

LEC-2b Demand Forecast II

9/5/2024

INDENG 250 2024 Fall
Introduction to Production Planning and Logistics Models
University of California, Berkeley

Huiwen Jia
Assistant Professor
Industrial Engineering & Operations Research

Demand Forecast

Problem

Time Series Models

- ARIMA

ARMA(p,q)

• **AR(p)**
Auto-regressive
p - AR order

$$X_t = \mu + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t$$

• **MA(q)**
Moving average
q - MA order

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

• **ARMA(p,q)**

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

X_t : observation in period t
 ε_t : error term in period t
 φ_i : i^{th} AR coefficient used to fit
 θ_i : i^{th} MA coefficient used to fit

Model Algorithm

General Supervised Learning Models

- Ensemble models

Supervised Learning - Ensemble Learning

"The wisdom of the crowd is the collective opinion of a group of individuals rather than that of a single expert."

"A group of predictors is called an ensemble. Therefore, this Machine Learning technique is known as Ensemble Learning."

"Ensemble methods work best when the predictors are as independent of one another as possible. One way to get diverse classifiers is to train them using very different algorithms. This increases the chance that they will make very different types of errors, improving the ensemble's accuracy."

Bagging **Stacking** **Boosting**

- Gradient boosting

- XGBoost (next lecture)

Decision Tree

Historical umbrella sale data and some factors.
The goal is to provide a forecast once given a new combination of the features.

| Temperature (°F) | Weekday | Raining | Sales |
|------------------|---------|---------|-------|
| 75 | Mon | No | 88 |
| 75 | Wed | No | 76 |
| 75 | Mon | No | 86 |
| 80 | Thu | No | 73 |
| 75 | Wed | No | 75 |
| 75 | Mon | No | 87 |

Decision Tree Structure:

```
graph TD
    Root[Raining?/No] --> Temp75[Temp=75]
    Root --> Weekday[Weekday not Mon]
    Temp75 --> Sales56[56, 57]
    Temp75 --> Sales76[76]
    Weekday --> Sales73[73, 77]
    Weekday --> Sales88[88]
```

Question 1: How do we determine the next node (starting from root)?

Question 2: Should we split at the current node? Or just stop?

ARMA(p,q)

- AR(p)

Auto-regressive
p - AR order

$$X_t = \mu + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t$$

- MA(q)

Moving average
q - MA order

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

- ARMA(p,q)

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

X_t : observation in period t

ε_t : error term in period t

φ_i : i^{th} AR coefficient need to fit

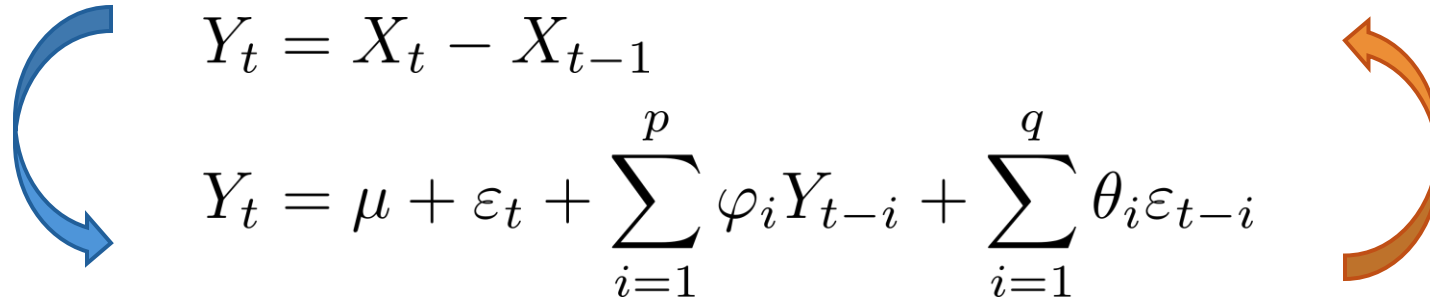
θ_i : i^{th} MA coefficient need to fit

ARI(integrated)MA(p,d,q)

- d is the degree of differencing, e.g., $d = 1$

$$Y_t(d) = Y_t(d-1) - Y_{t-1}(d-1)$$

$$Y_t(d) = (1 - \mathcal{B})^d X_t \text{ where } \mathcal{B} \text{ is a backshift operator}$$


$$Y_t = X_t - X_{t-1}$$
$$Y_t = \mu + \varepsilon_t + \sum_{i=1}^p \varphi_i Y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

ARI(integrated)MA(p,d,q)

- d is the degree of differencing, e.g., $d = 1$

$$Y_t(d) = Y_t(d-1) - Y_{t-1}(d-1)$$

$$Y_t(d) = (1 - \mathcal{B})^d X_t \text{ where } \mathcal{B} \text{ is a backshift operator}$$

$$Y_t = X_t - X_{t-1}$$
$$Y_t = \mu + \varepsilon_t + \sum_{i=1}^p \varphi_i Y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

Order selection & Parameter Estimation:

- Maximum Likelihood Estimation– maximize logarithm of the probability of the observed data
- ACF & PACF (Partial Auto-Correlation Function)
- Information Criterion (p,q) – (not comparable under different d)

- Akaike's Information Criterion (AIC)
- corrected AIC
- Bayesian Information Criterion

