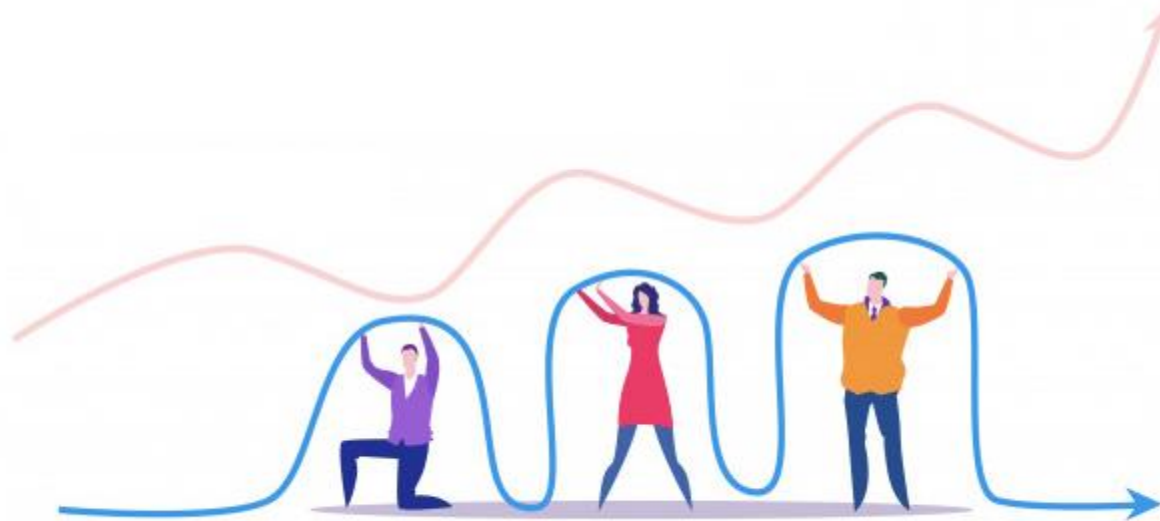




Module 5

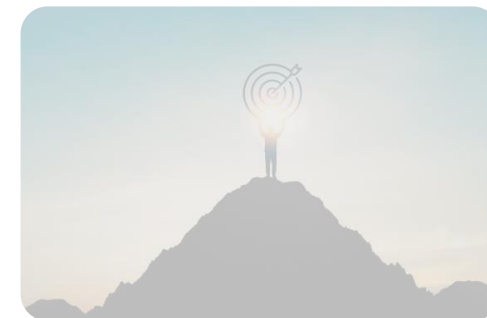
Linear Regression (ML approach)





