



Introduction to Data Science

(Lecture 16)

Dr. Mohammad Pourhomayoun
Assistant Professor
Computer Science Department
California State University, Los Angeles





Random Forest

Introduction

- Before Talking bout Random Forest Algorithm, we have to briefly define two new concepts:
 - Ensemble Learning
 - Bootstrapping

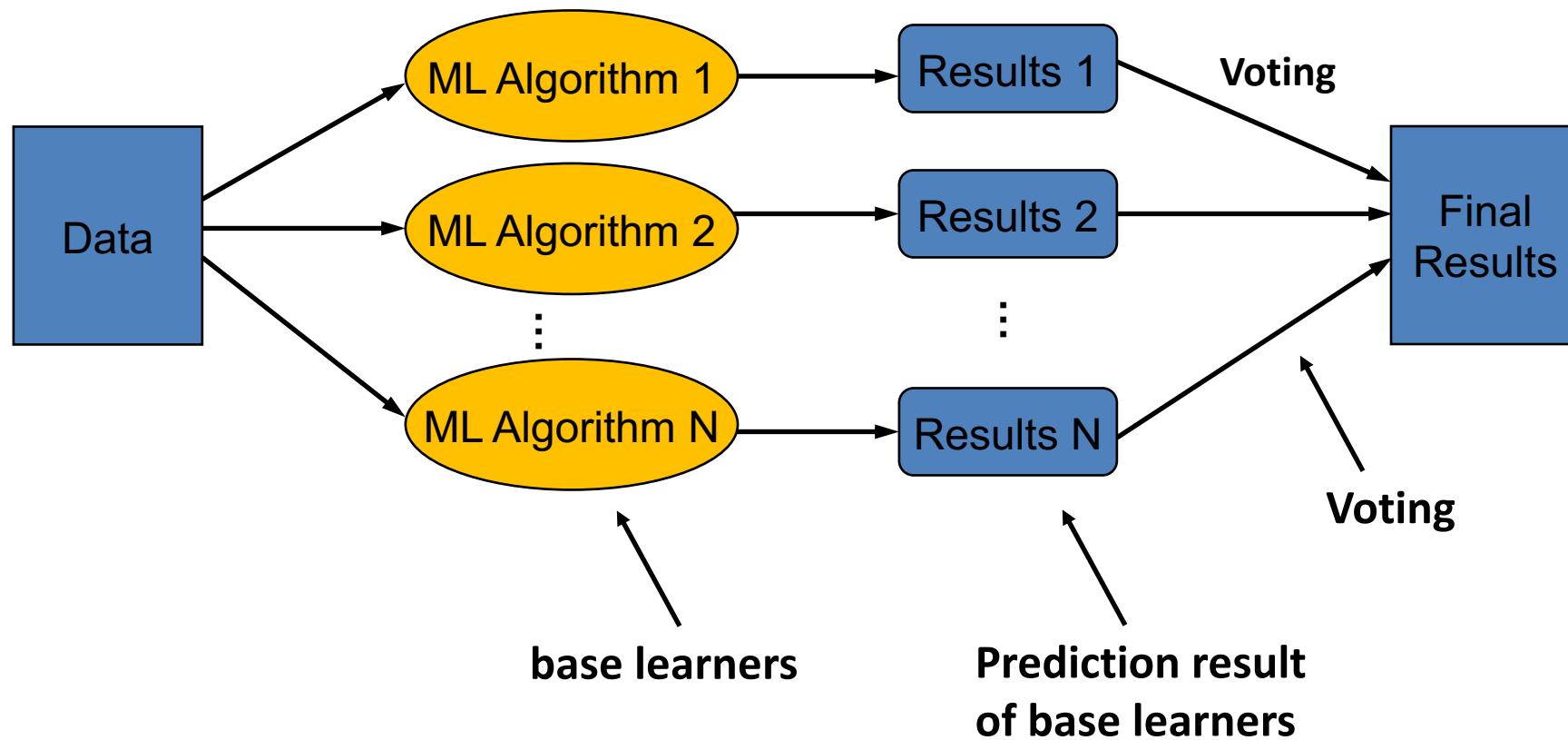


Ensemble Learning

- **Ensemble Learning** is a popular and effective approach to improve the accuracy and performance of a machine learning problem.
- **Ensemble Learning** uses a group of machine learning algorithms (called base learners), and then combine the results of them using some techniques such as Voting to achieve higher accuracy.
 - **Example:** Constructing a **Strong Classifier** by combining several **Weak Classifiers!**
 - Each learner (e.g. a classifier) **alone may have very poor performance**. But, a group of them **together can achieve very accurate results**.