$$\text{AIC} = -2 \log(L) + 2(p + q + k + 1),$$

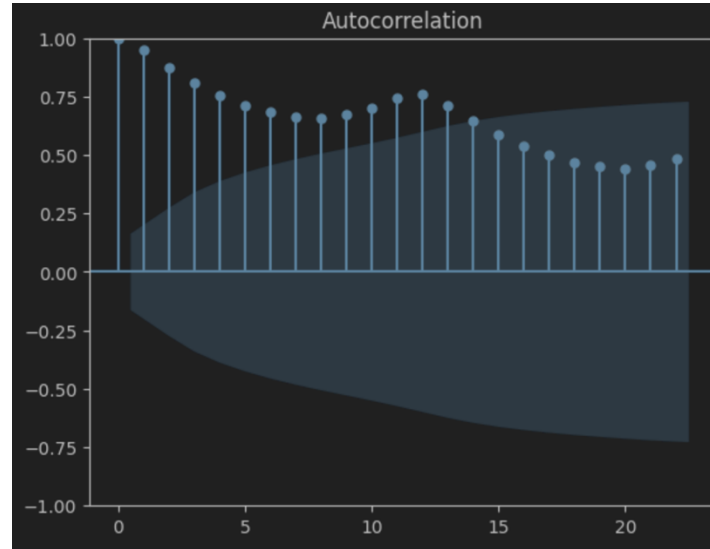
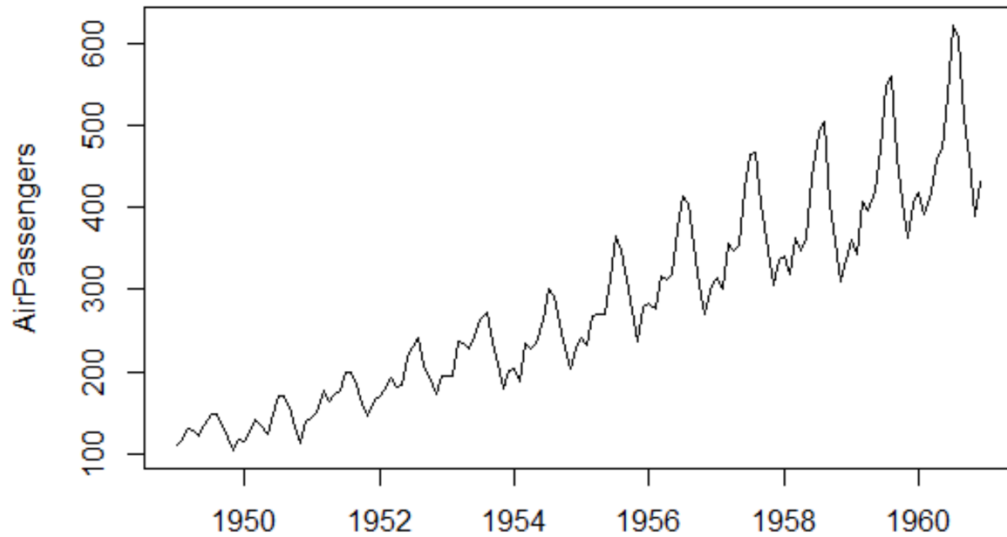
$$\text{AICc} = \text{AIC} + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2},$$

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p + q + k + 1).$$

$$k = \mathbb{1}(\mu \neq 1)$$

ACF & PACF

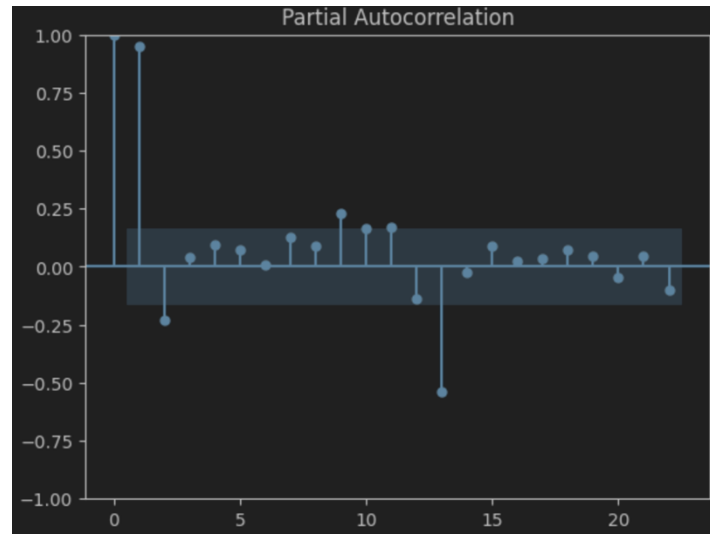
Monthly International Airline Passengers



ACF:

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2},$$

Peaks at multiples of 12 (12, 24, ...) suggest a yearly cycle



PACF:

Peaks at lags 1 and 12 suggest potential autoregressive terms related to the monthly and yearly patterns in the data.

$$\phi_k = \frac{\text{cov}(X_t, X_{t-k} | X_{t-1}, X_{t-2}, \dots, X_{t-k+1})}{\sqrt{\text{var}(X_t | X_{t-1}, X_{t-2}, \dots, X_{t-k+1}) \cdot \text{var}(X_{t-k} | X_{t-1}, X_{t-2}, \dots, X_{t-k+1})}}$$

Highlighting direct correlations.

Supervised Learning - Ensemble Learning

“The wisdom of the crowd is the collective opinion of a group of individuals rather than that of a single expert.”

“A group of predictors is called an ensemble. Therefore, this Machine Learning technique is known as Ensemble Learning.”

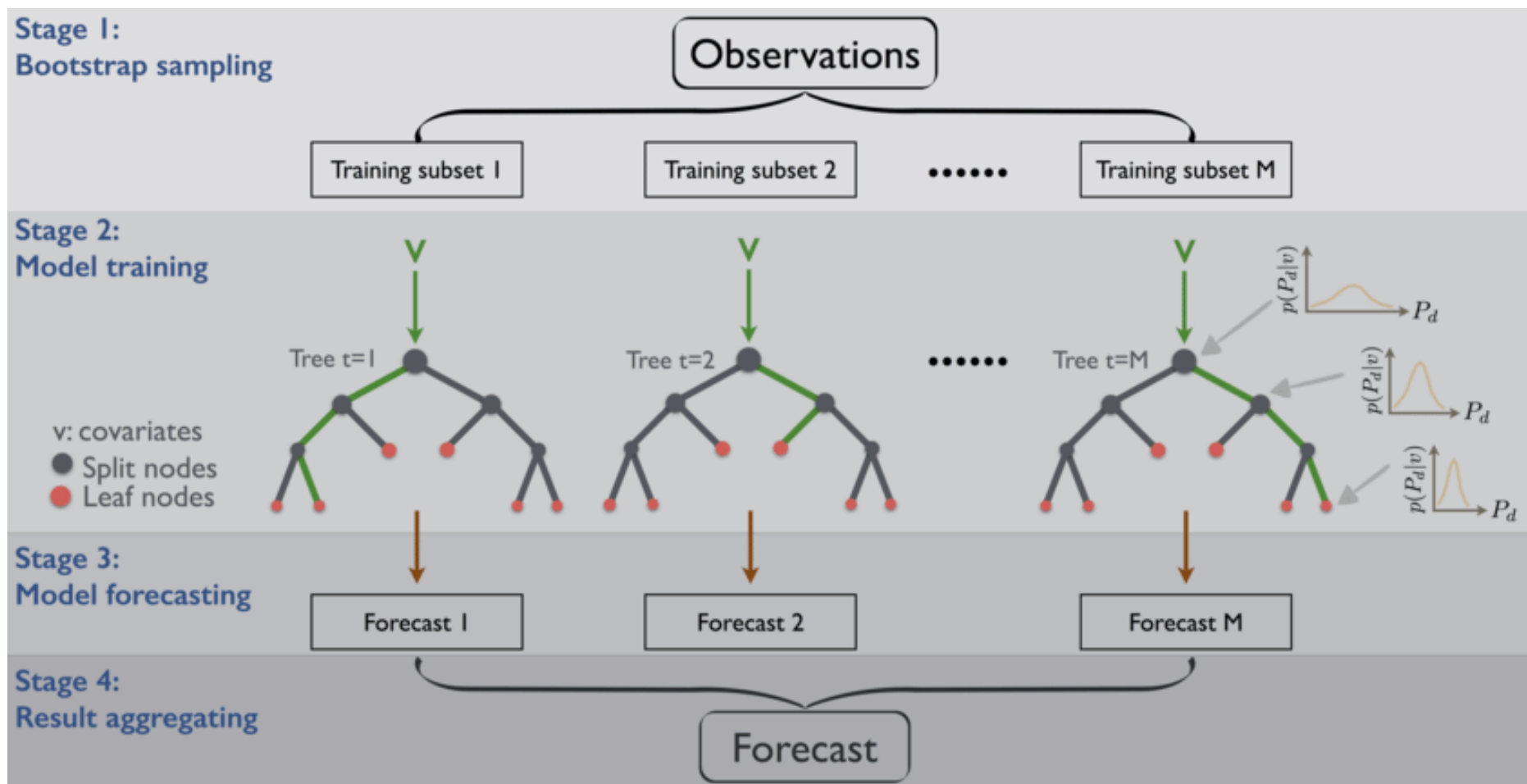
“Ensemble methods work best when the predictors are as independent of one another as possible. One way to get diverse classifiers is to train them using very different algorithms. This increases the chance that they will make very different types of errors, improving the ensemble’s accuracy.”

