Random Forest

By:
Kylee LaPierre, Dennis Kelly, Beth Vander Hoek, & Stephanie Levia

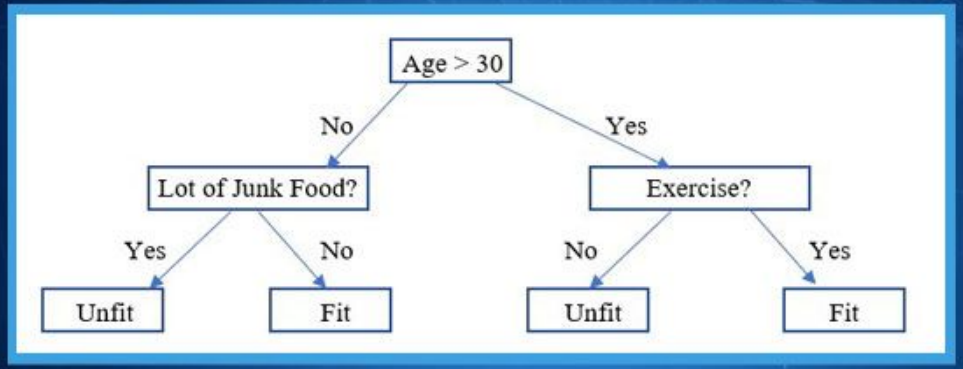
What is Random Forest?

- **Supervised, non-parametric** (parameters are adjustable AND can change) algorithm
- Can be **used for either classification or regression**, though the former seems more common
- **Ensemble method**, which means its composed of many smaller, less accurate, and most importantly, independent learners. In our case these are decision trees
- Prediction of the random forest is aggregated prediction of all constituent decision trees - “**wisdom of crowds**”

How does the Random Forest model work?

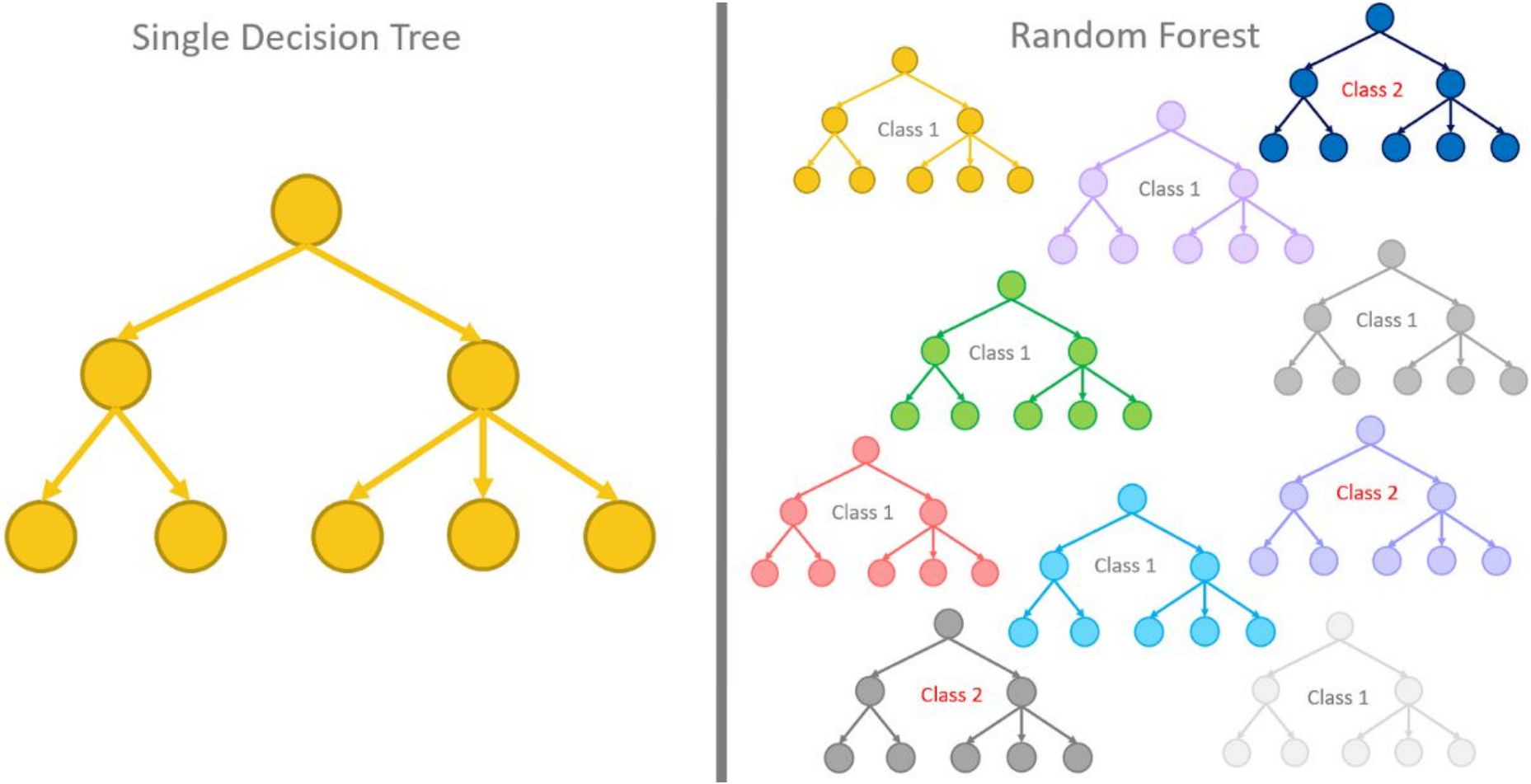
- Random Forest is just a number of *randomly created, different decision tree models*
  - Like many trees in a forest
- Each decision tree model makes a prediction
- Majority rules! Either mode (categorical) or mean (continuous) for overall Random Forest predicted result

