



CS 405/605 Data Science

Dr. Qianqian Tong

Random Forest

Recap Decision Tree

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

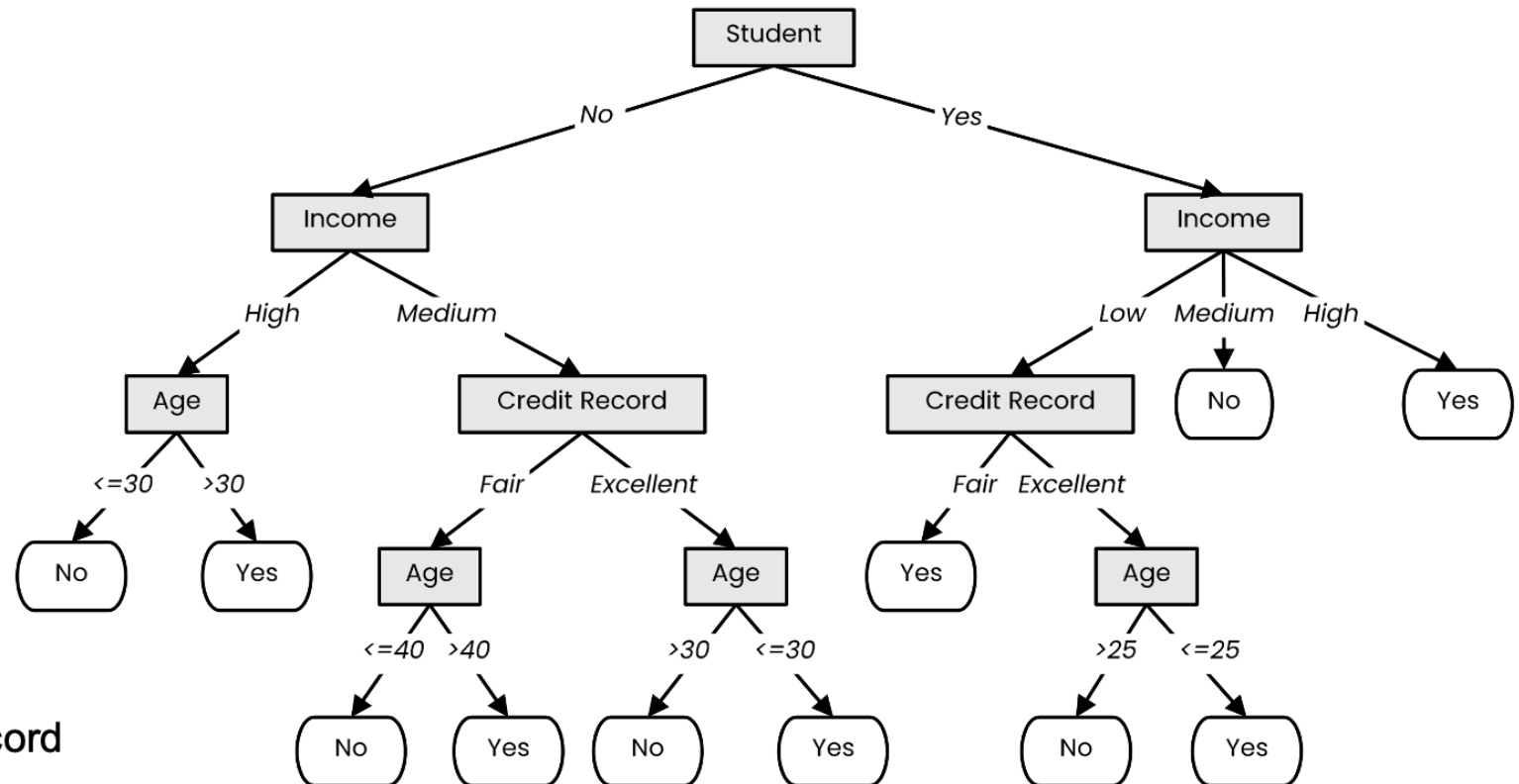
Who to loan?



- Not a student
- 45 years old
- Medium income
- Fair credit record



- Student
- 27 years old
- Low income
- Excellent credit record



Random Forest

Recap Decision Tree

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

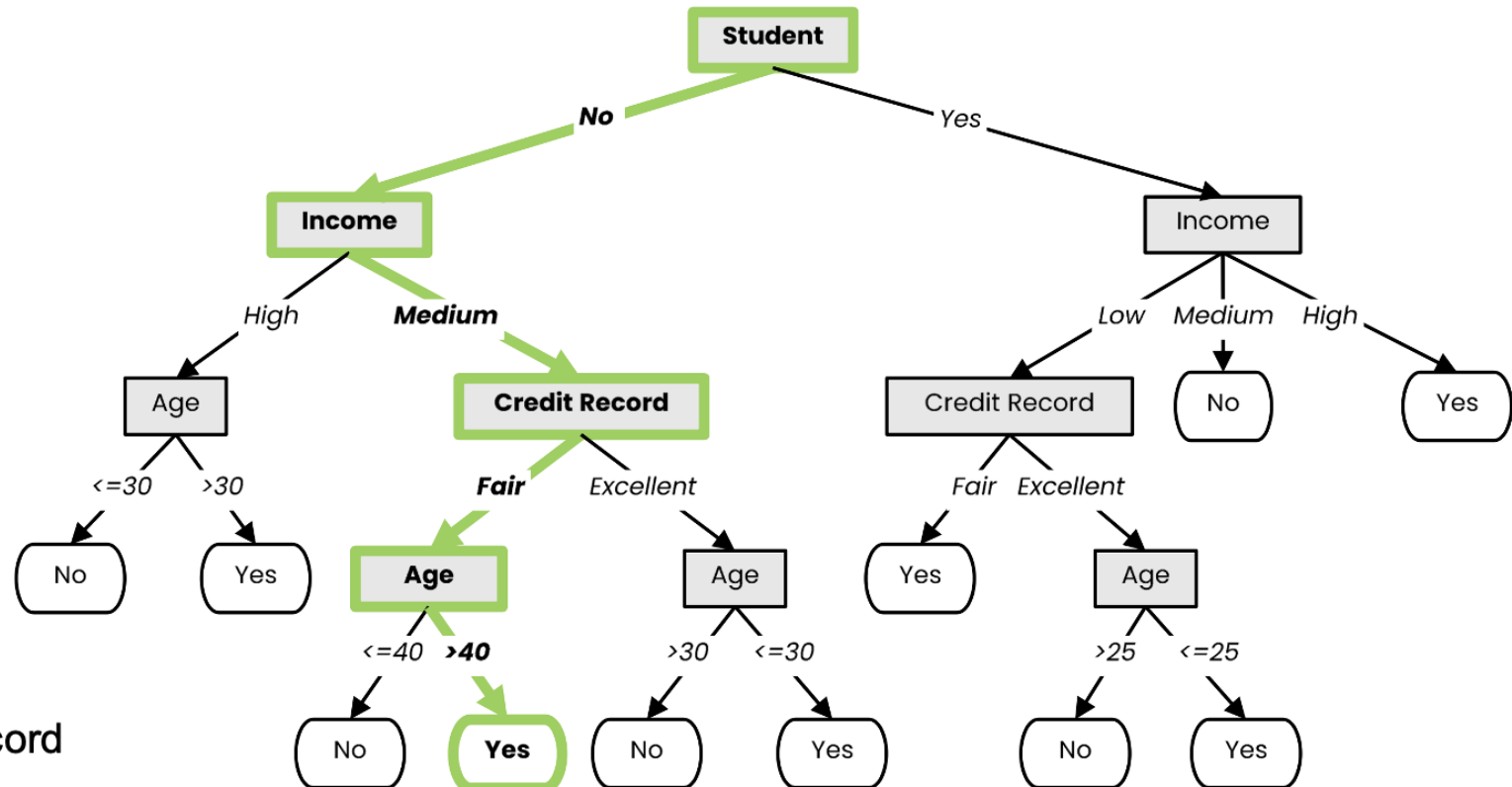
Who to loan?