Bagging

Stacking

Boosting

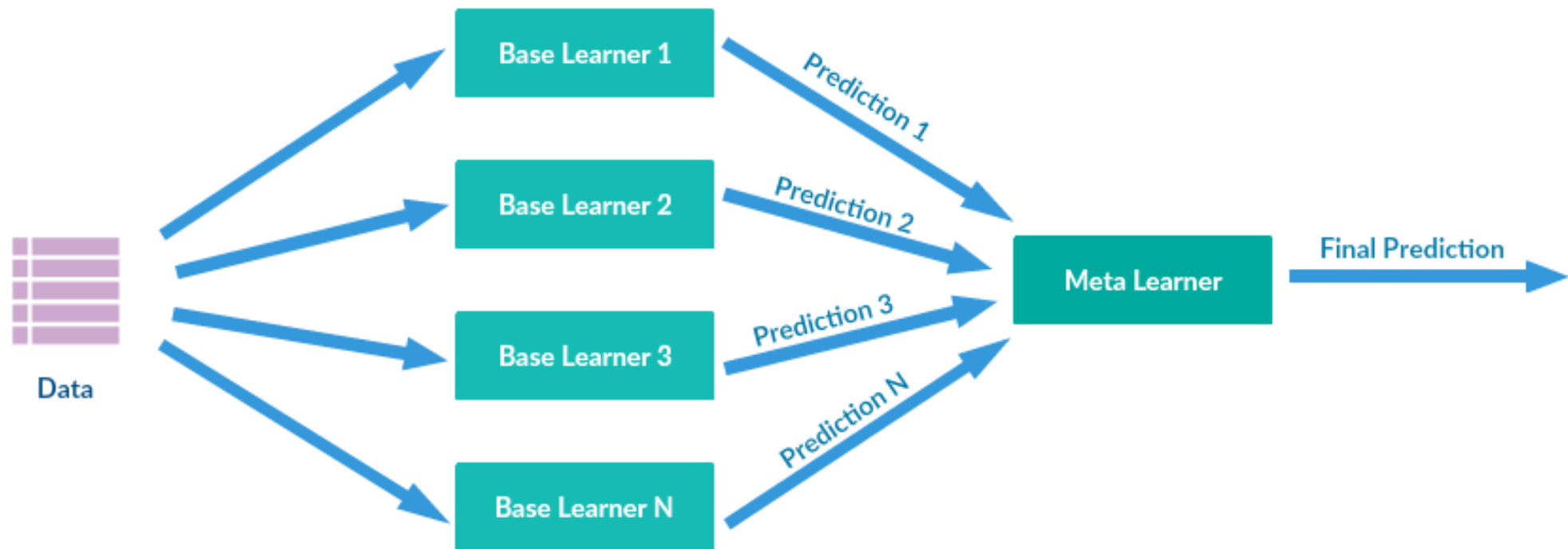
Ensemble Learning - Bagging



E.g.
Random
Forest

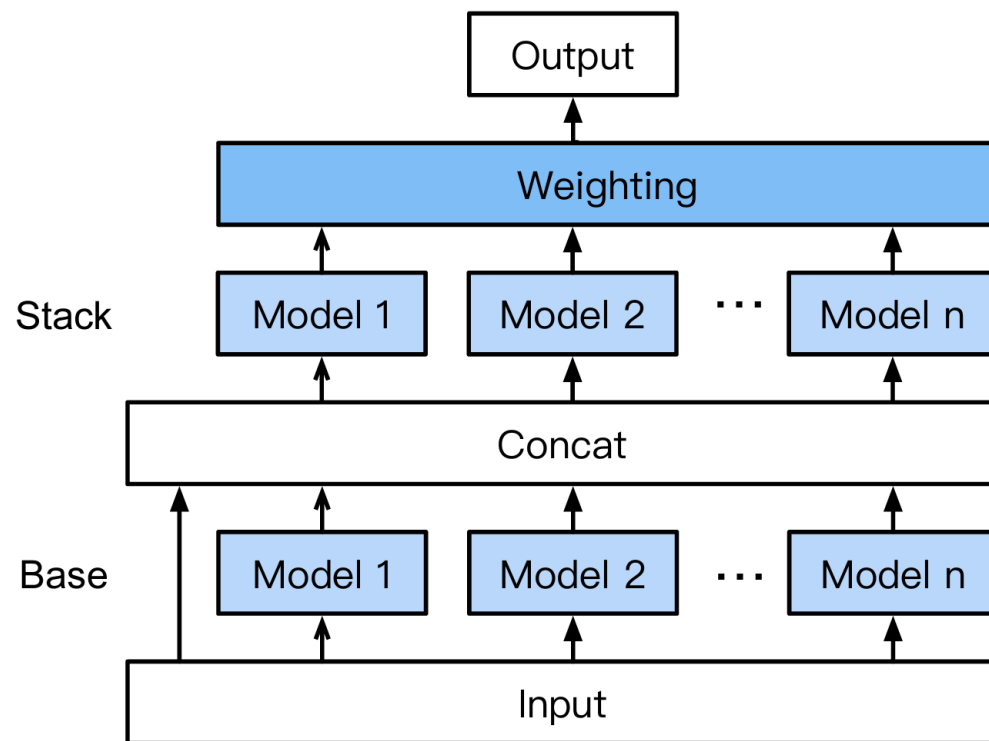
- Bootstrap: sample with replacement
- Training are independent
- Hard voting classifier (for classification)
- Averaging or weighted averaged (for regression)

