Comparison of Random Forest and Decision Tree in Gender Classification

Gulce Alper
Gulce.Alper@city.ac.uk

Explaination and Motivation

- Comparison of binary classification results in gender determination (Female/Male) by using Random Forest and Decision Tree in the light of given physical characteristics
- Determining the physical features that affect gender the most
- Evaluation of former classification studies to gain comprehensive insight

Exploratory Data Analysis

The initial data analysis steps and visuals are listed below:

- Dataset used in the study: Gender Classification Dataset, from Kaggle
- The dataset itself consists of 5001 rows and 8 columns in total with 1 target (gender) column and 7 feature columns including 5 binary (long_hair, nose_wide, nose_long, lips_thin, distance_nose_to_lip_long) and 2 continuous (forehead_width, forehead_height) variables.
- Binary columns show the value 1 if the physical feature in question is present, and 0 otherwise. Target column contains categorical Female/Male values. Continuous columns include values in cm.

In the next steps, a few visualizations were used to have information about the structure of the data.

- As a first step, missing values were checked and it was found that there were no missing values.
- Figure 1 provides information about the distributions of the continuous variables by gender. It can be said that the each variable has a balanced distribution on gender.
- Figure 2 provides an outline of the continuous variables with their descriptive statistics.
- In Figure 3, a correlation matrix was calculated to evaluate whether there was a high correlation between the variables. When looking at the results, it is clearly seen that there is no high relationship between any variables which is good.
- Lastly, in Figure 4 we can see the the proportion of values for binary variables. It can be seen that the dataset is generally balanced, except for the long hair variable.

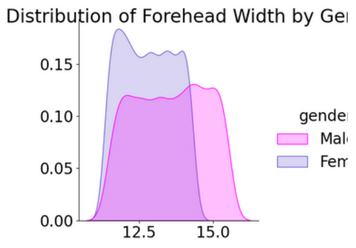


Figure 1. Displots

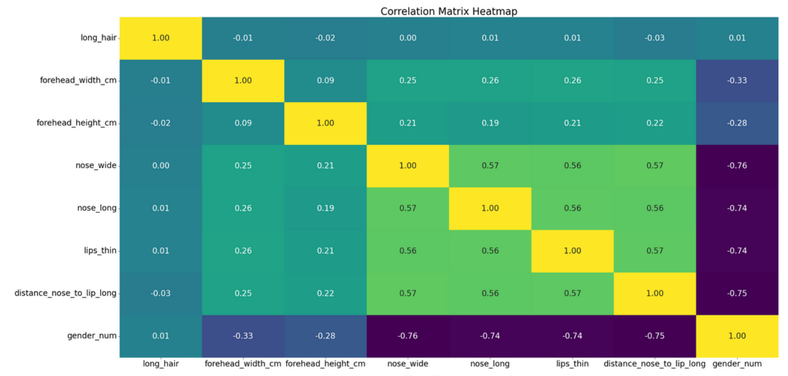


Figure 3. Correlation Matrix

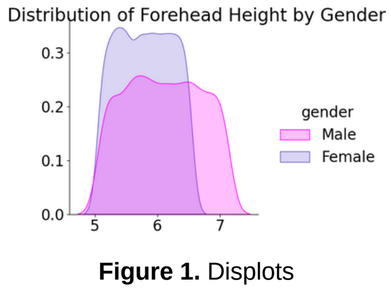


Figure 2. Barchart

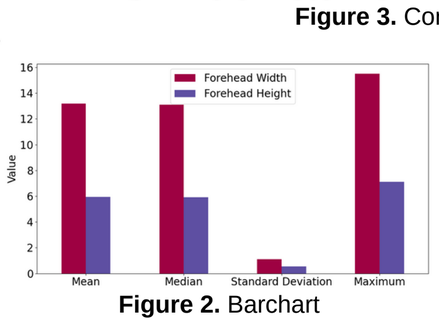
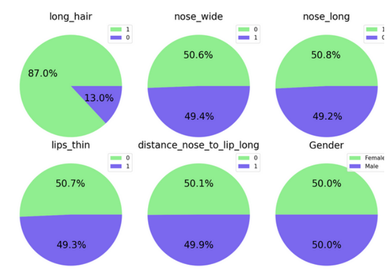


Figure 4. Piecharts



Random Forest (RF)

Random Forest or in other words Random Decision Forest is one of the common classification algorithms in the field of machine learning. RF is a ensemble supervised learning algorithm. RF is used to rank the importance of variables in a classification problem. [2]

Advantages

- Overfitting can be prevented by increasing the number of trees
- Hyperparameters are applicable and easy to understand
- It can handle binary, continuous, and categorical data.[1]
- Resistant to outliers

Disadvantages

- Training time is more than other models due to its complexity. Whenever it has to make a prediction, each decision tree has to generate output for the given input data. [1]
- High complexity

Decision Tree (DT)

Decision Tree is also one of the supervised learning algorithms for classification problems. The most significant feature of DTC is its ability to change the complicated decision making problems into simple processes, thus finding a solution which is understandable and easier to interpret. [3]

Advantages

- Can easily handle with missing values
- Needs tiny data preparation
- Can be easily applied and interpreted
- Resistant to outliers

Disadvantages

- Risky to make changes in dataset
- Possibilty for overfitting
- Classifying the continuous data may prove to be expensive in terms of computation, as many trees have to be generated to see where to break the continuum. [4]

Hypothesis Statement

- A Random forest is ensemble of decision trees so it helps in predicting data accurately**.** [2]
- Random Forest training takes more time than Decision Tree due to several trees
- Random Forest is more reliable in evaluating the feature importance.

