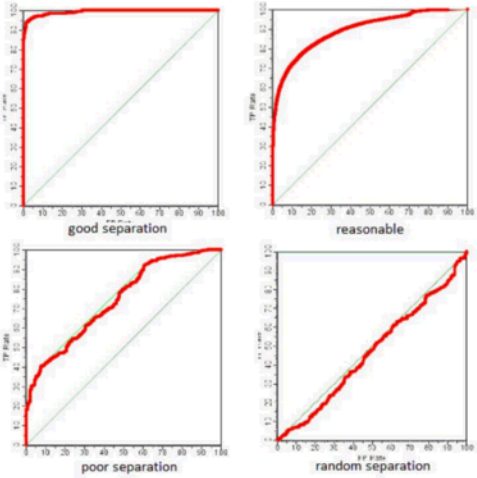
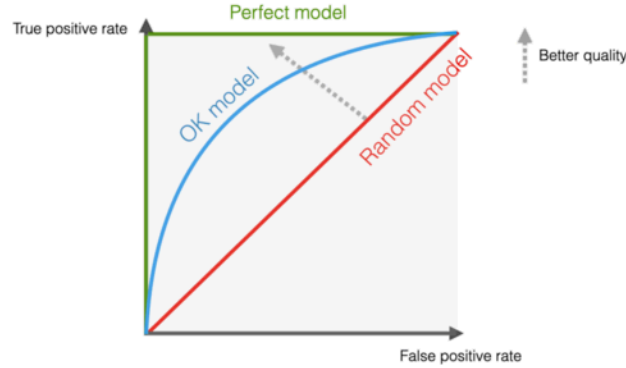
ROC Curve

The ROC curve illustrates this trade-off between the TPR and FPR





Perfect model is correct in all the predictions, all the time

Random model cannot distinguish between the two classes, and its predictions are no better than random guessing.

Each point on the curve corresponds to a combination of TPR and FPR values at a specific decision threshold.

Evaluation Methods (Classification problem)

Given the confusion matrix below.

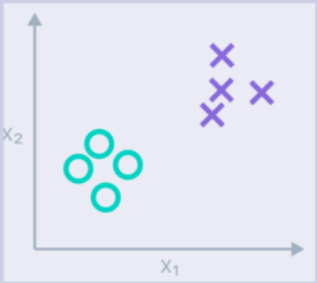
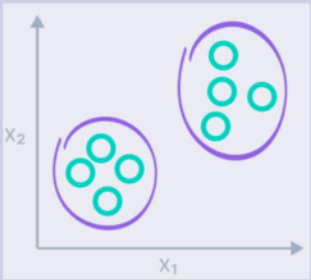
n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

a. Calculate the accuracy of a classification model
b. Calculate the Precision and Recall values
c. Calculate F1-score

Evaluation Methods (Classification problem)

If a confusion matrix shows that there are 50 true positives (TP), 20 false positives (FP), 120 true negatives (TN), and 10 false negatives (FN), what is the accuracy of the model?

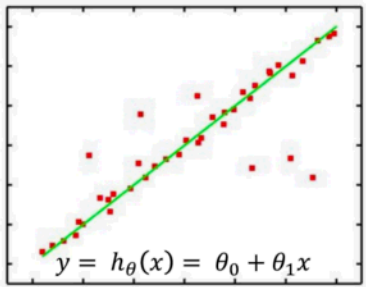
Machine Learning

Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Data is classified based on the training dataset	Assigns properties of given data to classify it
Used for prediction	Used for analysis
A known number of classes	An unknown number of classes
	

Linear Regression vs. Logistic Regression

Linear Regression

- The main goal of Linear Regression is to find the relationship between a dependent variable and independent variables.
- Used to solve regression problem
- Continuous dependent/continuous or discrete target variables
- Hypothesis is a linear straight line (might not capture the complexity of some datasets)



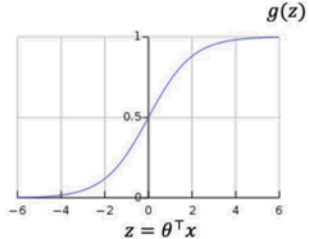
Logistic Regression

- The primary purpose of Logistic Regression is to predict a binary outcome (0 or 1) based on the input features, which is interpreted as a probability
- Used to solve classification problem
- Discrete/categorical dependent/target variables
- Hypothesis is a S-curve (Sigmoid)
- It does not work well when the data has high dimensionality or many correlated features

$$h_\theta(x) = g(\theta^\top x)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}$$



Linear Regression

- **Hypothesis representation**

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- **Cost function**

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- **Gradient descent**

Repeat until convergence

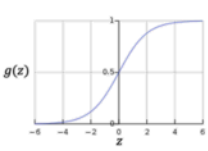
$$\{\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\}$$

- The main purpose of the Gradient Descent algorithm is to minimize the loss function by adjusting the model's parameters.
- A smaller learning rate can result in slower convergence but more precise updates.
- A very large learning rate, the algorithm may oscillate around the optimal solution or diverge

