contributed by protime feature, and the percentage of missing values for this feature is about 43.2%, resulting in a protime that is not a proper feature to select for our learning model. In that case, we filled in the missing values with the median of each feature to avoid losing information on the data. The summary of the label class as shown in Figure 1 illustrates that 79% of patients were alive and 21% of those were dead

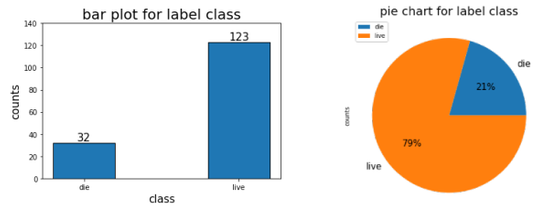

Figure 1: Data summary of label class for Hepatitis dataset

in this dataset. There are 19 attributes consisting of 6 continuous numerical features and 13 binary categorical features. From the description of every feature in the data frame, the average age of patients in this dataset is about 41.2 and there are 139 male patients and only 16 female ones. The numerical features distribution is shown in Figure 2.

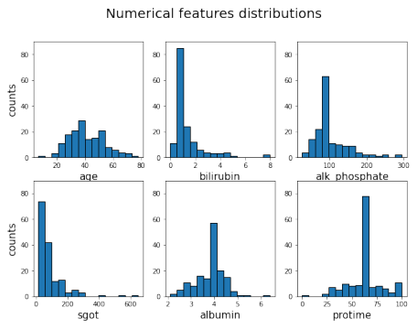

Figure 2: Features distributions for Hepatitis dataset

Attributes	Correlation
class	1.000000
ascites	0.469334
albumin	0.455927
bilirubin	0.424523
spiders	0.389137
varices	0.362385

Table 1: Top 5 Correlation features with class for Hepatitis Dataset

In order to select the best features for our machine learning model, we made a correlation matrix between each feature and label, and the top 5 correlation values with class and features were shown in Table 1. From the table, we chose the best two features(ascites and albumin) which have a high correlation with our target class.

2.3 Diabetic Retinopathy Debrecen dataset

The shape of this dataset is 1151 rows $\times$ 20 columns. There are 19 features containing 3 categorical features, 16 numerical features, and 1 label class. From Figure 3 we concluded that there are 611 people in this dataset who had signs of Diabetic Retinopathy while 540 of them showed no signs. After searching for some materials and reading some data analysis[5] on this dataset, we found that attributes 2-7 have very similar distributions and attributes 8-15 have many outliers in those distributions. Due to the lack of names for attributes in the raw dataset, we assigned each feature with proper names. Attribute names from *ma_level_a* to *ma_level_f* represent the number of MAs found and from *exudate_level_a* to *exudate_level_h* represent the sets of points for *exudates* which contain the same information as *ma_level_a* to *ma_level_f*. For the feature selection part, we found out correlation values

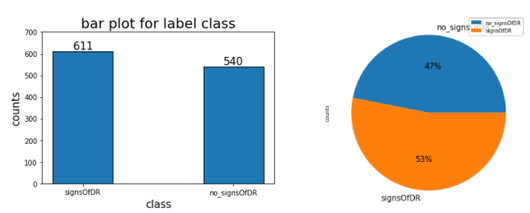

Figure 3: Data summary of label class for Diabetic Retinopathy dataset

between each feature and the label. Table 2 shows that feature *ma_level_a* has the highest correlation value which is about 0.2926 and *ma_level_b* has the second-highest one that is 0.2663. However, if we look at the correlation heatmap in Figure 4, we find that *ma_level_a* and *ma_level_b* show high levels of correlation with correlation 1 which means these two features contain the same information. Therefore, through the correlation heatmap, we selected *ma_level_a* and *exudate_level_g* to be the features for our models.

Attributes	Correlation
class	1.000000
ma_level_a	0.292603
ma_level_b	0.266338
ma_level_c	0.234691
ma_level_d	0.197511
exudate_level_g	0.184772

