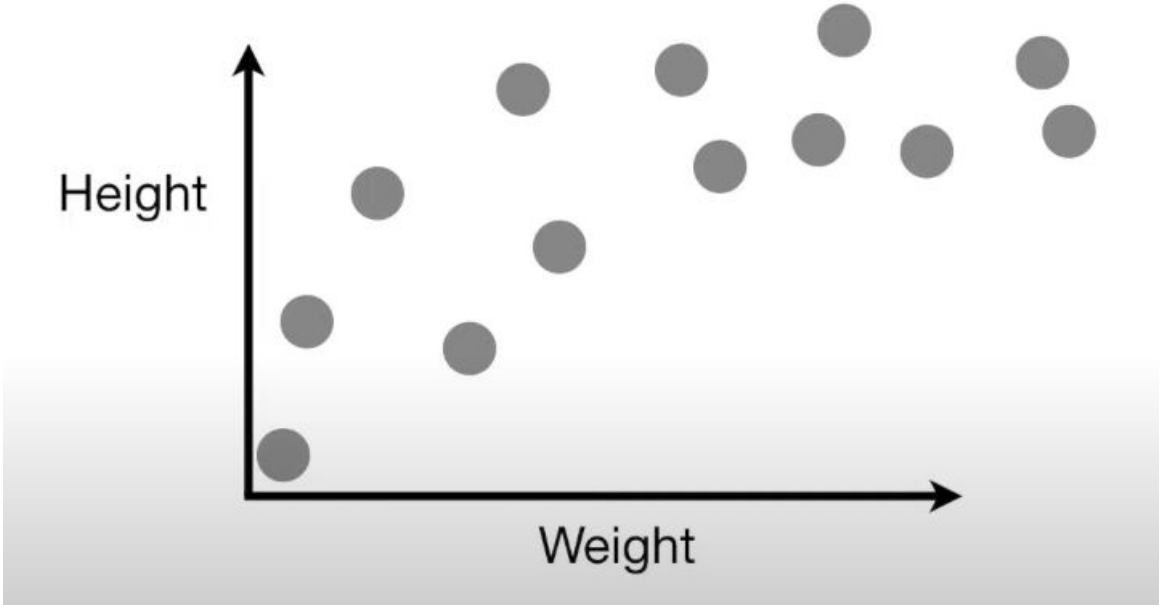
ANTICIPARE LA CRESCITA CON LE NUOVE COMPETENZE SUI BIG DATA – EDIZIONE 2

Operazione Rif. PA 2019-11596/RER "Anticipare la crescita con le nuove competenze sui Big Data", approvata dalla Regione Emilia-Romagna con DGR n° 789 del 20 maggio 2019 e co-finanziata dal Fondo Sociale Europeo PO 2014-2020

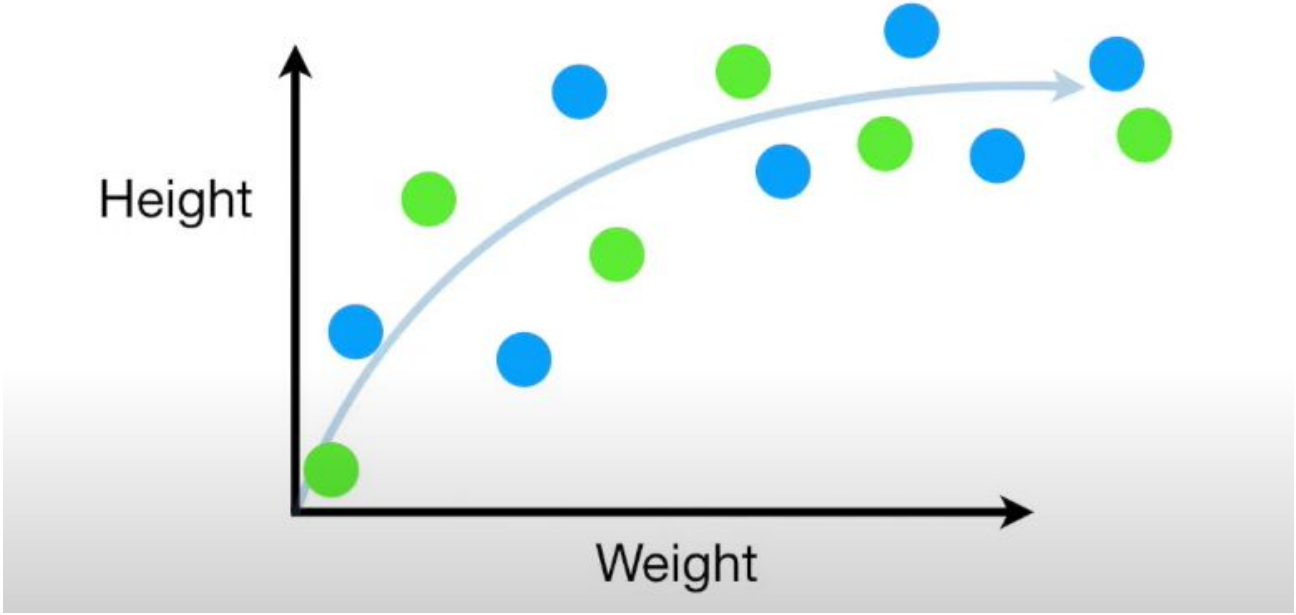
Programma della lezione

- Bias e Variance
- Decision tree
- Random Forest
- Gradient Boosting

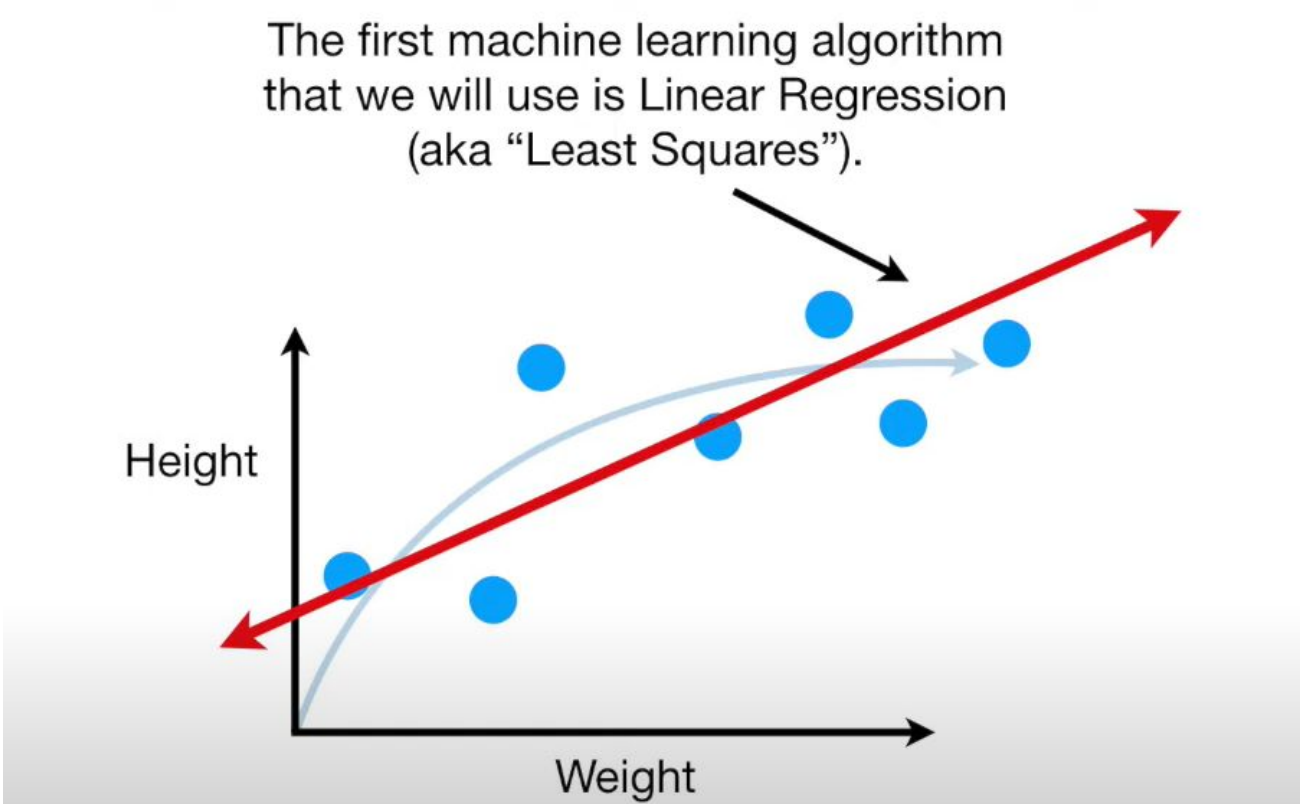
Esempio: vogliamo prevedere il peso dei ratti, data l'altezza