Logistic Regression

- **Hypothesis representation:**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^{zT}x}}$$



- **Cost function**

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\text{Cost}(h_\theta(x^{(i)}), y^{(i)}))$$

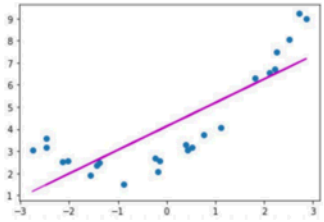
$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log\left(h_\theta(x)\right) & \text{if } y = 1 \\ -\log\left(1 - h_\theta(x)\right) & \text{if } y = 0 \end{cases}$$

- **Gradient descent**

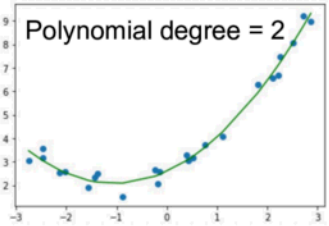
Repeat until convergence

$$\{\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\}$$

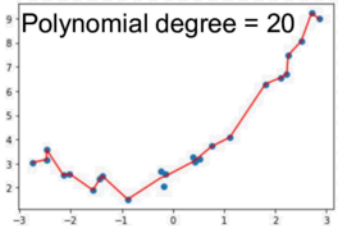
Under-Fitting vs. Over-Fitting



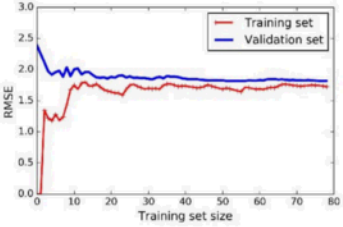
Under-fitting model
High bias
Small variance



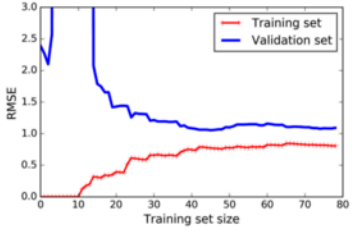
Good-fitting model
Small variance
Small bias



Over-fitting model
Small bias
High variance



"**generalization**" mean the ability of an AI system to be trained on one dataset and perform well on different, unseen datasets

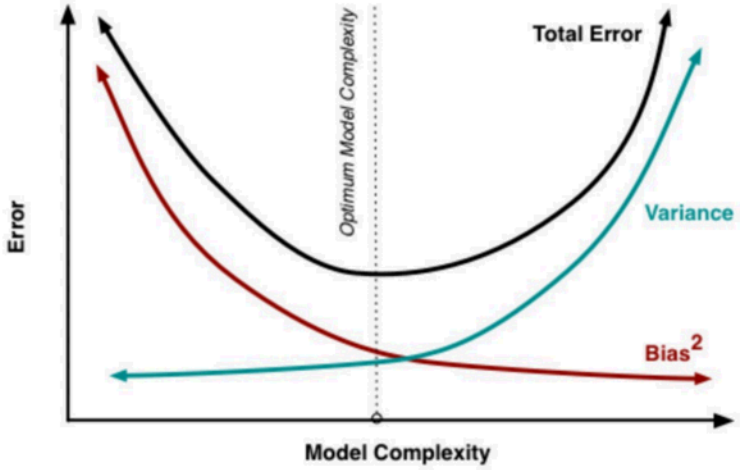


Under-fitting is when the model's error on both the training and test sets is very high.

Over-fitting is when the model's error on the training set is very low but the model's error on the test set (i.e., unseen samples) is large.

The Bias/Variance Trade-off

The bias-variance tradeoff is the tradeoff between underfitting (high bias) and overfitting (high variance) when building a model.



Increasing model complexity (adding more independent variables) will increase its variance and reduce its bias.

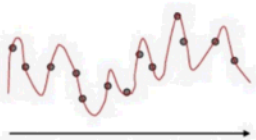
Regularization

To solve the overfitting problem, regularization technique can be used.

- The minimization

$$\min_f |Y_i - f(X_i)|^2$$

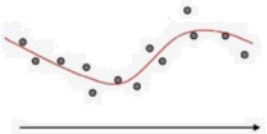
may be attained with zero errors.
But the function may not be unique.



- Regularization

$$\min_{f \in H} \sum_{i=1}^{n} |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$

- Regularization with smoothness penalty is preferred for uniqueness and smoothness.
- Link with some RKHS norm and smoothness



L1 regularization on least squares:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \alpha \sum_{n=1}^{N} |\theta_n|$$

L2 regularization on least squares:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \alpha \sum_{n=1}^{N} \theta_n^2$$

L2 regularization	L1 regularization
Computational efficient due to having analytical solutions	Computational inefficient on non-sparse cases
Non-sparse outputs	Sparse outputs
No feature selection	Built-in feature selection

Normalization

The process of transforming the columns in a dataset to the same scale to make the training model less sensitive to the scale of features.