COMPARISON BETWEEN DECISION TREE AND NAÏVE BAYES ON WIND DATASET

KASHVI SHAH |210032143| INM431-Machine Learning

Description and motivation of the problem

- We will be building two models that is Naïve Bayes and Decision Tree for a classification task to use supervised learning.
- The models will be applied to a wind speed dataset and will be using the same methodology used in paper [5], which will be considered as our main paper.
- The dataset contains 11 synoptic meteorological stations with their wind speeds(knots) to predict if the station has a lower target or not.

Data Set and Exploratory Analysis

- The dataset is Wind data which consists of wind speed calculated on 11 different meteorological stations from the year 1961-1978 in the Republic of Ireland.
- This dataset's target is converted to two class nominal features by calculating mean and classifying instances with target values that are lower as 'P' and all other values as 'N'. But in this coursework, changes the labels to 0 and 1 using label encoder.
- The dataset consists of 6574 instances and 15 features which does not have any missing values.
- Figure 1 consists of table with all the feature predictors with their information about count, mean, standard deviation, min, and max.
- The Correlation heatmap looks into the correlation between the variables. As we can see almost all the stations are highly correlated but this seems to be a coincidence as the stations cannot be connected (shown in figure 2).
- The pie chart depicts all the stations and their percentage which shows that BEL station had the highest contribution in our data(figure 3).
- Box plot help to analyze distribution of each feature and some outliers. This distribution helps us in understanding the how binary class is related to each of the stations.(figure 4)
- Standardization is done in MATLAB, as features might have varying scales.

	count	mean	std	min	25%	50%	75%	max
year	6574.0	1969.500304	5.188131	1961.00	1965.00	1969.50	1974.00	1978.00
month	6574.0	6.523274	3.448871	1.00	4.00	7.00	10.00	12.00
day	6574.0	15.728628	8.800335	1.00	8.00	16.00	23.00	31.00
RPT	6574.0	12.363715	5.619610	0.67	8.12	11.71	15.92	35.80
VAL	6574.0	10.646448	5.268602	0.21	6.67	10.17	14.04	33.37
ROS	6574.0	11.660103	5.007765	1.50	8.00	10.92	14.67	33.84
KIL	6574.0	6.306275	3.605407	0.00	3.58	5.75	8.42	28.46
SHA	6574.0	10.456880	4.935739	0.13	6.75	9.96	13.54	37.54
BIR	6574.0	7.092254	3.968683	0.00	4.00	6.83	9.67	26.16
DUB	6574.0	9.796834	4.977272	0.00	6.00	9.21	12.96	30.37
CLA	6574.0	8.494420	4.499000	0.00	5.09	8.08	11.42	31.08
MUL	6574.0	8.495818	4.167778	0.00	5.37	8.17	11.21	25.88
CLO	6574.0	8.707268	4.503615	0.04	5.33	8.29	11.63	28.21
BEL	6574.0	13.121007	5.835037	0.13	8.71	12.50	16.88	42.38

