AI for Business: Insights from Corporate Data

Topic 5: Foundations of Machine Learning

Miao Liu

Boston College

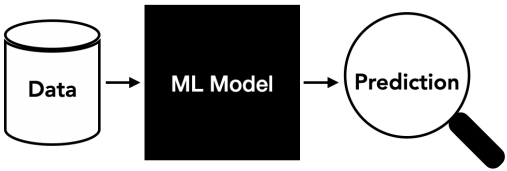
February 17, 2026

Overview: Topic 5

Overview: Topic 5

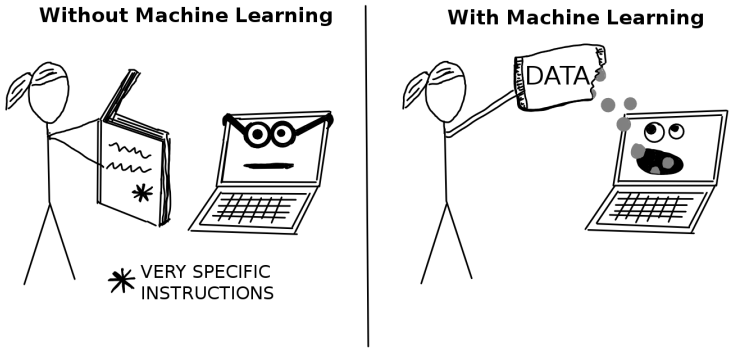
What is Machine Learning?

Machine Learning is a Prediction Problem



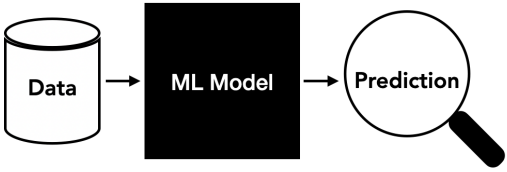
- But what's new?

What's new: Data in the driver's seat



Overview: Topic 5

The Learning Problem



- Let predicting target be Y and Data (covariates) be X. Their relationship is:

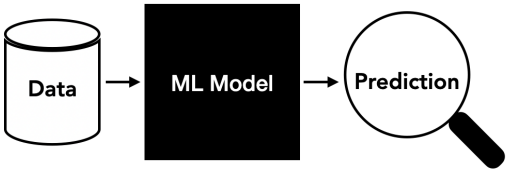
$$Y = f(X) + e$$

e is the error term and is normally distributed:

$$e \sim N(0, \sigma^2).$$

- What are the sources of e?
  - Unobservables + Randomness of the World
  - It is Irreducible Error

The Learning Problem

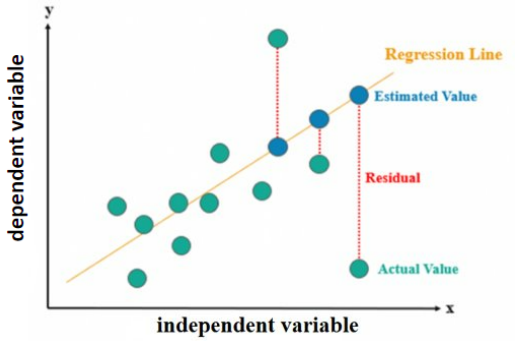


$$Y = f(X) + e$$

- Machine learning is about finding (read: training the data to learn) the best $f(X)$
- Denoting the trained model as $\hat{f}(X)$
- With $\hat{f}(X)$ at hand, we can predict for any new X_0

$$\hat{Y} = \hat{f}(X_0)$$

Let's Use the Simplest Model to Illustrate: Linear Regression

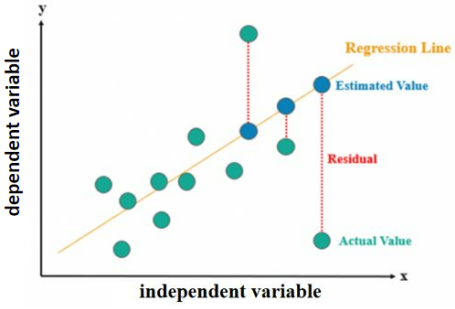


$$Y = f(X) + e$$

$$\hat{Y} = \hat{f}(X_0)$$

- Can you find Y, X, $\hat{f}(X_0)$, X_0, $\hat{Y}$?

