ABSTRACT

In this assignment, we performed KNN and decision tree classification for the hepatitis and the Messidor dataset. We used cross-validation to set the hyperparameters in each model and reported the evaluation measures (accuracy, F-score, etc.) on the test (unseen) data. In the case of the hepatitis dataset, when we investigated the performance of the KNN model and the Decision Tree model, we got the accuracy of 0.93% and 0.966%, respectively. However, we should consider that in imbalanced datasets, accuracy cannot be a perfect measure to evaluate the model. Hence, other measures such as F-score are also reported. Moreover, in the case of the Messidor dataset, the accuracy for the KNN model and the Decision Tree model were 0.685 and 0.68, respectively.

We investigated the effect of increasing K on decision boundaries; we observed that small values of K cause the model to become overfitted to the train data and make the model unable to perform well on the test data. Also, we saw that a large value of K is problematic since the decision boundaries are unreasonably smooth. This magnifies the importance of setting the hyperparameter using cross-validation to find the best model. We also investigated the effect of increasing depth in decision boundaries when implementing a decision tree. In this case, increasing depth could make the model follow the pattern within the train data and lose its generalization. However, decreasing depth could lead to more robust and smooth boundaries.

INTRODUCTION

Correct diagnosis of a disease is one of the most important problems in medicine, and a lot of researchers have recently used computational intelligence in diagnosing different diseases (Neshat, et al., 2012). Here, we plan to work with two datasets, the hepatitis dataset and the Diabetic retinopathy dataset. The main goal of this study is to see if we can classify the patients based on the measurements and status of different features. In the case of the hepatitis dataset, the goal is to determine whether patients with hepatitis will either live or die. In the case of the Diabetic retinopathy dataset the goal is to explore if there is existence of diabetic retinopathy. In this assignment, we use KNN and decision tree algorithm to do the classification procedure. In case of the first dataset, for KNN and decision tree we got the accuracy of 0.93% and 0.966%, respectively. However, the number of patients per group (die or live) was not balanced, leading to a misleading interpretation of results. To overcome this problem, Safdari (2022) suggested applying Symmetric Minority Over-Sampling Technique prior to model building to solve the imbalance problem by generating artificial samples for the less pronounced group.

For the Diabetic retinopathy dataset, the accuracy for KNN and decision tree classifiers was 0.685 and 0.68, respectively. To further improve the results, other methods of artificial intelligence could be used (Neshat et al.,2012; Alyoubi, 2021). Furthermore, applying dimension reduction using Principal Component Analysis on the features and doing the classification based on the principal components can turn up accuracy up to 80% (Rathi 2021).

DATASETS

The hepatitis dataset used in this work is obtained online from the "UCI machine learning repository." Based on the dataset, the Hepatitis patients' results are classified into either "Live" or "Die". The dataset consists of 14 nominal features, six categorical features, and a class label attribute. The empirical distribution of each attribute per class is illustrated in supplementary figure 1. The most significant observation, in this case, is that there is a total of 155 instances in this dataset, out of which 123 belong to the class "Live", and the remaining 32 belong to the class "Die". We infer that there is a data imbalance that can adversely affect the classification procedure.

The second dataset used in this work is “Diabetic Retinopathy Debrecen Data, " obtained online from the Messidor program partners. This dataset consists of 1151 instances, and the features are obtained from the Messidor Image dataset to predict whether an image has potential signs of Diabetic Retinopathy (DR). The patients' results based on the datasets are classified into two classes, either "1" or "0". The dataset consists of 16 nominal features, three multi-valued features, and a class label attribute. The empirical distribution of each attribute per class is illustrated in supplementary figure 2. It is noteworthy to mention that there are 611 subjects with signs of DR and 540 with no signs in this dataset.

It is noteworthy to mention that although we are using these datasets, we should care about the **ethical concerns** that might exist. One of the main issues here is that even if participants are aware that their information is considered publicly available and can be used in research, they might be **unaware of the type of research** and findings that can result from their information. And the other thing to mention is that we are not sure if in acquiring the **data equality** has been achieved or not; we need to make sure there are no overrepresented populations if we want to introduce the best classifier at the end – this might not work for the other unseen populations. (Howe et al. 2021).