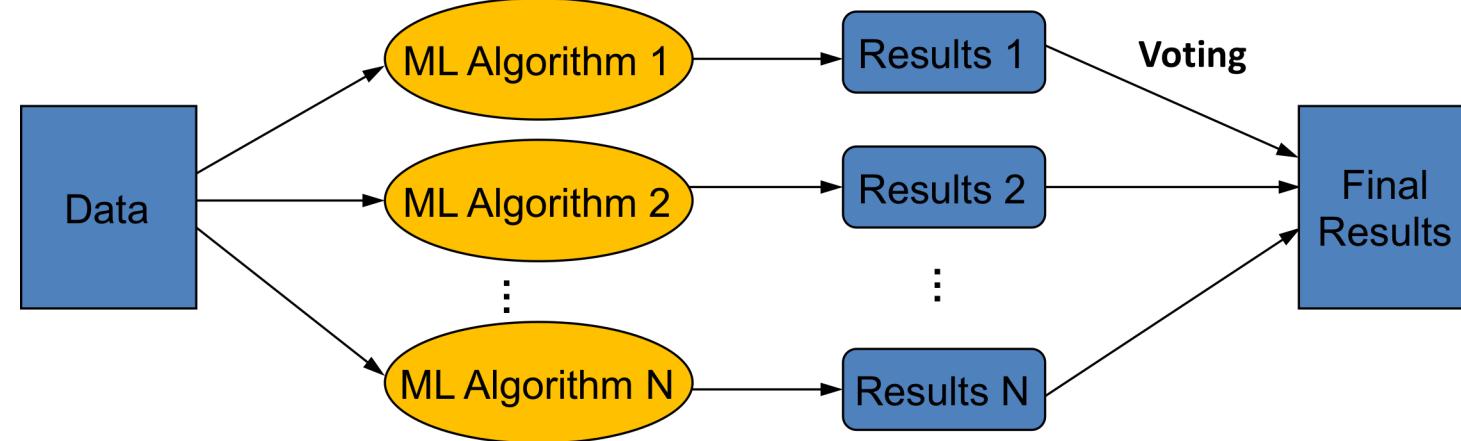


An Example of Ensemble Learning



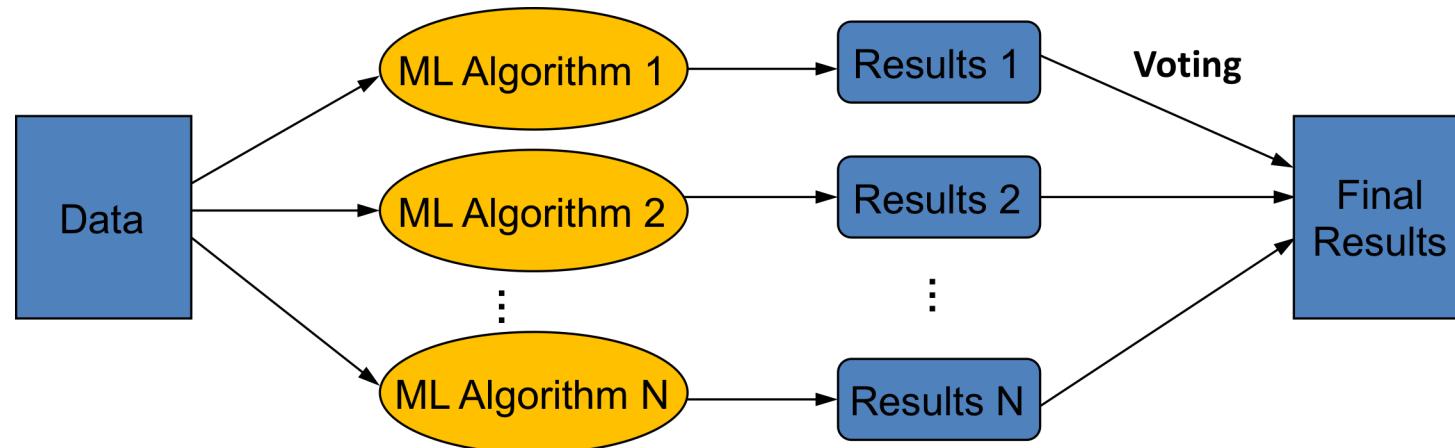
An Important Note about Ensemble Learning

- The key in designing ensembles is **diversity** and **not necessarily high accuracy** of the base classifiers.
- Members of an ensemble group should vary in the examples they misclassify, so that they cover each other's mistakes!



