

Tera

Módulo 4: Aprendizado de Máquina Supervisionado

Aula 21: Random Forests & Ensembles

Recapitulando...

- Aprendemos como alguns algoritmos de ML funcionam
- Aprendemos o fluxo geral de aprendizado supervisionado e algumas etapas cruciais necessárias
- Hoje vamos investigar algumas limitação dos métodos que conhecemos
- E aprender métodos poderosos que mitigam alguns desses problemas!

`<code> ... </code>`

Árvores de Decisão

- São métodos flexíveis, intuitivos e interpretáveis
- Porém nem sempre geram os modelos mais precisos
- E sofrem muito de problemas de variância e overfitting
- São mais efetivos quando usados como base para métodos mais poderosos

Ensembling

- Usamos uma técnica simples de **ensembling** para combinar múltiplos classificadores
- Essa técnica é chamada **bagging**
- Vamos entender melhor como e por que ela funciona com um cenário hipotético

Cenário

- Você quer decidir se você deve **investir um startup promissora S**
- Basicamente você quer prever o sucesso dessa startup
- A Startup opera em uma área pouco familiar para você, então você **contrata 5 especialistas** para auxiliar na sua decisão

Especialistas

Especialista em media social



Ex-CEO de uma startup adquirida



Operador da bolsa



Ex-funcionário da startup



Investidor da startup

T

Especialistas

60% de acerto



70% de acerto



75% de acerto



65% de acerto



70% de acerto

Especialistas

- Todos têm seus pontos **fortes** e **vieses**
- Você se sentiria mais confortável com a opinião de um deles ou de todos eles?
- Se você somente decidir por **unanimidade**, sua chance de acerto é maior que 99.9% !!
- Se você decidir por **maioria**, sua chance de acerto ainda é de 90%!