PREPROCESSING

Usually, the data collected from patient's records are not completely clear. Therefore, data cleaning is essential for developing machine learning models (Willemink MJ, 2020). By far, the most common means of dealing with missing data is listwise deletion; in the case of the hepatitis dataset, if we remove the missing data blindly, we will lose about 50% of the data that we have, which can cause severe data insufficiency - if we remove the subjects with incomplete data the number of instances will shrink to 80 subjects (13 die/67 live). Another approach to handle missing data is to delete a particular feature if it has more than a specific number of missing values (Dong, Yiran, 2013). In this case, the feature called "potime" has the most significant number of missing data (67 missing data), but the information revealed by the normalized histograms (Figure 1) exhibits the importance of this feature in classification as the histogram of instances in die, and live groups seem to have a clear distinction. Hence, rather than removing attributes or observations with missing data, mean/median imputation is chosen in this work to deal with missing values. However, the fact that this approach underestimates the standard deviation should not be neglected; despite this fact, Safdari et al. (2022) reported that the mean imputation still works well in the Hepatitis dataset. Although we question their approach since in histograms drawn in figure 1, you can see a skewness in the distribution of some attributes, which suggests median imputation can be a more statistically meaningful approach to deal with missing data rather than mean imputation.

On the other hand, as we already know about the class/label of each subject, we can use this fact as prior knowledge and improve our estimations for the missing values. Consequently, for a numerical feature, we calculate the median for the class label "die" and "live" individually of non-missing values to replace missing values based on the class labels methodology called similar case imputation. It is noteworthy to mention that the mode of feature used to do similar class imputation for categorical attributes.

The other step toward data processing is the conversion of categorical attributes into numerical attributes, and the further action is data normalization. The KNN algorithm should not be biased towards features with higher magnitude. To overcome this problem, one option is to do Standard-Scaling; however, using this method of normalization has an assumption of normality which is often ignored by the publishers (just like in the case of Safdari et al. (2022)); we tried to check this assumption in our numerical attributes, using QQ-plots (Figure 4). We can clearly see that the empirical distribution of some attributes deviates from the normal distribution. So, we highly suggest that other normalization methods, such as same-range normalization, would be applied to the attributes of this dataset.

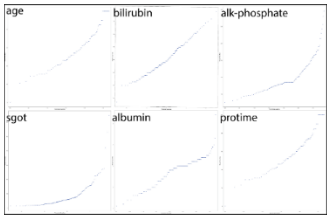


Figure 1 QQ plots

$$xi_{normalized} = \frac{\text{xi} - \min(\text{x})}{\max(\text{x}) - \min(\text{x})} * 100$$

Where xi is the original feature value, and $xi\_normalized$ is the normalized value of the attribute.

In the second dataset, no missing data is found. The first column (called: '0') within the dataset contains the quality assessment result, which describes the quality of a color fundus image. We have four subjects with low-quality data; we exclude the data of those four patients from our further analysis (1147 subjects remain in the study) and also remove this feature vector. No normalization was applied to this dataset when we built KNN or decision tree models.

METHODS AND RESULTS

THE EVALUATION PARAMETERS

One of the most straightforward measures to report the performance of a classifier is accuracy. However, accuracy might not be an informative measure when the data is imbalanced. For instance, in the hepatitis dataset which contains 80% of instances in the "live" class and 20% in the "die" class; a naïve approach of classifying every instance to be a majority class would give a high predictive accuracy of 80% (Chawla, N.V., 2009); however, this description fails to reflect the fact that 0 percent of minority examples are identified.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

$$Error(or\ Cost) = 1 - Accuracy$$

In place of accuracy, other evaluation metrics are frequently adopted in the research community to provide comprehensive assessments of imbalanced learning problems such as F-score. The other measures used in this assignment are defined as follows:

$$Precision = \frac{TP}{TP+FP}; \quad Sensitivity = \frac{TP}{TP+FN}$$

$$Recall = \frac{TP}{TP+FN}; \quad F1 = 2\ \frac{Recall*Precision}{Recall+Precision}$$

Intuitively, precision measures exactness. In contrast, recall is a measure of completeness. The F-Measure metric combines precision and recall measuring classification effectiveness in terms of a ratio of the weighted importance on either recall or precision. As a result, F-Measure provides more insight into the functionality of a classifier than the accuracy metric (H. He, 2009).

KNN

KNN is the simplest classification method applied in machine learning domains. With the KNN method, the test data were labeled based on their similarities. The K-value is the main factor in the KNN method. In this lazy algorithm, the chosen number of K-values indicates the number of neighbors that should be selected.

KNN – HPEATITIS DATASET

In the hepatitis dataset, however, imbalanced class sizes (low occurrence of "die" class label) can lead to practical problems with KNN. Classification of imbalanced data is biased towards the large categories. One way to compensate for this unbiased manner is using Symmetric Minority Over-Sampling Technique (SMOTE). This method uses the KNN algorithm in creating new synthetic samples to balance the class distribution of the dataset (Safdari et al., 2022 and Chawla et al., 2002). However, applying this algorithm is out of the scope of this assignment.