Table 2: Top 5 Correlation features with class for Diabetic Retinopathy dataset

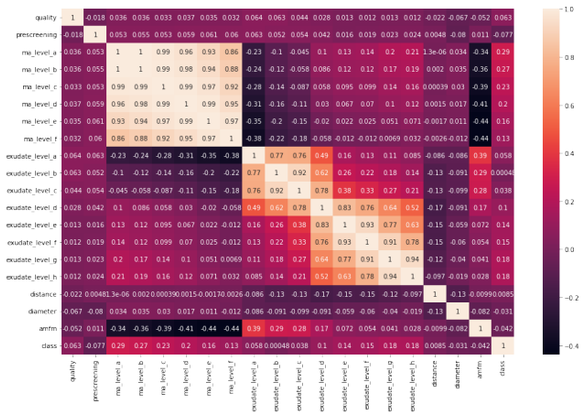

Figure 4: Correlation heatmap for Diabetic Retinopathy dataset

3 Results

3.1 Summary

We tested for different values of the hyperparameters, the change of accuracy is worth investigating, in order to find the best choice of the value of the hyperparameters, we designed an experiment to see the difference in the accuracy of training data and testing data. We also selected different numbers of features, to see which selection provides us with the highest accuracy. Impact for different distance/cost functions for both models were also tested. We use K-fold cross-validation to test which model is better for these datasets respectively. Finally, we analyzed the decision boundary plots for each model with different hyperparameters values.

3.2 K-Nearest Neighbors Result

Firstly, we implemented the KNN model based on the code provided in tutorials. There are two distance functions to be tested: Euclidean distance and Manhattan distance. Our implementation has time complexity $O(DN^2)$, where N is the number of samples and D is the number of features (i.e. dimensions). We can also use KD-Tree to find the K-Nearest Neighbors, but due to our limitation and the size of datasets, we choose to use a relatively unsophisticated method. We then ran our model on the Hepatitis Dataset, setting the test data size to be 0.3 of the whole dataset. To fix the feature selection, we select "ascites" and "albumin" as our two features. For $k \in \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$, we ran a test for each k to test the accuracy of training data and test data for both KNN models using Euclidean distance and Manhattan distance for the same dataset. (See Figure 5, 6)

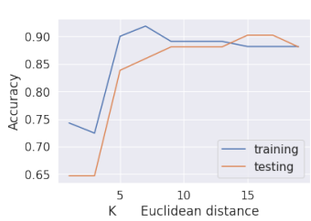

Figure 5: Accuracy for different k's using Euclidean distance for Hepatitis Dataset

Figure 6: Accuracy for different k's using Manhattan distance for Hepatitis Dataset

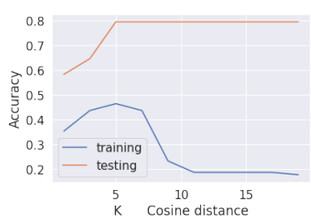

Figure 7: Accuracy for different k's using Cosine distance for Hepatitis Dataset

From the figures above we can clearly see that the plot is very similar for both figures, which means that the choice of these two distance functions makes a very small difference for this dataset, this is because one is the square of another, but Euclidean distance does provide more changes in the curve when k changes. So we test another distance function, cosine distance (Figure 9), we can see that the accuracy decreased a lot as k increased from 5 for the training data. Compared with the other two distance functions, it performs worse in

terms of accuracy, moreover, we noticed that using cosine distance requires more running time, and we think this is because the computation is more complicated. Starting from $k = 3$(since $k = 1$ is overfitting), as k increases, the accuracy is increasing for the test data, the increase is about 3% from $k = 3to9$, but the accuracy train data slightly decreased by about 5%. So for this model, our best choice for k should be around the intersection, which can be $7, 9$. The figures with $k \in \{x|x = 2m + 1, 0 \leq m \leq 29\}$, features selected are "ma_level_a" and "exudate_level_g ", can also be drawn for Diabetic Retinopathy Debrecen dataset(Figure 8, 9, 10). We can learn from the figures that for this particular dataset, the trend is similar to the ones above, we can see that for the model that used cosine distance, the accuracy varied a lot as k changes and compared with the other two distance functions. And the accuracy is overall low, since features and class are poorly correlated. By choosing $k = 11$, we can achieve a relatively high accuracy for both distance functions. However, it is a larger dataset, so actually, we have more cases to learn from, which is closer to the real-world machine learning situation.

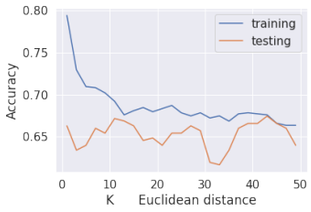

Figure 8:
Accuracy for different k's using Euclidean distance for Diabetic Retinopathy Debrecen dataset

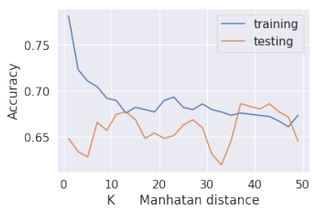

Figure 9:
Accuracy for different k's using Manhattan distance for Diabetic Retinopathy Debrecen dataset

