

1)Student Registration NO:

2)Project Methodology:

2.1) Data Collection: The dataset "forestdata.csv" contains information about forest fires.This dataset contain different attributes such as collector ID, c.score, l.score, rain, tree age, surface litter, wind intensity, humidity, tree density, month, time of day and fire(target class/column).

2.2) Exploration of DataSet: First explored the dataset using histograms to visualize the each attribute. Missing values were handled by replacing the mean of the respective attributes in which they occurred.

2.3) Preprocessing: After fixing the null values. The column 'time.of.day' was cleaned and transformed into numerical values for further process.

2.4)DataSet Splitting: The dataset was split into training(60%), validation(20%), and testing(20%) sets.

2.5) Model Selection: Three classification models were trained and evaluated(Logistic Regression, Decision Tree, Neural Network)

2.6) Models Comparison: The accuracy, precision, and F1 score were calculated for each model individually. The results of all models shown in one table at the end of all work.

3)Variables:

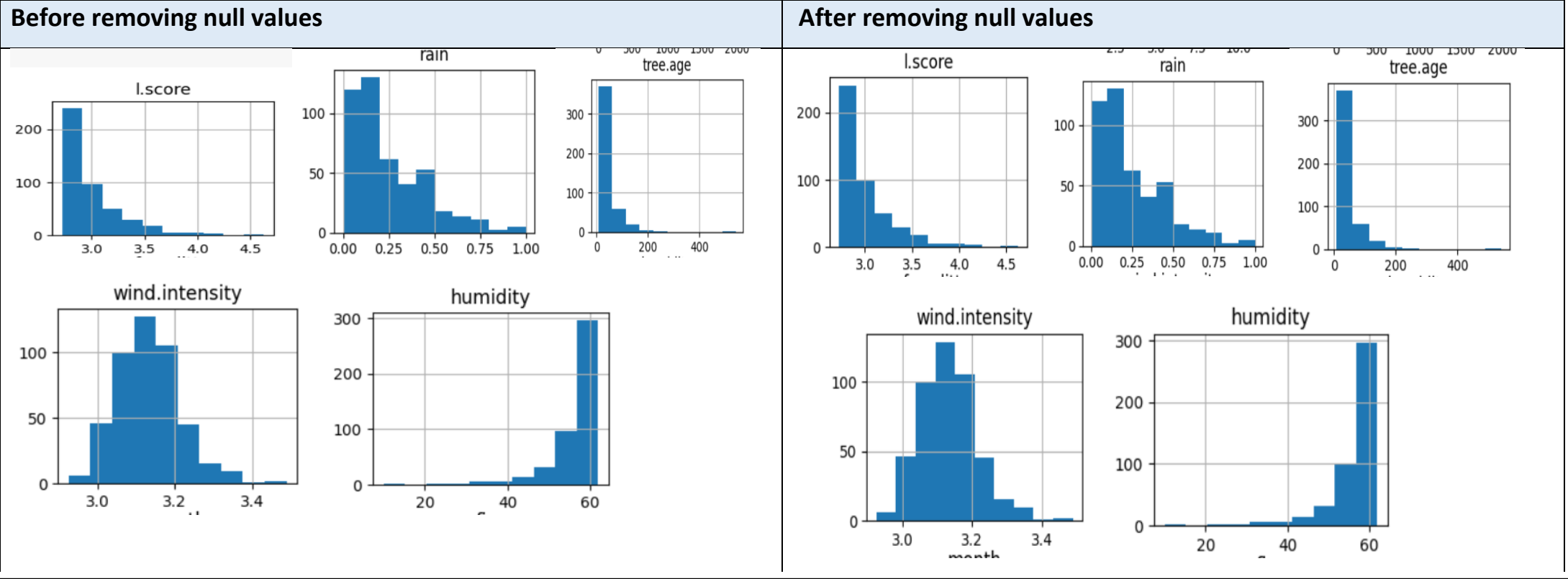
Different variables are used for loading, splitting, and mention other process of project:

- df: was use to save the loaded dataset(forestdata.csv).
- X: Independent variables (features) used for training and testing the models.
- y: dependent variable (target) used for training and testing the models.
- X_train, y_train: split training data stored in this variables.
- X_test, y_test: split test data stored in both variables.
- X_train, X_val, y_train, y_val: Further split the training set into training and validation sets for model selection.
- LR_model: Logistic Regression model save in this variable.
- DTmodel: used to store the instance Decision Tree Classifier model.
- NN_model: MLP Classifier model instance save in NN_model.
- trainAcc, : used this variable to store the training accuracy of each model one-by-one.
- validationACC: this variable is used to store the validation accuracy of each model.
- testAcc: similarly of trainAcc variable is used to store the testing accuracy of models.
- Precision: precision variable is used to store the precisions.
- F1: save the f1-score of all models that used in the project.
- Cm : cm variable is used to display the confusion matrix of the logestic regression model.
- All features (collector_id, c_score, l_score, rain, tree_age, surface_litter, wind_intensity, humidity, tree_density, month, time_of_day:): all features are used to save the input taken by the user.

4)Data Preparation:

The `isnull().sum()` method identified missing values in each column. The `fillna()` method replaced missing values with the mean value of the respective column.

The `time.of.day` column was initially a categorical variable with values like 'morni7ng', 'afternoon', 'night', and 'morning'. First replace the 'morni7ng' into ‘morning’ because it has wrong spellings. To make all process suitable for numerical analysis, the values were replaced with numerical labels (1 for afternoon, 2 for morning, and 3 for night). Different column contain null values l.score(1), rain(1), tree.age(1), wind.intensity(1), humidity(3).this columns are contain null values and there counts are mentioned.



5)Model training and Hyper Parameters:

Model	Hyperparameters	Accuracy	F1 Score	Precision
MLP Classifier	hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, max_iter=200	0.85	0.82	0.85
Decision Tree	criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1	0.88	0.86	0.85
Logistic Regression	penalty='l2', C=1.0, solver='lbfgs'	0.85	0.82	0.85

6) Final Model and results:

The choice of the final model to test depends on several factors, including the accuracy, precision, and F1 score of each model. In this case, the Decision Tree model has the highest accuracy, precision, and F1 score, making it the best choice for testing but select logestic regression model as a final model because the training accuracy of the decession tree model is 100% and it may overfit the model . The model was fit to the training data using the `fit()` method, which optimizes the model parameters to minimize the prediction error. The hyperparameters of the model were not explicitly tuned in this example, but they can be adjusted to improve the model's performance further.

