Analysis of Hyper-parameter tuning on titanic dataset using Decision Tree

Salman Ahmad Khan
Department of Computer Science
Bahria University Islamabad, Pakistan
MS Data Science (2nd Semester)
Reg No: 01-249212-013
E-mail: Salmannahmed123@gmail.com

***Abstract* – Titanic disaster occurred on April 15, 1912 (100 years ago), killing about 1500 passengers and crew members. The unfortunate incident still compel the researchers and analysts to understand what can have led to the survival of some passengers and demise of the others. With the help of Machine learning algorithm and a dataset consisting of total 891 datapoints having 712 datapoints is used for the training of model in the train set and 418 rows for the testing of the model. The Decsion tree alogothim is used for classification to dertemine the label of survived and not survived passenger in this mishap. the main purpose of this task to tune the hyperparmaters to minimize the cost function and get best result of the model. Tools used for this project are VSCode using Python and different dependencies or libraries used such as Numpy, Pandas, Matplotlib, Seaborn and machine learning library Scikit-learn.**

***Keywords* – Supervised learning, Classification technique, Decision Tree, Hyper parameterss, Machine learning, Numpy, Pandas, Matplotlib, Scikit-learn, VScode, Jupyter, Anaconda.**

I. INTRODUCTION

Machine learning is used as a predictive approach to unseen data. In machine learning, the model or algorithm first learns how to perform the task by training the dataset, then testing [1]. In the Supervised learning approach, both the input and output data will be given to the model to learn from the data and predict continuously. for the prediction of continuous output problems, we used the Regression model while for categorical or classification problems, we use Classification algorithms [2]. Desiosion tree algorithms is used in this task. Decision tree learning is the method of construction of a decision tree from class-labeled training tuples. A decision tree can be considered as a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. . [3] The main objective of this work is to analyze the hyper parameter to minimize the cost function and get the better model and result for this titanic dataset.

The Major steps involves in this project include of machine learning that are:
1. Data Collection
2. Data prepossessing and EDA
2. Choose algorithm or Classifier (in this case Decision tree classifier)
3. Creating object of the model
4. Train the model by training dataset (using different ratio)
5. Making prediction on unseen
6. Evaluation of the model

II. DATASET

In this project, the dataset is taken from the online source Kaggle, which is an open-source repository. The dataset contains 891 data points in which PassengerID, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare , Cabin and Embarked are the attributes. Out of these 5 attributes, 4 are in numerical attributes while the predictive output attribute "Species" are in categorical form. Figure.1 shows the exploration of the dataset and the relationship among all attributes and shows the dispersion of the sample point of the dataset. Fig.2 shows the overall data distributions.

III. IMPLEMENTATION

We used Anaconda software(Visual Studio Code) for data manipulation, wrangling, and built a model using different libraries of Numpy, Pandas, Matplotlib, Seaborn, and Scikit Learn. We keep training testing ratio is 80-20 percent, in which 80 percent is training and 20 for testing. The Desicion tree algorithm is used for the classification of the titanic set to predict the survived and not survived. The standard Scalar approach has been used for the normalization of the data.

IV. Detailed Analysis of variation in Hyper-parameter

In this task, we do some analysis on basis of variation in decision tree hyper parameter, which are Criterion, Splitter and Maximum depth number. We have the two values of Criterion on which we check the accuracy, precision, Recall and Confsuin Matrix. In descion tree Criterion has two different approches; Entropy and Gini. We also have two variations in splitter hyper-parameters which is "Best" and "Random". We will use this splitter value in combination with criterion and maximum depth with different combinations. We used 2 and 3 as the values of Max Depth with use different values with criterion to check this hyper-parameter impact on accuracy. We also use the max depth is 2 because we want to visualize the decision tree so 2 is the best depth to visualise it. We also use split data into 20 percent test and 80 percent train in all the scenarios.