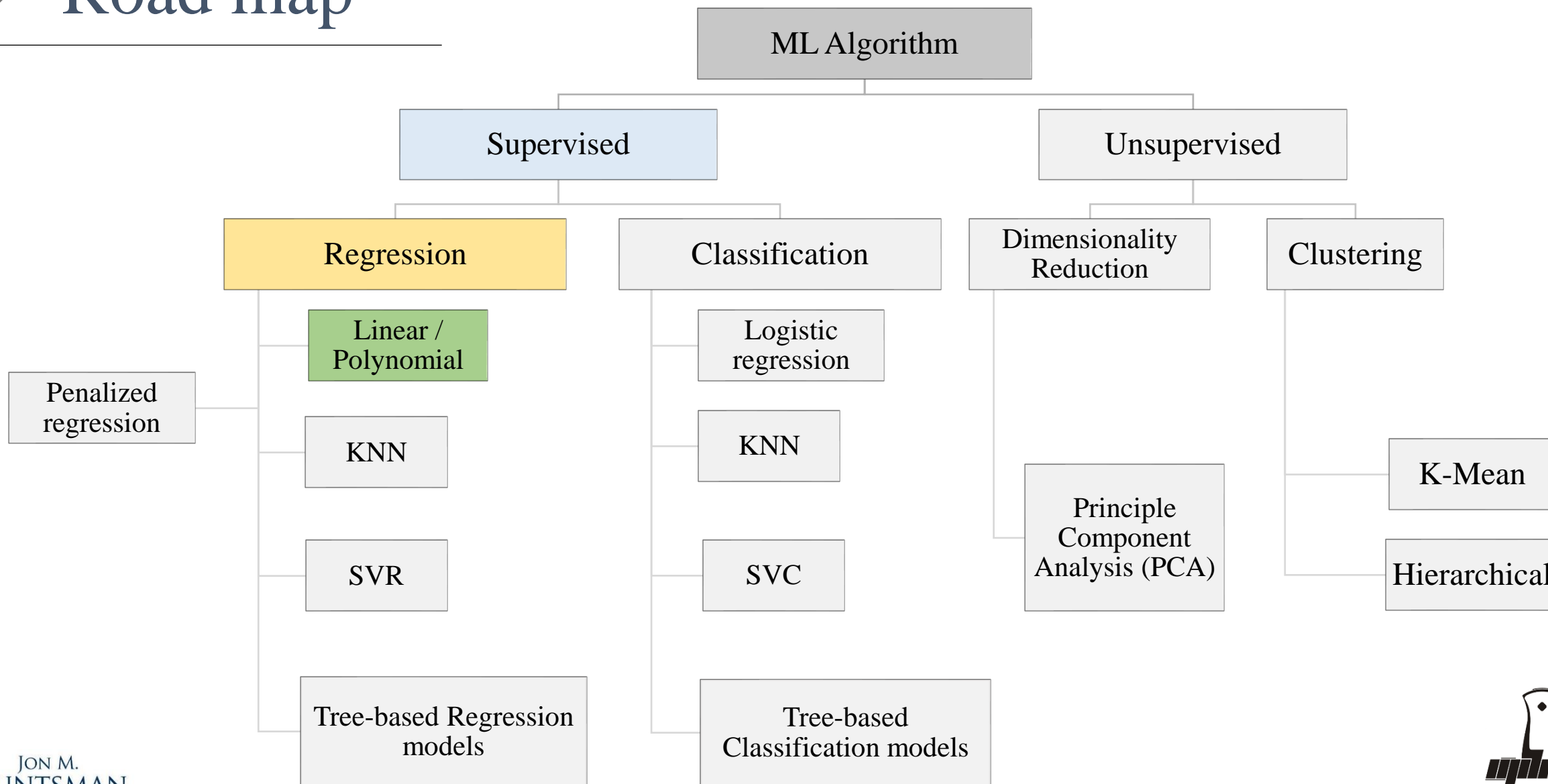
Class Modules

- Module 1- Introduction to Deep Learning
- Module 2- Setting up Machine Learning Environment
- Module 3- Linear Regression (Econometrics approach)
- Module 4- Machine Learning Fundamentals
- **Module 5- Linear Regression (Machine Learning approach)**
- Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)
- Module 7- Logistic Regression
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)





Road map



Part I

Linear regression: Machine Learning approach



Linear Models

The linear model is an important example of a **parametric** model

- We have a collection of labeled examples $\{(X_i, y_i)\}_{i=1}^N$, where
 - N is the size of the collection
 - X_i is the **D-dimensional** feature vector
 - y_i is a real-valued target

$$f_{w,b}(X) = \mathbf{W}X + b$$

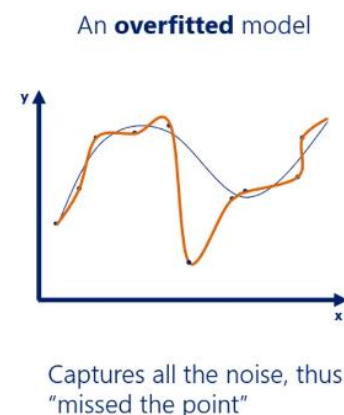
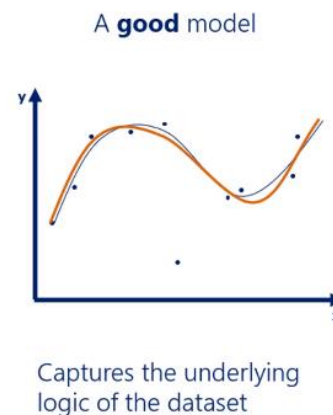
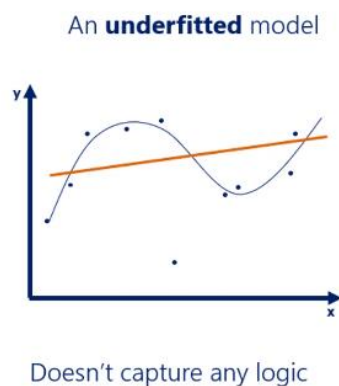
The model $f_{w,b}$ is a linear combination of features and parameterized by **W** and **b**

- **W** is a D-dimensional vector of parameters
- **b** is a real number

→ Linear Models (cont'd)

$$f_{w,b}(X) = \textcolor{red}{W}X + \textcolor{green}{b}$$

- The model is specified with $D+1$ parameter.
- We estimate the parameters (W^*, b^*) by fitting the model to training data.
- Although it is almost never correct, a linear model often serves as a **simple** and **interpretable** approximation the unknown true $f(X)$.
- It may seem overly simplistic, but linear regression is extremely useful both conceptually and practically.
- Linear regression models rarely overfit.