Ensemble Learning - Stacking



- Base models are trained on different portions of the training data.
- A meta learner is then trained to generate the final output.
- Most cases use multiple types of models.

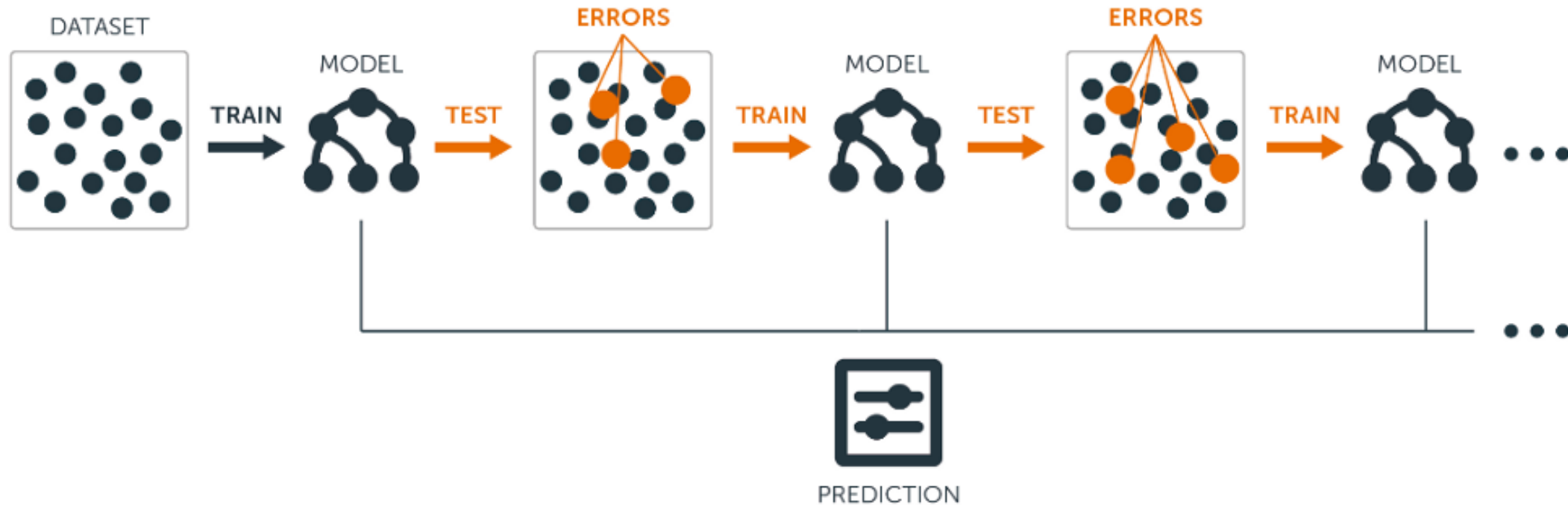
Ensemble Learning – Stacking (with Layers)



Amazon AWS
AutoGluon, an open source AutoML library

Figure 2. AutoGluon's multi-layer stacking strategy, shown here using two stacking layers and n types of base learners.

Ensemble Learning - Boosting



E.g, AdaBoost, Gradient Boosting, XGBoost

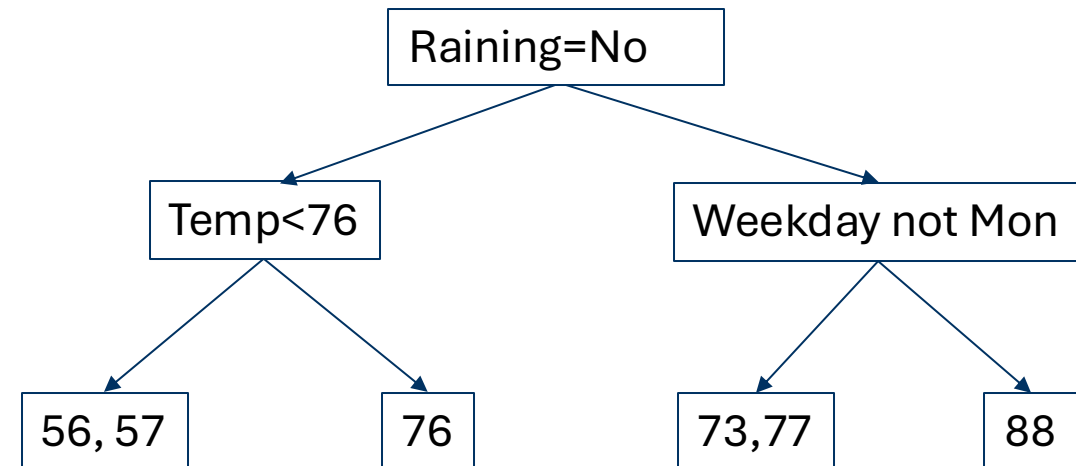
- The models are trained sequentially.
- Training datasets are dependent on the model trained in the previous round.
- Outputs of models trained in different rounds are summed together with weights.

Decision Tree

Historical umbrella sale data and some factors.
The goal is to provide a forecast once given a new combination of the features.



| Temperature (F) | Weekday | Raining | Demand |
|-----------------|---------|---------|--------|
| 76 | Mon | Yes | 88 |
| 76 | Wed | No | 76 |
| 75 | Mon | No | 56 |
| 88 | Fri | Yes | 73 |
| 75 | Wed | Yes | 77 |
| 74 | Mon | No | 57 |



Question 1: How do we determine the next node (starting from root)?

Question 2: Should we split at the current node? Or just stop?