MODEL TRAINING AND TESTING

In the first step, we put 20% data (30 samples) aside as unseen and then built the model based on the other 125 samples. We created several models, each using a different number of neighbors to make classification (we changed k from 1 to 50). The train set was used to build the model, and then the test data was used to evaluate the model's performance; in the following figure, you can see the accuracy of the KNN model for both test and train data.

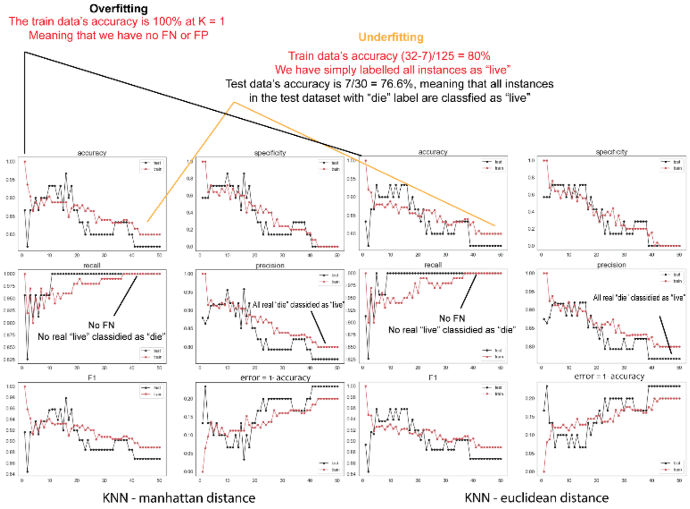


Figure 2 Model performance on the train and the test set – Hepatitis dataset

One can see that when we use the train data to evaluate the model, the model shows a better performance than when we introduce test data to the model. The reason for this observation is that the train data was previously used to build the model itself, so we expect to see a high accuracy when we introduce the same data and ask the model to classify it. On the other hand, when it comes to test (unseen) data, as this part of data has not been seen in the model, we cannot be that accurate.

CROSS-VALIDATION

To fine-tune the hyperparameter and increase the generalizability of the KNN tree, we splitted our train set (125 samples) into train set and validation set (100 and 25 samples respectively) and used 5-fold cross-validation. The figure below shows mean classification error and variance with the K ranging from 1 to 20 using two different distance measures. For each distance measure, we chose the best K that minimizes the validation error; after finding the best hyperparameter, we used all the 125 instances to train a KNN model with that specific value of K and tested the model using the 30 instances of unseen data. The confusion matrix of the result is shown in the table. We also got performance metrics including accuracy, specificity, recall, precision, F1 score, error.

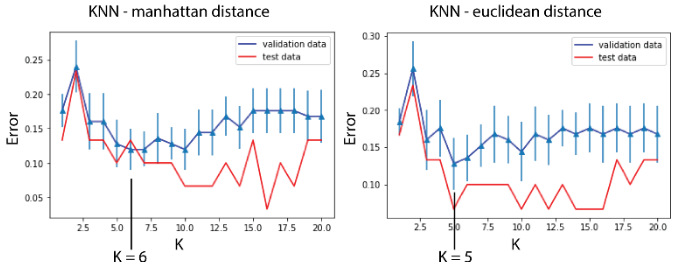
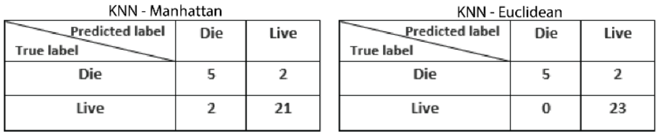


Figure 3 Finding the best hyperparameter for KNN- hepatitis dataset

Results on unseen data:

We can see that in this case, using Euclidean distance to find the nearest neighbors has led to better results:



KNN - Manhattan

Predicted label True label	Die	Live
Die	5	2
Live	2	21

KNN - Euclidean

Predicted label True label	Die	Live
Die	5	2
Live	0	23

Figure 4 Confusion matrix - KNN - hepatitis dataset

Measure Distance	Accuracy	Specificity	Recall	Precision	F1	Error
Euclidean	0.933	0.714	1	0.92	0.958	0.066
Mahattan	0.866	0.714	0.913	0.913	0.913	0.13

As mentioned before, one should take into consideration that because of having biased classes, accuracy cannot be considered as a perfect measure for evaluation.

DECISION BOUNDARY

In other to plot the decision boundaries, we needed to find two features and build the KNN model based on those attributes. We calculated each feature vector's correlation with the class label column to decide which attributes to choose. The two features that had the highest magnitude of correlation were selected as the features needed to build the model; these two features were “Albumin” and “Protime”. Moreover, if one pays attention to the normalized histograms, he can realize that these features are the ones in which the normalized histograms of the two classes are highly discriminated.