Primo step: Dividere il dataset in training e test set

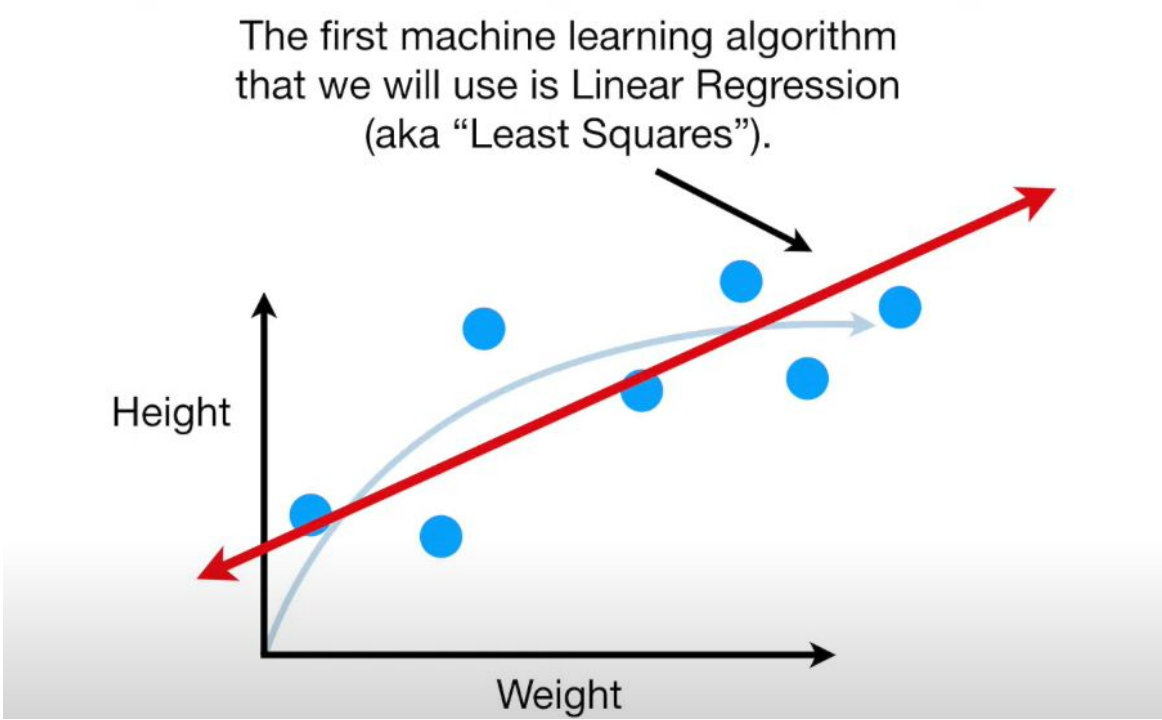


Secondo step: applico modelli di machine learning

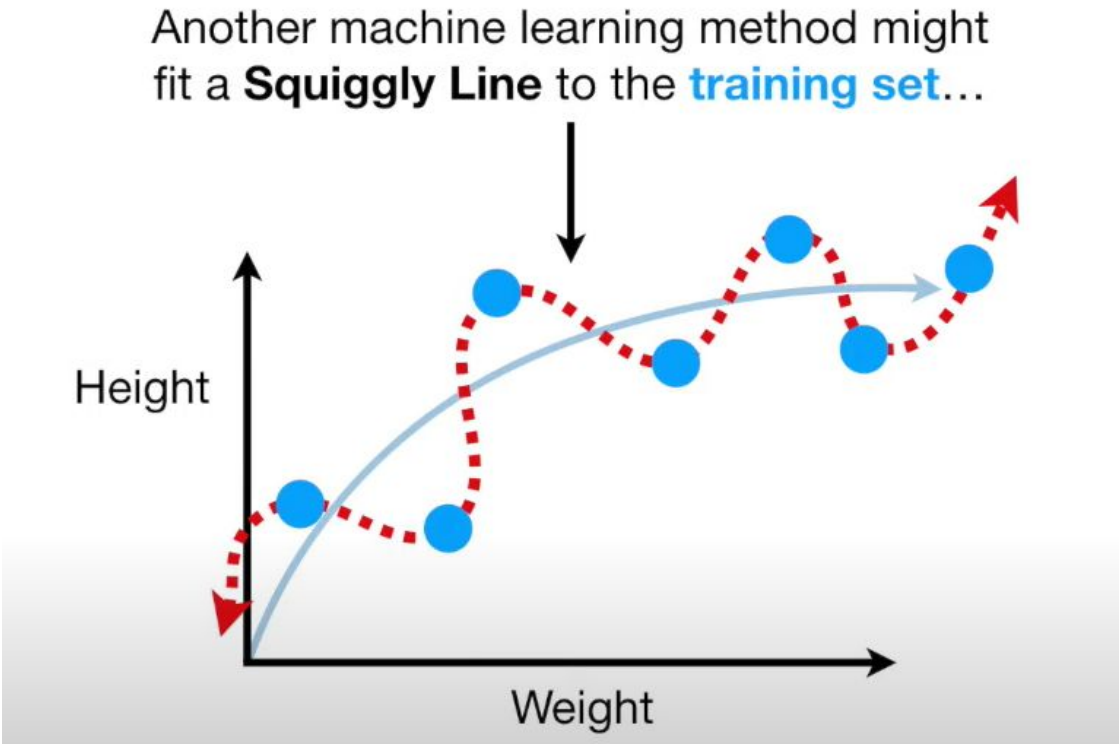


Bias

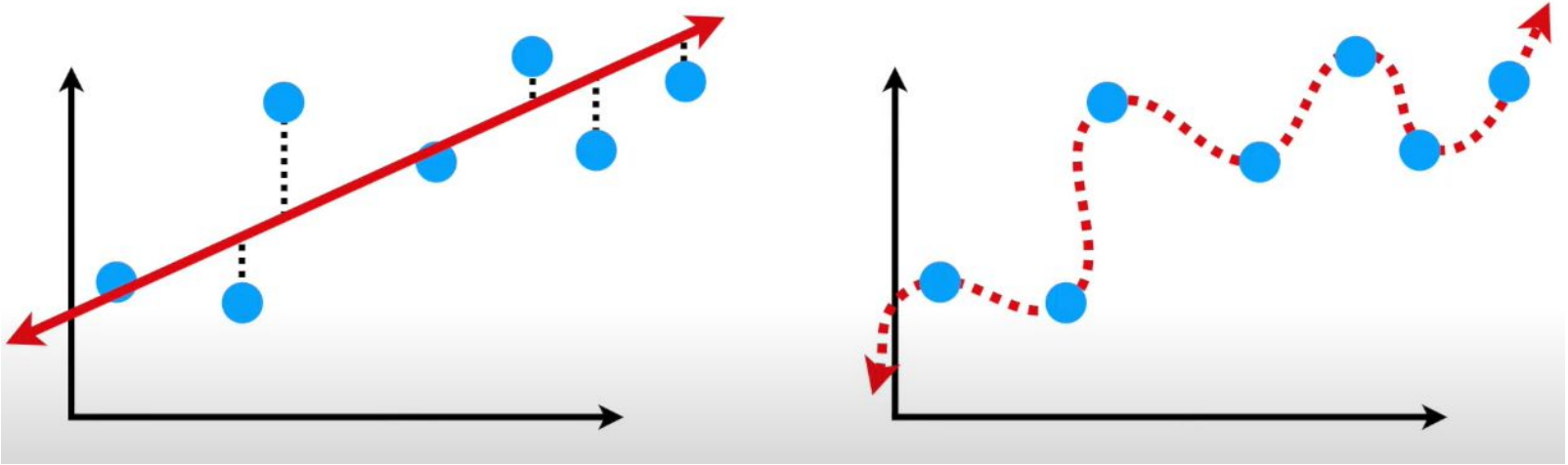
- Bias = incapacità di un modello di catturare la vera relazione tra le variabili
- Esempio: regressione lineare ha un alto bias
- Quando si ha un alto bias, significa che c'è underfitting