How to determine and split a node?

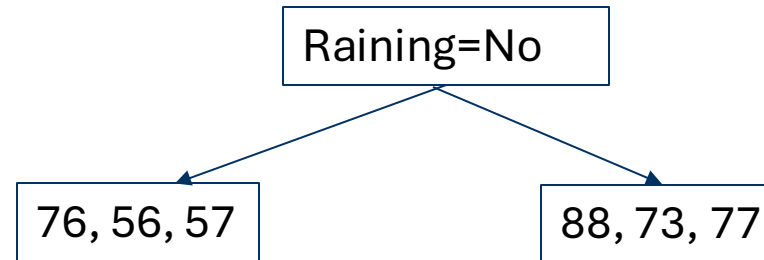
Measure of impurity (for regression) is deviance (over **average**).

| Temperature (F) | Weekday | Raining | Demand |
|-----------------|---------|---------|--------|
| 76 | Mon | Yes | 88 |
| 76 | Wed | No | 76 |
| 75 | Mon | No | 56 |
| 88 | Fri | Yes | 73 |
| 75 | Wed | Yes | 77 |
| 74 | Mon | No | 57 |

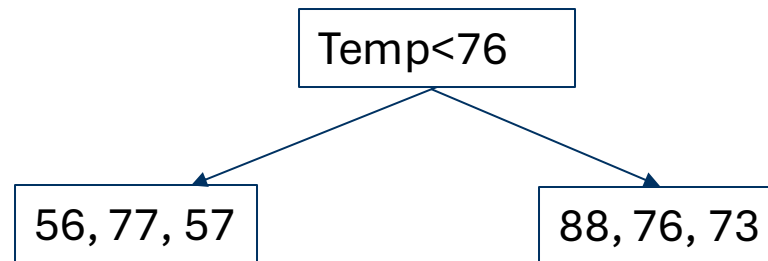
88, 76, 56, 73, 77, 57

Equivalent to LSE!

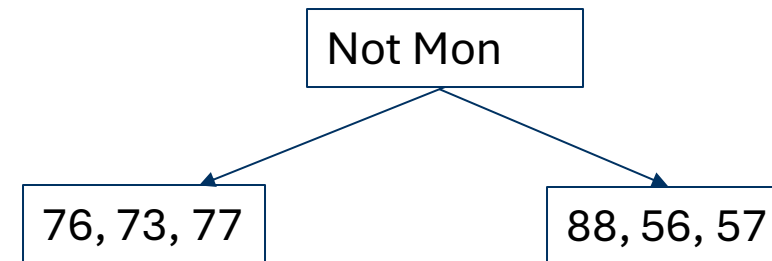
Deviance = 774.84



Deviance = 254 + 120.67 = 374.67



Deviance = 280.67 + 126 = 406.67



Deviance = 8.67 + 662 = 670.67

Gradient Boosting Decision Tree (GBDT) – Iteration 0

GBDT is a Boosting model.

$$F_{m-1}(x) = \sum_{k=1}^{m-1} \alpha_k f_k(x)$$

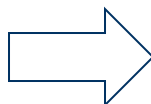
Iteration m:

$$(\alpha_m, f_m(x)) = \arg \min_{\alpha, f(x)} E_{y,X} \psi(y, F_{m-1}(x) + \alpha f(x))$$

$$F_m(x) = F_{m-1}(x) + \alpha_m f_m(x)$$

F0 = Initial Model = Taking the mean

| Temperature (F) | Weekday | Raining | Demand |
|-----------------|---------|---------|--------|
| 76 | Mon | Yes | 88 |
| 76 | Wed | No | 76 |
| 75 | Mon | No | 56 |
| 88 | Fri | Yes | 73 |
| 75 | Wed | Yes | 77 |
| 74 | Mon | No | 57 |



| Temperature (F) | Weekday | Raining | Demand | F0 | PR0 |
|-----------------|---------|---------|--------|------|-------|
| 76 | Mon | Yes | 88 | 71.2 | 16.8 |
| 76 | Wed | No | 76 | 71.2 | 4.8 |
| 75 | Mon | No | 56 | 71.2 | -15.2 |
| 88 | Fri | Yes | 73 | 71.2 | 1.8 |
| 75 | Wed | Yes | 77 | 71.2 | 5.8 |
| 74 | Mon | No | 57 | 71.2 | -14.2 |



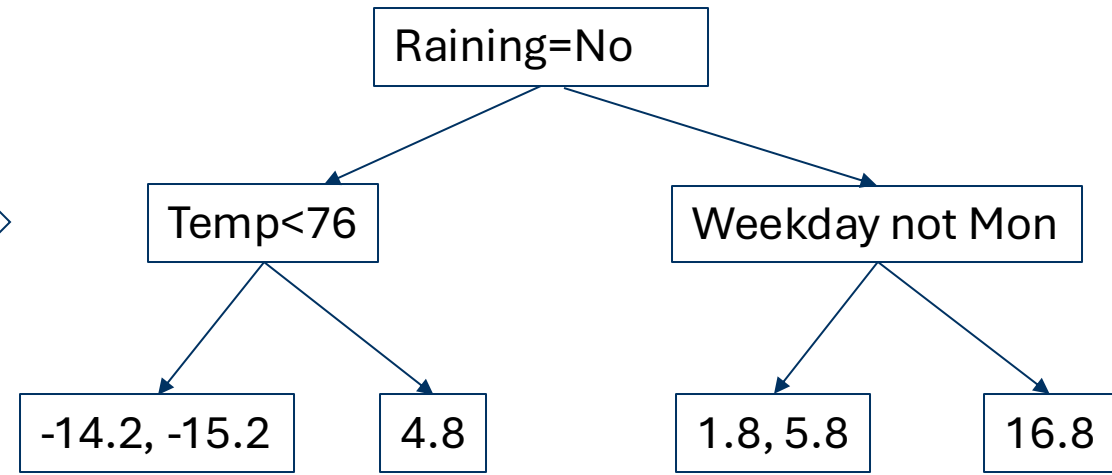
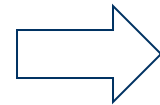
$$y_i - \hat{y}_i$$

Pseudo Residual (PR) = True Value – Predicted Value

GBDT – Iteration 1 - Structure

Fit PR0 into a decision tree (up to four leaves)