1) Analysis with Criterion='Entropy', splitter='best' and max='non' (Default): Using the above comibation of hyper-paramaeter having Entropy criterion is used with Best spliiter and max-depth number allotted by alogorithms. Using these comibation we get the accuracy value 0.8156, precision value 0.8035 and recall value is 0.6716 , which is shown in fiq.1;

```
Accuracy is : 0.8156424581005587
Precision is :0.8035714285714286
 Recall is :0.6716417910447762
[[101  11]
 [ 22  45]]:is the Confusion Matrix
```

Fig. 1. Default hyper-parameter Results

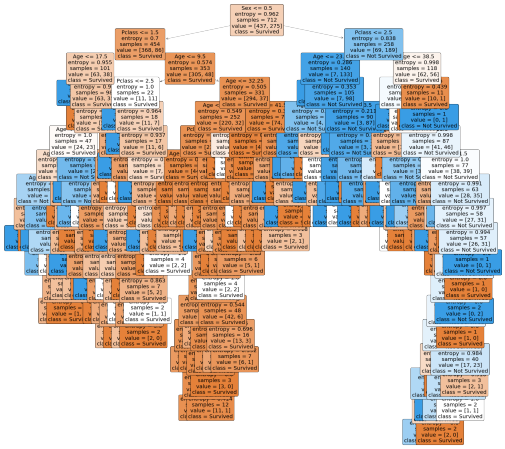


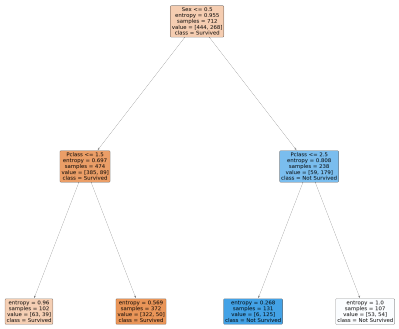
Fig. 2. Default Descion Tree

We can also get huge default decision tree from that hyper-parameter. Base on the tree and other results such as accuracy etc , we conclude that , the hyper-parameter used by default is not suitable and best result giver for this data set and we need to tune our hyper parameters.

2) Analysis with Criterion='Entropy', splitter='best' and Max Depth= '2': In this case, we manually choose our depth of tree which is 2 and recalculate the accuracy , precision, recall and also tree. The accuracy is 0.7653, precion value is 0.7105 and Recall value is 0.7297.

```
Accuracy is : 0.7653631284916201
Precision is :0.7105263157894737
 Recall is :0.7297297297297297
[[83 22]
 [20 54]]:is the Confusion Matrix
```

Fig. 3. Criterion='Entropy', splitter='best' and Max Depth= '2'



Based on the above result, we conclude that this combinations is also not well fitted for this data set.

3) Analysis with Criterion='Entropy', splitter='random' and Max depth='2': We use the above combination of the hyper-parameter we have the same accuracy as we use the best splitter. So using random split just have the impact on the depth of the tree but at the end when we talk about performance in this scenario performance is same. we tuned two hyper parameters this time, random splitter and max depth value is two.

```
Accuracy is : 0.8268156424581006
Precision is :0.8169014084507042
 Recall is :0.7631578947368421
[[90 13]
 [18 58]]:is the Confusion Matrix
```

Fig. 4. Result: Criterion='Entropy', splitter='random' and Max depth='2'

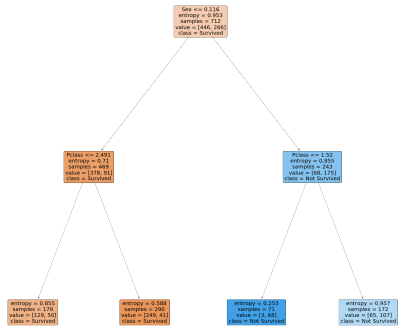


Fig. 5. Criterion='Entropy', splitter='random' and Max depth='2'