Figure 1

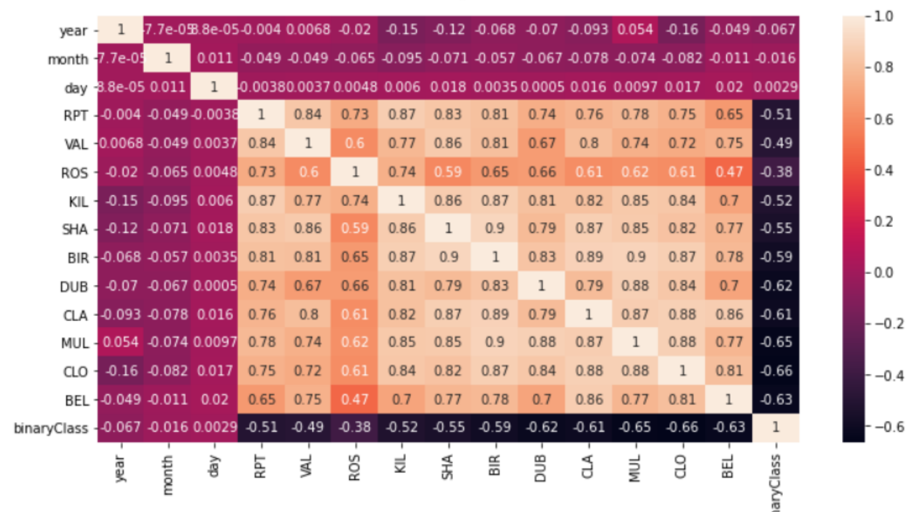

Figure 2 – Heatmap of correlation

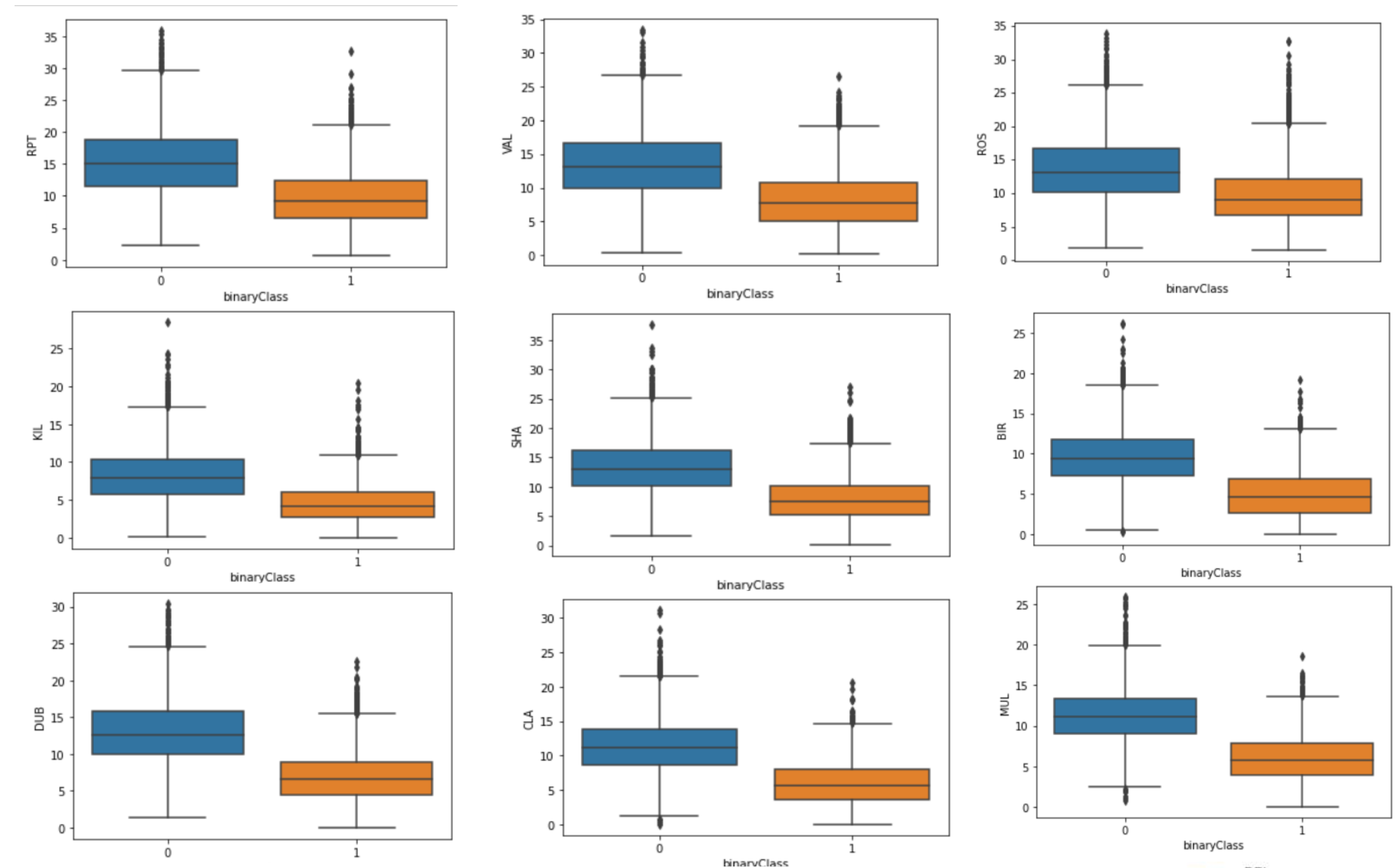

Figure 4 – Box plots for all stations

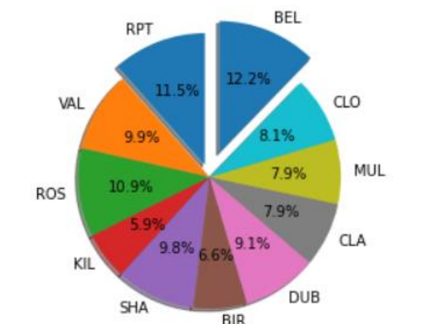

Figure 3 – Pie chart

Summary of both the models

Naïve Bayes

- The Naive Bayes classifier is one of the most simple yet more sophisticated techniques of classification. It combines information from predictions in order to get accurate results for our model.[1]
- It classifiers assume that there are no dependencies amongst attributes. This assumption is called class conditional independence.[3]
- It is a statistical approach of inductive-inferencing in the classification problem that uses bayes theorem.[2] It is used to find the maximum probability as we need to classify our data on the basis of the class that has the highest probability value.