An Important Note about Ensemble Learning

- In other word, if we have several classifiers that are pretty accurate but they all misclassify the **same samples**, then ensemble learning will not achieve any better results! Therefore, most ensemble approaches, seek to promote diversity among the models they combine.



- We will talk a lot more about Ensemble Learning in CS4662/CS5661!

Bootstrapping

- **Bootstrapping** or **Bootstrap Sampling** is an important step in many Ensemble Learning methods (such as Random Forest).
- **Bootstrapping**: Suppose we have a **Training Dataset S** of **size N** . Bootstrapping generates L new training sets S_1, S_2, \dots, S_L each of **size M** , by sampling from S randomly and **with replacement**.
 - This type of sampling is called **Bootstrapping** or **Bootstrap Sampling**.
 - The bootstrap training sets S_1, S_2, \dots, S_L may have overlap with each other.
 - By sampling with **replacement**, some data sample may be repeated in each S_i .



Example for Bootstrap Sampling

Dataset “S”

1
2
3
4
5
6
7
8
9
10



S_1

7
4
5
5
8
9
2
8

S_2

9
3
9
9
10
7
4
6

S_3

4
1
1
10
8
3
7
6

...

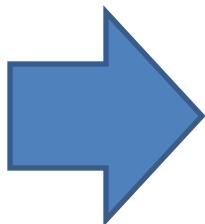
Random Forest

- **Random Forest** (also called Random Decision Forest) is an ensemble learning method, that operates by constructing several decision trees at training stage, and then combining the prediction results.
- First proposed by **Leo Breiman** in 2001.
- Random Forest algorithm resolves the well-known problem of **overfitting** in decision tree algorithms by reducing variance and instability.