- Not a student
- 45 years old
- Medium income
- Fair credit record
- Yes



- Student
- 27 years old
- Low income
- Excellent credit record



Random Forest

Recap Decision Tree

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

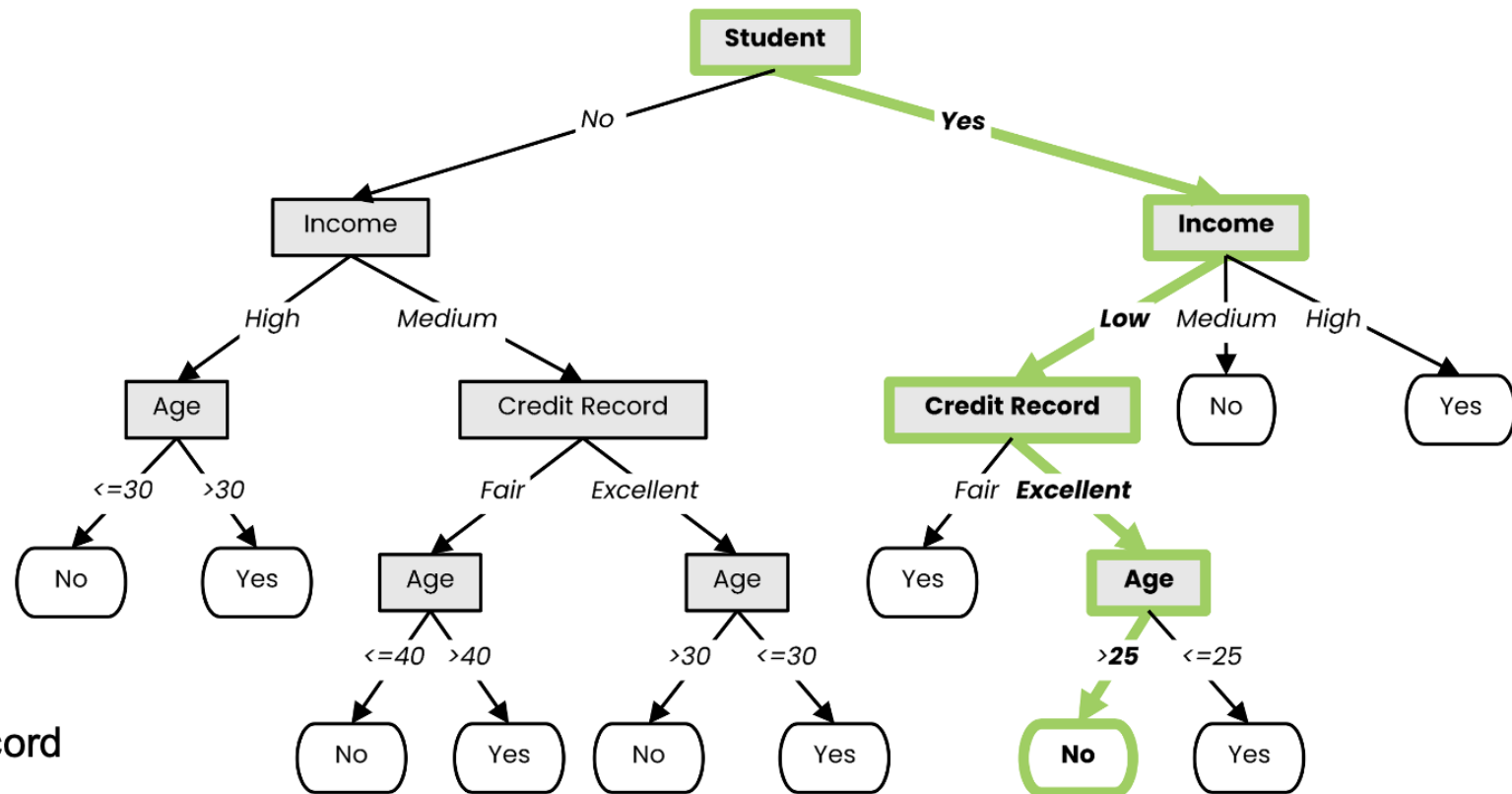
Who to loan?



- Not a student
 - 45 years old
 - Medium income
 - Fair credit record
- Yes



- Student
 - 27 years old
 - Low income
 - Excellent credit record
- No



Random Forest

Recap Decision Tree

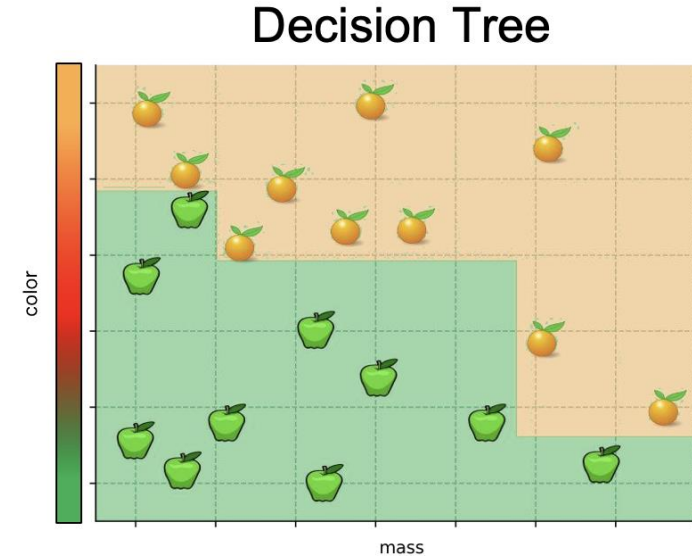
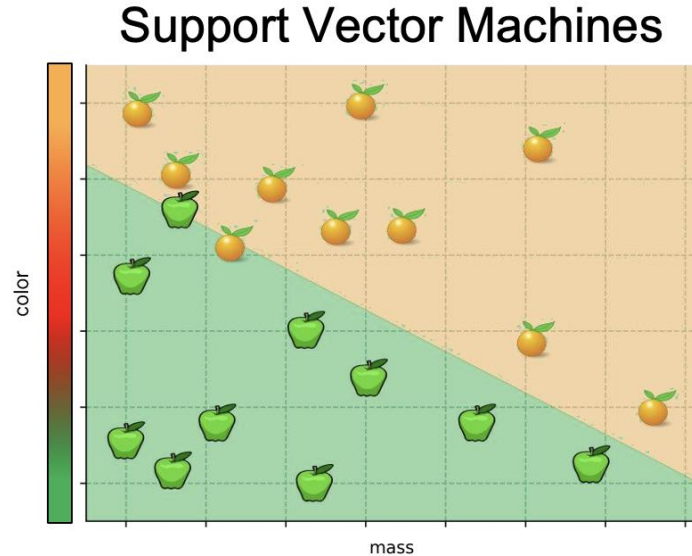
- Basic step: choose an attribute and, based on its values, split the data into smaller sets
 - Recursively repeat this step until we can surely decide the label
- We use labeled data to obtain a suitable decision tree for future predictions
 - We want a decision tree that works well on unseen data, while asking as few questions as possible