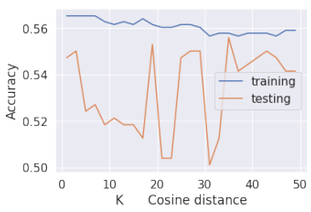

Figure 10:
Accuracy for different k's using Cosine distance for Diabetic Retinopathy Debrecen dataset

Then we can draw the figures of Decision Boundary using KNN model for each dataset, we set the distance function to be Euclidean distance, and $K = 7, 11$ respectively. (Figure 11, 12) For Hepatitis Dataset, we can conclude that in general, most of the area is green,

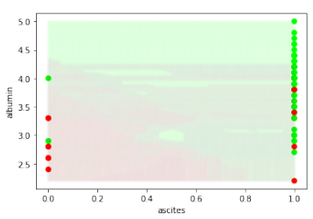

Figure 11:
Decision Boundary for Hepatitis Dataset, k = 7

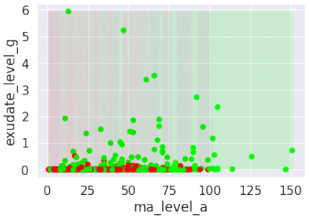

Figure 12:
Decision Boundary for Diabetic Retinopathy Debrecen dataset, k = 11

which means the model is more likely to predict "1", in particular, on the right and top, it is more likely to predict the "1" class. The cases lie in triangle starts from origin has a higher change to be classified as "0". For the Retinopathy Debrecen dataset, we can conclude that in general, green is still more likely to be as a classified result, and at many places there are overlaps, which shows that in these areas it's more likely to be misclassified, in particular, when ma_level_a is large, it's more likely to be classified as "1" class, the cases with lower ma_level_a value have a higher chance to be classified as "0".

3.3 Decision Tree Result

For the Decision Tree Algorithm, we select "ascites" and "albumin" for Hepatitis Dataset, "ma_level_a", and "exudate_level_g" for Diabetic Retinopathy Debrecen dataset. We implemented the model by following the notes via the course website. Instead of applying only a single type of the cost functions, in our experiments, we obtained the results from running multiple controlled groups. For instance, in order to present a better demonstration of the impact of different cost functions on the same model, we decided to keep the datasets(both training and testing) and the hyperparameter which is the max depth same through every distinct trial. After running the experiment multiple times based on a random split of the datasets, we discovered that the distribution varies in different trials. In most cases, the resulting test accuracy is obtained by setting Gini-index and entropy cost as the cost

function follows the same pattern. However, the test accuracy of training a model using misclassification cost as the cost function is usually different from the above two. Further, the second controlled experiment was designed for testing how different max depths can influence both the training and testing accuracy. Due to the limited amount of the data and running time, we restricted the max depth to at most 10 and the results are demonstrated in the following figures for Hepatitis and Diabetic Retinopathy Debrecen datasets(from figure 13 to 18). After running multiple times of the experiment, it is obvious that in most cases,

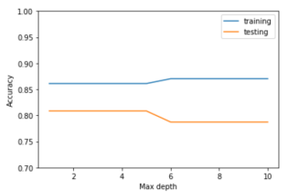

Figure 13:
Train/Test Accuracy using entropy as cost function for Hepatitis dataset

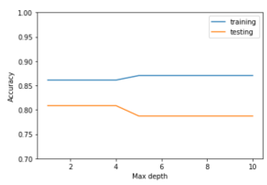

Figure 14:
Train/Test Accuracy using gini index as cost function for Hepatitis dataset

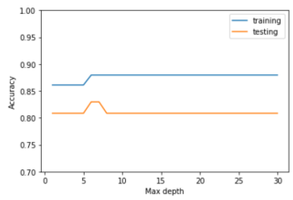

Figure 15:
Train/Test Accuracy using Missclassification Cost as cost function for Hepatitis dataset

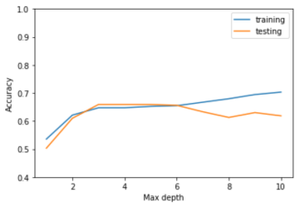

Figure 16:
Train/Test Accuracy using entropy as cost function for Diabetic Retinopathy Debrecen dataset

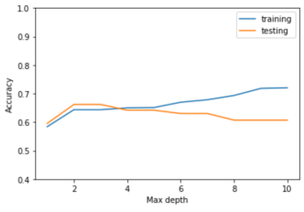

Figure 17:
Train/Test Accuracy using gini index as cost function for Diabetic Retinopathy Debrecen dataset