→ The optimization problem

The optimization problem is defined as:

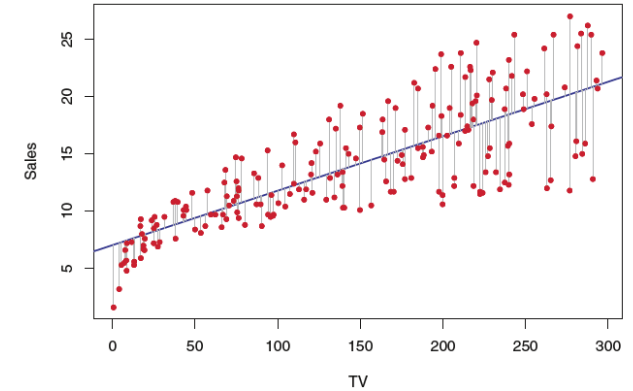
$$\text{Min}_{\mathbf{w}, \mathbf{b}} \text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \frac{1}{N} \sum_{i=1}^N \left(y_i - f_{\mathbf{w}, \mathbf{b}}(X_i) \right)^2$$

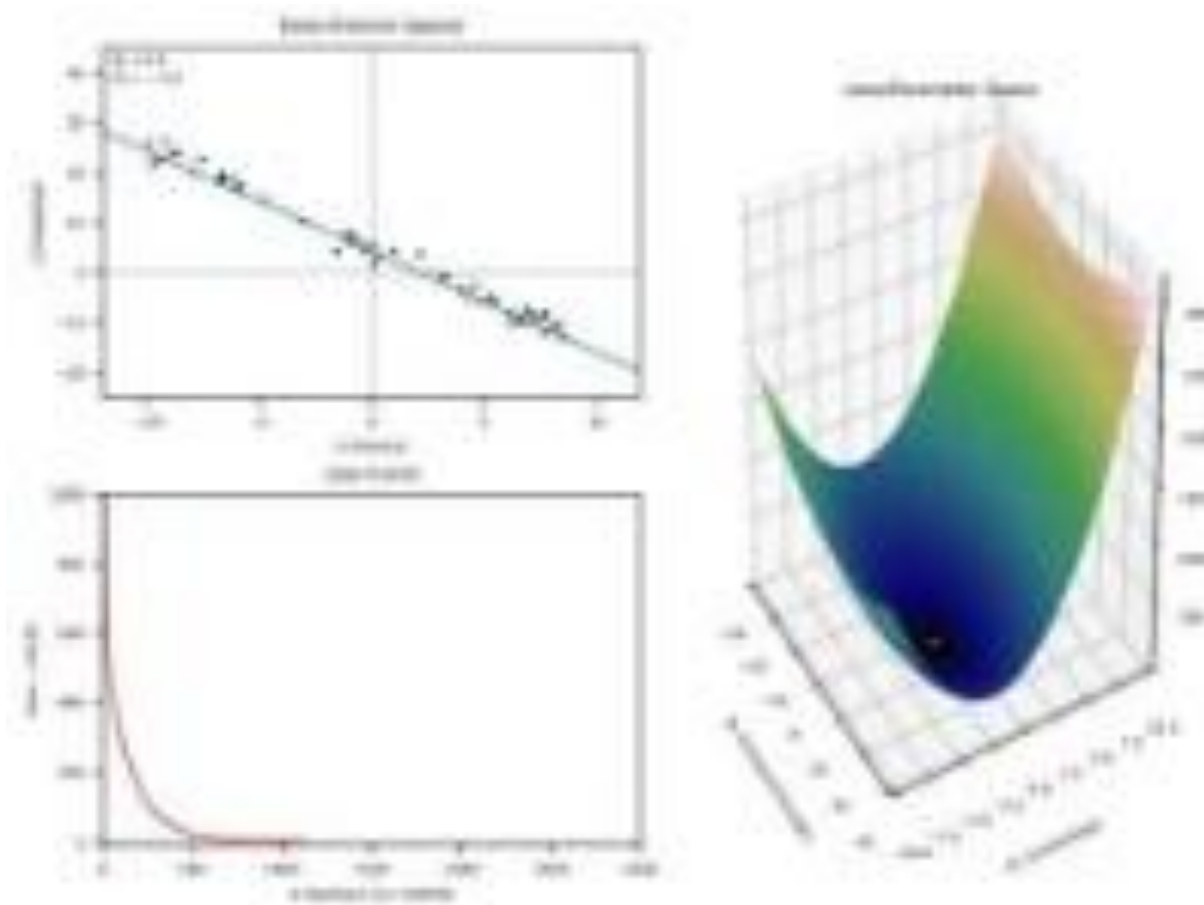
$\left(y_i - f_{\mathbf{w}, \mathbf{b}}(X_i) \right)^2$ is called the **objective function**, or the **loss function**! Or the **squared error loss**.