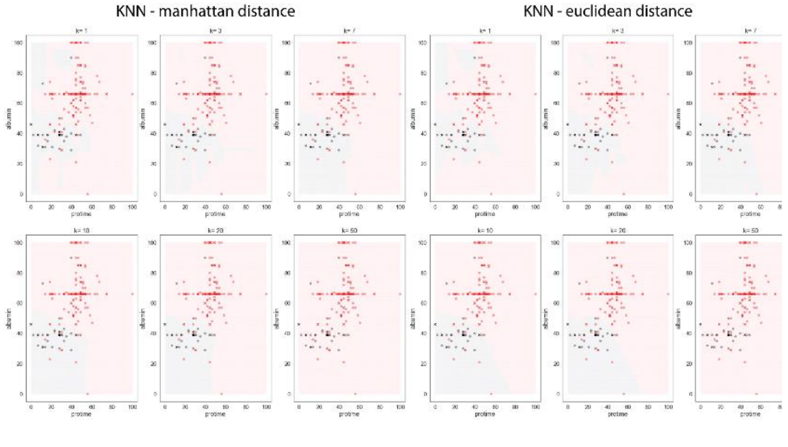


Figure 5 Decision boundary - KNN - Hepatitis dataset

We can see that as we increase K the decision boundaries are becoming smoother. Presumably, when K is equal to 1 the model is just good at predicting the train data itself, while not being good when we introduce new data to it. On the other hand, when we increase K to 50, we can see that the model assumes every introduced instance as “live”, in other words the model has become so simple and cannot catch up with the information revealed by the attributes.

KNN – MESSIDOR DATASET

MODEL TRAINING AND TESTING

In the second dataset, the same procedure has been done, with some minor differences. Here, we have 1147 instances remaining after preprocessing; we put aside 197 instances as test(unseen) data and the other 950 for training; the model was built using the training set. Once the measures were calculated when we introduced the train data to the model, and once they were calculated when the test data was introduced to the model. Then again, we can see that the models have a better performance on the train data as the model has observed the data before. On the other hand, we can see the effect of choosing K on model performance. We can see that at small values of K, the model is highly attached to the train data and cannot have a good generalization, while high values of K, make models be underfitted. The best K is somewhere in the middle that we need to find it using cross-validation.

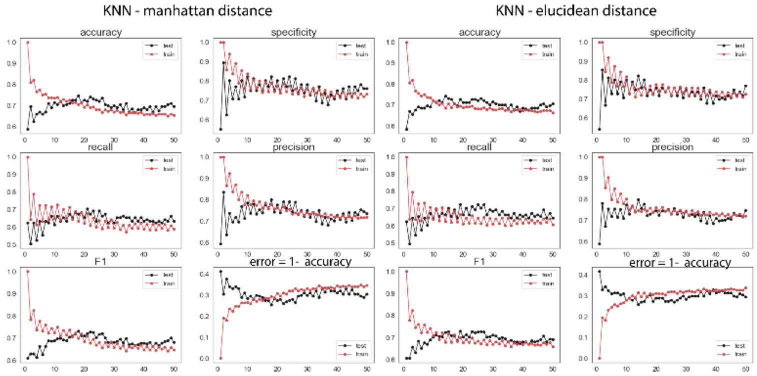


Figure 6 Model performance on the train and the test set – Messidor dataset

CROSS-VALIDATION

We did cross-validation to find the best value for hyperparameter (K). We put aside 197 unseen data, then splitted the remaining 950 instances into 10 folds and did the 10- fold cross-validation to find the best K; the results are as follow:

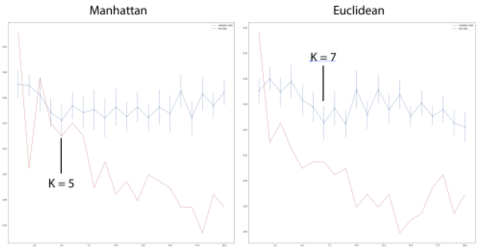
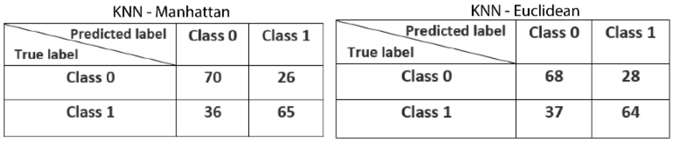


Figure 7 Finding the best hyperparameter for KNN- Messidor dataset

Results on unseen data:



KNN - Manhattan

Predicted label / True label	Class 0	Class 1
Class 0	70	26
Class 1	36	65

KNN - Euclidean

Predicted label / True label	Class 0	Class 1
Class 0	68	28
Class 1	37	64

Figure 8 Confusion matrix - KNN - Messidor dataset

Measure / Distance	Accuracy	Specificity	Recall	Precision	F1	Error
Euclidean	0.685	0.729	0.643	0.714	0.677	0.314
Mahattan	0.67	0.708	0.633	0.695	0.663	0.329

DECISION BOUNDARY

To show the decision boundary of the KNN in the 2D plane, we select the two most important features, the number of MAs detected at confidence levels 0.6 and the number of exudate pixels and refitted the KNN model to out data. For k = 1 the boundary is attached to the train data, and as we increase K the model is decision boundaries are becoming smoother.