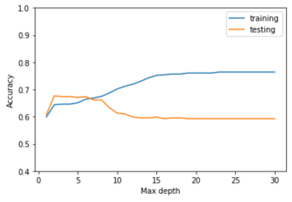

Figure 18:
Train/Test Accuracy using Missclassification Cost as cost function for Diabetic Retinopathy Debrecen dataset

the training accuracy is higher than the testing accuracy as we increase the max depth of the tree. Additionally, by observing and comparing the results of different trials, we found that unlike the results from the same experiment in the KNN algorithm, no matter how we randomly split the datasets, the graph shows that the training accuracy is always increasing or maintaining the same value. This significant observation leads to the fact that the training accuracy by using the Decision Tree Algorithm can not decrease as we increase the value of the max depth. Nevertheless, in some line graphs, we can see that the testing accuracy might decrease as we continue to increase the value of max depth. This occurs due to the overfitting of the training data. Apart from this, we also integrated the observations from both experiment1 and experiment2. Back to the six figures we have obtained by running the experiment2 on two datasets, it can be seen from figure 13 and 14, figure 16 and 17, that the pattern of using Gini-index and entropy cost as the cost function presents a huge similarity. This satisfied the interpretation we made in the previous experiment.

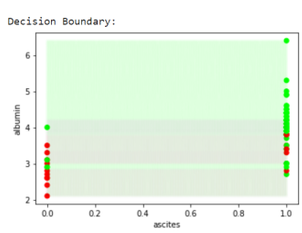

Figure 19:
Decision Tree Decision Boundary for Hepatitis Dataset

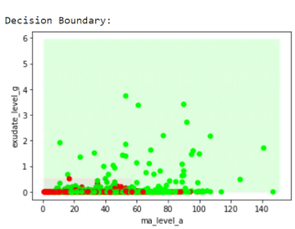

Figure 20:
Decision Tree Decision Boundary for Diabetic Retinopathy Debrecen dataset

The above two figures are the decision boundary graphs obtained by running Decision Tree Algorithm about these two features respectively on each dataset. For drawing the decision boundary and plotting graph, we implemented our own model of cross-validation in order to find the best max depth of the corresponding tree and demonstrate the decision boundary graph. We considered entropy cost as the cost function. It is apparent that the decision boundary graph of the Decision Tree has a significant difference compared to the decision boundary graph of KNN. Unlike the Voronoi Diagram visualization of KNN, the decision boundary of the Decision Tree looks more linear and clean.

3.4 Comparision

Using K-Nearest Neighbors (KNN), We achieved 89.8% accuracy for the Hepatitis Dataset and 69.1% accuracy for the Diabetic Retinopathy Debrecen dataset. Using Decision Tree, We achieved 93% accuracy for the Hepatitis Dataset and 71.8% accuracy for the Diabetic Retinopathy Dataset. (See Table 3)

	Hepatitis Dataset	Diabetic Retinopathy
K-Nearest Neighbors	89.8%	69.1%
Decision Tree	93%	71.8%

Table 3: Two models' best accuracy on different datasets

4 Discussion and Conclusion

From this project, we learned how to apply both K-Nearest Neighbors (KNN) and Decision Trees algorithms into real-world problems to predict relative targets, and we compared these two models through many experiments. There are several differences in predictive accuracy, time complexity, and performance on various data between these two models. Therefore, choosing an appropriate model for the classification problem is significant. To improve the performance of our learning models, we would like to investigate some methods like pruning to solve overfitting problems. Moreover, feature selection is an important part of machine learning and we only used the correlation method to select features in this project. However, there are many ways to select the most important features such as wrapper methods and embedded methods and so on which can be learned and used to develop a better performance for our models.

5 Statement of Contributions

Chuqi Wang: Dataset loading and preprocessing, validation and testing, write-up contribution. **Jiechen Zhang**: Dataset preprocessing, implementation of KNN algorithms, model training, validation and testing, write-up contribution. **Yubai Zhang**: Implementation of Decision Tree algorithms, model training, validation and testing, write-up contribution.

References

[1] Hepatitis. UCI Machine Learning Repository, 1988.

[2] Diabetic Retinopathy Debrecen Data Set. UCI Machine Learning Repository, 2014.

[3] Michal Haindl, Petr Somol, Dimitrios Ververidis, and Constantine Kotropoulos. Feature selection based on mutual correlation. In *Iberoamerican Congress on Pattern Recognition*, pages 569–577. Springer, 2006.

[4] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. 2007.

[5] TG Tiago. Machine learning on the diabetic retinopathy debrecen dataset. *knowledge-Based System60*, pages 20–27, 2016.