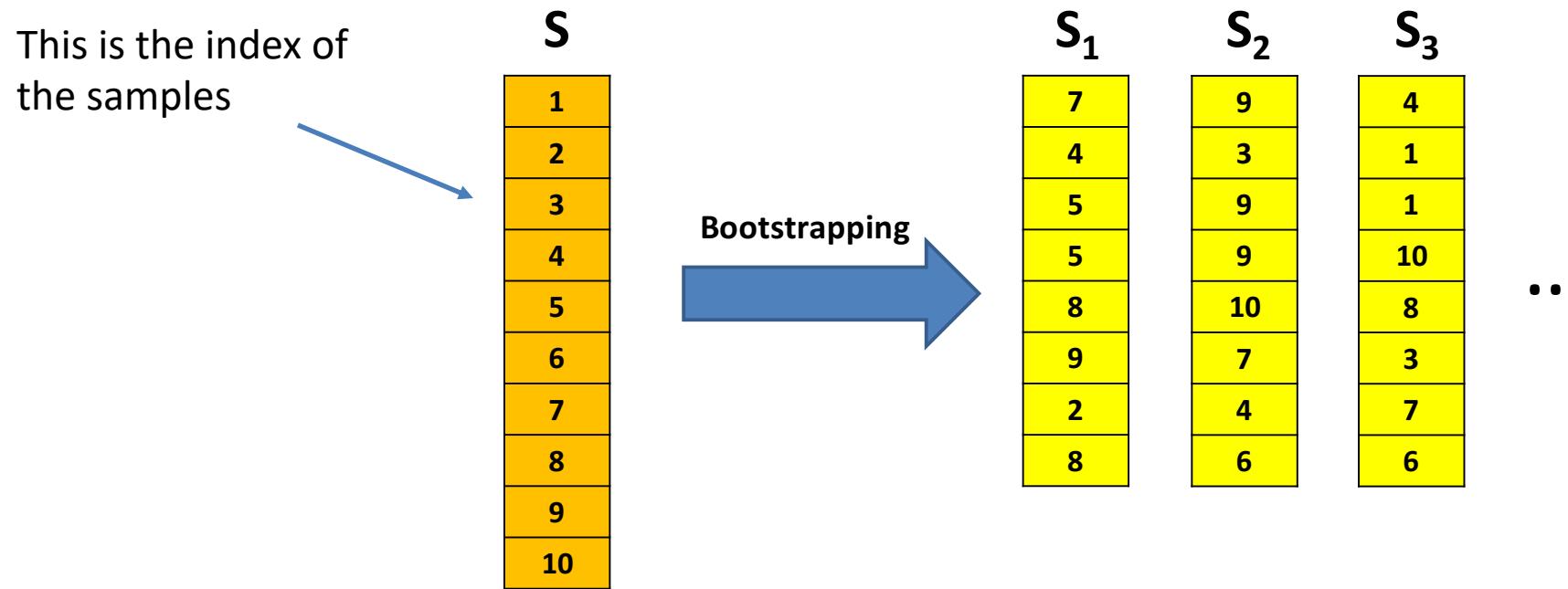
Random Forest

- **Strategy 1:** Random Forest uses **Bootstrap Sampling (randomly sampling of data)** to generate several training datasets and train a decision tree for each one.
- **Strategy 2:** Random Forest uses a **random selection of Features** to split on at each node of each decision tree.
- Both of these approaches will help the algorithm build a more accurate predictor, which is also robust to overfitting!



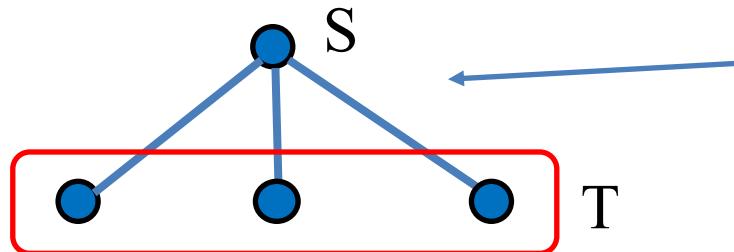
Random Forest Algorithm

- **Step1 (Bootstrapping):** Suppose we have a training set with N samples and D features. Random Forest first uses Bootstrap Sampling (randomly sampling of data) to generate L training datasets S_1, S_2, \dots, S_L .



Random Forest Algorithm

- **Step2 (Training):** The L new training sets S_1, S_2, \dots, S_L will be used to train L decision trees. **HOWEVER, for each branch split, the algorithm first randomly selects a small number of features, and then use the **best of them** for splitting.**
- In other word, unlike regular decision tree that selects the best feature in the entire feature set, random forest select the best feature in a very small random subset of features! **Thus, the features that are used in each tree **may differ from another tree!****



At each node, this is the best feature selected from a **small random subset of features**.

- **Note:** If d is the size of random feature subset at each node, the d is much smaller than the size of the entire feature set ($d \ll D$, e.g. $d = \sqrt{D}$).

Random Forest Algorithm

- **Step3 (Base Learner Prediction):** Given a new unknown data sample, all **trained** decision trees (L trees) make their prediction for the new sample.
- **Step4 (Voting):** The final decision will be made using **Voting** method. The final prediction is based on the majority vote of the L decision trees.



Pseudo Code for Random Forest

Given a training set S

For $i = 1$ to L do:

- Build subset S_i by sampling with replacement from S .

- Learn tree T_i from S_i :

- At each node of tree T_i :

- Choose best split from a random subset of d features.

Each trained tree makes prediction about a new sample.

Make final prediction according to majority vote of the L trees.



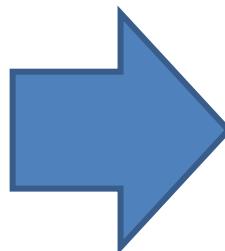
Advantages of random forest

- One of the most accurate classification algorithms!
- Very Robust to Noise and Overfitting.
- It can handle big data including hundreds of features.
- It can handle missing value!



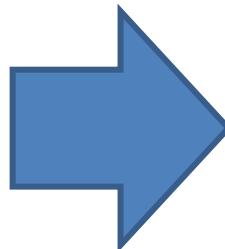
Why does Random Forest work so well?

- **Question1: Why is Random Forest much better than a single Decision Tree?**
- It can reduce variance and resolve the overfitting problem.



Why does Random Forest work so well?