Random Forest

Recap Decision Tree

Decision Boundaries

- Decision trees produce non-linear decision boundaries



Random Forest

Recap Decision Tree

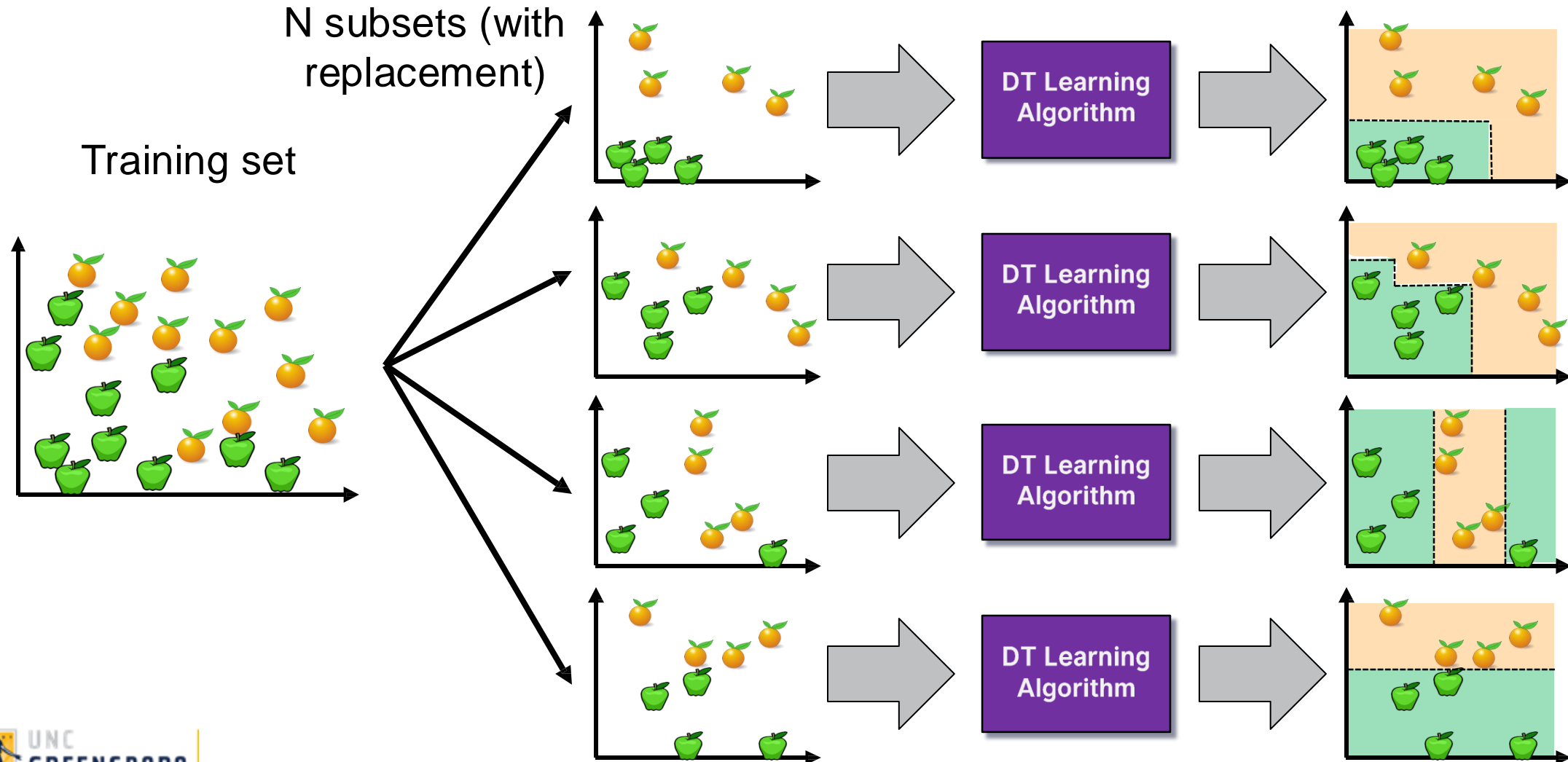
- Decision trees represent a tool based on a tree-like graph of decisions and their possible outcomes
- Decision tree learning is a machine learning method that employs a decision tree as a predictive model
- While decision trees classify quickly, the time for building a tree may be higher than another type of classifier
- Decision trees suffer from a problem of errors propagating throughout a tree
A very serious problem as the number of classes increases
- **Decision Trees have very high variance**

Random Forests (Ensemble learning with decision trees)

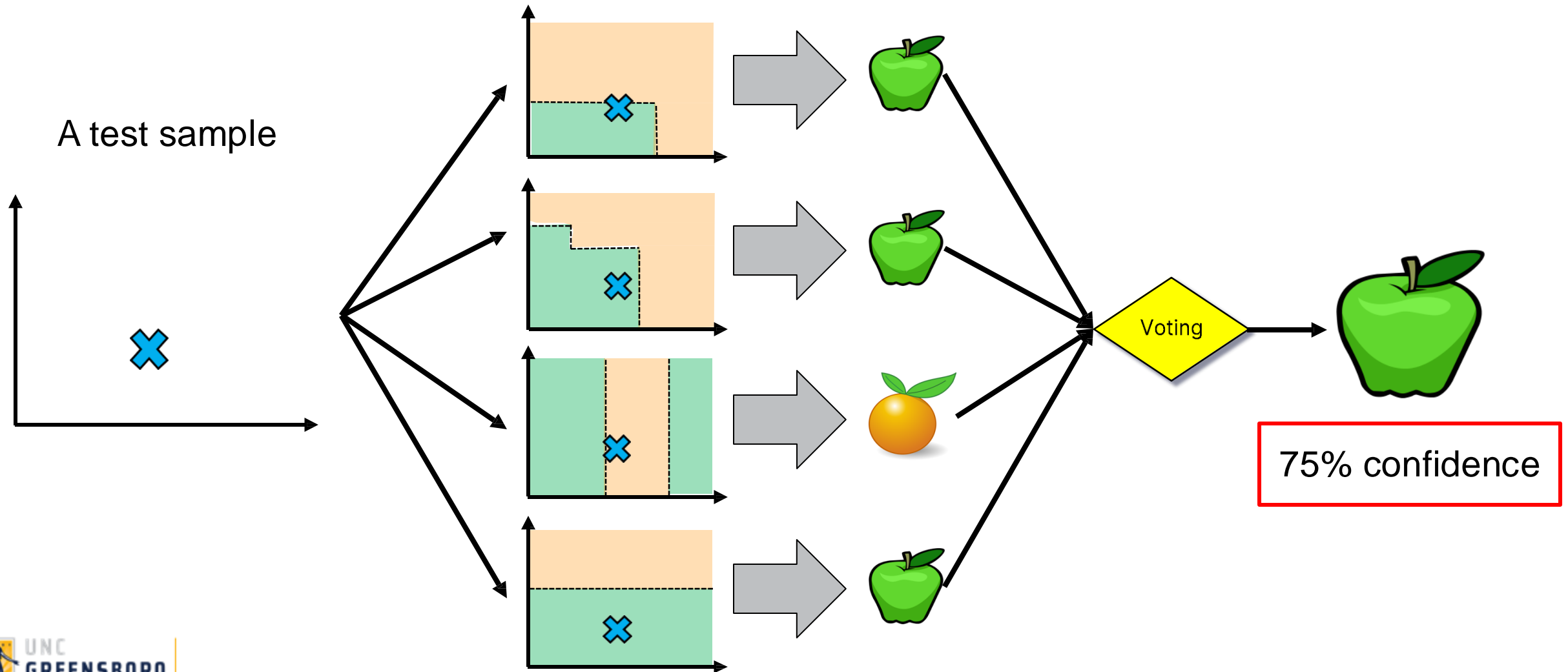
Random Forest

- Random Forests:
 - Instead of building a single decision tree and use it to make predictions, build many slightly different trees and combine their predictions
- We have a single data set, so how do we obtain slightly different trees?
 1. Bagging (**B**ootstrap **A**ggregating):
 - Take random subsets of data points from the training set to create N smaller data sets
 - Fit a decision tree on each subset
 2. Random Subspace Method (also known as Feature Bagging):
 - Fit N different decision trees by constraining each one to operate on a random subset of features

