COMP 551 Mini Project 1:

KNN model and Decision Tree

Firzana Sadik (260915094)
Neshma Metri (260847343)
Xinxin Lu (260772336)

Abstract

In this project, we implemented and explored the performance of K-Nearest Neighbor (KNN) model and Decision Tree model in on two distinct health datasets Diabetic Retinopathy Debrecen (DRD) dataset (Antal and András 20-27) and Hepatitis dataset (Dua and Graff). We preprocessed the data and tested the performance our two models - KNN and Decision tree - on each dataset. We tested the performance of both models in different conditions and make a comparison. We tested the accuracy of models by training with different parameters and using different cost functions (distance functions). Also, we measure the model performance in different scaling, and impact of noisy features on both models. Besides, we compared the model performance with stratified data and regular data. We found that both models have better accuracy with stratified data. Scaling will not affect Decision Tree performance and noisy features don't affect the model performance too much compared to KNN. KNN is more sensitive to scaling and noisy features. Normalizing data and removing noisy features will improve the KNN performance.

Introduction

Machine learning can be widely used in medical system to predict and test disease. In our project, we implemented Decision trees and KNN model and trained on the DRD dataset and hepatitis dataset. The former dataset has data describing Retinal images to determine DR in patients; while the latter has the data collected from ~150 patience diagnosed with acute or chronic Hepatitis.

KNN model is a simple approach in machine learning which make prediction by finding the similar instances in stored training dataset. Finding the similar instances can be applied with different similarity measure approach such as Euclidean distance function, Manhattan distance and Minkowski distance function. KNN has a parameter as K which indicate the number of nearest neighbours the model needs to look at for comparison.

Decision Tree model uses a tree like structure to decide the class of a given datapoint with given features . Decision Tree model has multiple cost functions which are used to split a node into one class or the other for a binary class mode. The cost functions we saw in class are Misclassification, Gini and entropy functions. Decision Tree is not sensitive to the scaling, so normalization the data will not lead to any improvement. Removing noisy features can improve the Decision tree, but it's not change as many as it in KNN. Besides, stratifying data can improve the performance of Decision Tree.

Datasets:

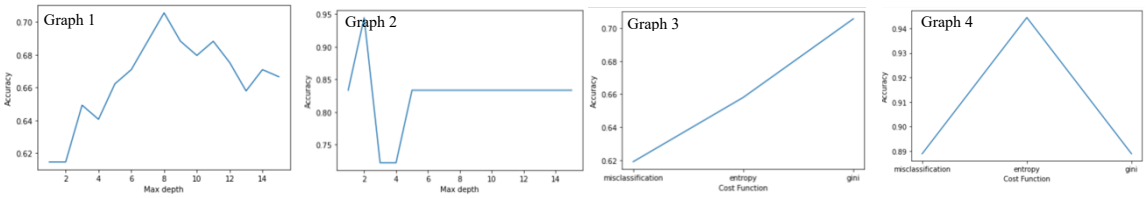
The Hepatitis dataset contains information of various symptoms faced by a total of 155 patients diagnosed by Hepatitis out of which 123 lived and 32 died. On the other hand, DRD dataset contains data of 1151 images from Messidor image set, labelled as with or without signs of DR (total 611 and 540 respectively) based on multiple descriptive and distance values. Each of the dataset contains 19 features.

The missing values from Hepatitis dataset marked as '?' were removed after removing any given noisy features. Noisy features were chosen based on the heatmap that showed the features that were correlated most with each other (Patil, 2018) and Boxplot of each feature with least variance. We soon realised that the correlation had more impact than the variance and hence we stuck to correlation. We removed Bilirubin, Fatigue, Protime, Liver Firm, Anorexia, Age and Histology from Hepatitis Dataset. Additionally, we removed Image quality, Pre-screening, a_0.9 and attr8 through attr15 from DRD Dataset.

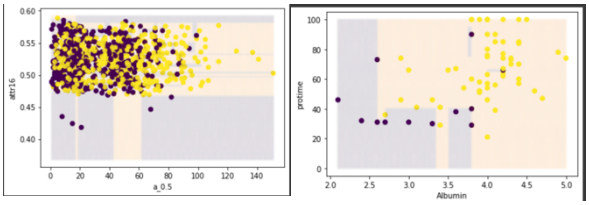
A major ethical concern is the possibility of an individual being identified whose data is recorded in the Hepatitis data because, unlike the DRD dataset, this dataset contains age and sex information of participants as well. However, this might not be as much of a concern if the patients name and other private information is anonymized when inputting the data itself.

