FOREST COVER PREDICTION

Prediction the type of forest cover using data analysis for a 30m x 30m patch of land in the forest

Internship ID: UMIP272882
Name: Yashwanth Reddy Kuchipudi

Dataset Description

This dataset is an analysis dataset from the forest department performed in the Roosevelt National Forest of northern Colorado.

Integer Classification of the forest cover types:

- Spruce/Fir
- Lodgepole Pine
- Ponderosa Pine
- Cottonwood/Willow
- Aspen
- Douglas-fir
- Krummholz

Approach

- Exploration of dataset
- Data Preprocessing
- Train- Test dataset split
- XGBClassifier & Random Forest model initialization and training
- Evaluation
- Hyper parameter tuning
- Comparison of results from both the models

Exploration of dataset

- The dataset was clean without any null or Nan values. So there was not much data pre-processing was needed.
- Anyway, feature scaling was done using StandardScaler to normalize the data

Train- Test dataset split

80-20 ration dataset split is opted for better training and testing of the model

XGBClassifier

- Its an advanced gradient boosting algorithm that builds trees sequentially, with each new tree correcting errors from the previous ones.

Why I chose this:

- works exceptionally well when features interact in non-linear ways
- It’s faster and more efficient than plain gradient boosting methods.

The no of rounds I chose was 100 as n_estimators and mlogloss as evalmetrics

Random Forest

- It's an ensemble learning method that builds multiple decision trees during training and outputs the mode (most common) class for classification tasks.
- As the dataset contains binary data and continuous, it's a good to select random forest as it works well with mixed data and reduce over fitting.

- No of trees was set to 100 as n_estimators which is the only parameter to the model along with the standard random state as 42.

Initial Test Results

The initial test results are as follows:

- XGBClassifier: 87.43
- Random Forest: 86.01

Hyperparameter tuning and model accuracy improvement for XGBClassifier

- To find out the best parameters which contributes more towards accurate prediction, Grid Search algorithm was used.
- A few other hyperparameters as also changed like max_depth, learning_rate, n_estimators, etc.
- With the better parameters, a slight improvement was seen in the model from 87.43 to 88.26

Hyperparameter tuning and model accuracy improvement for Random Forest

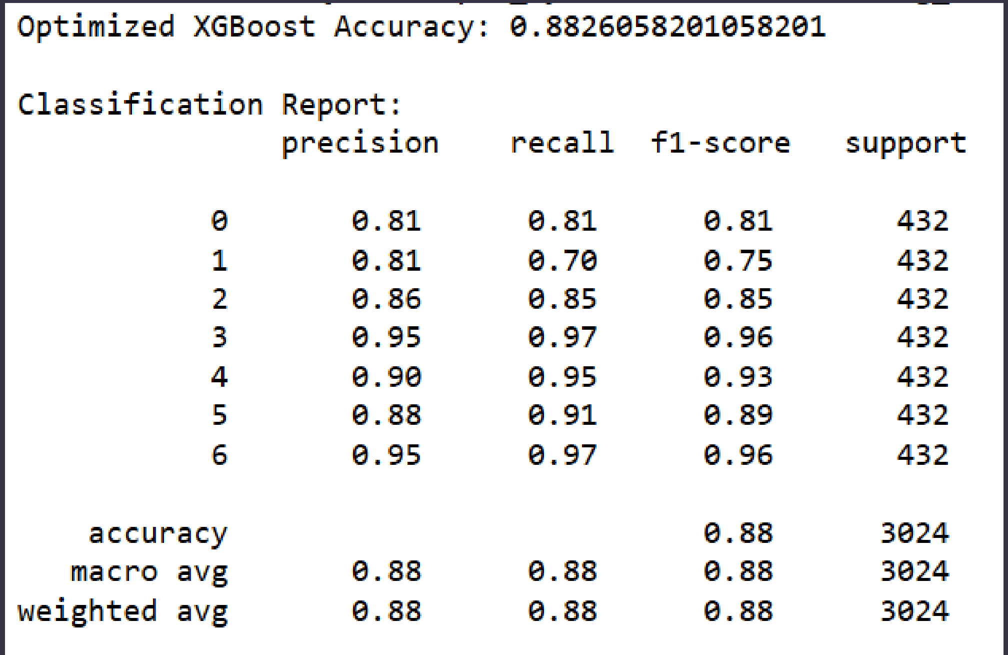
- Same was applied to Random Forest model too. parameters like n_estimators, max_features, min_samples_leaf, etc were changed.
- The model was optimized from 86.01 to 86.97

Other optimizing methods- Feature Extraction

- Feature extraction was also tried but when best feature say 10-15 are chosen from the data, the performance was drastically reduced to around 66%.
- Even larger no of best parameters were also tried 25-30 but of no use.
- This method was removed as it showing poor results.