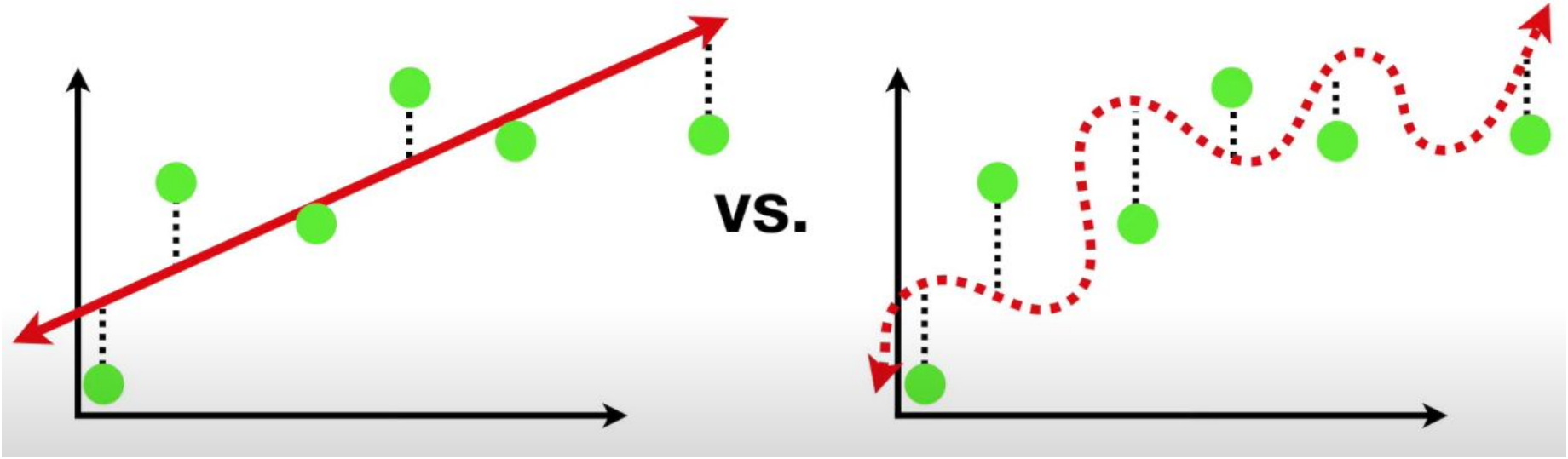
Secondo step: applico modelli di machine learning



Paragoniamo i risultati dei due modelli nel training set

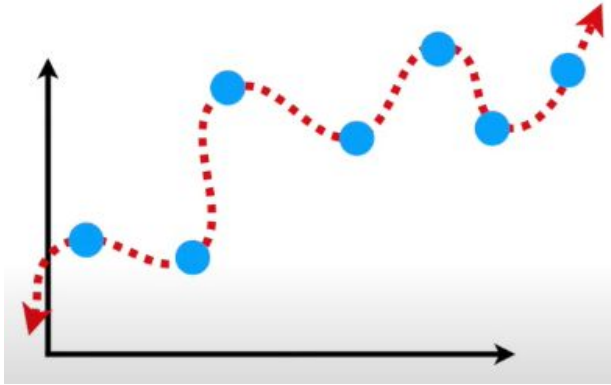


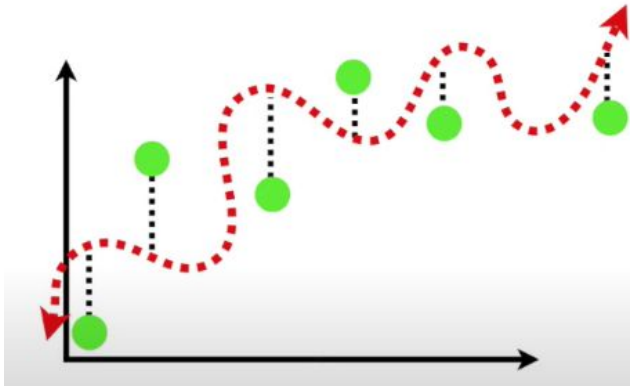
Paragoniamo i risultati dei due modelli nel testing set



Variance

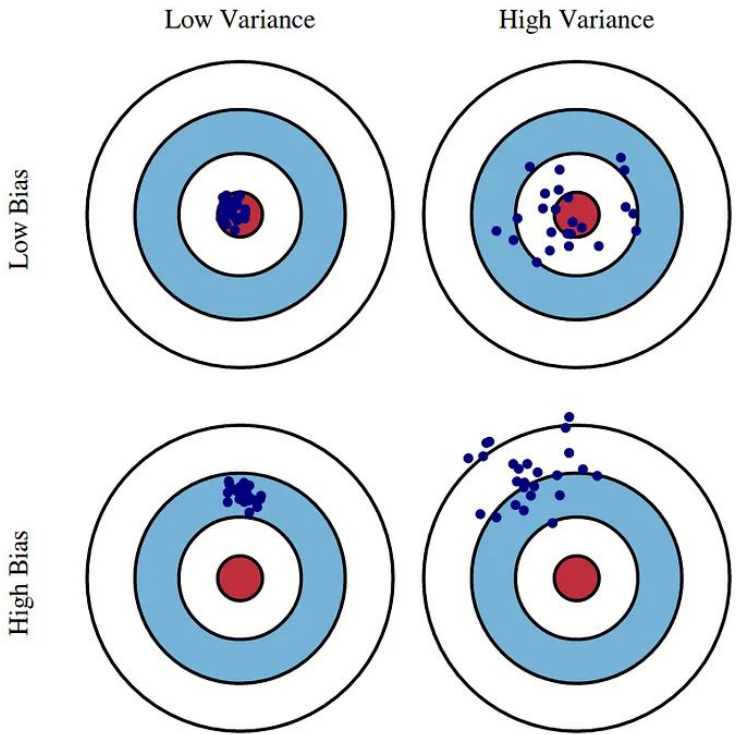
- Variance = differenza di performance del modello in training e test set
  - Alta varianza significa che il modello ha una buona performance nel training set, ma le prestazioni sono scarse in nuovi dati
  - Quando si ha un’alta variabilità, significa che c’è overfitting



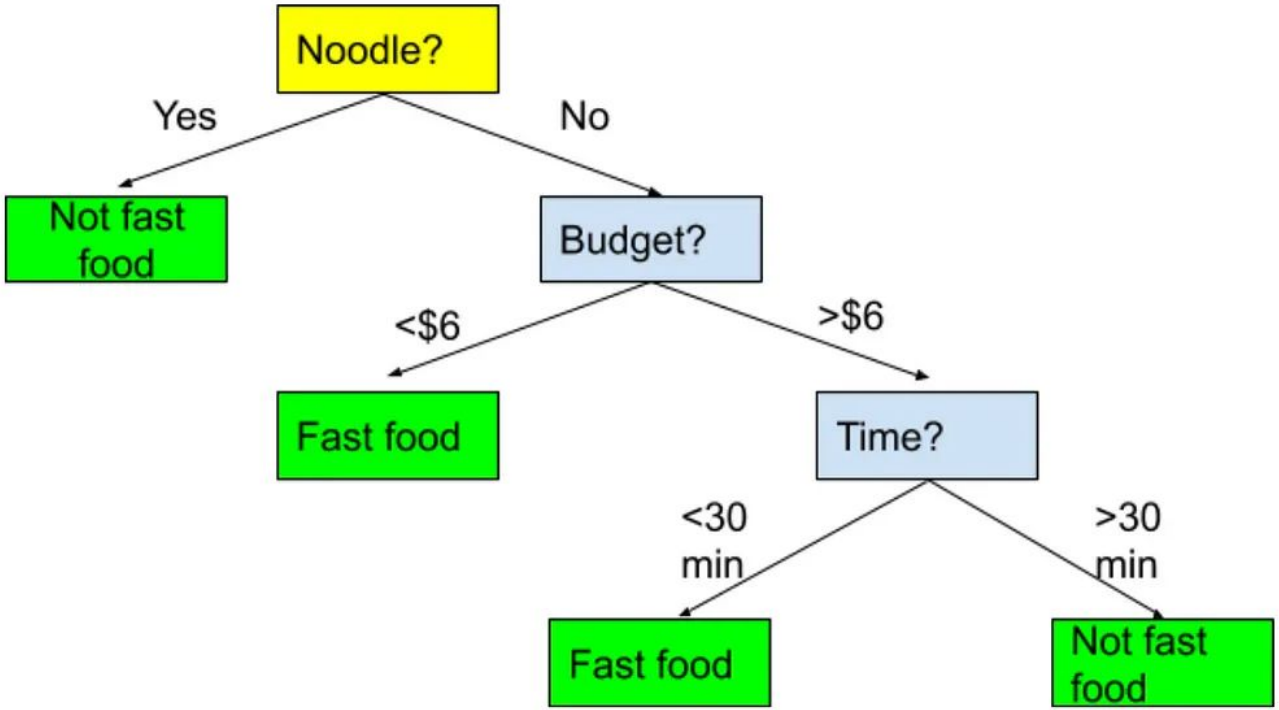


Bias-Variance trade-off

- Bias = eccessiva semplificazione del modello
- Variance = differenza di performance del modello in training e test set
- L’ideale sarebbe avere un modello con basso bias e bassa variabilità