- Question2: Why is Random Forest much better than a set of Decision Trees together?
- In other word, Why does using a subset of features at each node (rather than all features) help a lot?



Why does Random Forest work so well?

- Question2: Why is Random Forest much better than a set of Decision Trees together?
- In other word, Why does using a subset of features at each node help a lot?
- Review:
 - The key of designing ensembles is **diversity!**
 - Members of the ensemble should be uncorrelated and vary in the examples they misclassify, so that they can compensate for each others mistakes!



Why does Random Forest work so well?

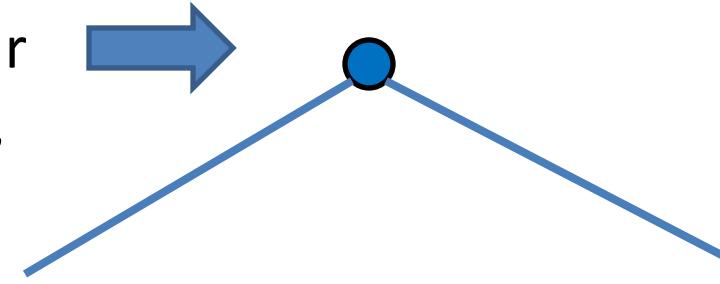
- **Question2: Why is Random Forest much better than a set of Decision Trees together?**
- **In other word, Why does using a subset of features at each node help a lot?**
- If each time we select from the entire feature set, every tree always selects the **best features one after another**, and consequently, the structure of all trees will be **the same**.
- On the other hand, Selecting only from a random subset of the features at each node, makes our trees **different**.
- Random Forest tries to generate the set of trees that are **different from each other**. Nonetheless, each one does **its best in its own way**.
- After voting, the trees will **compensate for each others mistakes**, and provide the best results together.



Example: Tree1 in Random Forest for Titanic

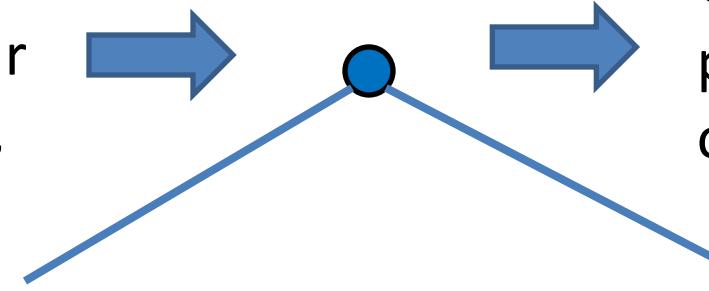
We have total 3 features

{age, gender, pclass} in our dataset. But, at this node,
let's limit our feature subset to {gender,pclass}



Example: Tree1 in Random Forest for Titanic

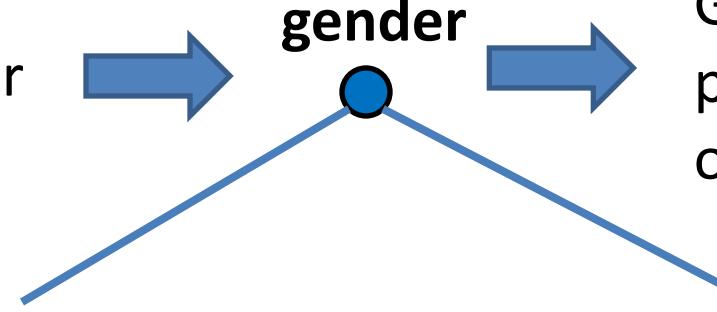
We have total 3 features **{age, gender, pclass}** in our dataset. But, at this node, let's limit our feature subset to **{gender,pclass}**



Gender is better than pclass, So I split based on gender!