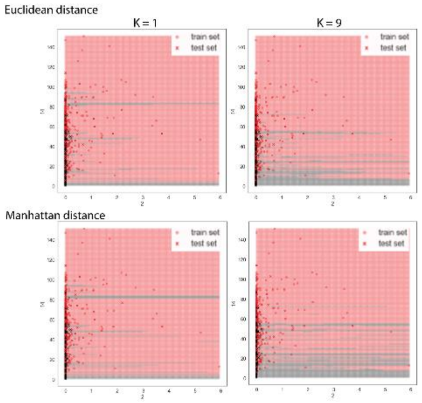


Figure 9 Decision boundary - KNN - Messidor dataset

DECISION TREE

Decision trees use a recursive, top-down greedy search algorithm that uses a feature selection scheme (e.g., information gain) to select the best feature as the split criterion at each node of the tree; a successor (leaf) is then created for each of the possible values corresponding to the split feature. As a result, the training set is successively partitioned into smaller subsets that are ultimately used to form disjoint rules pertaining to class concepts. These rules are finally combined so that the final hypothesis minimizes the total error rate across each class. The problem with this procedure in the presence of imbalanced data is two-fold. First, successive partitioning of the dataspace results in fewer and fewer observations of minority class examples resulting in fewer leaves describing minority concepts and successively weaker confidences estimates. Second, concepts that have dependencies on different feature space conjunctions can go unlearned by the sparseness introduced through partitioning. Here, the first issue correlates with the problems of relative and absolute imbalances, while the second issue best correlates with the between-class imbalance and the problem of high dimensionality. In both cases, the effects of imbalanced data on decision tree classification performance are detrimental. In the following sections, we evaluate the solutions proposed to overcome the impact of imbalanced data.

DECISION TREE – HEPATITIS DATASET

MODEL TRAINING AND TESTING

We first showed the classification accuracy in our train set and test set with the change of the hyperparameter, depth, using three kinds of cost function. It can be seen in the figure below; the train set classification accuracy always keeps increasing with depth and reaches 1. While the test set classification accuracy is always lower than classification accuracy in train set, which shows overfitting. Although a deeper tree can fit the train, set better, but it can also decrease the generalizability to unseen data.

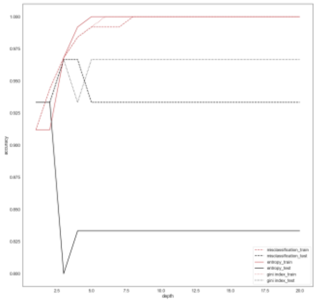


Figure 10 Train and test error - Decision Tree - Hepatitis dataset

CROSS-VALIDATION:

To fine-tune the hyperparameter and increase the generalizability of the decision tree, we split our train set (125 samples) into train set and validation set (100 and 25 samples, respectively) and used 5-fold cross-validation. Figure shows mean classification error and variance with the depth ranging from 1 to 10 using three cost functions. We choose the depth with the lowest classification error for each cost function. Then we test their performance on the test set (30 samples). The confusion matrix of the test result is shown in the table. We can then get performance metrics including accuracy, specificity, recall, precision, F1 score, and error, shown in the figure. As you can see, the performance of the misclassification rate is the best among all the three cost functions.

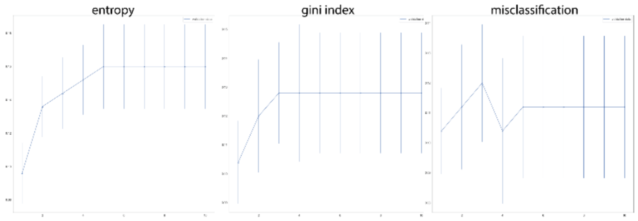
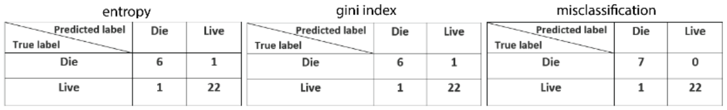