Bagging at training time



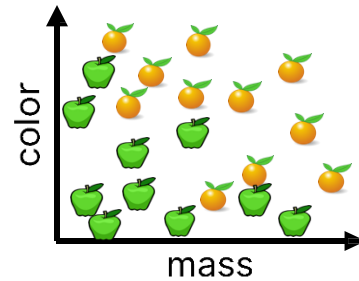
Bagging at inference time



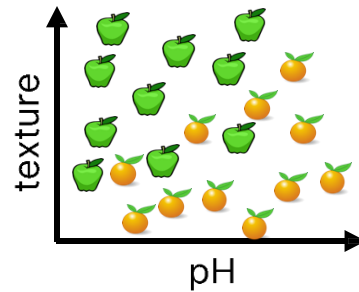
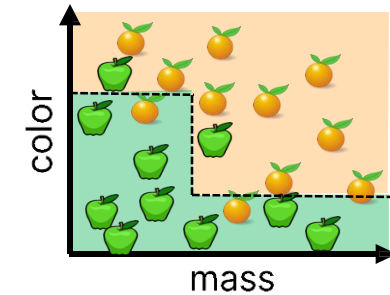
Random Subspace Method at training time

Training data

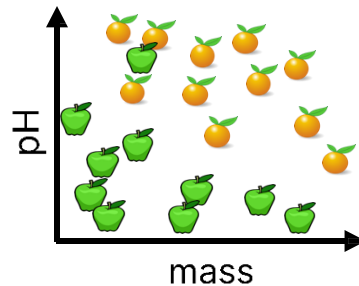
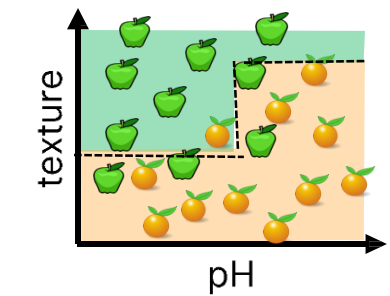
Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	Apple
121	Orange	Rough	3.9	Orange
85	Red	Smooth	3.3	Apple
101	Orange	Smooth	3.7	Orange
111	Green	Rough	3.5	Apple
...				
117	Red	Rough	3.4	Orange



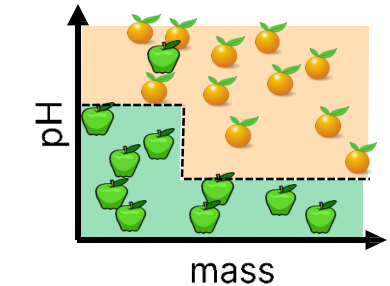
DT Learning Algorithm



DT Learning Algorithm



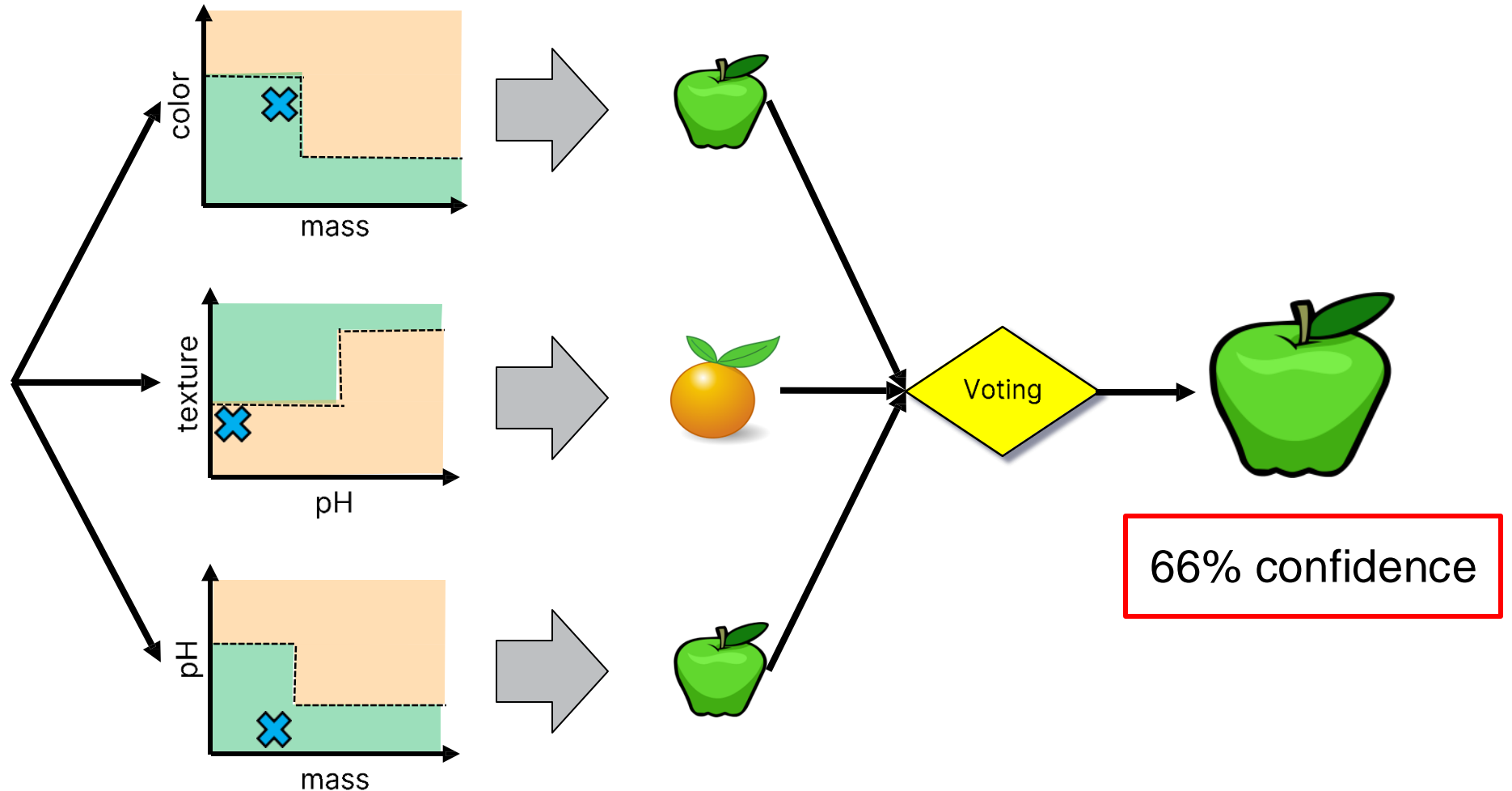
DT Learning Algorithm



Random Subspace Method at inference time

A test sample

87 Red Smooth 3.1



Random Forest

Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	Apple
121	Orange	Rough	3.9	Orange
85	Red	Smooth	3.3	Apple
101	Orange	Smooth	3.7	Orange
111	Green	Rough	3.5	Apple
...				
117	Red	Rough	3.4	Orange