Advantages:

- Naïve bayes has a very simple design process because unlike neural networks, it does not have to set many parameters.
- It removes irrelevant features to improve the performance of the model.
- Computational time is short.
- It returns probabilities as a result which can be applied to a variety of tasks.
- Although this model is based on the conditional independence assumption, even if this independence does not hold, it will perform.

Disadvantages:

- On some datasets, the accuracy is less.
- It gives zero probability when an attribute in the test data set is not present with a target in the training dataset.
- Sometimes it doesn't work when real-life problems are in question.

Decision Tree

- Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value.[4]
- It is used by the training model which predicts the class of the target variable by learning from prior data.
- Attribute selection is a problem faced by the decision tree when the model has to identify the root node for each level.[1]
- Here data is modelled as hierarchy of connected nodes where each node splits into sub node according to parameters.

Advantages:

- Produces nearly to accurate results.
- It takes very little time for data preparation and while implementing this model.
- Different measures can be used to find the best split attribute.
- Comparatively to other models, DT takes up less memory.
- DT is simple, fast, and comprehensible.

Disadvantages:

- Overfitting problem usually occurs in DT models
- Decision trees can have a significantly more complex representation for some concepts due to replication problems.[4]

Hypothesis Statement

- As per few research papers, its is proven that naïve bayes model works better than decision tree model.
- As per our main paper [5], it can be said that performance metrics of naïve bayes had higher values than Decision Tree.

Description of the choice of training and evaluation methodology

Choice of Training:

- The original data set is split into 85% for training and testing as we are using K-fold cross-validation and 15% of the data will be unseen by the model.
- In training, we are applying 7 folds for partitioning as per the paper [5].
- Models are trained with and without the hyperparameters involved and both scenarios are trained to see if there is any increase in accuracy.

Evaluation Methodology:

- Evaluating the results with validation accuracies, confusion matrices, ROC curve, and AUC score, and other performance metrics. The baseline model will use the features from the paper [5].
- The optimized model with better accuracy for both DT and Naïve bayes with hyperparameter optimization will be done.
- As mentioned, that 15% of the dataset will be unseen, which will be predicted and we can take note of training and test times.
- These models will run with their best parameters on the test data.

METRICS	DT	NB
Train Acc	83.89%	81.16%
Test Acc	83%	80.22%
Val. Acc	83.72%	81.17%
Avg AUC	91.80%	89.40%
Precision	82.97%	80.14%
Recall	82.76%	80.10%
F1 score	82.86%	80.16%
Error	16.28%	18.83%