Decision Tree

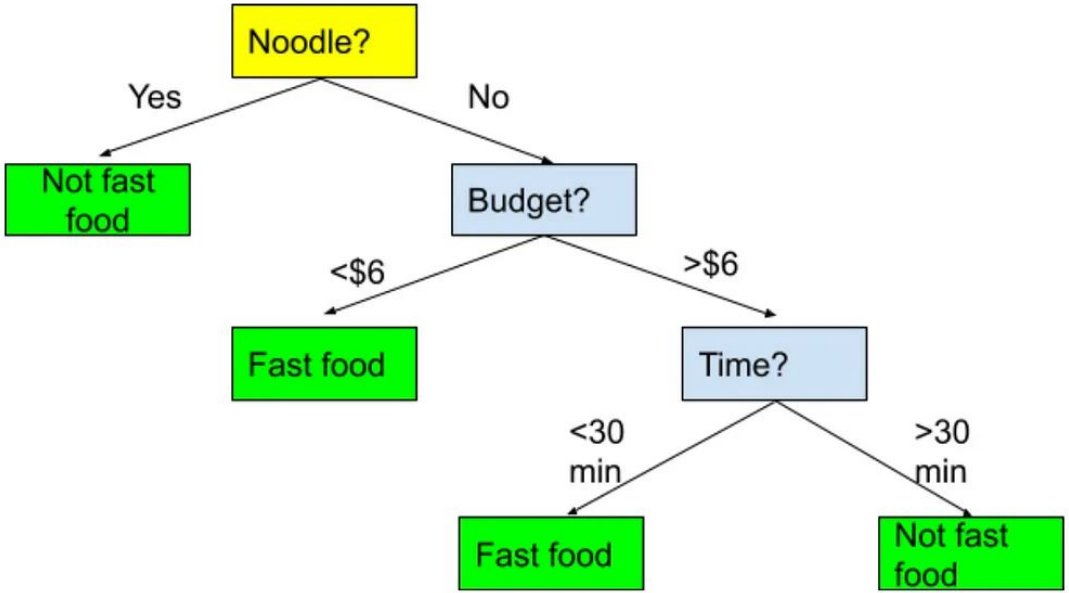


Decision Tree

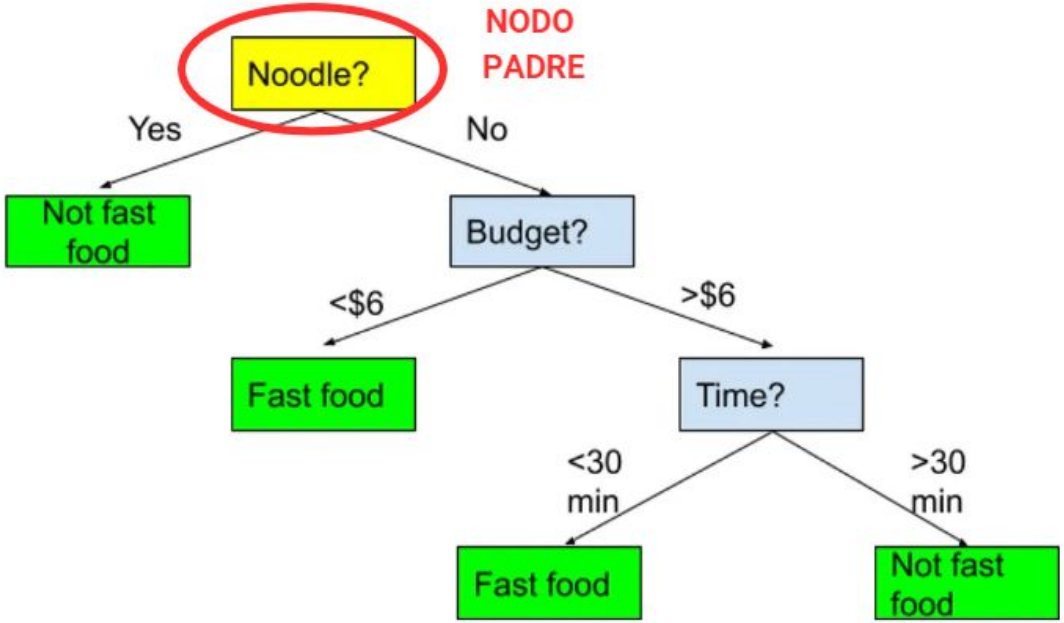
- Possono essere applicati a problemi di regressione e classicazione
- Ci sono due tipi di alberi decisionali:
  - Classification tree
  - Regression tree

Terminologia per i Decision Tree

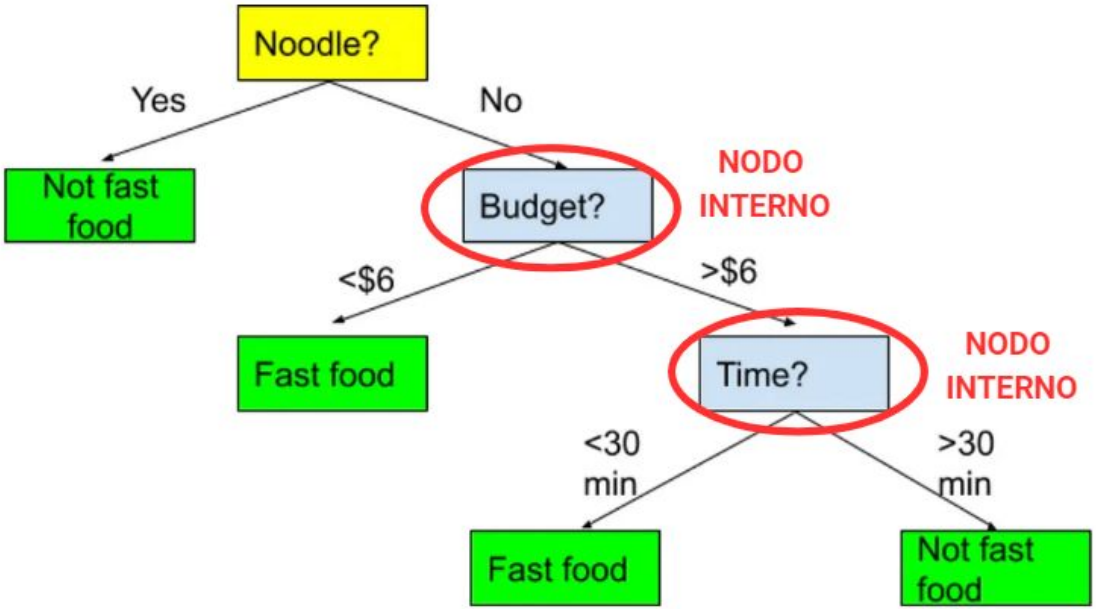
- Nodo padre
- Nodo interno
- Nodo foglia



