Module-04, Python for Machine Learning

Classification Algorithms (Random Forest)

Dostdar Ali
Instructor

Data science and Artificial Intelligence
3-Months Course
at
Karakaroum international Univrsity

February 11, 2024

Table of Contents

Random Forest

- **Definition:** Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes.
- **History:** Proposed by Leo Breiman in 2001. Improved upon the overfitting issues of individual decision trees.
- **Working Principle:** Constructs a collection of decision trees and combines their predictions to improve accuracy and control overfitting.
- **Examples:**
  Image classification.
  Predictive maintenance in manufacturing.

Random Forest

- **Definition:** Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes.
- **History:** Proposed by Leo Breiman in 2001. Improved upon the overfitting issues of individual decision trees.
- **Working Principle:** Constructs a collection of decision trees and combines their predictions to improve accuracy and control overfitting.
- **Examples:**
  Image classification.
  Predictive maintenance in manufacturing.

Random Forest

- **Definition:** Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes.
- **History:** Proposed by Leo Breiman in 2001. Improved upon the overfitting issues of individual decision trees.
- **Working Principle:** Constructs a collection of decision trees and combines their predictions to improve accuracy and control overfitting.
- **Examples:**
  Image classification.
  Predictive maintenance in manufacturing.

Random Forest

- **Definition:** Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes.
- **History:** Proposed by Leo Breiman in 2001. Improved upon the overfitting issues of individual decision trees.
- **Working Principle:** Constructs a collection of decision trees and combines their predictions to improve accuracy and control overfitting.
- **Examples:**
  Image classification.
  Predictive maintenance in manufacturing.

Random Forest: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Prediction:**

$$\text{Prediction}(X) = \frac{1}{N} \sum_{i=1}^{N} \text{Prediction}_{T_i}(X)$$

where $T_1, T_2, \ldots, T_N$ are individual decision trees in the forest.
- **Training:** Bootstrap samples and use a random subset of features at each split for each tree.

Random Forest: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Prediction:**

$$\text{Prediction}(X) = \frac{1}{N} \sum_{i=1}^{N} \text{Prediction}_{T_i}(X)$$

where $T_1, T_2, \ldots, T_N$ are individual decision trees in the forest.

- **Training:** Bootstrap samples and use a random subset of features at each split for each tree.

Random Forest: Mathematical Formulation

- **Problem:** Binary classification problem with features $X = \{X_1, X_2, \ldots, X_n\}$.
- **Prediction:**

$$\text{Prediction}(X) = \frac{1}{N}\sum_{i=1}^{N} \text{Prediction}_{T_i}(X)$$

where T_1, $T_2, \ldots,$ T_N are individual decision trees in the forest.
- **Training:** Bootstrap samples and use a random subset of features at each split for each tree.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

Random Forest Example

- **Problem:** Predict whether a loan application will be approved based on various features.
- **Working Principle:** Ensemble method combining multiple decision trees.
- **Training:** Train several decision trees on different subsets of the data and features.
- **Example:**
  - Train Tree 1 on a random subset of data and features.
  - Train Tree 2 on another random subset.
  - ...
- **Prediction:** Aggregate predictions from all trees (e.g., voting for classification).
- **Advantages:** Reduces overfitting, improves accuracy, handles missing data well.

- **Problem:** Binary classification based on a single feature (X).
- **Data:**

Example	Feature X
1	3
2	4
3	2
4	5
5	1

- **Random Forest:**

Decision Tree	Prediction
T_1: $X \leq 3$	0
T_2: $X > 3$	1

- **Final Prediction:**

$$\text{Prediction} = \frac{1}{2}(\text{Prediction}_{T_1} + \text{Prediction}_{T_2})$$

Random Forest: Hand Calculations

- **Problem:** Binary classification based on a single feature (X).
- **Data:**

Example	Feature X
1	3
2	4
3	2
4	5
5	1

- **Random Forest:**

Decision Tree	Prediction
T_1: $X \leq 3$	0
T_2: $X > 3$	1

- **Final Prediction:**

$$\text{Prediction} = \frac{1}{2}(\text{Prediction}_{T_1} + \text{Prediction}_{T_2})$$

- **Problem:** Binary classification based on a single feature (X).
- **Data:**

Example	Feature X
1	3
2	4
3	2
4	5
5	1

- **Random Forest:**

Decision Tree	Prediction
T_1: $X \leq 3$	0
T_2: $X > 3$	1

- **Final Prediction:**

$$\text{Prediction} = \frac{1}{2}\left(\text{Prediction}_{T_1} + \text{Prediction}_{T_2}\right)$$

Random Forest: Hand Calculations

- **Problem:** Binary classification based on a single feature (X).
- **Data:**

Example	Feature X
1	3
2	4
3	2
4	5
5	1

- **Random Forest:**

Decision Tree	Prediction
T_1: $X \leq 3$	0
T_2: $X > 3$	1

- **Final Prediction:**

$$\text{Prediction} = \frac{1}{2}\left(\text{Prediction}_{T_1} + \text{Prediction}_{T_2}\right)$$

Python Code: Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assuming 'X' is feature matrix and 'y' is target variable
X_train, X_test, y_train, y_test = ...
...train_test_split(X, y, test_size=0.2)

# Creating and training the model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Making predictions
predictions = model.predict(X_test)
# Evaluating accuracy
accuracy = accuracy_score(y_test, predictions)
```

Great Job
Thank yo