Bagging +
Random Subspace Method +
Decision Tree Learning Algorithm



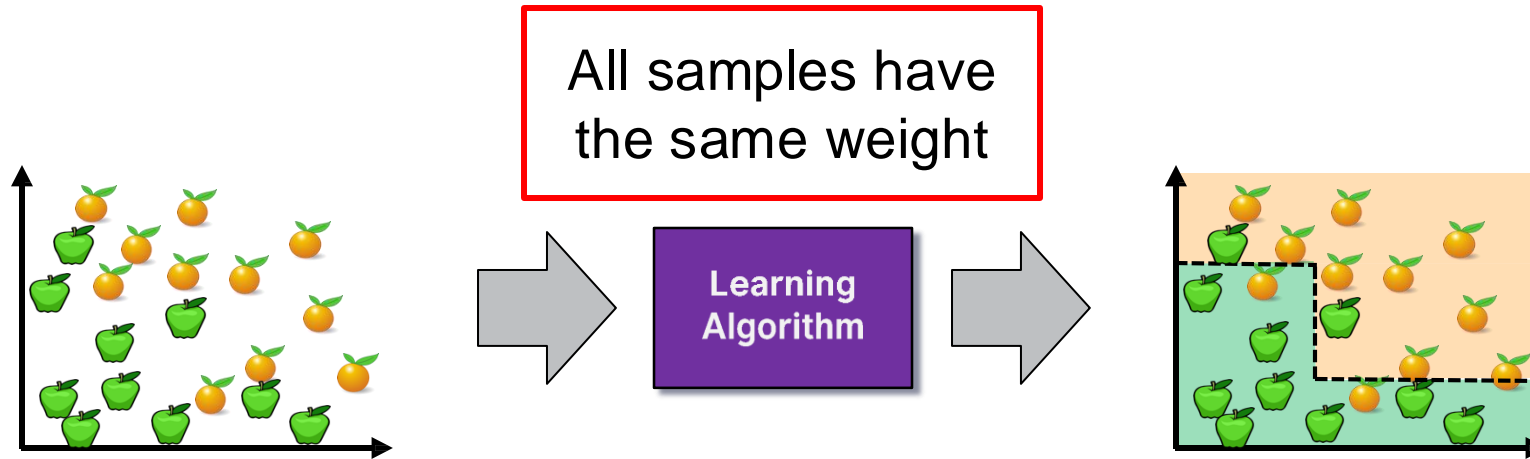
History of Random Forests

- Introduction of the Random Subspace Method
 - “Random Decision Forests” [Ho, 1995] and “The Random Subspace Method for Constructing Decision Forests” [Ho, 1998]
- Combined the Random Subspace Method with Bagging. Introduce the term **Random Forest** (a trademark of Leo Breiman and Adele Cutler)
 - “Random Forests” [Breiman, 2001]

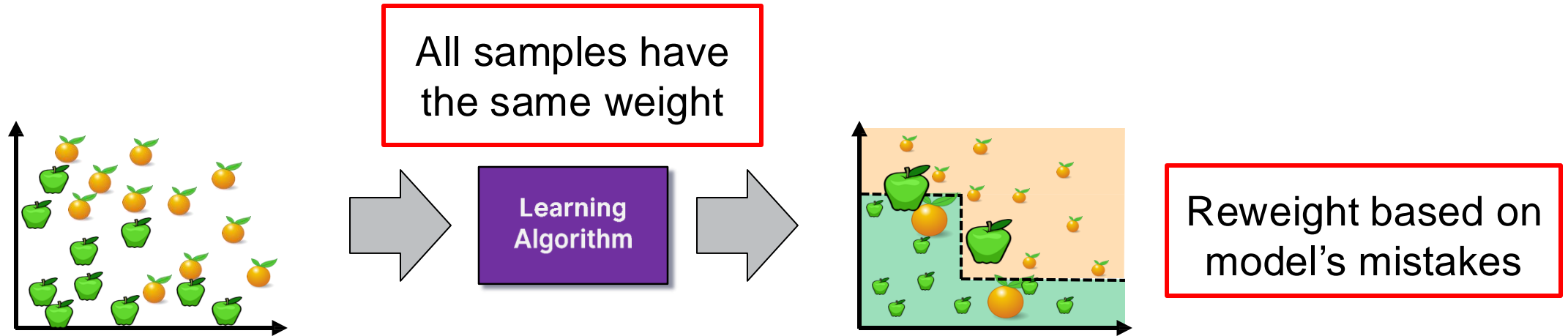
Ensemble Learning

- Ensemble Learning:
 - Method that combines multiple learning algorithms to obtain performance improvements over its components
- **Random Forests** are one of the most common examples of ensemble learning
- Other commonly-used ensemble methods:
 - **Bagging**: multiple models on random subsets of data samples
 - **Random Subspace Method**: multiple models on random subsets of features
 - **Boosting**: train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples

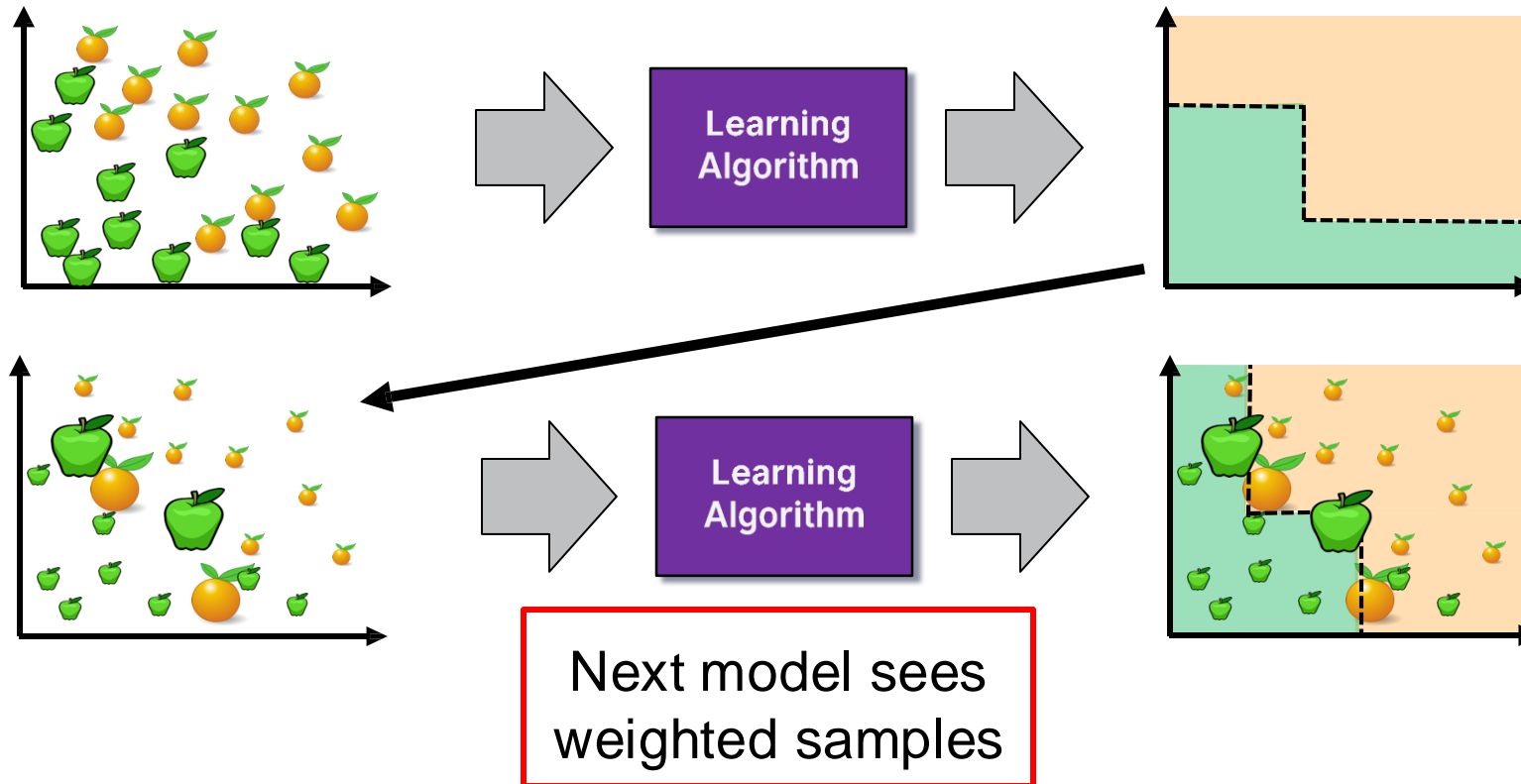
Boosting



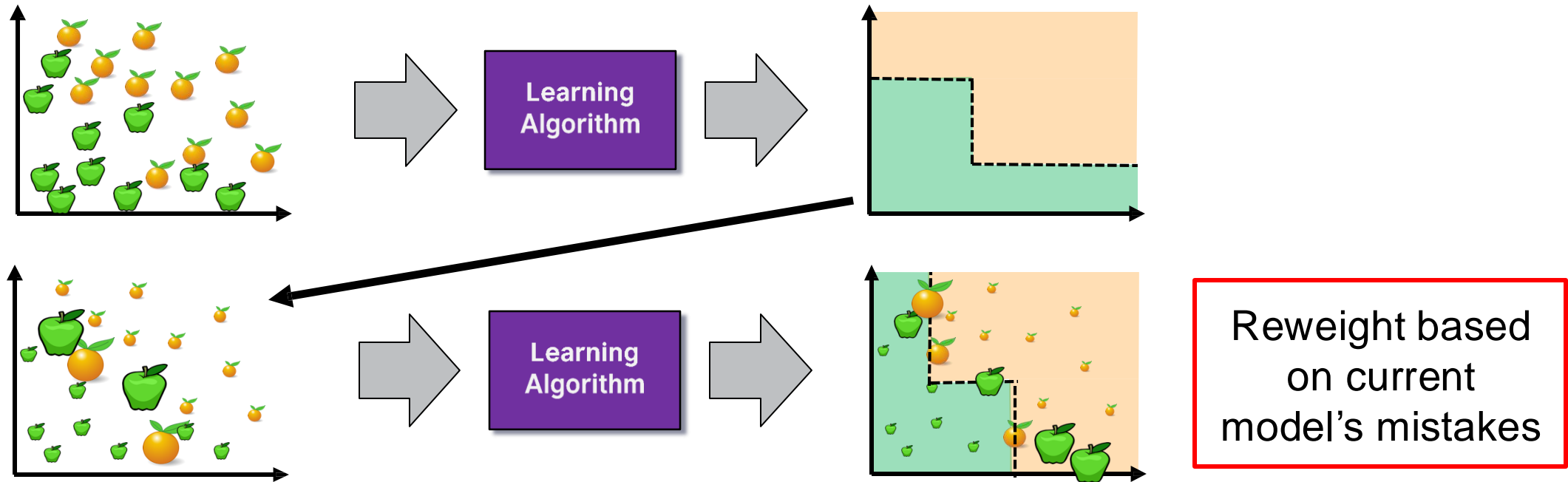
Boosting



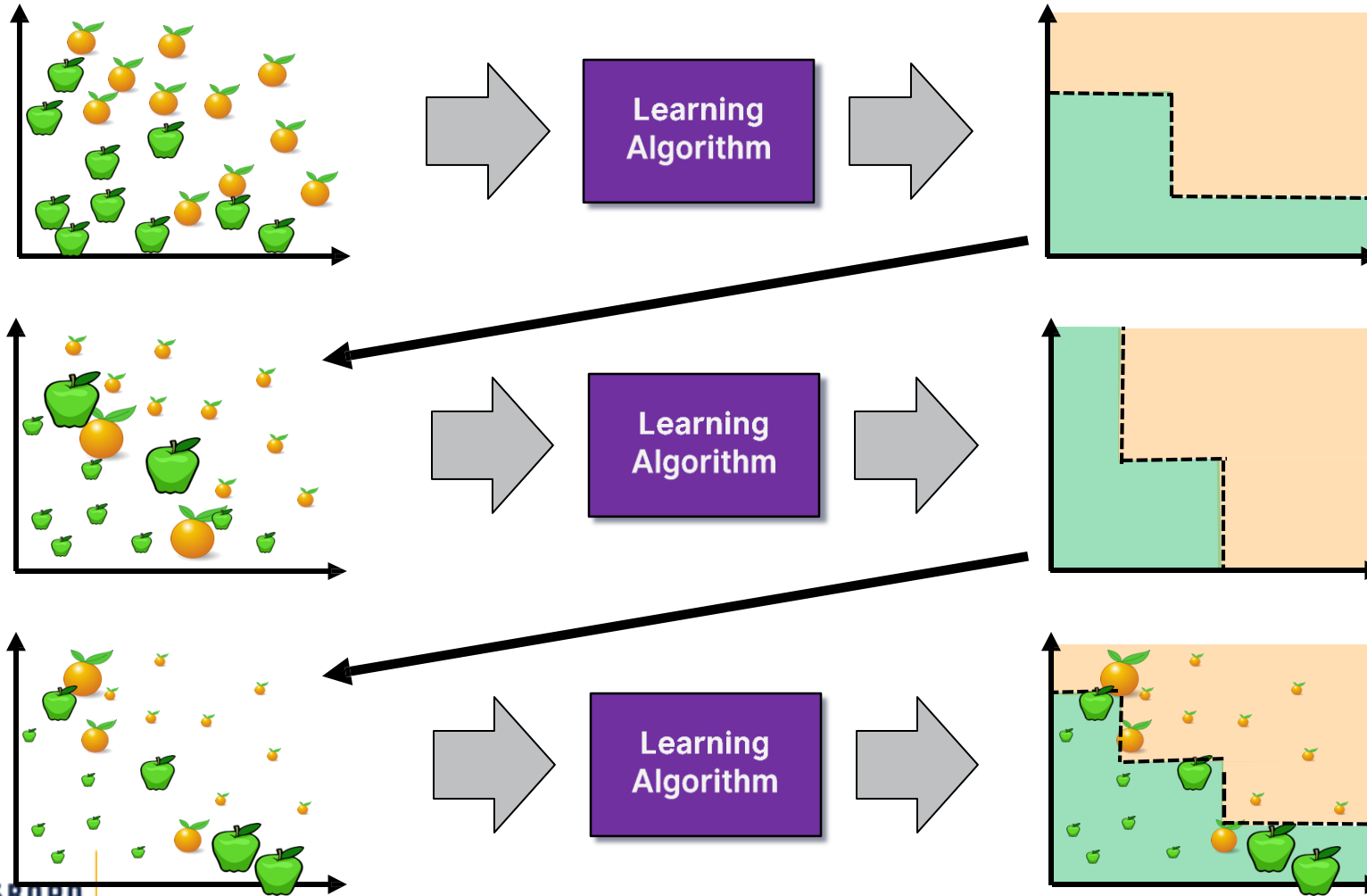
Boosting



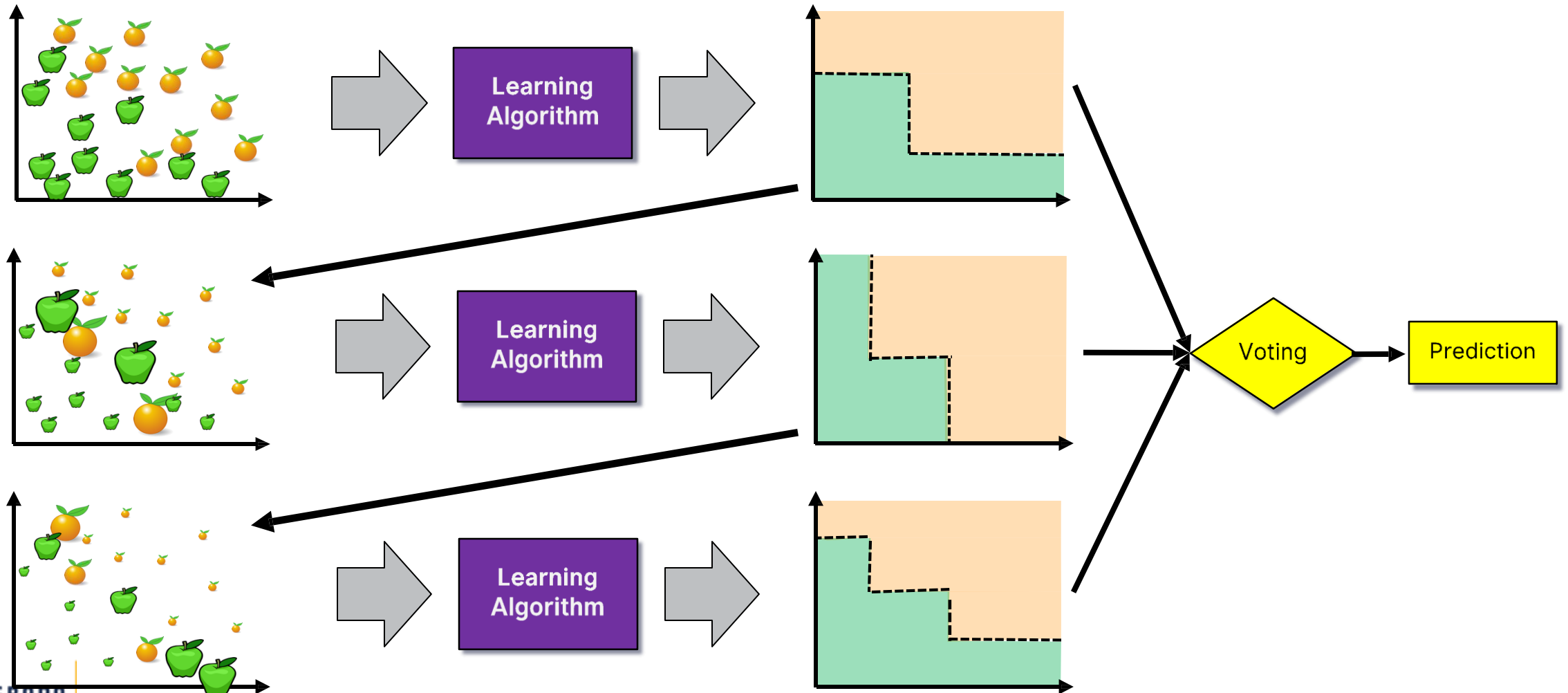
Boosting



Boosting



Boosting



Random Forest – scikit-learn

Objective:

Predict the class of wine based on its chemical composition and compare the performance of a Decision Tree classifier with that of a Random Forest classifier.

```
from sklearn import datasets
import pandas as pd

# Load the wine dataset
wine = datasets.load_wine()
df = pd.DataFrame(data=wine.data, columns=wine.feature_names)
df['class'] = wine.target
print(df.head())
```

```
from sklearn.model_selection import train_test_split
```

```
X = wine.data
y = wine.target
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	\
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	

	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	\
0	3.06	0.28	2.29	5.64	1.04	
1	2.76	0.26	1.28	4.38	1.05	
2	3.24	0.30	2.81	5.68	1.03	