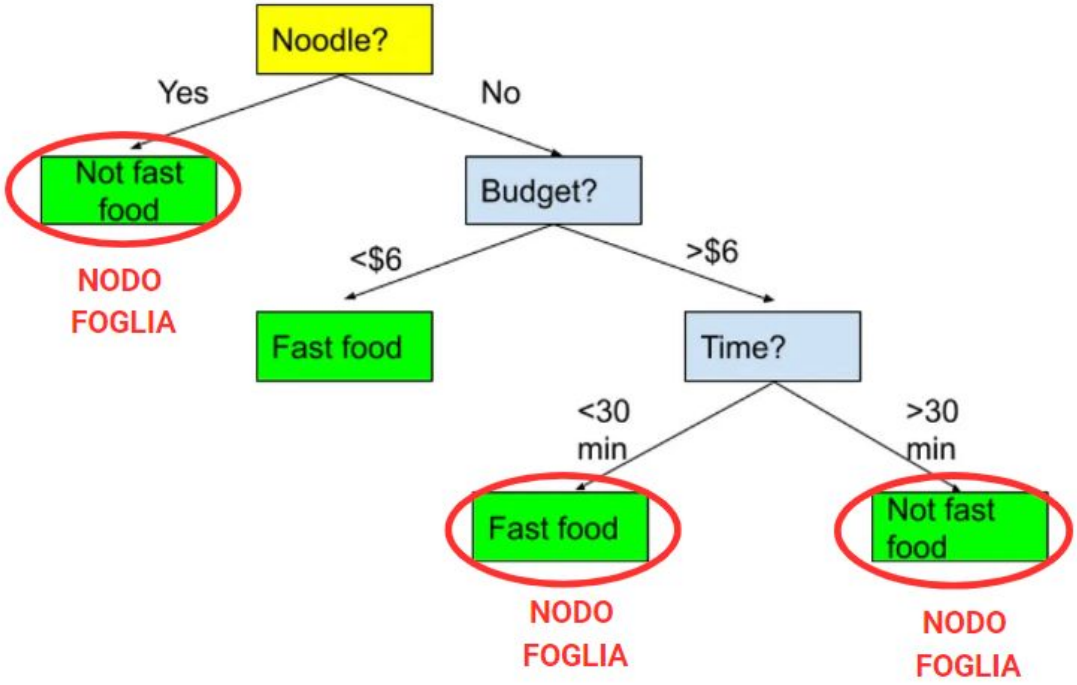
Decision Tree



Decision Tree

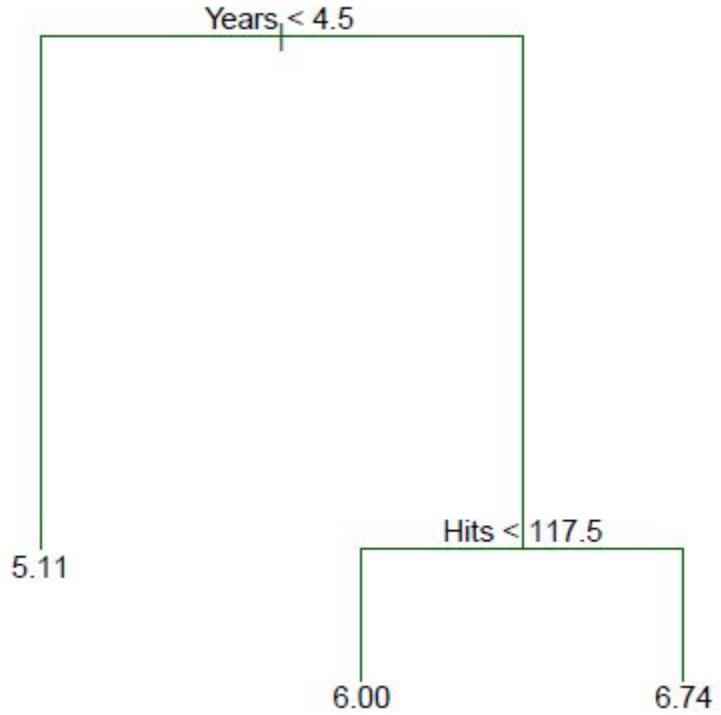


Decision Tree



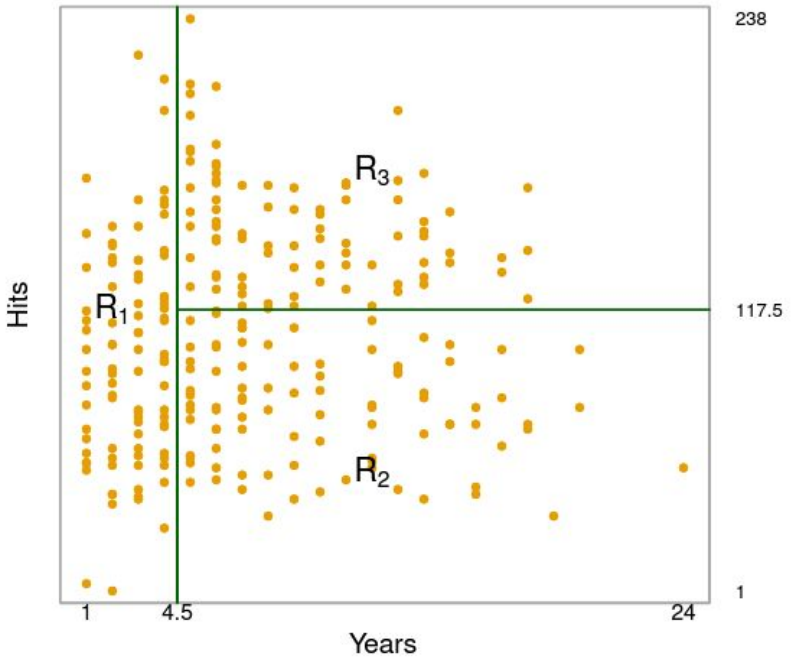
Regression tree: vogliamo prevedere il salario

- Il modello apprende una sequenza di domande if then dove ciascuna domanda coinvolge:
  - Feature
  - Punto di divisione



Risultati

- Queste domande permettono di segmentare lo spazio in 3 regioni:
  - R1: Years<4.5
  - R2: Years>=4.5 e Hits<117.5
  - R3: Years>=4.5 e Hits>=117.5



Fase di training del decision tree

1) Divido lo spazio del predittore in un numero finito di regioni
    - Esempio: **R1,R2 e R3**
2) Per ogni osservazione che cade in una delle regioni, ottengo la stessa previsione, che è la media dei valori delle variabili risposte di training in R

Regression Tree

- L’output finale è quantitativo
- Come ottengo la previsione per un’osservazione del test set?
  - Calcolo la **media delle variabili risposta di training** nella regione a cui l’osservazione del test set appartiene
- L’obiettivo è trovare le regioni che minimizzano il RSS

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

Classification tree

- L’output finale è qualitativo
- Come ottengo la previsione per un’osservazione del test set?
    - Calcolo la **moda delle variabili risposta di training** nella regione a cui l’osservazione del test set appartiene

Classification tree - Gini impurity

L’obiettivo è trovare le regioni che minimizzano il **Gini impurity**

- Misura per quantificare impurità/grado di disordine del nodo
- Se è pari a 0, il nodo è **puro**
  - Le osservazioni del nodo appartengono a un’unica classe
- Se è vicino a 1, il nodo è **impuro**
  - Le osservazioni del nodo appartengono a più classi

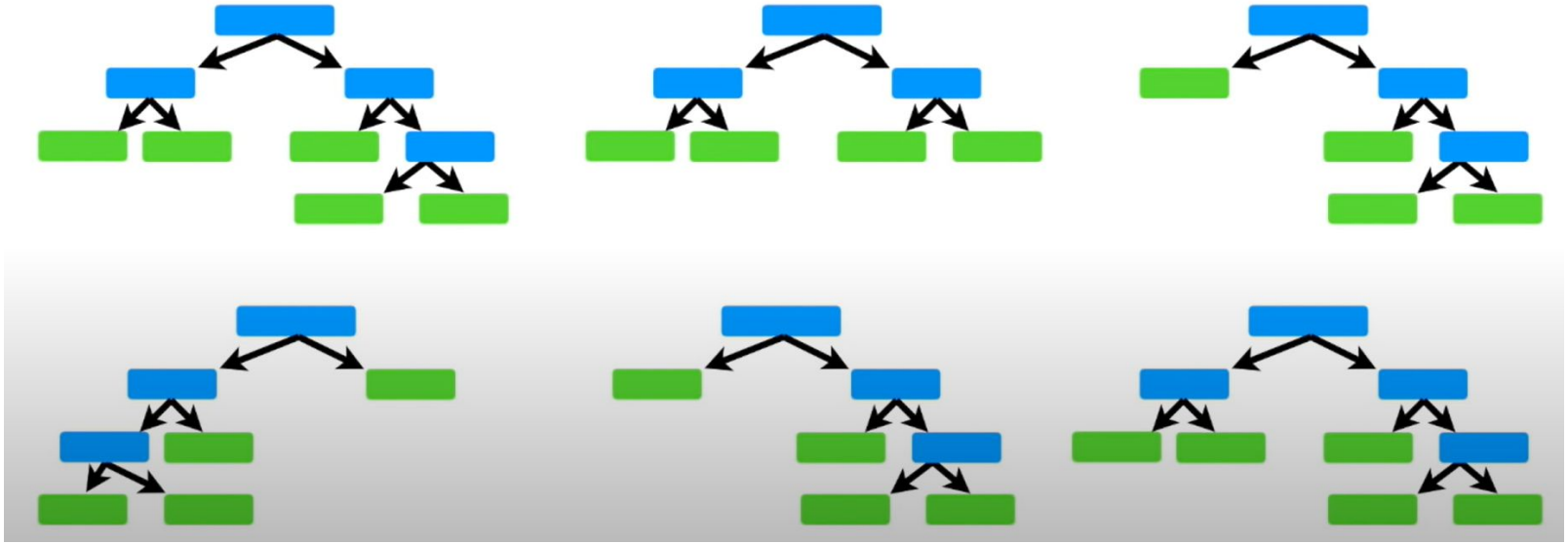
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Vantaggi e Svantaggi

- Facile da spiegare, usare e interpretare
- Si può mostrare il grafico e può essere interpretato anche da una persona senza conoscenze tecniche
- Veloce da allenare
- Ma … non sono allo stesso livello in termini di accuratezza
    - Non sono flessibili quando bisogna classificare nuovi campioni