Figure 11 Finding the best depth for DT- hepatitis dataset



entropy

Predicted label / True label	Die	Live
Die	6	1
Live	1	22

gini index

Predicted label / True label	Die	Live
Die	6	1
Live	1	22

misclassification

Predicted label / True label	Die	Live
Die	7	0
Live	1	22

Figure 12 Confusion matrix - DT – Hepatitis dataset

Measure / *Cost*	*Accuracy*	*Specificity*	*Recall*	*Precision*	*F1*	*Error*
Entropy	*0.933*	*0.857*	*0.956*	*0.956*	*0.956*	*0.066*
Gini-index	*0.933*	*0.857*	*0.956*	*0.956*	*0.956*	*0.066*
Misclassification	*0.966*	*1*	*0.956*	*1*	*0.977*	*0.033*

DECISION BOUNDARY

We show the decision boundary of decision trees with different depths (1,2,8, and 20) and cost functions. As you can see, the decision boundary becomes more complex when we increase the depth showing a decrease in generalizability. And the decision boundary will not change when we continue to increase the depth if the growth of the decision tree is completed (Please compare the decision boundary when depth is 8 and 20). We have shown that cost entropy or Gini index somehow share the same classification accuracy curve.

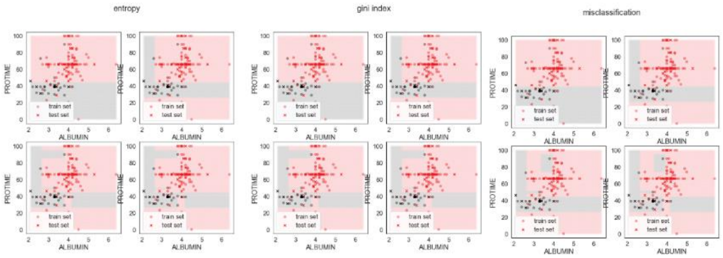


Figure 13 Decision boundary - DT - Hepatitis dataset

DECISION TREE – MESSIDOR DATASET

MODEL TRAINING AND TESTING

Again, we studied the influence of different depths of the decision tree on its classification performance. We use the misclassification rate as our cost function and change the depth from 1 to 20 and fit the tree. And then, we can test the performance of the decision tree using unseen samples (197 instances). We compared the performance in the train set and test set. It can be seen in the figure below that with the increase of depth, the test set accuracy doesn't change much and fluctuates around 0.65, while train set classification accuracy keeps increasing to 1, showing overfitting.

The figure also shows the influence of depth and cost function on the test set classification accuracy. Cross-entropy and Gini index somehow share a similar pattern. Their accuracy of prediction on the test set reaches the peak when the depth is 2 and then decreases. Using misclassification rate as our cost function, the accuracy first drops and then increases.

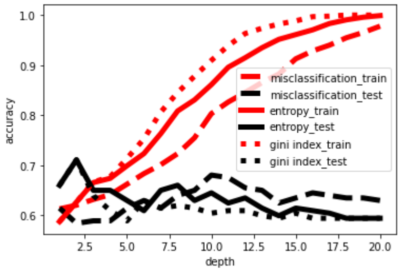


Figure 14 Train and test error - Decision Tree - Messidor dataset

Compared to the first dataset, the second dataset is much larger and has more complex features, which makes it more challenging to grow a good decision tree to have a high classification accuracy on the test set. The test set classification accuracy also shows greater fluctuation with the change of the depth.

CROSS-VALIDATION:

The same approach of 10-fold cross validation was applied at this stage:

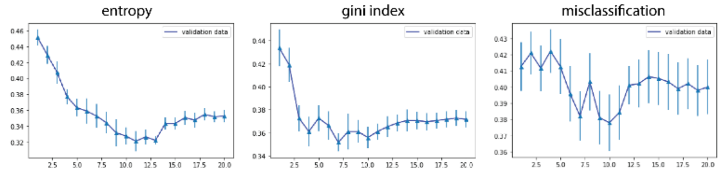
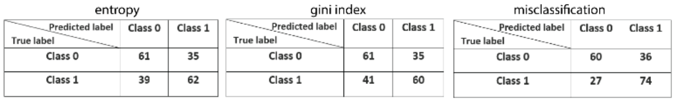


Figure 15 Finding the best depth for DT- Messidor dataset

The best depth for Gini index, entropy, and misclassification are 7, 11, and 10, respectively.



entropy

Predicted label True label	Class 0	Class 1
Class 0	61	35
Class 1	39	62

gini index

Predicted label True label	Class 0	Class 1
Class 0	61	35
Class 1	41	60

misclassification

Predicted label True label	Class 0	Class 1
Class 0	60	36
Class 1	27	74

Figure 16 Confusion matrix - DT – Messidor dataset

Measure Cost	Accuracy	Specificity	Recall	Precision	F1	Error
Entropy	0.642	0.635	0.613	0.639	0.626	0.375
Gini-index	0.614	0.635	0.594	0.631	0.612	0.385
Misclassification	0.68	0.62	0.732	0.6727	0.702	0.319

DECISION BOUNDARY

To show the decision boundary of the decision tree in the 2D plane, we select the two most important features, the number of MAs detected at confidence levels 0.6 and the number of exudate pixels to refit the decision tree using these two features. We show the decision boundary at a depth of 1,3, and 25 using different cost functions.

