|  |
| --- |
| **Machine Learning Report** |

**Gaurav Rai**

College of Engineering

Northeastern University

Toronto, ON

*rai.ga@northeastern.edu*

**Abstract**

In this report, I explored the performance of supervised learning algorithms on two interesting datasets. The three learning algorithms include Decision Tree, Neural Network and Boosting The two datasets (<https://github.com/raiigauravv/LAB6105/blob/main/titanic.csv>) and (https://github.com/raiigauravv/LAB6105/blob/main/ship\_data.csv). The learning curve, model complexity and training time of each algorithm on both datasets have been explored and analyzed.

**1) Datasets1**

Firstly, I used the above dataset and applied Decision Tree algorithm on it. I imported libraries like pandas, NumPy, matplotlib libraries to perform operations on the dataset. I checked and cleaned the dataset by looking for missing null values. Later, I differentiated algorithms between dependent and independent variables(X,y). Performed train\_test\_split to create validation set and used decision tree classifier to train and test the data. I tried to fit the training model and to check its accuracy using accuracy\_score library. The accuracy of our testing dataset was around 73% and training dataset was around 98%. Here we can clearly see that for training\_dataset accuracy is quite high whereas for testing\_dataset it is very low. Therefore, our model is overfitted and to avoid this we will use pruning. After performing pruning the accuracy score of testing\_dataset is 73% and training\_dataset is 80%.

Secondly, I applied Neural Network algorithm on it, and I got amazing results again with the dataset1. I created a simple feedforward neural network model using kerals.models and Sequential library. Later, I compiled the model using ‘adam’ optimizer and use loss function as “sparse\_categorical\_crossentropy” for the categorical crossentropy loss function without the need for one-hot encoding, moving on I trained the model and set epoch=50 (defines the number of times the entire data set has to be worked through the learning algorithm,) batch\_size=32 (32 samples from the training dataset will be used to estimate the error gradient before the model weights are updated) and validation\_split=0.1 (Fraction of the training data to be used as validation data.) to get better accuracy. I evaluated the model later and got training data accuracy around 77% and testing data accuracy around 73% for dataset1 and for dataset2 I got accuracy around 79% for both training and testing dataset.

Lastly, I applied Boosting algorithm on it, and I got decent results. I created a decision tree classifier and split the values. Later, I used the adaboosted classifier to get an estimate accuracy value then I fit this boosted classifier into the training dataset. I made predictions on the testing dataset and got the accuracy around 79%.

Table 1: The basic feature of both datasets.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Data Set Characteristics | Attribute Characteristics | Associated Tasks | Number of RangeIndex | Number of Columns |
| Dataset1 | Multivariate | Real | Classification | 891 | 12 |
| Dataset2 | Multivariate | Real | Classification | 690 | 25 |

* 1. **Data characteristics**

The ROC curve of our decision tree is shown below in the graphical form. I have also visualized the alpha w.r.t impurity of the leaves. Later, we have displayed the accuracy vs alpha for both training and testing sets.

A graph of a line

Description automatically generated

Figure 1:Dataset1: ROC curve of decision tree.

A graph with a line

Description automatically generated

Figure 2:Dataset2: ROC curve of decision tree.

A graph of an alpha

Description automatically generated with medium confidence

Figure 3: Dataset1 alpha w.r.t impurity of the leaves.

A graph of a graph

Description automatically generated

Figure 4: Dataset 1 accuracy vs alpha for both training and testing sets

A graph of a graph

Description automatically generated with medium confidence

Figure 5: Dataset2 alpha w.r.t impurity of the leaves.

A graph of a graph

Description automatically generated

Figure 6: Dataset 2 accuracy vs alpha for both training and testing sets

* 1. **Why are these interesting datasets?**

The datasets are interesting because of their practicality and smoothness. I tried performing various algorithms some worked, and few were not, but after cleaning the data most algorithms started running smoothly.

**2) Decision Tree with Pruning**

I chose post-pruning by controlling the cost complexity and impurities of the tree which helps with the accuracy of training and testing dataset. I used post- pruning because the decision tree has very large depth and will show overfitting of model. After performing pruning the accuracy score of testing\_dataset is 73% and training\_dataset is 80% for dataset1, whereas pruning the accuracy score of testing\_dataset is 75% and training\_dataset is 81%.

**3) Conclusions**

In this report, I have analyzed the performance of 3 supervised learning algorithms on dataset 1 and I have got the phenomenal results with the accuracy for both training and testing dataset.

A diagram of a number of samples

Description automatically generated with medium confidence

Figure 7: Dataset1: Final Decision Tree after Pruning

A diagram of a number of samples

Description automatically generated with medium confidence

Figure 8: Dataset2: Final Decision Tree after Pruning