Why quadratic loss function? Why not using absolute value or cube?

1. More convenient (well-behaved derivative).
2. There exists a closed form solution.

The solution to this optimization problem is \mathbf{w}^* and \mathbf{b}^* . Now we can make predictions!





Linear Regression Evaluation Metrics

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

Adjusted

versions

$$Adjusted R^2 = 1 - (1 - R^2) * \frac{n - 1}{n - k - 1}$$

$$AIC = \frac{2K}{N} - \frac{2 \ln(\hat{L})}{N}$$

$$BIC = \ln(N) K - 2 \ln(\hat{L})$$

- **AIC**: Akaike information criterion
- **BIC**: Bayesian information criterion
- K : number of estimated parameters
- \hat{L} : Maximum value of the likelihood function

Part II

Linear regression: Polynomial regression

➔ Polynomial regression model

The polynomial regression model is a special case of multiple linear regression models!

- Create new variables $X_1 = X$, $X_2 = X^2$, ... etc and then treat as **multiple linear regression**.
- Not really interested in the coefficients; more interested in the fitted function!

$$\hat{f}(X) = f_{\mathbf{w},b}(X) = b + w_1x + w_2x^2 + \dots w_dx^d$$

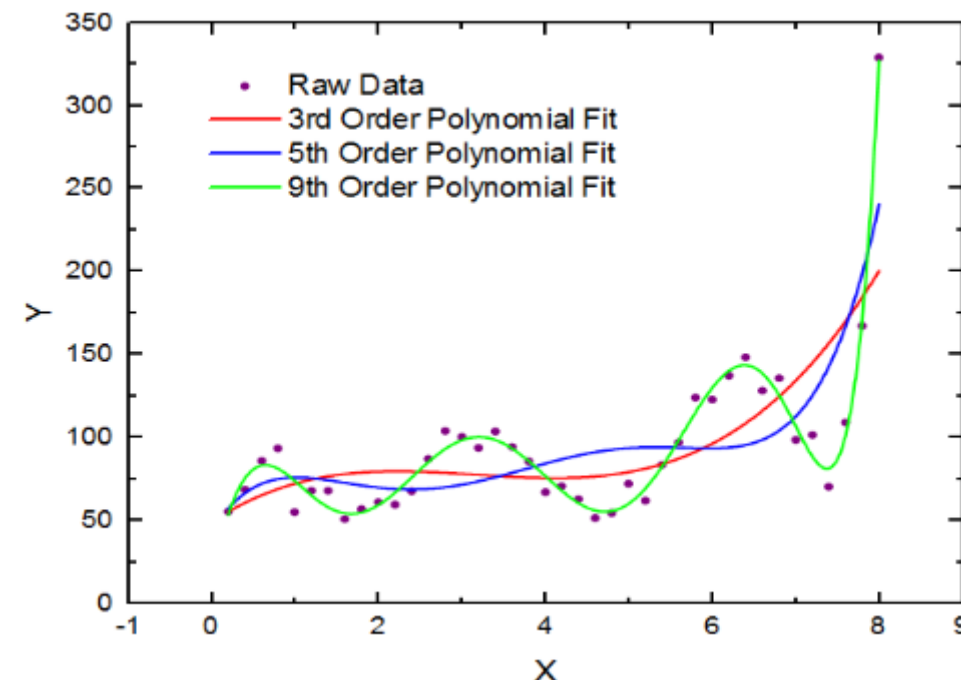
- \mathbf{W} is a d -dimensional vector of parameters
- b is a real number
- d is the polynomial degree of the model (we either fix the d at some reasonably low value, else use **cross-validation** to choose d)