Figure 5: Performance Metric

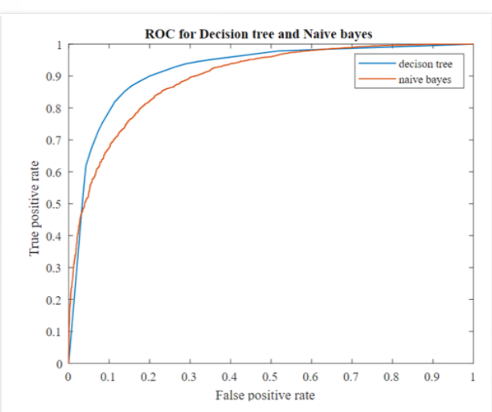

Figure 6: ROC curve

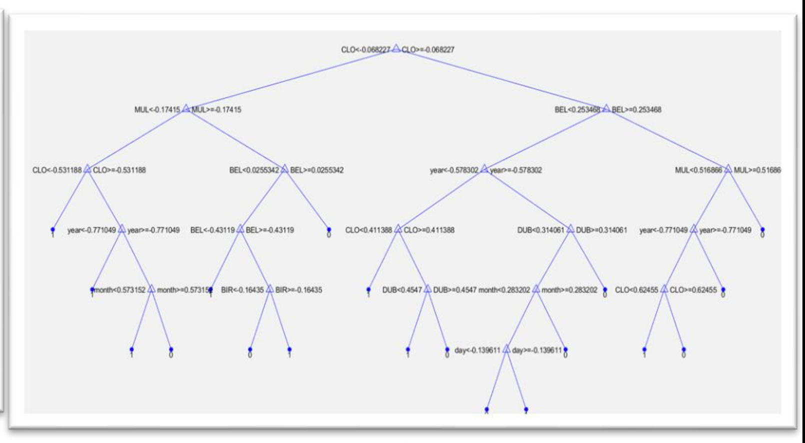

Figure 7: Optimized Decision Tree

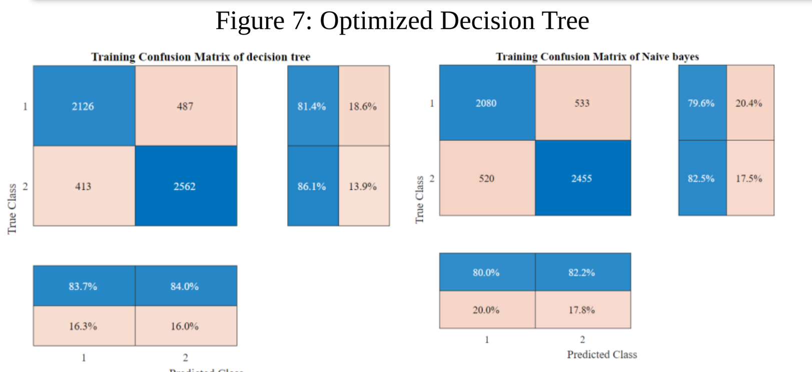

Figure 8: Confusion Matrix

Choice of parameters and experimental results

Naïve Bayes:

- The baseline model was fit in using a normal method which had decent accuracy.
- To improve the model, we tried using a kernel estimator which yields more accurate predictions, but in our model, it hardly changes the accuracy result.
- From this, our first hypothesis is proven false.

Decision Tree:

- Using the CART Algorithm, we are trying to fit the model.
- Using optimized hyperparameters, we found optimal parameters by running them on our DT model and then changing them according to the best model, we get significant change in accuracy of the model.
- As result we get better accuracy for DT than NB, proving our second hypothesis wrong.

Analysis and Evaluation of results

- Both the models were worked on 85/15 data set and 15% of the data set was kept unseen by the model, for it to be used against the training data.
- Figure 5 represents the accuracy, F1 score and other performance metrics which was calculated after the hyperparameters were added. As seen from the table the accuracy for the DT classifier is roughly 3% higher than that of the NB classifier proving our hypothesis wrong. F1 score is roughly 2% higher in DT classifier than NB.
- Before adding any hyperparameters it was noticed that the tree was too deep and very difficult to understand. By adding the hyperparameters, it not only resolves this problem but also increases the DT model accuracy over NB model. Tree figure obtained can be visualized in Figure 7.
- **Decision Tree**: Hyperparameters are basic parameters that control the learning process and thereby help the learning process of the model. For our DT model, we used this to get a tree that is not that deep and more understandable. Our model's hyperparameters usually gives a number for minimum leaf size which will be added to the new model. MinLeafSize is basically a minimum number of samples required to be at leaf node. The CART algorithm is used to maximize the split criterion gain over all possible splits. For DT, we have used 7-fold cross-validation as in one of the methodologies used in our main paper [5]. After tuning in the hyperparameters we saw that our test error is around 0.162.
- **Naïve bayes** : Hyperparameters tuned in for the NB model is a kernel, which is basically a probability distribution name for smoothing density estimate. For NB, we use 7-fold cross-validation as mentioned in the main paper. After tuning the hyperparameters we get the test error around 0.19.
- In the figure 5, we see that precision, accuracy, recall, F1score, and AUC are shown for both DT and NB and it is observed that DT and NB have done well in all the metrics but for our coursework, we can safely say that DT is better than NB. ROC curve is shown in figure 6.
- Figure 8 gives us confusion matrix, for both the both the models generated by the training model.

CONCLUSION: As from above analysis we can conclude that Decision Tree classifier works better on this dataset than naïve bayes classifier. And both our hypothesis fails as before any optimization, both the model gave similar accuracy but after optimization decision tree gives better accuracy.

Lessons learned, future work

Lesson learned:

- Optimizing and comparing the model has been a very useful and interesting exercise.
- Both the models have worked well and given quite a good accuracy.
- Tuning in the hyperparameters has helped both models increase their accuracy.

Future Directions:

- For Naïve bayes, we can implement Max- redundancy Minimum relevance for feature selection.
- Exploring the performance of Random Forests would be an extension of the Decision tree model.
- Training other models and combining all the results to see which works best on this particular dataset.
- Developing further skills in MATLAB and continue adding knowledge to our personal data science material.

References:

[1] Yadav, K., & Thareja, R. (2019). Comparing the Performance of Naive Bayes And Decision Tree Classification Using R. International Journal of Intelligent Systems and Applications, 11, 11-19.

[2] S Rahmadani et al 2018 J. Phys.: Conf. Ser. 978 012087

[3] Ashari, Ahmad & Paryudi, Iman & Tjoa, A Min. (2013). Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. International Journal of Advanced Computer Science and Applications. 4. 10.14569/IJACSA.2013.041105.

[4] Sayali D. Jadhav, H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques", International Journal of Science and Research (IJSR)

[5] hanam, Jobeda Jamal & Foo, Simon. (2021). A comparison of machine learning algorithms for diabetes prediction. ICT Express. 7. 10.1016/j.icte.2021.02.004.