Let's Use the Simplest Model to Illustrate: How to find $\hat{f}()$?



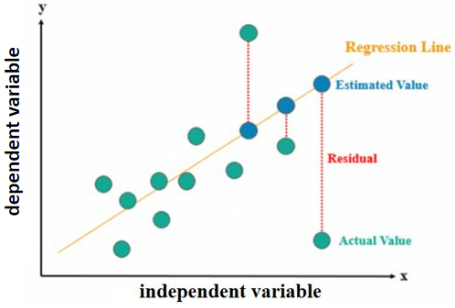
- Linear Regression for $Y = f(X) + e$:

$$Y = \beta_0 + \beta_1 X + e$$

- finding $\hat{f}()$ becomes finding $\hat{\beta}_0$ and $\hat{\beta}_1$ so that:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

Let's Use the Simplest Model to Illustrate: How to find $\hat{\beta}_0$ and $\hat{\beta}_1$?



- We want $\hat{\beta}_0$ and $\hat{\beta}_1$ such that e^2 is as small as possible across data points:

$$\text{Minimize } \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Another interpretation: Reduce e all the way till it becomes **Irreducible Error**

Model Performance Evaluation

- Linear regressions minimize prediction error e^2:

$$\text{Minimize } \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- The idea of minimizing prediction errors carries over to other ML models
- The Mean Squared Error for any new data x is given by:

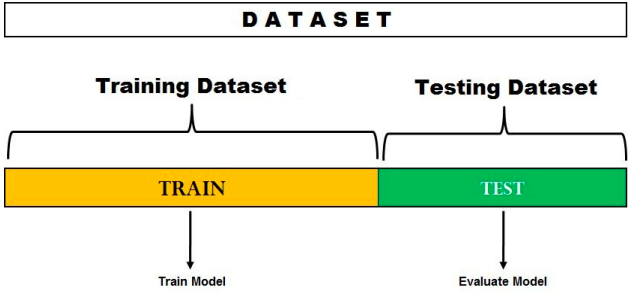
$$\text{MSE}(X) = \mathbb{E}\left[(Y - \hat{f}(X))^2\right].$$

- Commonly used to evaluate ML model performance

Model Performance Evaluation

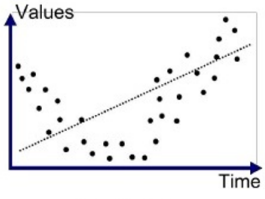
- In ML, we partition data into training and testing datasets
- Training dataset to train ML model; Testing dataset to evaluate model by

$$\text{MSE}(X) = \mathbb{E}\left[(Y - \hat{f}(X))^2\right]$$

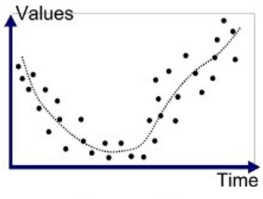


Fundamental Trade-off of Machine Learning

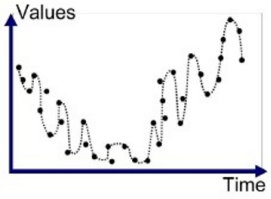
- Trade-off between Under vs. Over-fitting
- Suppose this is our stock price data:



Underfitted



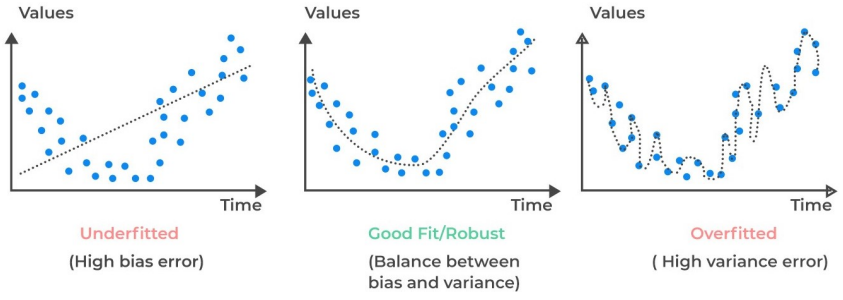
Good Fit/Robust



Overfitted

Fundamental Trade-off of Machine Learning

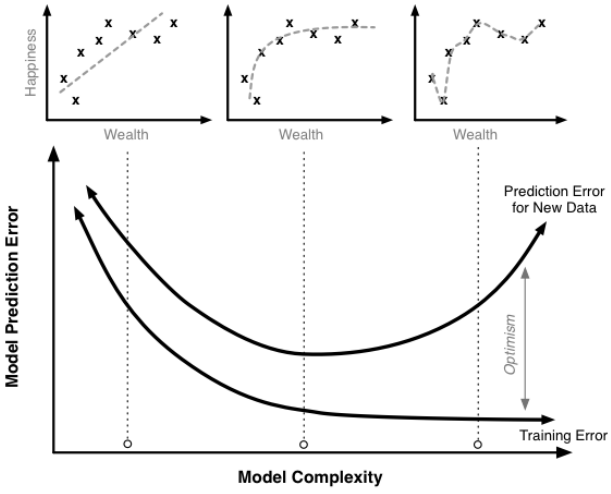
- Also called Bias and Variance Trade-off
- **Bias**: Difference between true model $f(X)$ and the estimated model $\hat{f}(X)$
- **Variance**: Variability of model prediction whenever new data comes in



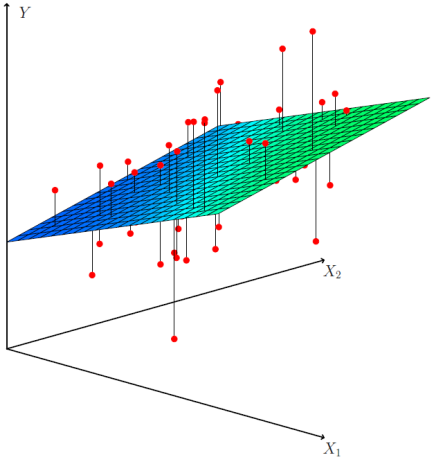
- High Bias: model too simple - high error on both training and test data
- High Variance: model too complex - low error on training; high error on test