It can be seen in figure below, the complexity of the decision increases with depth. And we can see the decision boundary is not correct when the depth is 1 and 3 using misclassification rate resulting in low classification accuracy compared with entropy and Gini index. With the increase of depth, the decision boundary using misclassification rate becomes more accurate.

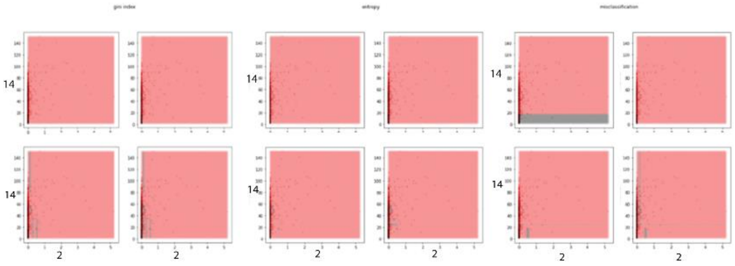


Figure 17 Decision boundary - DT – Messidor dataset

DISCUSSION AND CONCLUSION

In this assignment, to deal with the missing features, in case of the Hepatitis dataset, we use data imputation. We also use cross-validation to help increase the robustness of our model to training data splitting and fine-tune the hyperparameter, k for KNN and depth for the decision tree. KNN classifies new samples according to their distance to the training data. Feature normalization is always needed. By increasing k, the decision boundary becomes smoother and simpler. But for the decision tree, it is not the case. The decision tree builds a tree based on the features and labels of the training data and does judgment at each node according to the feature value of new samples. So, feature normalization is not needed. And increasing the depth of the decision tree can involve more features and values in the judgment, making the decision boundary more complex. We also compare the effect of different distances (Euclidean and Manhattan) on KNN and cost functions (misclassification rate, cross-entropy, and Gini index) on the decision tree. The results of cross-entropy and Gini index are somehow similar because of their similar cost curves. By setting k to 1 for KNN and increasing the depth of the decision tree, we can perfectly fit our models to training data but also cause overfitting. We can use cross-validation to fine-tune these hyperparameters to make our models more generalized to new samples. Now we just impute missing features using the mean value; we can try other methods such as generating missing features based on other features.

STATEMENT OF CONTRIBUTIONS

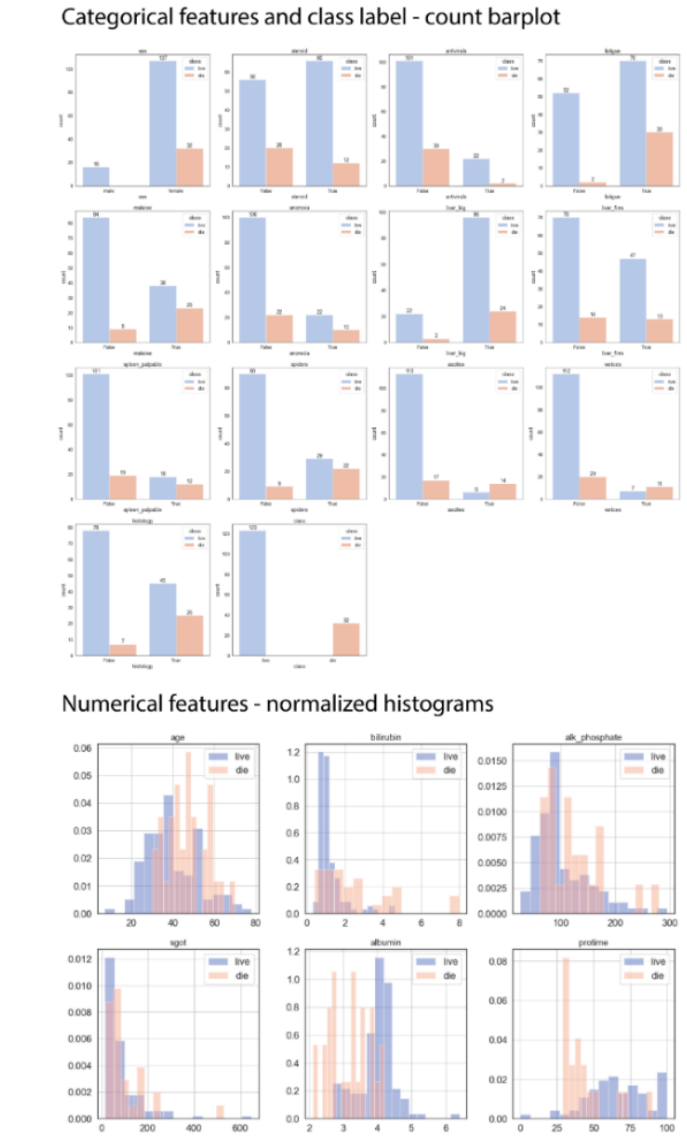
Equal contribution: Asa Borzabadi Farahani - Yunhan Li - Gurucharan Marthi Krishna