Results Comparison (Best Performance)

XGBClassifier (Best Model)



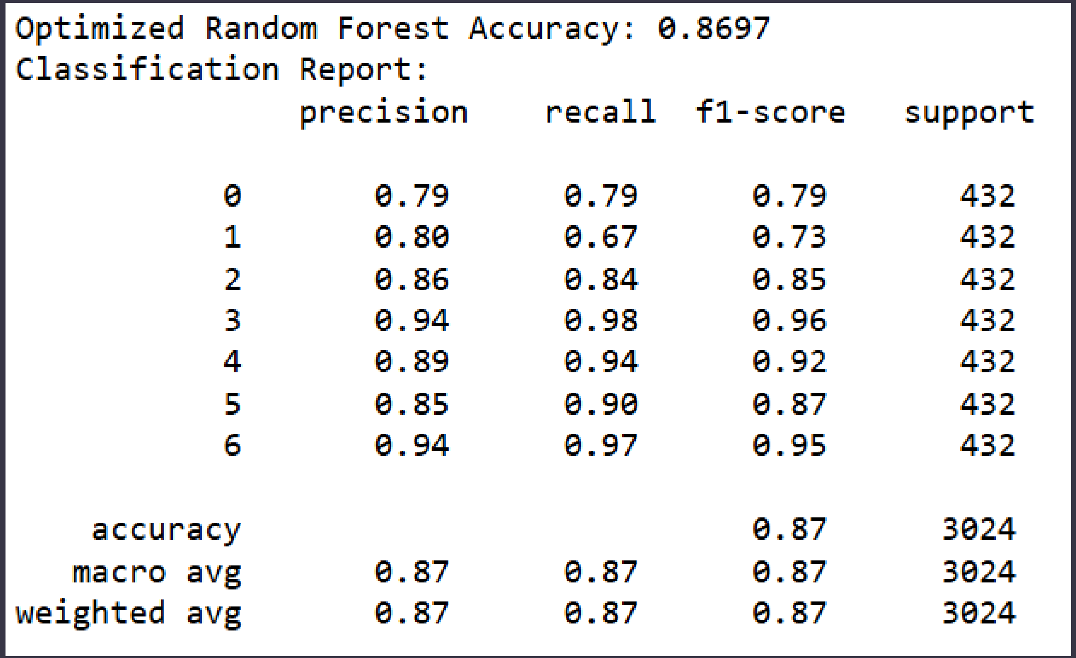
```
Optimized XGBoost Accuracy: 0.8826058201058201

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.81      0.81       432
           1       0.81      0.70      0.75       432
           2       0.86      0.85      0.85       432
           3       0.95      0.97      0.96       432
           4       0.90      0.95      0.93       432
           5       0.88      0.91      0.89       432
           6       0.95      0.97      0.96       432

    accuracy                           0.88      3024
   macro avg       0.88      0.88      0.88      3024
weighted avg       0.88      0.88      0.88      3024
```

Random Forest



```
Optimized Random Forest Accuracy: 0.8697
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.79      0.79       432
           1       0.80      0.67      0.73       432
           2       0.86      0.84      0.85       432
           3       0.94      0.98      0.96       432
           4       0.89      0.94      0.92       432
           5       0.85      0.90      0.87       432
           6       0.94      0.97      0.95       432

    accuracy                           0.87      3024
   macro avg       0.87      0.87      0.87      3024
weighted avg       0.87      0.87      0.87      3024
```

Results Comparison (Best Performance)

XGBClassifier (Best Model)

A pretty good classification was performed by this model and it’s a better one compared to the latter.

Forest classes with good precision and recall scores were:

- 4 - Cottonwood/Willow
- 5 – Aspen
- 7 - Krummholz

Random Forest

A slightly low performance by this model but a reasonable score is predicted.

Forest classes with good precision and recall scores were:

- 4 - Cottonwood/Willow
- 7 - Krummholz