Random Forest

Combina un insieme di alberi decisionali e aggrega i risultati ottenuti da ogni albero (media per la regressione, moda per la classificazione)



Step 1 - Creare un boostraped dataset

- **Seleziono in modo casuale** dei campioni dal dataset originale
- La dimensione del boostraped dataset è la stessa del dataset originale

Original Dataset

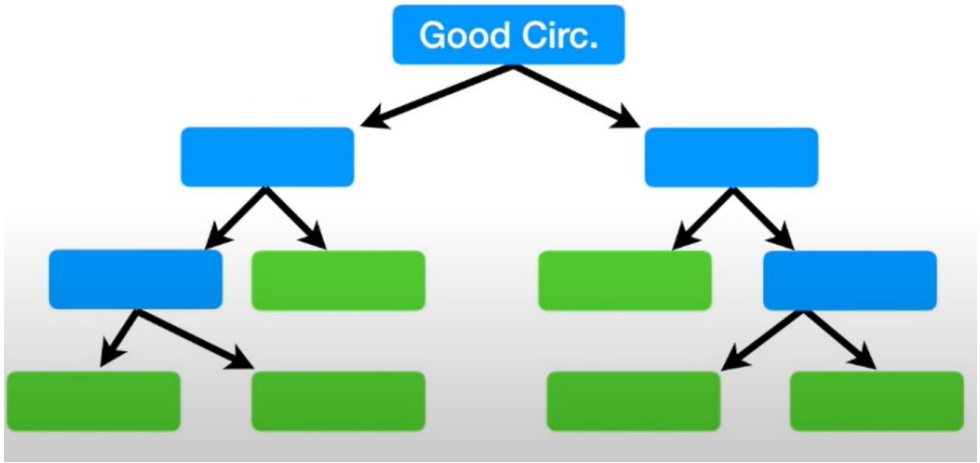
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Step 2 - Creo un albero decisionale usando il boostraped dataset

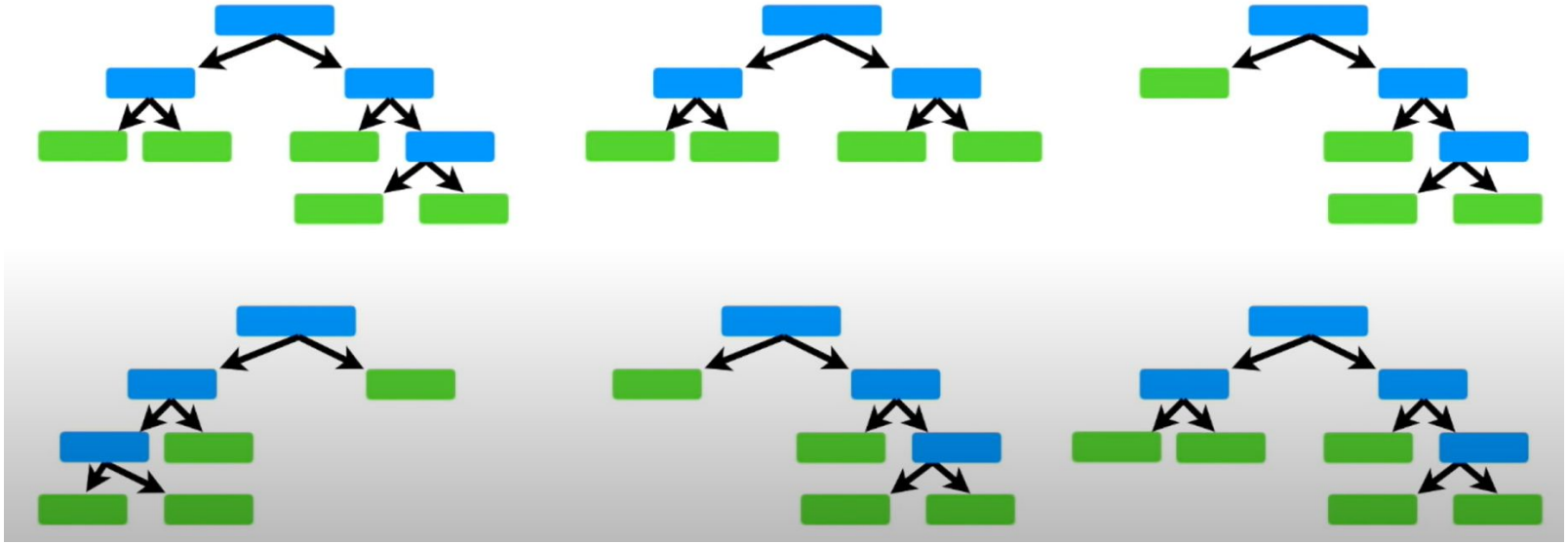
- Considero un sottoinsieme di variabili in ogni passo
- Seleziono una variabile in modo casuale



Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Ripeto lo **Step 1** e **Step 2** per gli altri alberi: creo il **bootstraped dataset** e costruisco un albero che considera un sottoinsieme di variabili

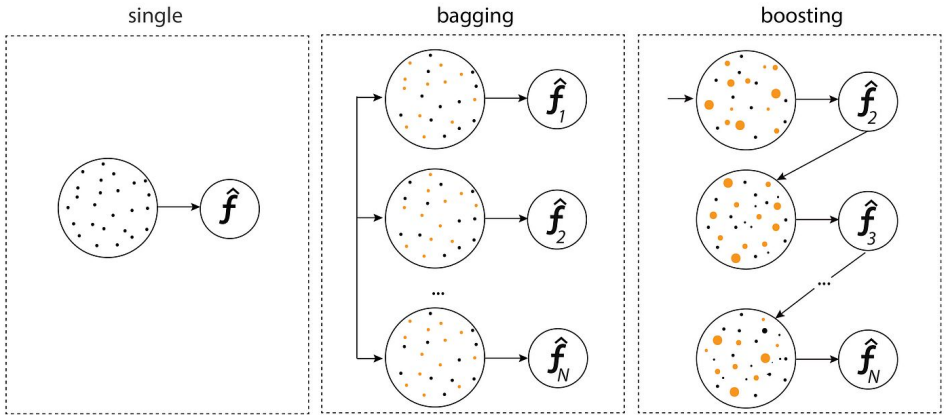


Vantaggi

- È più efficiente di un singolo albero decisionale
- Riduce la variabilità, e quindi riduce l’overfitting
- Utilizzare un insieme di alberi decisionale rende il modello più robusto

Gradient Boosting

- Perché questo nome?
  - **Gradient** perché utilizza il gradiente dell'errore di perdita per minimizzare il suo valore
  - **Boosting** perché è una tecnica ensemble che combina più modelli deboli per creare un modello forte



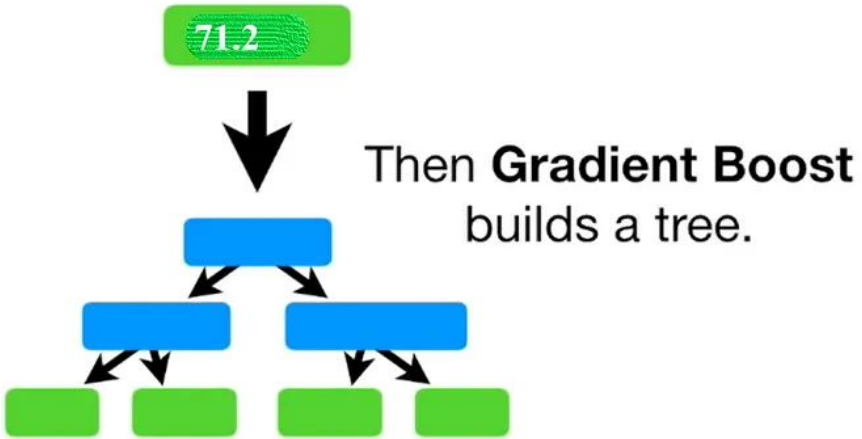
Gradient Boosting

- Perché questo nome?
  - **Gradient** perché utilizza il gradiente dell'errore di perdita per minimizzare il suo valore
  - **Boosting** perché è una tecnica ensemble che combina più modelli deboli per creare un modello forte
    - A differenza del random forest, i modelli sono allenati in sequenza

Step 1 - Inizia con un modello base che fa previsioni

Formula un'ipotesi iniziale per i pesi di tutti i campioni (71.2), che sarà pari alla media del peso all'inizio