➔ The optimization problem

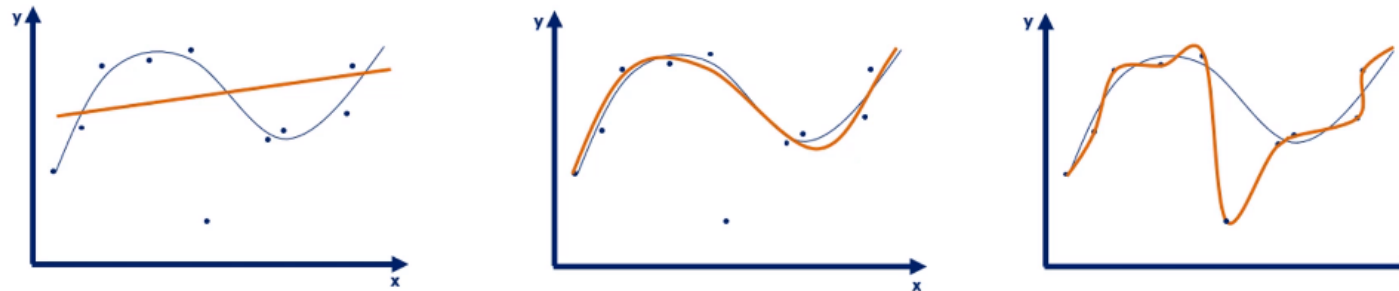
The optimization problem is defined as:

$$\text{Min}_{w,b} \text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \frac{1}{N} \sum_{i=1}^N (y_i - f_{w,b}(X_i))^2$$

- We use the same **loss function** as in linear regression!
- The solution to this optimization problem is w^* and b^*
- We use cross validation to optimize d^*
- Now we can make predictions!



➔ Tuning the hyperparameter d!