Example: Tree1 in Random Forest for Titanic

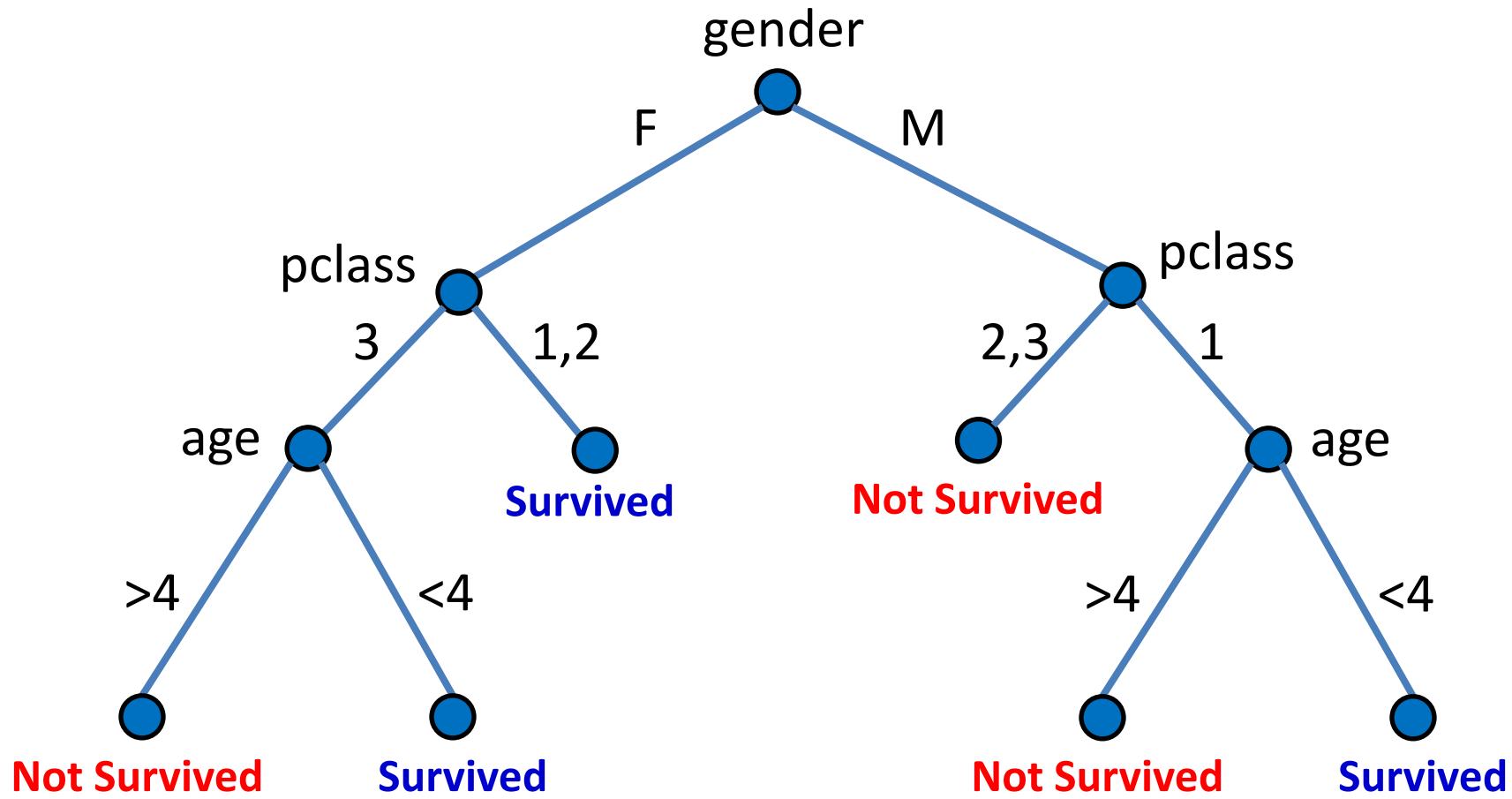
We have total 3 features **{age, gender, pclass}** in our dataset. But, at this node, let's limit our feature subset to **{gender,pclass}**



Gender is better than pclass, So I split based on gender!

And similarly for the rest of the tree ...

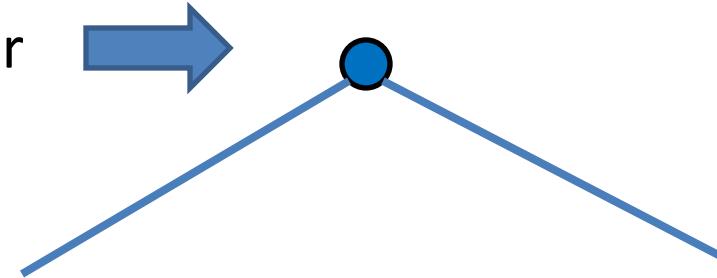
Example: Tree1 in Random Forest for Titanic