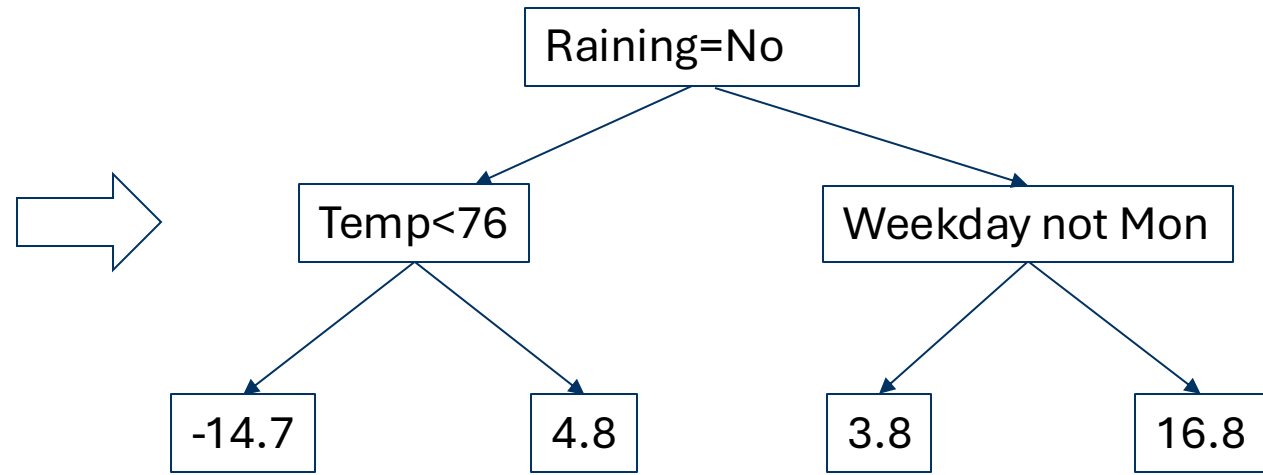
| Temperature (F) | Weekday | Raining | PR0 |
|-----------------|---------|---------|-------|
| 76 | Mon | Yes | 16.8 |
| 76 | Wed | No | 4.8 |
| 75 | Mon | No | -15.2 |
| 88 | Fri | Yes | 1.8 |
| 75 | Wed | Yes | 5.8 |
| 74 | Mon | No | -14.2 |



Pseudo Residual (PR) = True Value – Current Predicted Value

GBDT – Iteration 1 - Leaf Weights

| Temperature (F) | Weekday | Raining | PR0 |
|-----------------|---------|---------|-------|
| 76 | Mon | Yes | 16.8 |
| 76 | Wed | No | 4.8 |
| 75 | Mon | No | -15.2 |
| 88 | Fri | Yes | 1.8 |
| 75 | Wed | Yes | 5.8 |
| 74 | Mon | No | -14.2 |



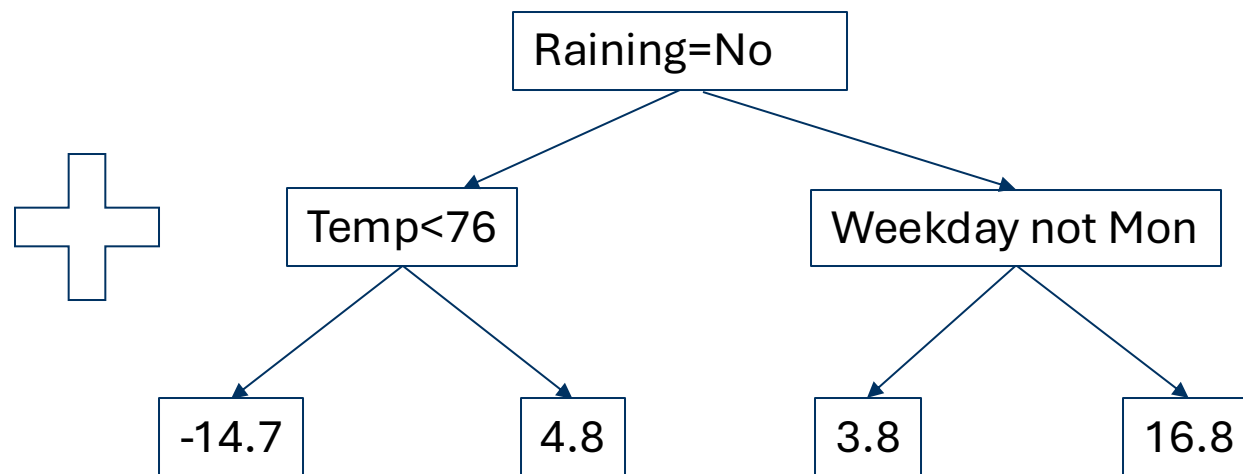
Averaging the residuals on each leaf → output of the tree

GBDT – Iteration 1 – Generate Output

$$F1(x) = F0(x) + \gamma_1 \times \text{Output of DT}(x)$$

Hyper parameter: Learning rate = 0.1

| Temperature (F) | Weekday | Raining | Demand | F0 |
|-----------------|---------|---------|--------|------|
| 76 | Mon | Yes | 88 | 71.2 |
| 76 | Wed | No | 76 | 71.2 |
| 75 | Mon | No | 56 | 71.2 |
| 88 | Fri | Yes | 73 | 71.2 |
| 75 | Wed | Yes | 77 | 71.2 |
| 74 | Mon | No | 57 | 71.2 |