4) Analysis with Criterion='Entropy', splitter='random' and Max depth='3: In this case, we manually set the depth of the tree which are 3, in which the random splitter and Entropy's criterion is used wit all remaining all hyper parameter with by default values. the result and the tree graph's shown below;

```
Accuracy is : 0.770949720670391
Precision is :0.7419354838709677
 Recall is :0.647887323943662
[[92 16]
 [25 46]]:is the Confusion Matrix
```

Fig. 6. Result : Criterion='Entropy', splitter='random' and Max depth='3'

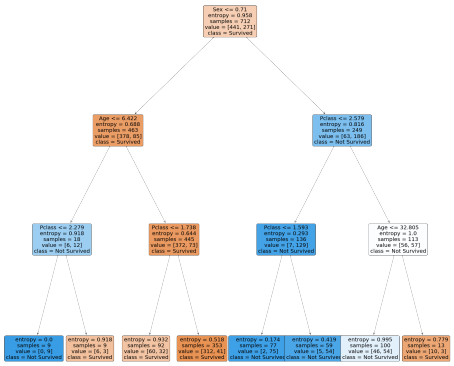


Fig. 7. Tree: Criterion='Entropy', splitter='random' and Max depth='3'

V. Discussion and Briefly Analysis

As the objective mentioned above , the main task to analyze the parameter and hyperparameter to check the impact on accuracy, prescion, recall and confusion matrix. The Criterion (Gini, Entropy), Splitter(Random and Best) and the Max depth no of the tree(default: none, 2 and 3) are comparatively checked and calculate the accuracy, prscion, recall value and confusion martix (some of them mentioned above). The summary of all this analysis are briefly explained in the following table;

TABLE I
Different Training and Testing Ratio

Criterion	**Splitter**	**'Max Depth**	**Accuracy**	**Precsion**	Recall
Entropy	Best	None	0.8156	0.8035	0.6716
Entropy	Best	2	0.7653	0.7105	0.7297
Entropy	Best	3	0.7709	0.7702	0.7037
Entropy	Random	none	0.7932	0.8382	0.6867
Entropy	Random	2	0.8367	0.8679	0.7189
Gini	Random	3	0.7709	0.7419	0.6478
Gini	Best	None	0.7821	0.7636	0.6176
Gini	Best	2	0.8044	0.7037	0.8382
Gini	Best	3	0.8234	0.7846	0.7434

VI. CONCLUSIONS

After completing this project by implementing the as well as via literature review, we conclude that, We varies the values of selected hyper-parameters for finding the best performance according to selected parameter. We have the value of test train split 70 percent for train and 30 percent for test.As we vary the the values we found we got different results of accuracy.For visualization we can see decision tree from which we can easily understand the distribution of leaf and also selection criteria of node.We got the maximum values of accuracy match when we use Entropy criteria , Random split and max depth is 2.

ACKNOWLEDGEMENTS

We get help in implementation from different literature sources and also borrow some code from online sources Kaggle **?**, Towadsdatascience, medium and geek for geeks [4] [5] [2].

REFERENCES

[1] I. El Naqa, M. J. Murphy, "What is machine learning?", *in machine learning in radiation oncology*, pp. 3–11, Springer, 2015.

[2] R. Sathya, A. Abraham, *et al.*, "Comparison of supervised and unsupervised learning algorithms for pattern classification", *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.

[3] A. Singh, S. Saraswat, N. Faujdar, "Analyzing Titanic disaster using machine learning algorithms", *in 2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 406–411, 2017, doi:10.1109/CCAA.2017.8229835.

[4] "ML: Label encoding of datasets in Python", , Sep 2021, URL: `https://www.geeksforgeeks.org/ml-label-encoding-o`

[5] Lnbalon, "Iris dataset EDA and classification analysis", , Aug 2017, URL: `https://www.kaggle.com/code/lnbalon/iris-dataset-`

VII. Appendix