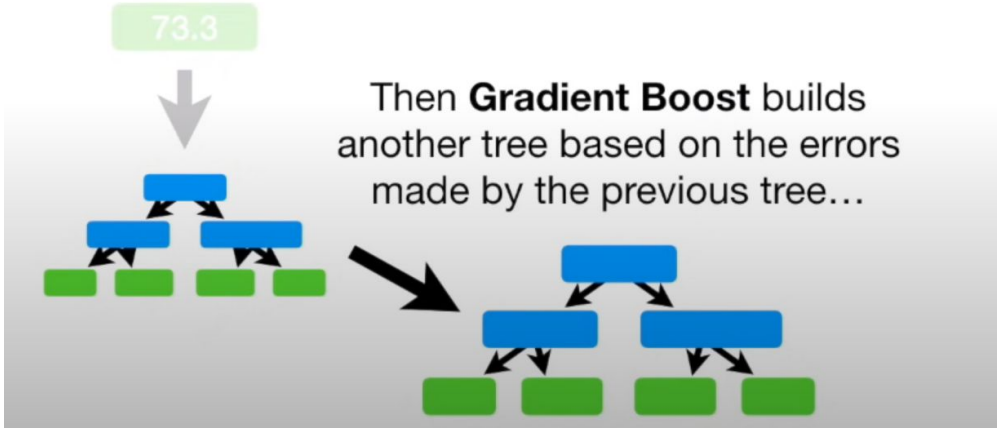
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57



Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

2.1 Errori = (Observed weights - Predicted Weight) = (Observed weights - 71.2)

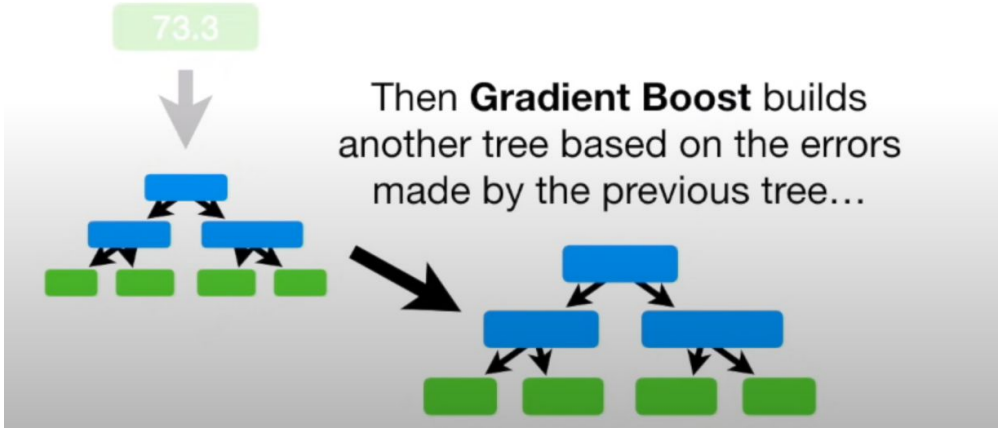
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57



Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

Errori = (Observed weights - Predicted Weight) = (88 - 71.2)

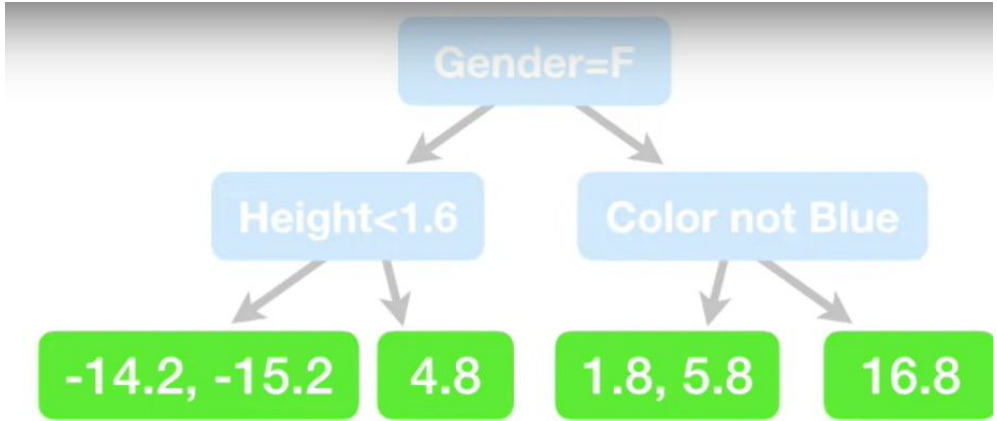
Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	



Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

2.2) Si allena il secondo albero a **prevedere i residui**, invece che i pesi

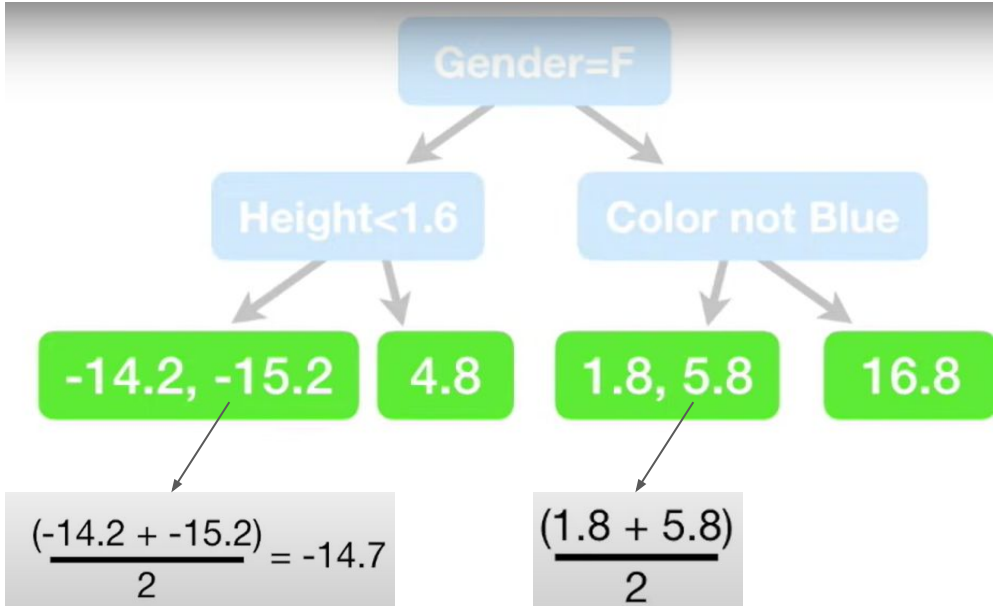
Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2



Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

2.3) Sostituiamo i residui con la **media dei residui** di quello specifico nodo foglia

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2



Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

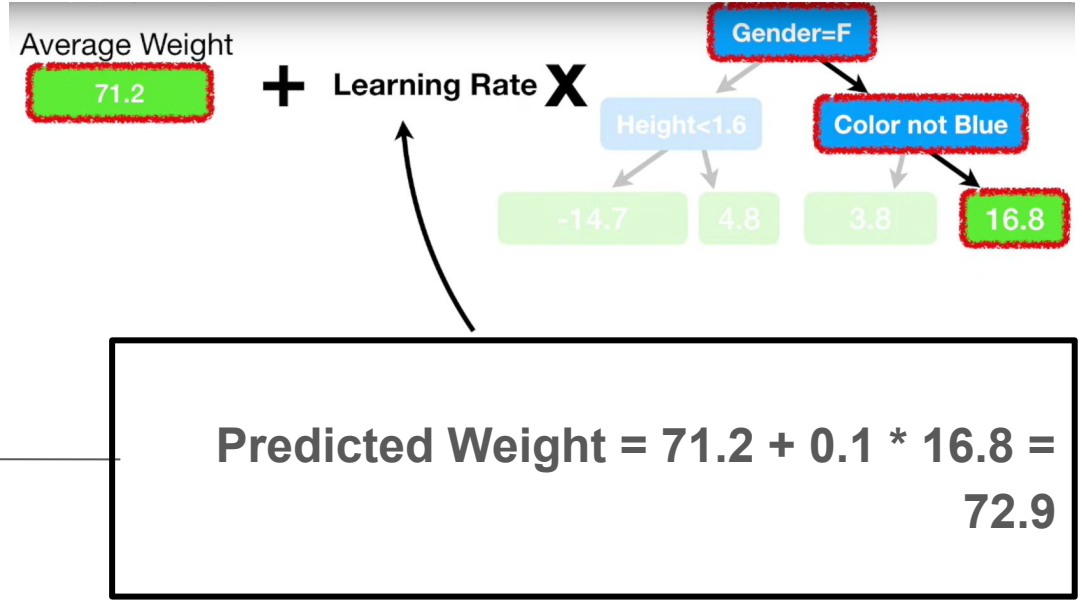
2.4) Combino il peso medio con il secondo albero decisionale per fare la **previsione del peso**

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

per fare la **previsione del peso**
dai dati di training

Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

2.4) Combino il peso medio con il secondo albero decisionale per fare la **previsione del peso**