3	3.49	0.24	2.18	7.80	0.86	
4	2.69	0.39	1.82	4.32	1.04	

	od280/od315_of_diluted_wines	proline	class
0	3.92	1065.0	0
1	3.40	1050.0	0
2	3.17	1185.0	0
3	3.45	1480.0	0
4	2.93	735.0	0



Random Forest – scikit-learn

Build and Train a Decision Tree Model

```
from sklearn.tree import DecisionTreeClassifier

dt_clf = DecisionTreeClassifier()
dt_clf.fit(X_train, y_train)

dt_pred = dt_clf.predict(X_test)
print("Decision Tree Accuracy:", accuracy_score(y_test, dt_pred))
```

Decision Tree Accuracy: 0.9444444444444444

Build and Train a Random Forest Model

```
from sklearn.ensemble import RandomForestClassifier

rf_clf = RandomForestClassifier(n_estimators=100)
rf_clf.fit(X_train, y_train)

rf_pred = rf_clf.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))
```

Random Forest Accuracy: 1.0

Random Forest – scikit-learn

Comparison of Feature Importance between Decision Tree and Random Forest

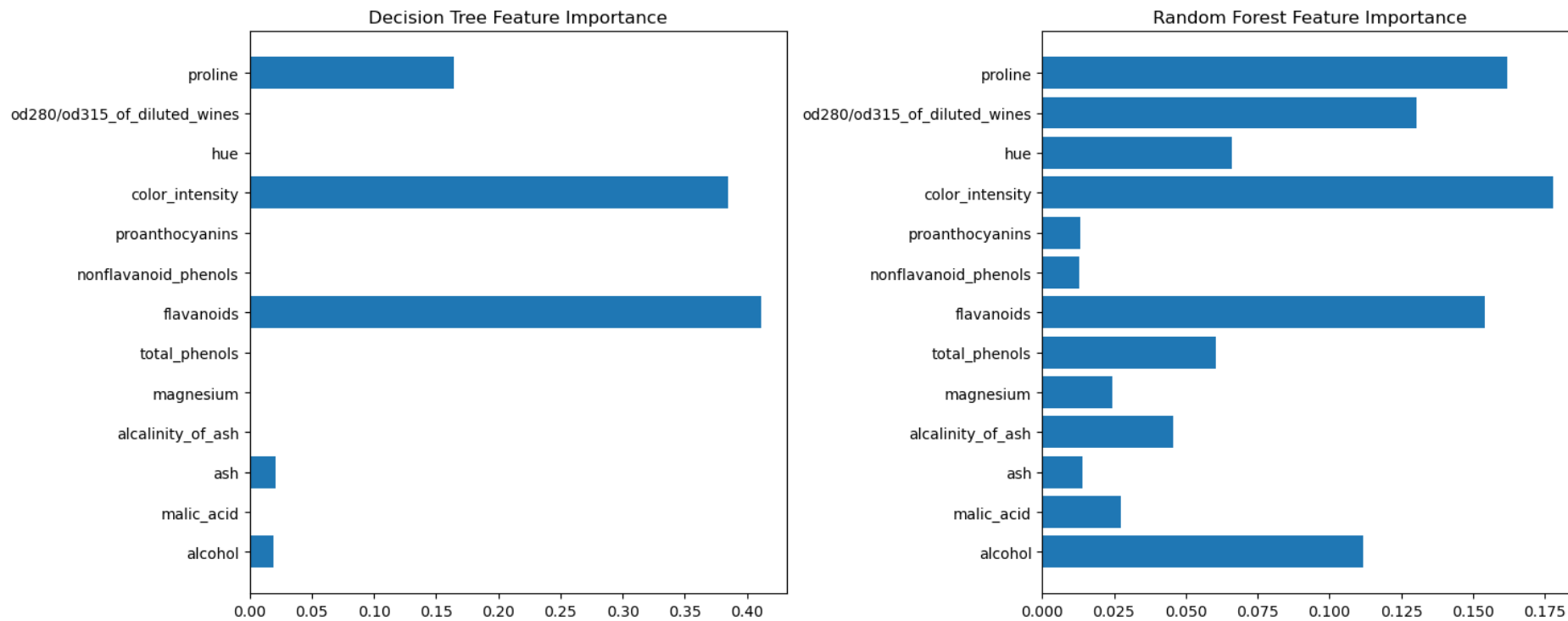
```
import matplotlib.pyplot as plt