$$F1((76, \text{Mon}, \text{Yes})) = 71.2 + 0.1 \times 16.8 = 72.9$$

$$F1((76, \text{Wed}, \text{No})) = 71.2 + 0.1 \times 4.8 = 71.7$$

$$F1((75, \text{Mon}, \text{No})) = 71.2 + 0.1 \times -14.7 = 69.7$$

$$F1((88, \text{Fri}, \text{Yes})) = 71.2 + 0.1 \times 3.8 = 71.6$$

$$F1((75, \text{Wed}, \text{Yes})) = 71.2 + 0.1 \times 3.8 = 71.6$$

$$F1((74, \text{Mon}, \text{No})) = 71.2 + 0.1 \times -14.7 = 69.7$$

GBDT – After Iteration 1

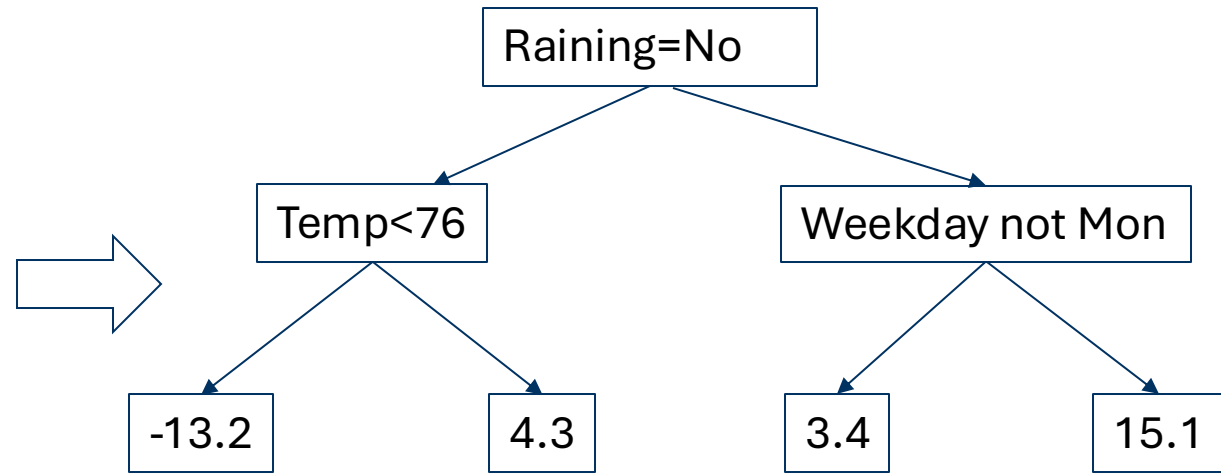
So after building the first DT, we obtain...

| Temperature (F) | Weekday | Raining | Demand | F0 | PR0 | F1 | PR1 |
|-----------------|---------|---------|--------|------|-------|------|-------|
| 76 | Mon | Yes | 88 | 71.2 | 16.8 | 72.9 | 15.1 |
| 76 | Wed | No | 76 | 71.2 | 4.8 | 71.7 | 4.3 |
| 75 | Mon | No | 56 | 71.2 | -15.2 | 69.7 | -13.7 |
| 88 | Fri | Yes | 73 | 71.2 | 1.8 | 71.6 | 1.4 |
| 75 | Wed | Yes | 77 | 71.2 | 5.8 | 71.6 | 5.4 |
| 74 | Mon | No | 57 | 71.2 | -14.2 | 69.7 | -12.7 |

GBDT – Iteration 2 – Structure & Leaf Weights

Fit PR1 into a decision tree (up to four leaves)

| Temperature (F) | Weekday | Raining | PR1 |
|-----------------|---------|---------|-------|
| 76 | Mon | Yes | 15.1 |
| 76 | Wed | No | 4.3 |
| 75 | Mon | No | -13.7 |
| 88 | Fri | Yes | 1.4 |
| 75 | Wed | Yes | 5.4 |
| 74 | Mon | No | -12.7 |



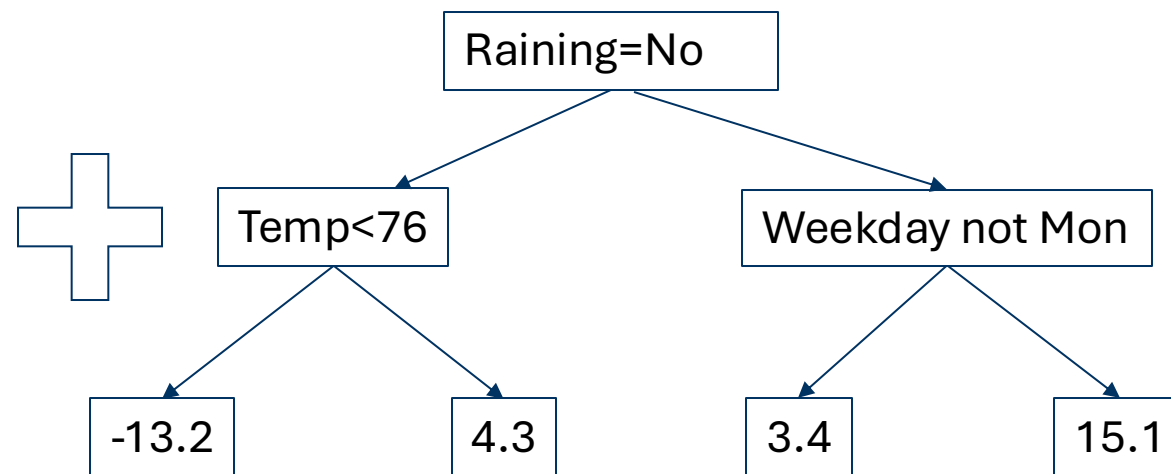
Averaging the residuals on each leaf...

GBDT – Iteration 2 – Generate Output

$$F2(x) = F1(x) + \gamma_2 \times \text{Output of DT}(x)$$

Hyper parameter - Learning rate = 0.1

| Temperature (F) | Weekday | Raining | Demand | F0 |
|-----------------|---------|---------|--------|------|
| 76 | Mon | Yes | 88 | 71.2 |
| 76 | Wed | No | 76 | 71.2 |
| 75 | Mon | No | 56 | 71.2 |
| 88 | Fri | Yes | 73 | 71.2 |
| 75 | Wed | Yes | 77 | 71.2 |
| 74 | Mon | No | 57 | 71.2 |