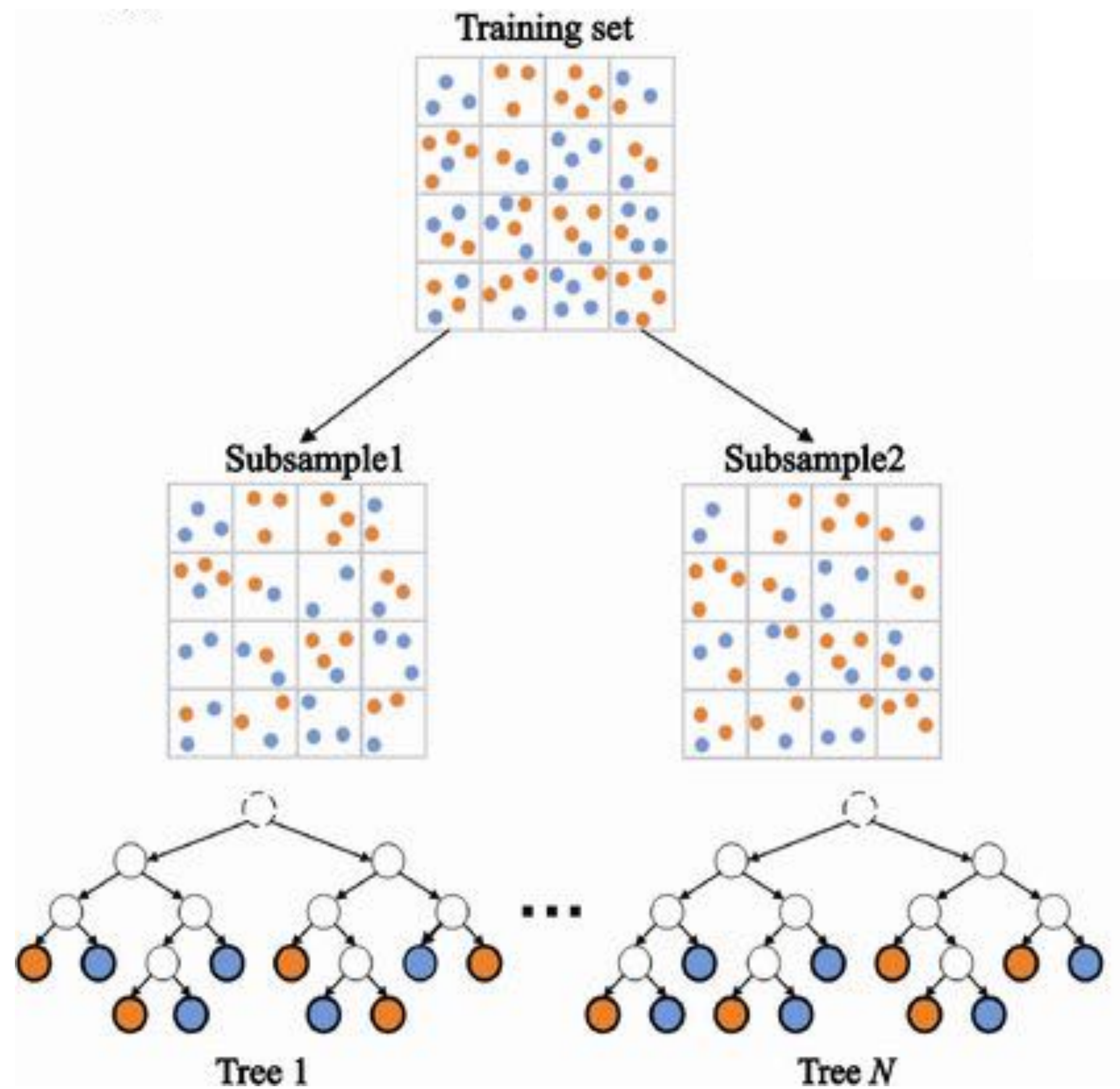
Especialistas

- Esses números assumem que as opiniões dos especialistas **são independentes**, o que raramente é verdade
- Intuitivamente, se todos especialistas fossem ex-funcionários da empresa S, sua confiança seria a mesma que no primeiro cenário?
- Diversidade é bom!
- Agora pense que ao invés de especialistas, temos cinco modelos treinados