Results

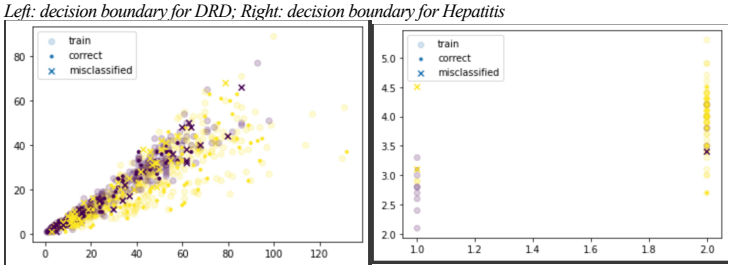
Decision Tree Model



The hyperparameter of interest with Decision Trees is the maximum depth of which the tree expands to. As part of the experiment, we tuned this parameter by finding the best accuracy resulting from K-fold cross validation. Specifically, we chose to do a 4-fold on DRD dataset, leaving us with 70% of the data to be validation and 5-fold for Hepatitis, which gives us 85% of validation set. Note that the K chosen for the folds were entirely based on the resulting accuracies. After, we decided upon the best cost function out of misclassification, Gini and entropy again based on the best performance. From Graphs 1 and 2, having tuned the parameters, we were able to find the optimal accuracy to be 70.6% with *max_depth* of 8 and 88% *max_depth* of 1 for DRD and Hepatitis dataset respectively. This was achieved with normalization, stratification, and the removal of noisy features. Graphs 3 and 4 shows the difference in accuracies for both DRD and Hepatitis data from using different cost functions. We can conclude that misclassification performs the worst compared to all the other cost functions.



Left: decision boundary for DRD; Right: decision boundary for Hepatitis

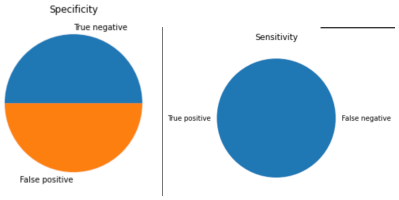


Left: misclassification plot for DRD; right: misclassification plot for Hepatitis

```
accuracy = 0.7056277056277056
precision = 0.7722772277227723
recall = 0.6341463414634146
[[ 78  23 101]
 [ 45  85 130]
 [123 108   0]]
```

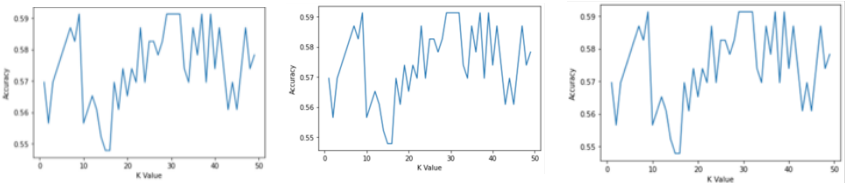
Confusion matrix for DRD; Specificity pie chart; Sensitivity pie chart

```
accuracy = 0.9444444444444444
precision = 0.9411764705882353
recall = 1.0
[[16  1 17]
 [ 0  1  1]
 [16  2  0]]
```

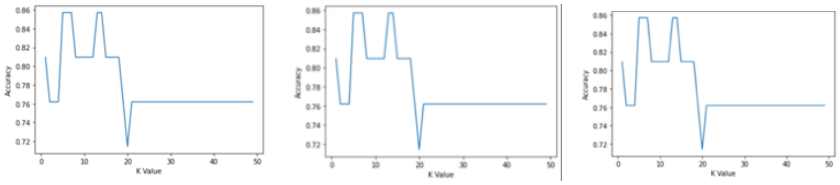


Confusion matrix for Hepatitis; Specificity pie chart; Sensitivity pie chart

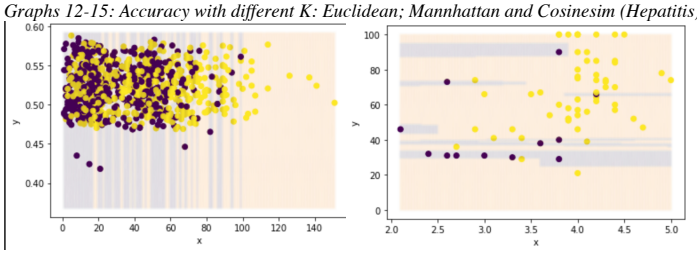
KNN Model



Graphs 9-11: Accuracy with different K: Euclidean; Mannhattan and Cosinesim (DRD)



Graphs 12-15: Accuracy with different K: Euclidean; Mannhattan and Cosinesim (Hepatitis)



Decision boundary plot: (Left) DRD: x is a_0.5, y is attr16; (Right)Hepatitis: x is albumin, y is protime

The KNN model also has a hyper parameter of which we tune to find the best accuracy. The "K" represents the number of nearest neighbors we evaluate to classify the test instance instead of simply the single most close training instance. The best *k* was found between a range of 1 and 20 and chosen from the accuracies resulting from the validation set. This particular *K* was then applied to the test set for the final accuracy. Graphs 9 to 11 are the results of using different *K* values for the validation set for each distance function. Overall, the cosine similarity function performed worst throughout all experiments and dataset. For DRD dataset, the best accuracy achieved was 65.4% using K=9. This was with the Manhattan distance function as well as having normalization, non-stratification, and removal of noisy features. For the Hepatitis dataset, we achieved a high 86.4% with K=5 for both Euclidian and Manhattan distance functions, and again including normalization, stratification and removal of noisy features.