# Plot feature importance for decision tree
plt.figure(figsize=(15, 6))
plt.subplot(1, 2, 1)
plt.barh(wine.feature_names, dt_clf.feature_importances_)
plt.title('Decision Tree Feature Importance')

# Plot feature importance for random forest
plt.subplot(1, 2, 2)
plt.barh(wine.feature_names, rf_clf.feature_importances_)
plt.title('Random Forest Feature Importance')
plt.tight_layout()
plt.show()
```

Random Forest – scikit-learn

Comparison of Feature Importance between Decision Tree and Random Forest



Random Forest

Summary

- Ensemble Learning:
 - Method that combines multiple learning algorithms to obtain performance improvements over its components
- Other commonly-used ensemble methods:
 - **Bagging**: multiple models on random subsets of data samples
 - **Random Subspace Method**: multiple models on random subsets of features
 - **Boosting**: train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples
- **Random Forests** are an ensemble learning method that employ **decision tree learning** to build multiple trees through **bagging** and **random subspace method**.
 - They rectify the overfitting problem of decision trees!

In-class Exercise

Objective: Predict the **species** of iris flowers based on the length and width of their sepals and petals.

1.Data Loading and Exploration

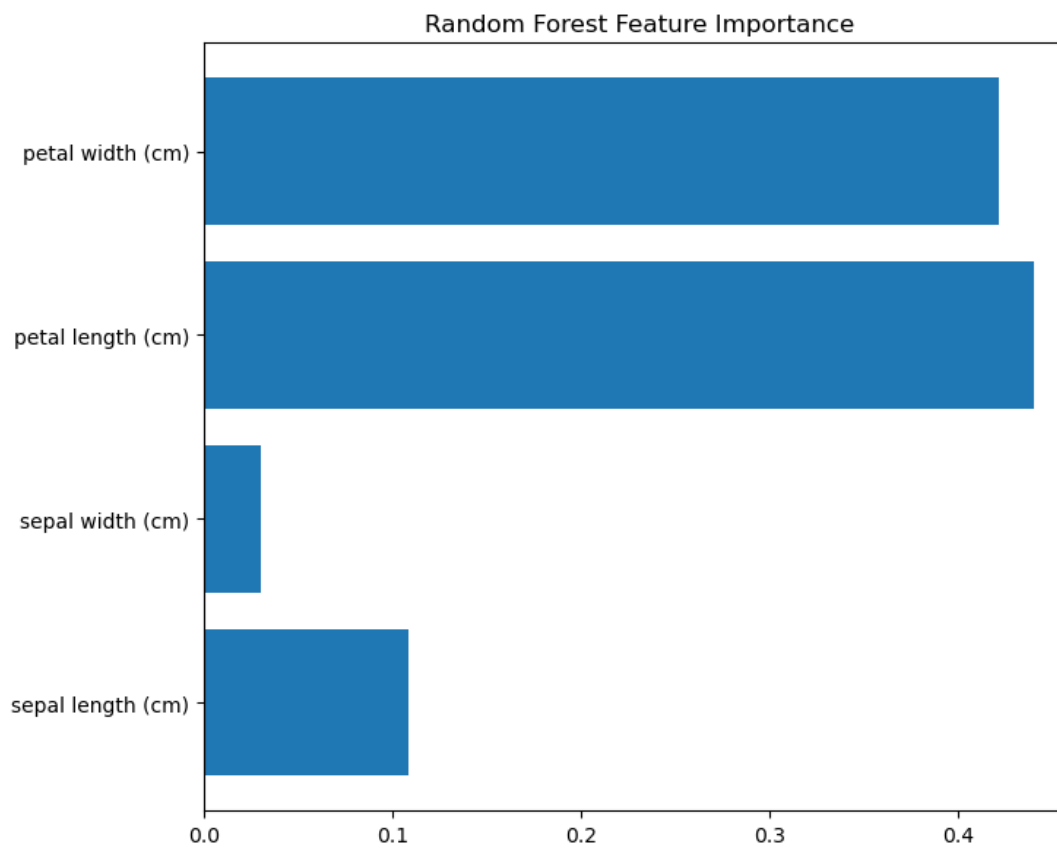
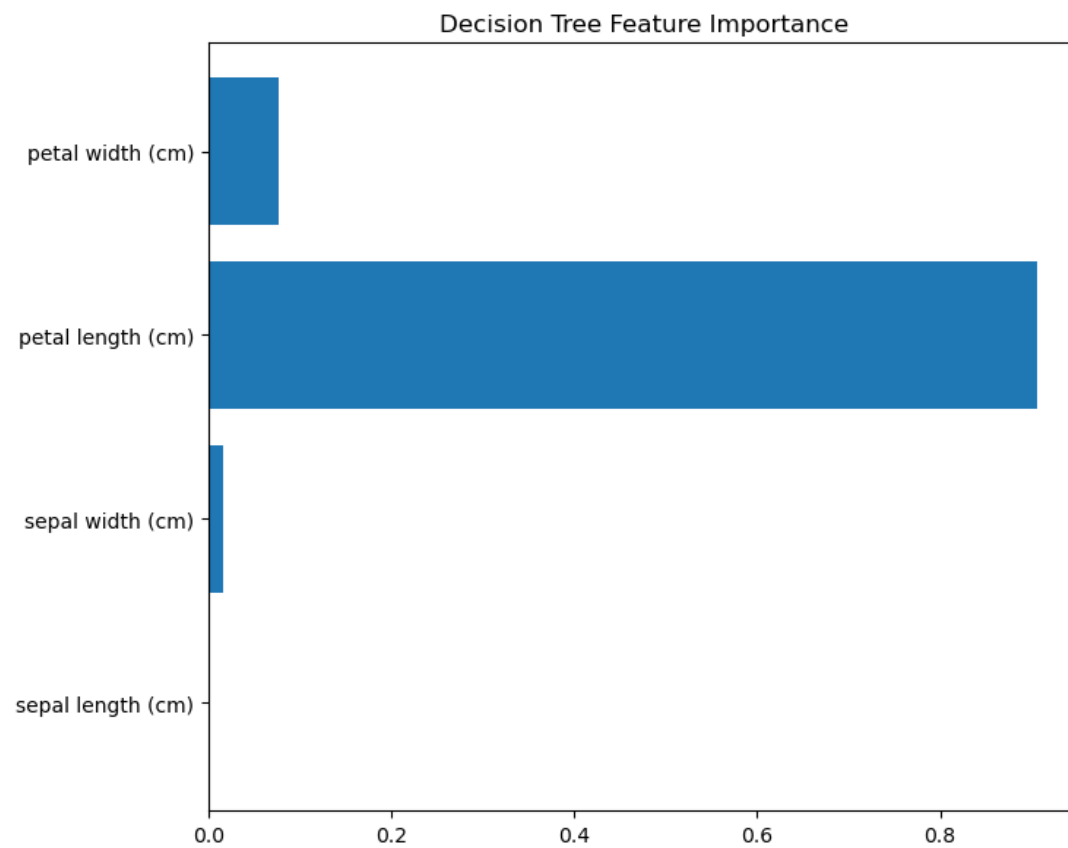
```
from sklearn.datasets import load_iris
import pandas as pd
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['species'] = iris.target
```

2.Split Data into Training and Testing Sets

3.Build and Train a Decision Tree Model & Random Forest Model

4.Evaluate the Model

5.Feature Importance comparison between decision tree and random forest.



In-class Exercise

- Explore Random Forest for Classifying Digits
- Compare SVM & DT & RF

```
from sklearn.datasets import load_digits  
digits = load_digits()  
X = digits.data  
y = digits.target
```