$$F2((76, \text{Mon}, \text{Yes})) = 72.9 + 0.1 \times 15.1 = 74.4$$

$$F2((76, \text{Wed}, \text{No})) = 71.7 + 0.1 \times 4.3 = 72.1$$

$$F2((75, \text{Mon}, \text{No})) = 69.7 + 0.1 \times -13.2 = 68.4$$

$$F2((88, \text{Fri}, \text{Yes})) = 71.6 + 0.1 \times 3.4 = 71.9$$

$$F2((75, \text{Wed}, \text{Yes})) = 71.6 + 0.1 \times 3.4 = 71.9$$

$$F2((74, \text{Mon}, \text{Yes})) = 69.7 + 0.1 \times -13.2 = 68.4$$

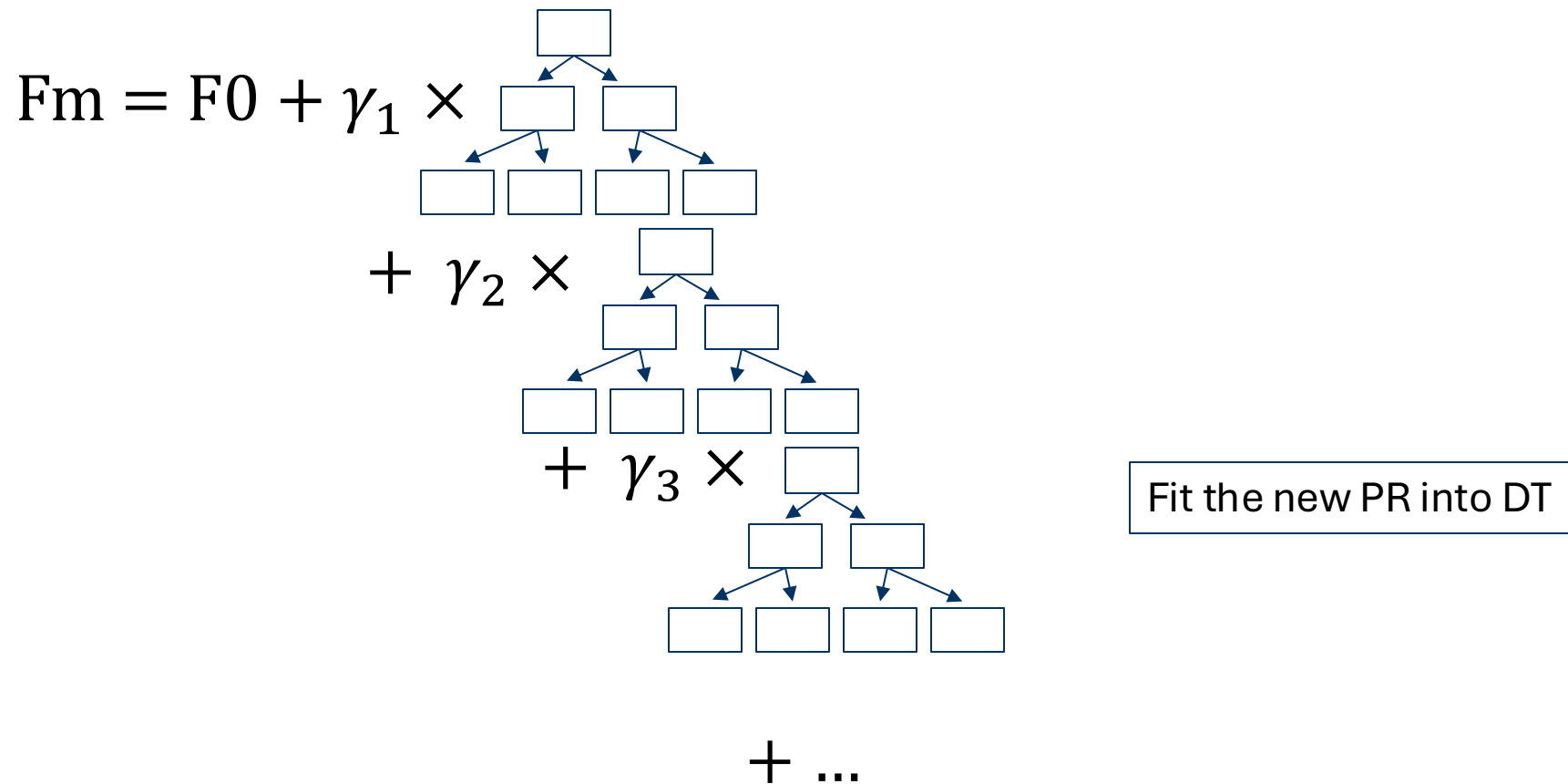
GBDT – After Iteration 2

So after building the second DT, we obtain...

| Temperature (F) | Weekday | Raining | Demand | F0 | PR0 | F1 | PR1 | F2 | PR2 |
|-----------------|---------|---------|--------|------|-------|------|-------|------|-------|
| 76 | Mon | Yes | 88 | 71.2 | 16.8 | 72.9 | 15.1 | 74.4 | 13.6 |
| 76 | Wed | No | 76 | 71.2 | 4.8 | 71.7 | 4.3 | 72.1 | 3.9 |
| 75 | Mon | No | 56 | 71.2 | -15.2 | 69.7 | -13.7 | 68.4 | -12.4 |
| 88 | Fri | Yes | 73 | 71.2 | 1.8 | 71.6 | 1.4 | 71.9 | 1.1 |
| 75 | Wed | Yes | 77 | 71.2 | 5.8 | 71.6 | 5.4 | 71.9 | 5.1 |
| 74 | Mon | No | 57 | 71.2 | -14.2 | 69.7 | -12.7 | 68.4 | -11.4 |

Notice the PR's are shrinking: Small steps towards the right direction!

GBDT – Boosting Structure



Stop until the pre-specified #DTs or the PR stops improving!

Why GBDT works?

Decision Tree

- The model trained in each iteration is a decision tree.

Why GBDT works?

Decision Tree

- The model trained in each iteration is a decision tree.

Boosting

- The models are trained sequentially.
- Training datasets are dependent on the model trained in the previous round.
- Outputs of models trained in different rounds are summed together with weights.

Why GBDT works?

Decision Tree

- The model trained in each iteration is a decision tree.

Boosting

- The models are trained sequentially
- Training datasets are dependent on the model trained in the previous round.
- Outputs of models trained in different rounds are summed together with weights

What does the term **gradient** mean here?



Gradient descent algorithm in continuous optimization.

*GD vs GBDT

- We will provide supplement materials for this page in a separate file.
- This part will not be included in the exam.
- Basically, if a page of which the title is indicated by *, then this page is just for sharing interesting (or supportive) knowledge and the content will not be covered in the exam.
- This page is one example, and the page regarding convex optimization in 2a lecture note is an example too.

*General Gradient Boosting Algorithmic Steps

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, number of iterations M .

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma).$$

2. For $m = 1$ to M :

1. Compute so-called *pseudo-residuals*:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n.$$

2. Fit a base learner (or weak learner, e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.

3. Compute multiplier γ_m by solving the following **one-dimensional optimization** problem:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)).$$

4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Why Cover GBDT in this Class?

Performance:

- Works exceptionally well in practice.
- Won a series of Kaggle competitions.
- More robust and explainable.

Theory:

Closely related to (derived from) optimization algorithms.

Take Away

This class:

- Time series forecast model
 - ARIMA
- Supervised learning model
 - Ensemble Learning
 - Gradient Boosting
 - *Gradient Descent
 - Gradient Boosting Decision Tree (Algorithm)

Next class:

- XGBoost
 - XGBoost Decision Tree (Algorithm)
 - *Newton's Method
- Inventory Management– EOQ