What is Decision Tree?



www.educba.com

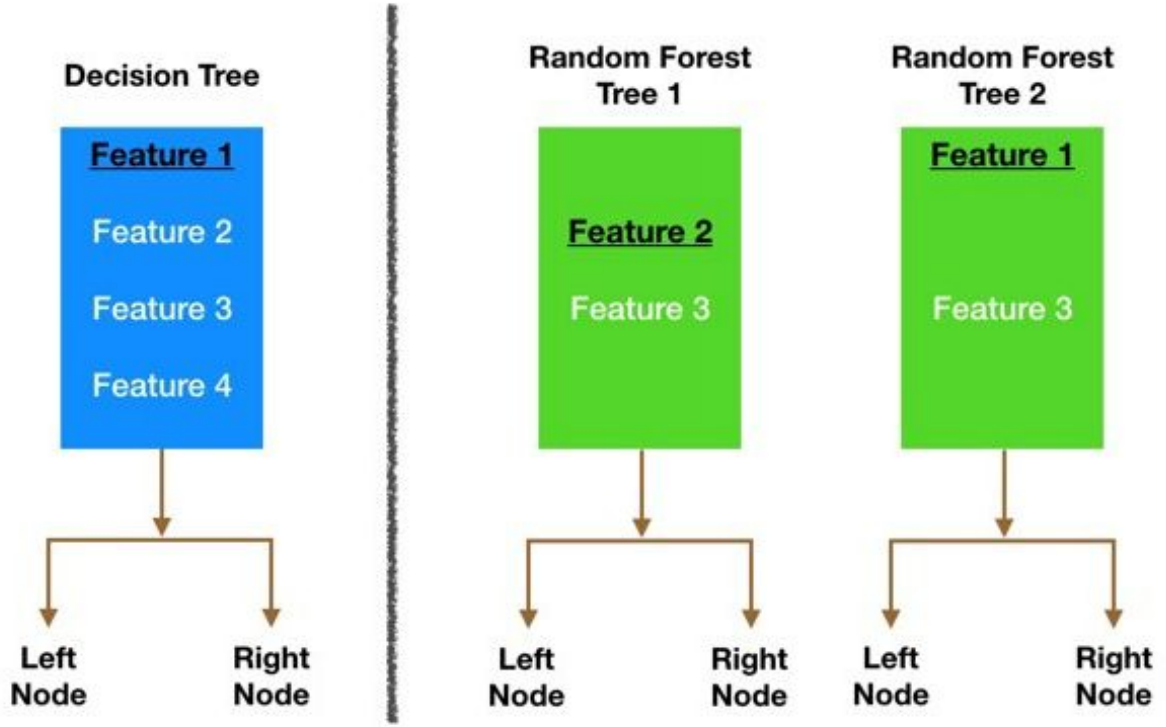
Source: Ghosal (2022). “What is Decision Tree?”, *EDUCBA*. https://www.educba.com/what-is-decision-tree/



Source: Silipo & Melcher (2019).“From a Single Decision Tree to a Random Forest”, *Towards Data Science.*
https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147

Random Forest is a number of ***randomly*** created, ***different*** decision tree models

- ***Randomness*** **leads to** ***different trees:***
  - "Bagging"(bootstrap agg.): each decision tree model takes random subset/sample of data, uses replacement (some data repeats)
    - Ex. #1: [1, 2, 3, 4] → [1, 1, 3, 3], #2: [1, 2, 3, 4] → [1, 2, 4, 4], etc.
  - 'Feature Randomness': can only divide data at each step using one of features to divide by from random subset(NOT all features)
    - Ex. All features to divide data by: gender, age, education, job
    - Features to choose from for tree #1, division #1: gender, job



Node splitting in a random forest model is based on a random subset of features for each tree.