Fundamental Trade-off of Machine Learning

- Bias and Variance Trade-off gives optimal model complexity



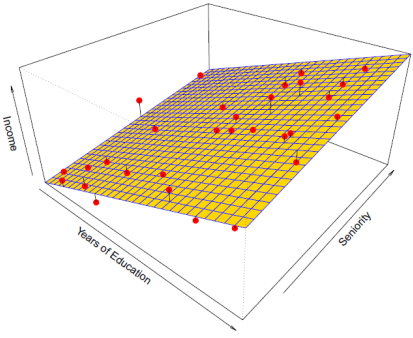
From One Signal to Many

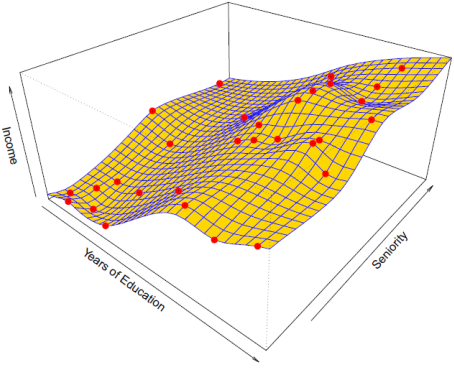


$$Y = \beta_0 + \beta_1 X + \beta_2 X + e$$

From One Signal to Many

- Under and Over-fitting in 3D





Overview: Topic 5

Classification Problem

Binary Classification

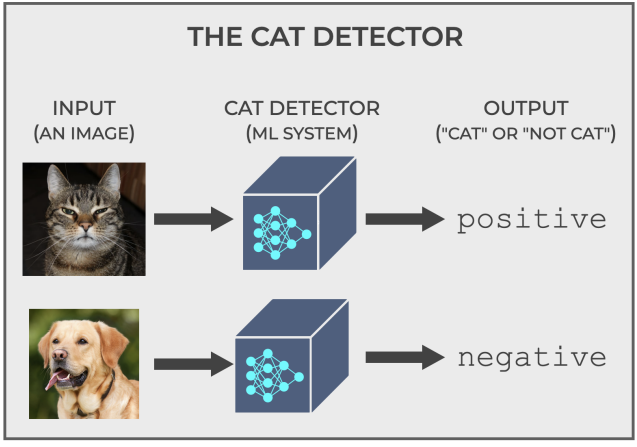
Spam
Not Spam

Cancer
Not Cancer

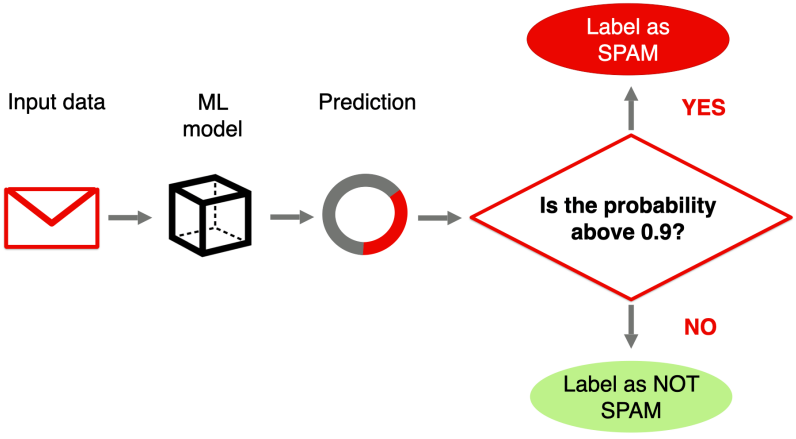
Positive Sentiment
Negative Sentiment

Fraud
Not Fraud

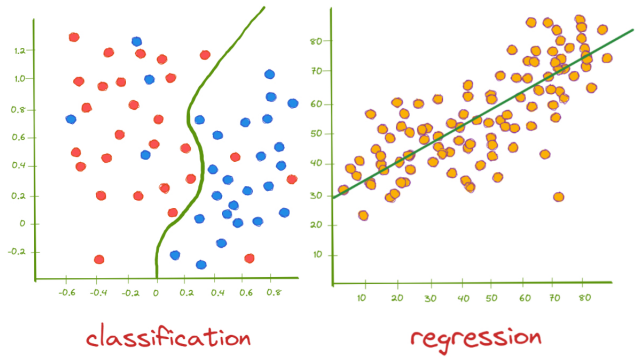
Classification Problem



Classification Problem



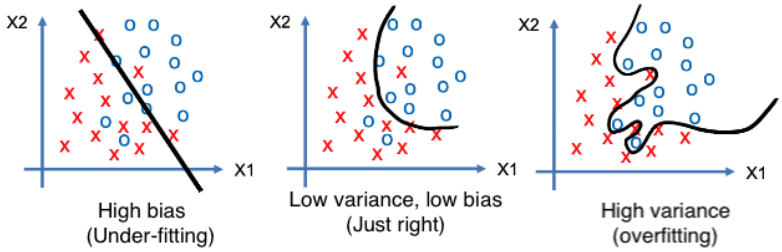
Classification Problem



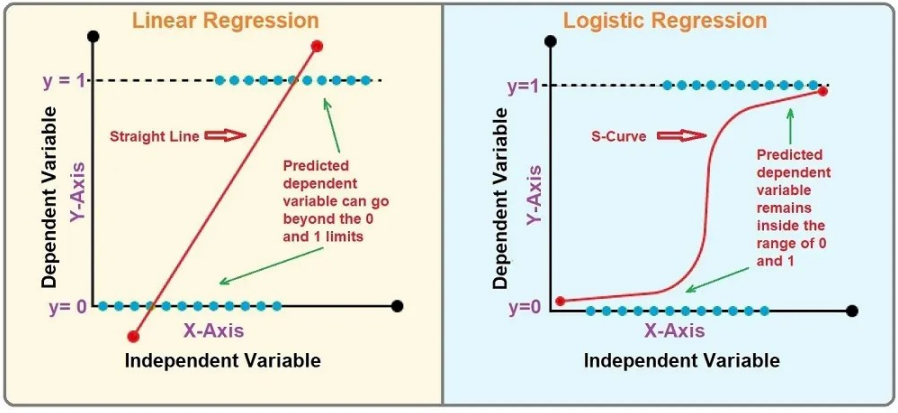
- Instead of predicting a value, we predict a probability

Classification Problem

- The Bias-Variance Trade-off of Classification

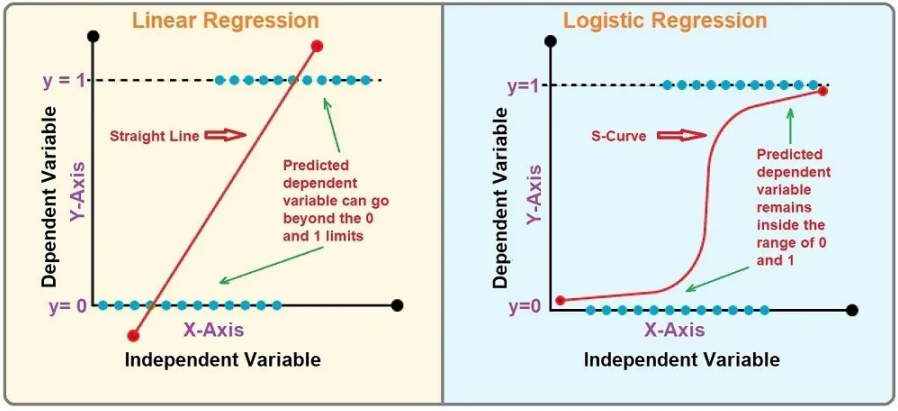


Logistic Regression



- One can estimate $p(X) = \beta_0 + \beta_1 X$
- The problem is that the predicted probability can exceed 0 and 1

Logistic Regression



- Instead of $p(X) = \beta_0 + \beta_1 X$, we estimate $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$
- Or, equivalently, $\log \frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X$

Logistic Regression

- The logistic regression is defined as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