Table 1. Design Choices for experiments

Design Choice	Description	Hypothesis
Normalization	The normalization used is the Min-Max scalar. This is to ensure that there is a common scale amongst the features to prevent any "weighing" on some features more than others. (2)	DT: It shouldn't be affected because it is robust to scaling of features KNN: Sensitive to feature scaling
Stratification	This is to ensure that when dividing a dataset into smaller datasets, the distribution of classes in each smaller dataset is same as the distribution of classes in the main dataset from which the smaller datasets are derived.	We expect both models to perform better since they can train and validate on equal distribution of classes and thus better predict the class for unseen data, which also have similar distribution.
Noisy Feature Removal	In our experiment, noisy features are any features that provide no "extra" information (features which correlate with each other and with the class) as well as features that decreases performance during the fine-tuning stages.	DT: It shouldn't be affected because it is robust to noisy features KNN: Sensitive to noise therefore accuracy should improve with removal.

Table 2. Accuracy Results between KNN and Decision Trees

Data		DRD Data		Hepatitis Data	
Model		**Decision Tree**	**KNN**	**Decision Tree**	**KNN**
Normalization	Y	70.56%	59.3%	94.44%	86.4%
Stratification	Y				
Noisy Features Removed	Y				
Normalization	Y	67.97%	63.6%	64.71%	69.6%
Stratification	N				
Noisy Features Removed	Y				
Normalization	Y	63.20%	49.8%	70.00%	62.5%
Stratification	N				
Noisy Features Removes	N				
Normalization	N	63.20%	63.2%	70.00%	62.5%
Stratification	N				
Noisy Features Removed	N				

Comparison between Decision Tree and KNN

Using Table 2, we will discuss how KNN and Decision Tree models vary beyond their differences in training and testing the data. The three key design choices we made on each model are normalization, stratification, and removal of noisy features. From the Table 1 our hypothesis on stratification was indeed correct in that overall, both models did perform better (except from KNN on DRD dataset). This was an interesting find considering that DRD dataset has more data points than Hepatitis. With normalization, we see that Decision Trees was not affected as much as KNN, as shown from the margins of only 7% but more than 10% for KNN. Lastly, with noisy features removed, KNN did perform better. And although this was excepted for, it was mostly surprising to see Decision Trees negatively affected with the removal of noisy features. For example, for hepatitis data, decision trees resulted with a 94% accuracy, but a low 70% accuracy without noisy features. Since keeping features can result in dropping more instances (as they have no values), the lack of training data may have influenced the model's ability to do better splits for the tree.

Discussion and Conclusion

A big, yet not surprising, take away for the KNN was that when we removed the noisy features from the dataset in turn reducing the feature space, our accuracy for KNN went from ~60% to ~85%
As for the decision tree, we saw that the model performance can decrease as the max depth gets too large or too small. Additionally, Decision Tree model had better accuracy with Gini and entropy functions compared to Misclassification. Scaling did not affect Decision Tree performance and noisy features don't affect the model performance too much compared to KNN.

The decision tree model is sensitive to data changes, and we saw this in the experiments we conducted. When using random data with no seed we got multiple 65% accuracies in row along with a 90% accuracy for the same model. This sensitivity to data changes can be avoided by further modifying using Bagging and Boosting methods as mentioned in the Kotsiantis' paper, "Decision trees: a recent overview."

Statement of Contributions

Neshma worked on the data processing and analyzing while Firzana and Xinxin oversaw implementing KNN and Decision Tree respectively. The experiments mentioned in the document, or otherwise, were suggested by all teammates and predominantly run or discussed during our regular meetings. All the sections of the report were divided amongst the teammates and were read and edited by the others.

References

Antal, Bálint, and András Hajdu. "An ensemble-based system for automatic screening of diabetic retinopathy." *Knowledge-based systems* 60 (2014): 20-27.

Dua, Dheeru, and Casey Graff. "UCI Machine Learning Repository: Hepatitis Data Set." *Archive.ics.uci.edu*. N.p., 2019. Web. 9 Feb. 2022.

Patil, Prasad. "What is exploratory data analysis." *Toward Data Science* (2018).