Example: Tree2 in Random Forest for Titanic

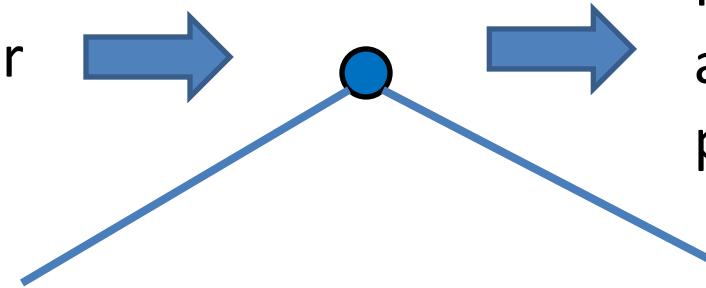
We have total 3 features

{age, gender, pclass} in our dataset. But, at this node, let's limit our feature subset to {age,pclass}



Example: Tree2 in Random Forest for Titanic

We have total 3 features **{age, gender, pclass}** in our dataset. But, at this node, let's limit our feature subset to **{age,pclass}**

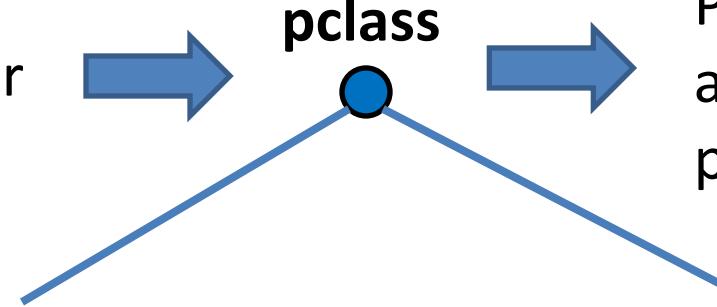


Pclass is better than age, So I split based on pclass!



Example: Tree2 in Random Forest for Titanic

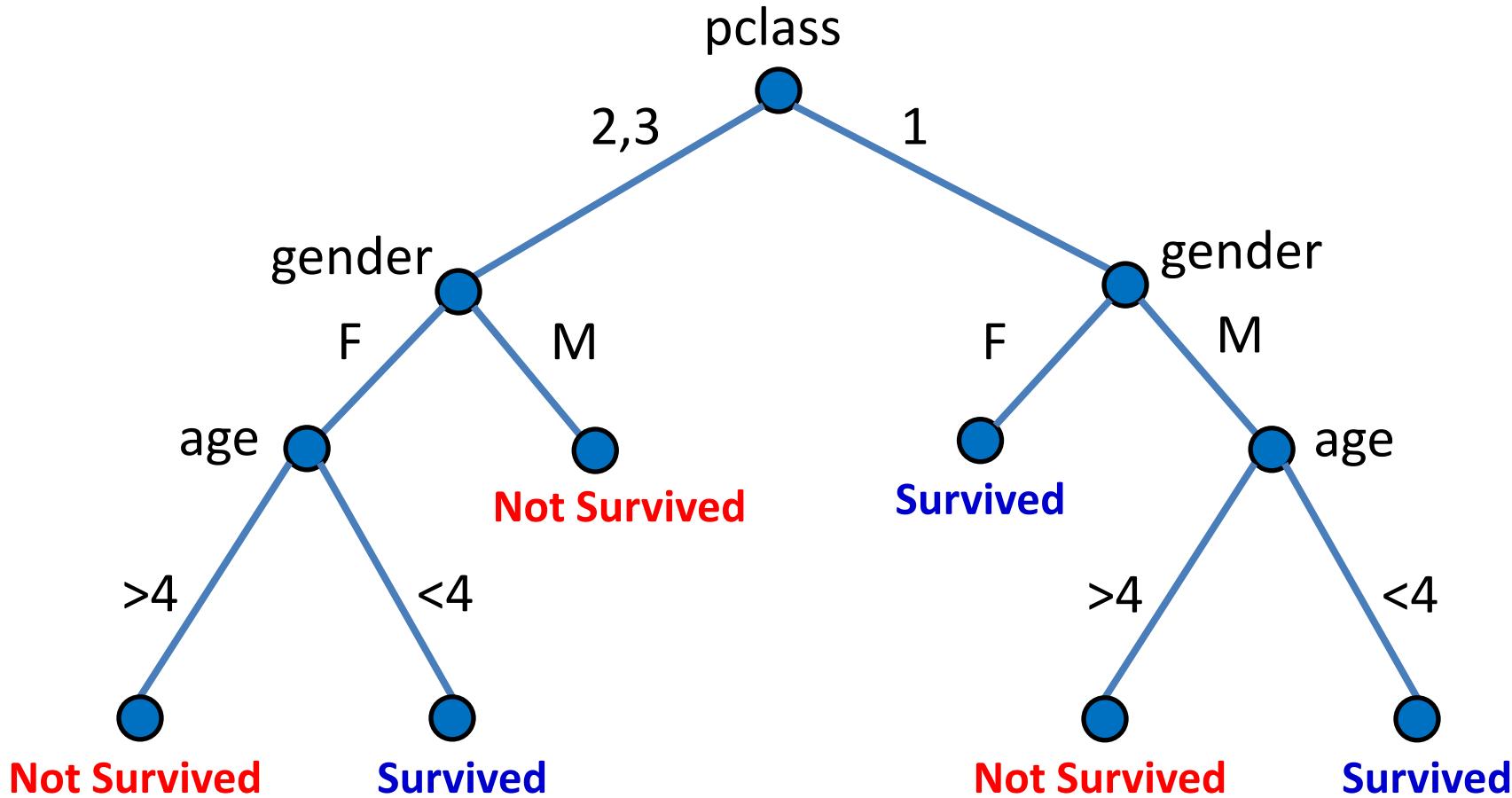
We have total 3 features **{age, gender, pclass}** in our dataset. But, at this node, let's limit our feature subset to **{age,pclass}**