REFERENCE

1. Safdari, R., et al., *Applying data mining techniques to classify patients with suspected hepatitis C virus infection.* Intelligent Medicine, 2022.
2. Chawla, N.V., et al., *SMOTE: synthetic minority over-sampling technique.* Journal of artificial intelligence research, 2002. **16**: p. 321-357.
3. Dong, Yiran, and Chao-Ying Joanne Peng. "Principled missing data methods for researchers." *SpringerPlus* vol. 2,1 222. 14 May. 2013, doi:10.1186/2193-1801-2-222
4. Willemink MJ, Koszek WA, Hardell C, et al. Preparing Medical Imaging Data for Machine Learning. *Radiology*. 2020;295(1):4-15. doi:10.1148/radiol.2020192224
5. Chawla, N.V., 2009. Data Mining for Imbalanced Datasets: An Overview, in: . pp. 875–886.. doi:10.1007/978-0-387-09823-4_45
6. H. He and E. A. Garcia, "Learning from Imbalanced Data," in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 21, no. 9, pp. 1263-1284, Sept. 2009, doi: 10.1109/TKDE.2008.239.
7. Neshat, M., Sargolzaei, M., Nadjaran Toosi, A., & Masoumi, A. (2012). Hepatitis disease diagnosis using hybrid case based reasoning and particle swarm optimization. *International Scholarly Research Notices*, *2012*.
8. Alyoubi, Wejdan L., Maysoon F. Abulkhair, and Wafaa M. Shalash. "Diabetic retinopathy fundus image classification and lesions localization system using deep learning." *Sensors* 21.11 (2021): 3704.
9. Rathi, Mrs Pooja. "Automated Diabetic Retinopathy Prediction using Machine Learning Classification Algorithms." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.11 (2021): 3020-3030.
10. Howe Iii EG, Elenberg F. Ethical Challenges Posed by Big Data. *Innov Clin Neurosci*. 2020;17(10-12):24-30. Published 2020 Oct 1.

SUPPLEMENTRARY FIGURES

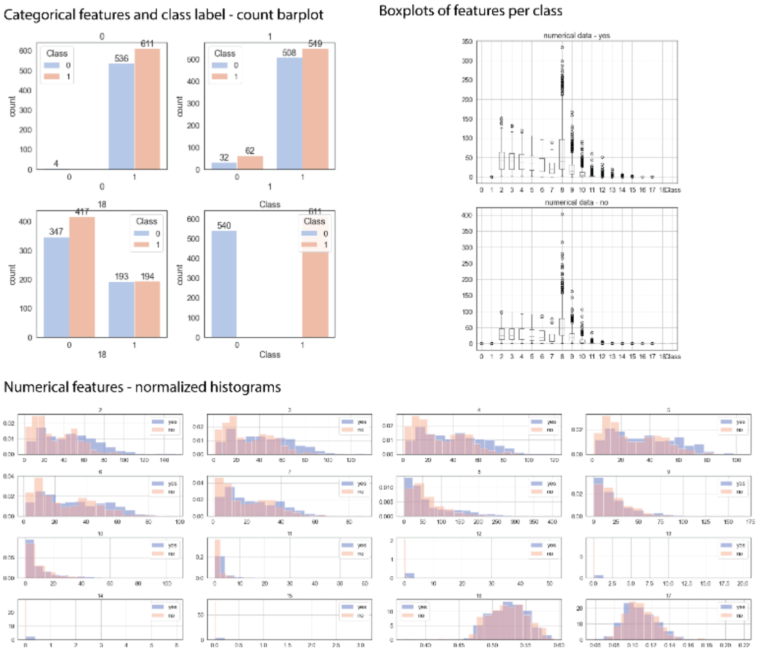
EMPIRICAL DISTRIBUTIONS FOR HEPATITIS DATASET – SUPLEMENTRARY FIGURE 1



Categorical features and class label - count barplot

Numerical features - normalized histograms

EMPIRICAL DISTRIBUTION FOR MESSIDOR DATASET SUPLEMENTRARY FIGURE 2



Categorical features and class label - count barplot

Boxplots of features per class

Numerical features - normalized histograms