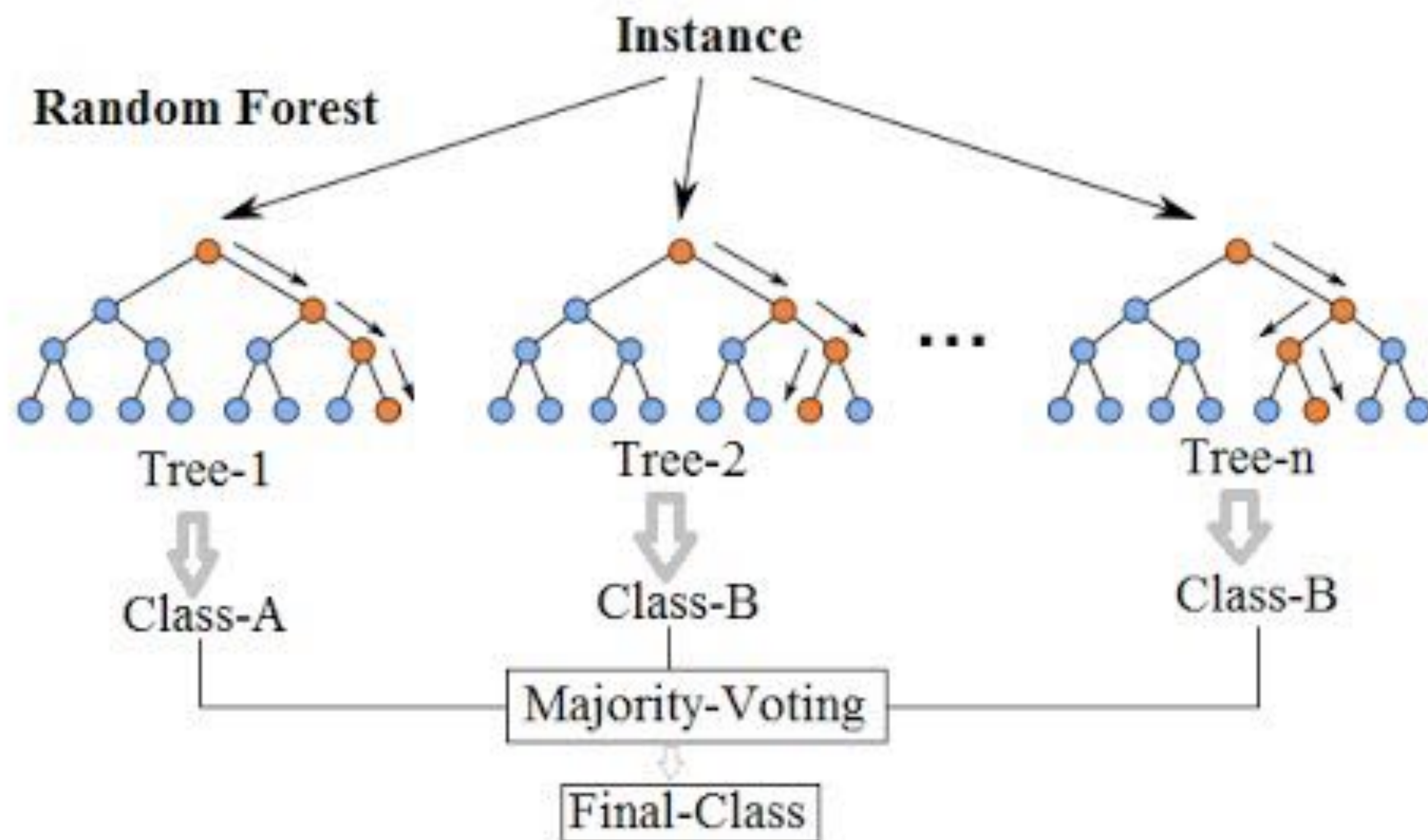


Florestas Aleatórias

- Podemos usar as **decisões de várias árvores** para fazer uma predição
- A diversidade entre as árvores é promovida usando **dados e atributos diferentes** pra treinar cada árvore
- Por fim, devemos combinar as decisões das árvores de alguma forma



Random Forest Simplified



`<code> ... </code>`

Ensembles

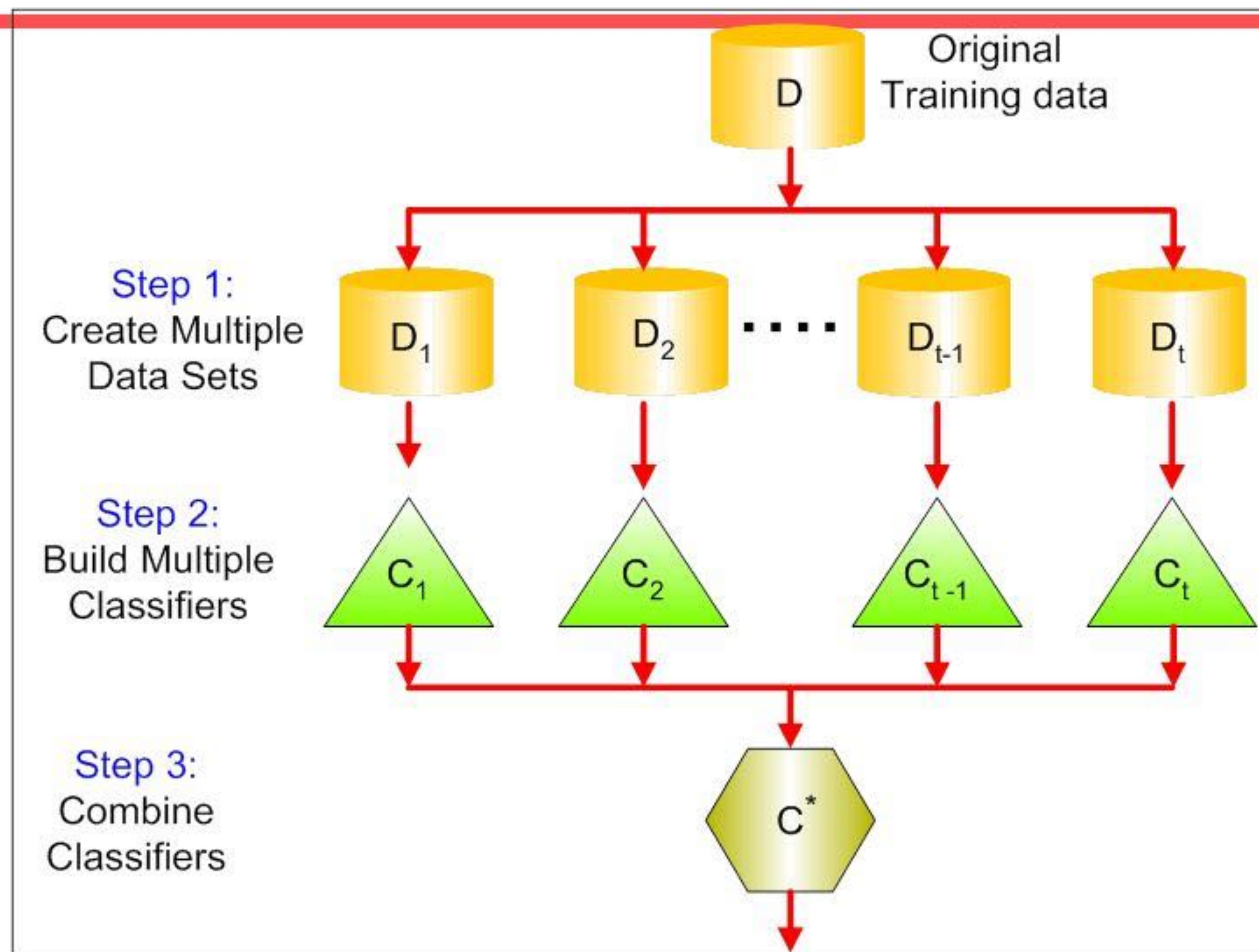
- *Random forests* são possivelmente os ensembles mais populares, porém o conceito vai muito além delas
- Se podemos combinar resultados de árvores de decisão diferentes, por que não **combinar quaisquer modelos diferentes?**
- Não só podemos, como existem inúmeras formas de fazê-lo!

