19-Extra_Trees_Classifier

October 20, 2024

1 Extra Tress Classifier

Extra Trees, short for Extremely Randomized Trees, is an ensemble learning technique that builds a collection of decision trees in a random manner. Like Random Forest, it constructs multiple decision trees, but it introduces additional randomness into the tree-building process. This results in an ensemble method that typically provides better performance and is less prone to overfitting than traditional decision trees.

Extra Trees can be used for both classification and regression tasks. The key idea is to leverage the randomness in both the selection of features and the thresholds for splitting nodes to create a diverse set of trees.

Extra Trees Classifier is a powerful and efficient ensemble method that excels in classification tasks. Its ability to introduce randomness while leveraging the strengths of decision trees makes it a valuable tool in the machine learning toolkit.

1.0.1 When to Use Extra Trees Classifier?

Extra Trees Classifier is particularly useful when:

- You want to improve the performance of a single decision tree or even Random Forest by introducing more randomness.
- Your dataset is large and complex, and you are looking for a robust model that can capture intricate patterns.
- You are interested in reducing overfitting compared to other tree-based models.
- You need an efficient model for classification or regression tasks that can handle high-dimensional data.

1.0.2 How Does Extra Trees Classifier Work?

The Extra Trees Classifier builds decision trees using the following steps:

1. **Bootstrapping**:
   - Unlike Random Forest, which samples the training data with replacement, Extra Trees uses the entire dataset for each tree, ensuring all data points are considered.
2. **Random Feature Selection:**
   - For each node in a tree, a random subset of features is selected. Instead of considering all features, the model randomly selects a subset of features to determine the best split.

3. **Threshold Selection**:
   - Instead of using the best split point based on the optimization criterion (like Gini impurity or information gain), Extra Trees selects the split threshold randomly from the possible values for the selected features. This means that the trees are built using more randomness, hence the name "Extremely Randomized."
4. **Tree Construction**:
   - The process of splitting nodes continues recursively until a stopping criterion is reached (e.g., maximum depth, minimum samples per leaf).
5. **Prediction**:
   - For classification tasks, predictions are made based on the majority vote of all the trees in the ensemble. For regression tasks, the average of the predictions from all trees is used.

1.0.3 Who Should Use Extra Trees Classifier?

- **Data scientists and machine learning engineers**: Who want to build robust classification models that leverage tree ensembles.
- **Kaggle competitors**: Extra Trees is often used in competitions for its high accuracy and efficiency.
- **Industries requiring fast and accurate models**: Suitable for applications in finance, healthcare, and customer analytics, where quick and reliable predictions are needed.

Advantages of Extra Trees Classifier:

- **High accuracy**: Typically provides better accuracy than single decision trees and sometimes outperforms Random Forest due to increased randomness.
- **Less prone to overfitting**: The additional randomness in feature selection and threshold determination helps reduce overfitting.
- **Fast training time**: Since all data points are used without bootstrapping and splits are chosen randomly, Extra Trees can train faster than other tree-based methods.
- **Feature importance**: Like other tree-based models, Extra Trees can provide insights into feature importance.

Disadvantages of Extra Trees Classifier:

- **Interpretability**: While feature importance is available, the ensemble nature makes it harder to interpret compared to a single decision tree.
- **Sensitivity to noise**: Although less than traditional decision trees, Extra Trees can still be sensitive to noise and outliers in the data.
- **Limited extrapolation**: The predictions can be less reliable for unseen data outside the range of the training data.

1.0.4 Real-World Applications of Extra Trees Classifier:

- **Credit scoring**: To assess risk based on various financial indicators.

- **Image classification**: Used in computer vision tasks to classify images based on features extracted from pixel data.
- **Customer segmentation**: For marketing strategies and customer behavior analysis.
- **Medical diagnosis**: In predicting patient outcomes based on clinical data.

[7]:
```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
from sklearn.ensemble import ExtraTreesClassifier

X, y = make_classification(n_features=11, random_state=21)

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,␣
 ↪random_state=16)

etc = ExtraTreesClassifier(random_state=21)
etc.fit(X_train, y_train)
```

[7]:
```
ExtraTreesClassifier(random_state=21)
```

[5]:
```
cross_val_score(etc, X_train, y_train, scoring='accuracy', cv=5, n_jobs=-1).
 ↪mean()
```

[5]:
```
0.9375
```

[8]:
```
param_grid = {
    'criterion': ['gini', 'entropy'],
    'n_estimators': [100, 250, 500],
    'min_samples_leaf': [5,15,25],
    'max_features': [3,5,7,9,11]
}

etc2 = GridSearchCV(etc,param_grid, cv=3, n_jobs=-1)

etc2.fit(X_train, y_train)
```

[8]:
```
GridSearchCV(cv=3, estimator=ExtraTreesClassifier(random_state=21), n_jobs=-1,
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_features': [3, 5, 7, 9, 11],
                         'min_samples_leaf': [5, 15, 25],
                         'n_estimators': [100, 250, 500]})
```

[10]:
```
etc2.best_params_
```

[10]:
```
{'criterion': 'gini',
 'max_features': 5,
 'min_samples_leaf': 5,
```

```
 'n_estimators': 500}
```

[12]:
```
etc2.best_score_
```

[12]:
```
0.9377967711301044
```