Description of the choice of training and evaluation methodology

1. Dividing the data set into 80% training and 20% test sets using the HoldOut method
2. A further seperation on train set by cross validation including train and validation sets
3. Fitting base model with hyperparameters on validation set and assesing performance
4. Implementing hyperparameter tuning on validation set in order to find best hyperparameters as well as the best model results
5. Applying best hyperparameters on the whole train set and evaluating the accuracies
6. Comparing the performance of test sets.

Choice of parameters, content and experimental results

Random Forest

- Fitting the model by using TreeBagger (Bootstrap Aggregating) function
- Selected hyperparameters : NumTrees[50, 100], MinLeafSize[5, 10], MaxNumSplits[10, 15], NumPredictorsToSample ‘all’, OOBPrediction ‘on’, OOBPredictorImportance ‘on’
- Applying cross validation by 5 KFold
- Visualising confusion matrix and feature importance
- Calculating performance metrics

Decision Tree

- Fitting the model by using fitctree function
- Selected hyperparameters : MinLeafSize[5, 10], MaxNumSplits[10, 15]
- Applying cross validation by 5 KFold
- Visualising confusion matrix
- Calculating performance metrics

Figure 5. Confusion Matrices

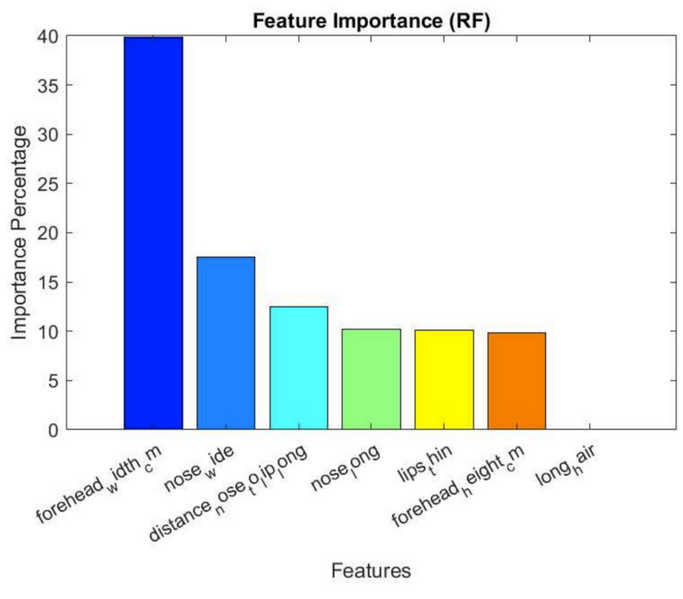


Figure 6. Feature Importance

	Precision	Recall	F1_Score
Random Forest	0.9733	0.9753	0.9743
Decision Tree	0.9548	0.9708	0.9627

Figure 7. Classification Metrics

Analysis and Evaluation of results:

The observed outputs for each step of the classifications are as follows:

- Considering the base model performance with current hyperparameters, both models performed an high accuracy with %95.88. This suggest that Random Forest generalizes better to unseen data. After performing hyperparameter tuning the performance of Random Forest boosted by %1.62 , higer than that of Decision Tree.
- As for the best accuracies calculated during hyperparameter tuning, it was observed that the tuned accuracy for Random Forest (%96.88) was lesser than the test accuracy (%97.50) . This means Random Forest might be overfitting the test data. This can be prevented by increase the number of trees.
- As stated in the hypothesis statement, Random Forest took approximately 12 seconds longer to train the data than the Decision Tree. Since Random Forest is an ensemble of data subsets of the decision tree, it is normal for multiple trees to take a long time to train.
- Random Forest can easily calculate feature importance by using OOBPermutedVarDeltaError parameter. This provides opportunity to identify features which have the most impact on the data. In Figure 6, we can see that the forehead width has the highest impact on gender.
- In Figure 5 and Figure 7, we can analyze the performance metrics and confusion matrices (row and column normalized) of each model.

Lessons learned and future work

Lessons learned

- For small sized datasets Random Forest may overfit the unseendata. It is always better to perform on large number of datasets.
- It is always better to be variation in the model metrics.
- Hyperparameter selection can significantly affect model performance.

Future work

- Check for variation on the data scores
- Try to use larger datasets
- Discover feature importance in order to remove features with high dominance on data

References

[1] R, Sruthi E. “Understand Random Forest Algorithms With Examples (Updated 2023)”. Analytics Vidhya, 17 June 2021

[2] Prajwala, T. R. "A comparative study on decision tree and random forest using R tool." International journal of advanced research in computer and communication engineering 4.1 (2015): 196-199.

[3] PRIYANKA; KUMAR, Dharmender. Decision tree classifier: a detailed survey. International Journal of Information and Decision Sciences, 2020, 12.3: 246-269.

[4] GUPTA, Bhumika, et al. Analysis of various decision tree algorithms for classification in data mining. International Journal of Computer Applications, 2017, 163.8: 15-19.