Ensembles são métodos que combinam **múltiplos modelos** para obter previsões que os modelos constituintes não seriam capaz de obter

Bagging

- RandomForest pode ser considerada uma forma **Bagging**
- Bagging consiste em combinar o resultado de n classificadores $[C_1, C_2, \dots, C_n]$, treinados respectivamente em n datasets $[D_1, D_2, \dots, D_n]$
- Cada dataset D_i é uma amostra, com reposição, de m elementos do dataset original D

General Idea



Bagging

- Cada modelo C_i é considerado um **weak learner**
- Com reposição significa que os D_i podem conter elementos em comum e até mesmo repetidos!
- O resultado dos modelos C_i podem ser combinados de diversas formas:
 - **Regressão:** média das predições
 - **Classificação:** classe majoritária das predições (hard voting),
média das probabilidades das predições (soft voting)

`<code> ... </code>`

O que mais?

- Em bagging, treinamos classificadores com amostras diferentes do dataset, porém com o mesmo algoritmo
- Por que não usar **diferentes algoritmos**?
- Podemos obter diversidade entre os *weak learners* utilizando diferentes:
 - Populações
 - Algoritmos
 - Parametrizações
 - Sementes aleatórias

Voting

- Quaisquer modelos podem ser combinados através de qualquer estratégia
- **VotingClassifier** do SKLearn pode ser usado para isso
- E essa combinação pode ser também usada em um novo *ensemble*!
- Sua criatividade é o limite

`<code> ... </code>`

Stacking

- Vamos supor que treinamos 5 modelos para prever inadimplência e queremos combiná-los com voting

- Obtivemos as seguintes predições para alguns exemplos:

C1	C2	C3	C4	C5	Maioria	Ytrue
0	0	1	0	1	0	1
1	0	0	0	0	0	0
0	0	1	0	1	0	1
1	1	1	1	0	1	0
0	1	1	0	1	1	1