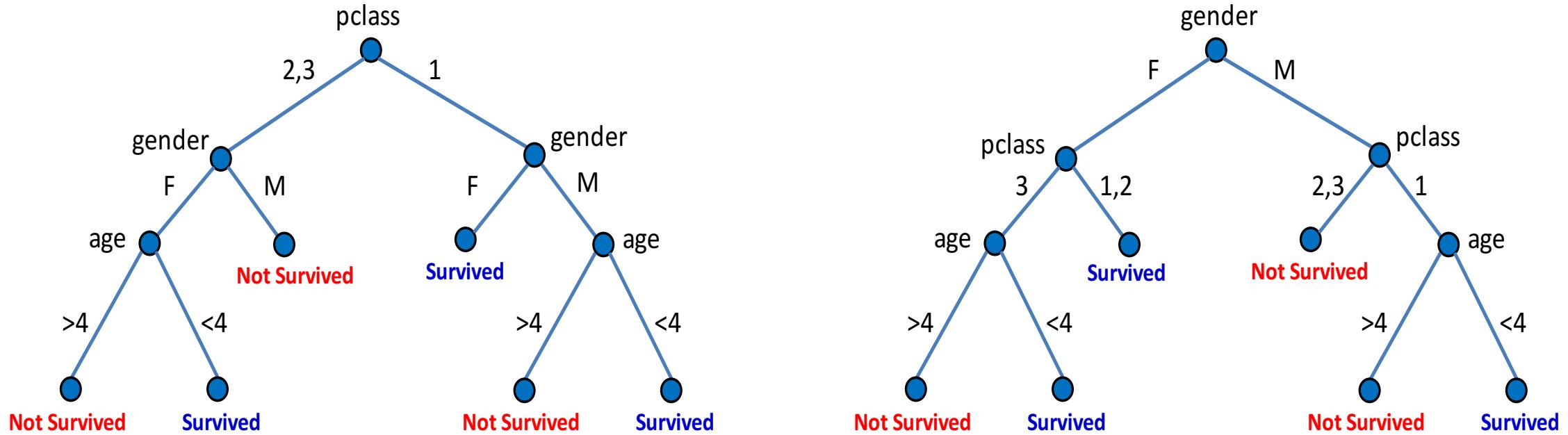
Pclass is better than age, So I split based on pclass!

And similarly for the rest of the tree ...

Example: Tree2 in Random Forest for Titanic



Example: Random Forest for Titanic



As we see, the 2 trees are completely different! And this is what we want!

Question: What is your prediction for a passenger who is adult female and has 2nd class ticket?



Thank You!

Questions?