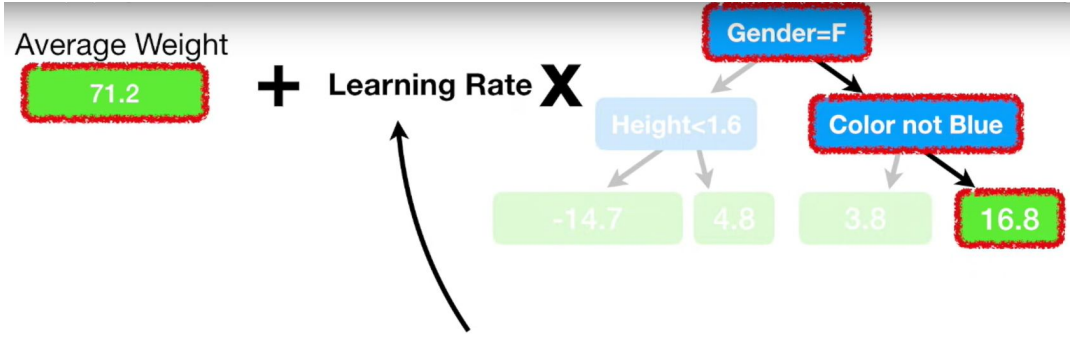


Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

Step 2 - Costruiamo un secondo albero che si basa sugli errori del primo albero

2.5) Calcolo gli errori (o residui)

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

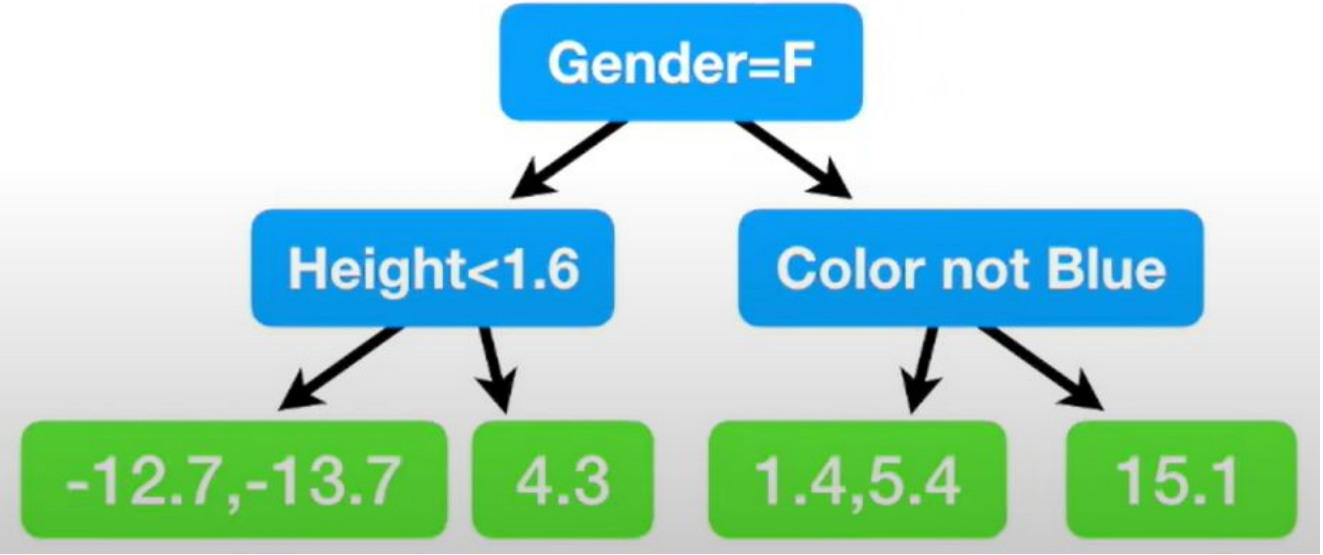


Predicted Weight = 71.2 + 0.1 * 16.8 = 72.9

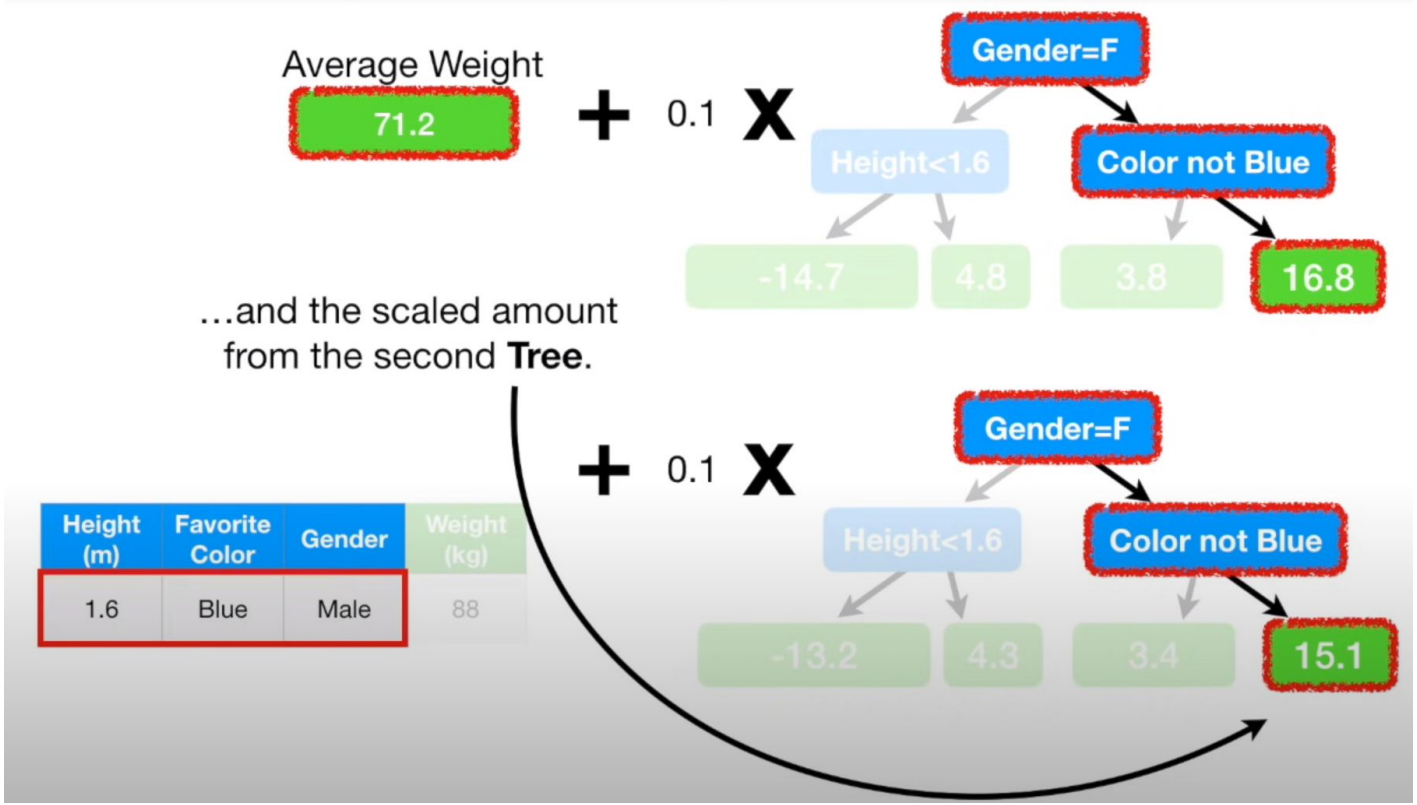
Residual = Observed Weight - Predicted Weight = 88 - 72.9 = 15.1

Step 3: Costruiamo un terzo albero che si basa sugli errori del secondo albero

- Lo alleniamo per fare la previsione degli errori
- Calcoliamo i pesi previsti
- Calcoliamo i residui, che sono la differenza tra pesi osservati e pesi previsti



Step 3: Costruiamo un terzo albero



71.2 +
(0.1 x 16.8) +
(0.1 x 15.1) =
74.4

Vantaggi

- Previsioni molto accurate
- Flessibile
- Ma … è più lento da addestrare, complicato da ottimizzare avendo tanti iperparametri e tende a overfittare
    - ma si può controllare limitando la profondità degli alberi, il numero di alberi, il learning

Curiosità: Differenza tra parametro e iperparametro

- **Parametro**
  - Variabile di configurazione interna al modello
  - appreso automaticamente dal modello durante il training
  - Esempi: pesi all'interno dell'albero decisionale sono parametri
- **Iperparametro**
  - Variabile di configurazione esterna al modello
    - Il cui valore non può essere stimato dal modello!
  - Viene impostato prima dell'addestramento del modello
  - Esempi: il learning rate, numero di alberi, profondità massima dell'albero, numero minimo di campioni necessari per dividere un nodo