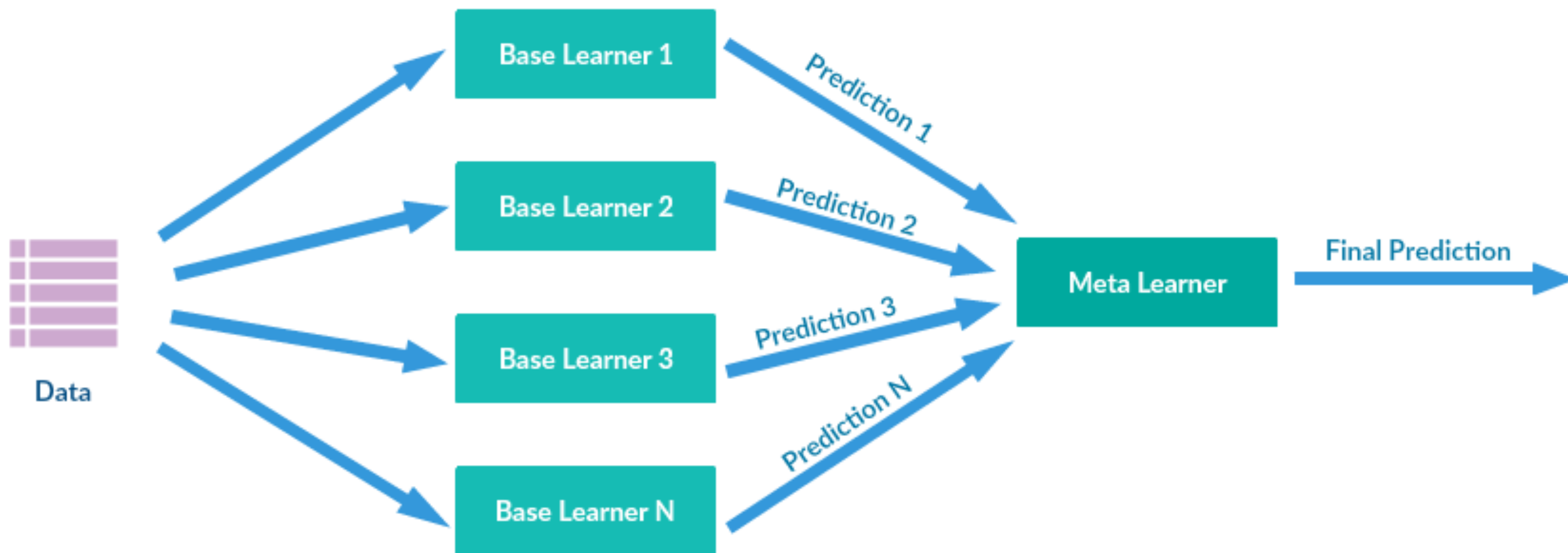
Stacking

- Acontece que votação não parece a melhor forma de combinar esses resultados
- Existem alguns comportamentos entre os classificadores que parecem mais informativos para determinar uma predição melhor Y
- Por que não usar as predições de $C1$, $C2$, $C3$, $C4$ e $C5$ como features X de um novo modelo tentando prever Y ?
- Isso é chamado de **Stacking**, e é extremamente poderoso!

Stacking

- Para cada exemplo no treino, podemos criar uma nova instância de treinamento $[P_1, P_2, \dots, P_n]$ com as previsões de cada um dos nossos modelos base $[C_1, C_2, \dots, C_n]$
- Esse novo modelo é um *meta-learner*, e ele vai aprender a melhor forma de combinar as previsões $[P_1, P_2, \dots, P_n]$
- Ao invés de usarmos as previsões binárias dos C_i como *features*, é comum usarmos as probabilidades

Stacking



`<code> ... </code>`

Ensembles

- Então por que não criar milhares de modelos para todos os problemas?
- **Restrição computacional** (treino longo e latência na predição), complexidade de implementação, dificuldade de manutenção e ganhos marginais a partir de um certo ponto
- Porém existem sim *ensembles* extremamente complexos e poderosos, especialmente em competições

Conclusões

- Técnicas de ensembles podem ser bem poderosas, porém aumentam consideravelmente a complexidade da sua solução
- São muito populares em competições pois abrem infinitas possibilidades de modelagem
- Elas reduzem mais erro por variância, mas não tanto erro por viés
- Na prática, porém, são difíceis de avaliar corretamente, colocar em produção e manter

DÚVIDAS?!