Source: Yiu, Tony(2019).“Understanding Random Forest”, *Towards Data Science.* https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Hyperparameters

Main parameters:

n_estimators

The **number of trees in the forest**. The **larger the better**, but also the longer it will take to compute. In addition, note that results will stop getting significantly better beyond a critical number of trees.

Max_features

The **size of the random subsets of features to consider** when splitting a node. The smaller the subset, the greater the reduction of variance, but also the greater the increase in bias.

Other parameters:

min_samples_split

min_samples_leaf

max_depth

max_leaf_nodes

max_samples

bootstrap

Advantages

Some advantages to using Random Forest include:

- The algorithm is **less biased** since there are multiple trees that are each trained on a subset of data.
  - Although more biased than 1 decision tree
- The algorithm is very **stable**.
- Works well with **both categorical and numerical** features.

Disadvantages

Some disadvantages to using Random Forest include:

- The **complexity** of using Random Forest.
- **Loss of interpretability** compared to simpler methods.
- Due to the complexity, they require **more time to train** compared to other algorithms.
- Impacted if there’s many outliers.

Classification Model Comparisons

#	Model	Data pre-processing		Impact from		Highlights
		Normalization	Scaling	Collinearity	Outliers	
1	Logistic Regression	Yes	No	Yes	Yes	• Highly descriptive with good accuracy • Reasonable computational requirements
2	Artificial Neural Networks	No	Yes	Yes	Yes	• High prediction accuracy • Self-extracts features • Heavy computational requirements for large datasets
3	Random Forest	No	No	No	Yes	• High prediction accuracy • Provides limited explainability • Works well with both continuous & categorical predictors
4	Naïve Bayes	NA	NA	Yes	Yes	• Applicable to categorical predictors only • Suitable for small train data
5	KNN	Yes	Yes	Yes	Yes	• Performs local approximation, no prediction formula

Data processing steps required

- **Remove** or **impute** missing values
- Decide whether or not to **drop** outliers, as these can influence the result
- That’s it! Random Forests are very accommodating!

Sample Code

```
#we are using diabetes dataset (used earlier in module for logistic regression exercise)
diabetes = pd.read_csv('diabetes.csv')
```

✓ 0.9s

```
#check to ensure no nulls in dataset (fill in/remove nas as needed)
#check to ensure all columns numeric or categorical (no unique string columns; categorical okay, no need for dummy)

d = diabetes
#no nas or unique string columns here so we're good to go!
```

✓ 0.3s

```
#split dataframe into independent (X) and dependent variables (y). Dependent variable is outcome

X = diabetes.drop('Outcome', axis=1).copy()

y = diabetes.Outcome.copy()
```

✓ 0.7s

```
#import sklearn train_test_split and create training and testing datasets by splitting up the data

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=0)
```

✓ 0.4s

```
#get Random Forest and train model

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

✓ 0.2s

```
RandomForestClassifier()
```

```
#check how well the model performed on dataset

model.score(X_test, y_test)
```

✓ 0.4s

```
0.8020833333333334
```

Appendix:

Resources

Video links

- https://www.youtube.com/watch?v=PHxYNGo8NcI&ab_channel=codebasics

  (video that explains tree decisions + code, the building block of random forest)

- https://www.youtube.com/watch?v=ok2s1vV9XW0&ab_channel=codebasics

  (helpful video to explain random forest + example code + how to change parameters to fine tune it by increasing # random tree models)

- https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

  (video series using random forest from start to finish)

Documentation links

- 1.11. Ensemble methods — scikit-learn 1.1.2 documentation

  (Scikit Learn documentation)

- ADAfaEPoV.pdf (cmu.edu)

  (Chapter 13 of this book is on Decision Trees, thorough but mathy)

- Random Forest Algorithm with Python and Scikit-Learn (stackabuse.com)

  (Advantages and disadvantages of using Random Forest)

- Hyperparameters of Random Forest Classifier - GeeksforGeeks

  (Hyperparameters that can be tuned to increase accuracy of the training model)

Examples and How-to guides

- Random Forest Regression in Python - GeeksforGeeks

  (Tutorial with sample code using Random Forest)

- Classification Models in Machine Learning | Classification Models (analyticsvidhya.com)

  (comparison of different classification models)

- Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning (freecodecamp.org)

- https://towardsdatascience.com/understanding-random-forest-58381e0602d2

  (How Random Forests works and why it's so effective)

Sample Code

- Sample codes using Random Forest models can be found in some the previous slides' articles and videos
  - This video and this article, for example
- Our created sample code can be found in GitHub repository as: sample-random-forest-code.ipynb
  - It uses diabetes.csv dataset from M09 Logistic Regression exercise