$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

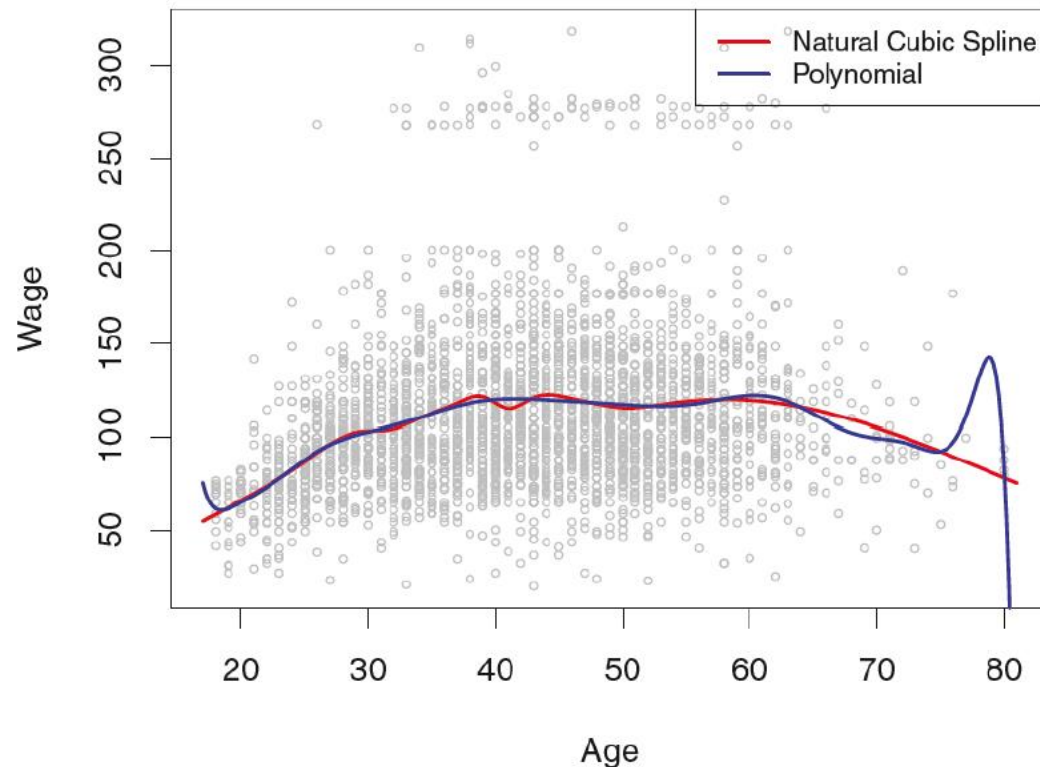
$$RMSE = \sqrt{MSE}$$

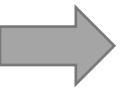




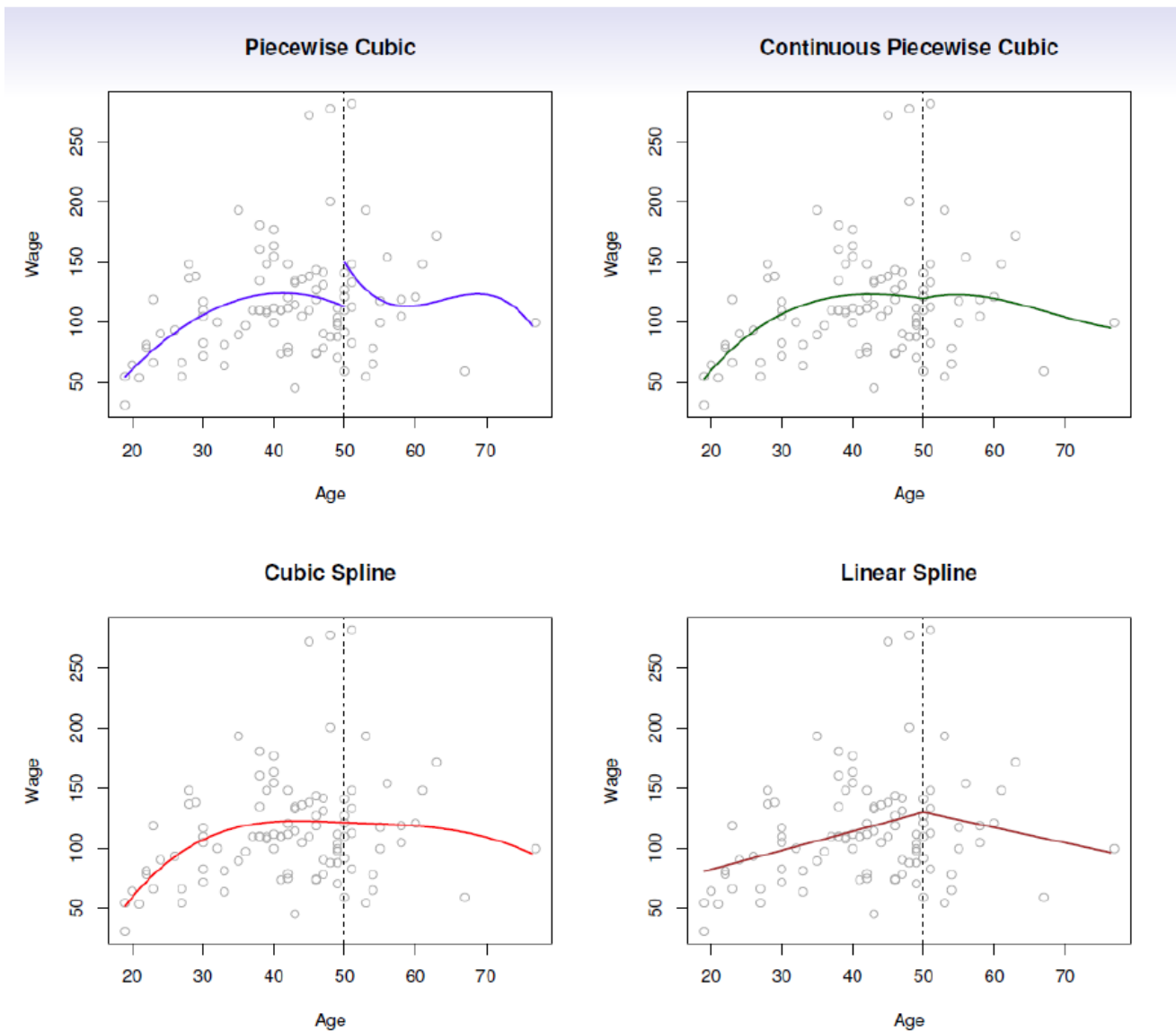
Caveats!

- In general, quadratic loss functions are **sensitive to outliers**
- Polynomials have **notorious tail behavior**
- Polynomials are **global fit**!
- **Solution**: piecewise polynomial, splines and local regressions.





Piecewise polynomials and splines!





Class Modules

- ✓ Module 1- Introduction to Deep Learning
- ✓ Module 2- Setting up Machine Learning Environment
- ✓ Module 3- Linear Regression (Econometrics approach)
- ✓ Module 4- Machine Learning Fundamentals
- ✓ Module 5- Linear Regression (Machine Learning approach)
- Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)
- Module 7- Logistic Regression
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)