- Alternatively, it can be expressed in terms of log-odds (logit):

$$\log \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X$$

Model Performance Evaluation for Classification Problem

- Recall the Mean Squared Error:

$$\text{MSE}(X) = \mathbb{E}\left[(Y - \hat{f}(X))^2\right]$$

- The corresponding metric for Classification problem:

$$\text{MSE}(X) = \mathbb{E}\left[\mathbb{I}(Y \neq \hat{f}(X))\right]$$

- It measures the expected proportion of classification error

Four Classification Outcomes

- True Positive (TP): Correctly classified positive cases.
- True Negative (TN): Correctly classified negative cases.
- False Positive (FP): Negative cases incorrectly classified as positive.
- False Negative (FN): Positive cases incorrectly classified as negative.

Actual \Predicted	**Predicted Pos**	**Predicted Neg**
Actual Pos	TP	**FN**
Actual Neg	**FP**	TN

Type I and II Errors

Actual \Predicted	**Predicted Pos**	**Predicted Neg**
Actual Pos	TP	**FN(Type II)**
Actual Neg	**FP(Type I)**	TN

- Type I Error (FP): False alarms (diagnosing a healthy patient as sick)
- Type II Error (FN): Missing critical cases (e.g., failing to detect a sick patient)
- There is often a trade-off between them
- Can we always reduce one without sacrificing the other?

Type I and II Errors

Actual \Predicted	Predicted Pos	Predicted Neg
Actual Pos	TP	**FN(Type II)**
Actual Neg	**FP(Type I)**	TN

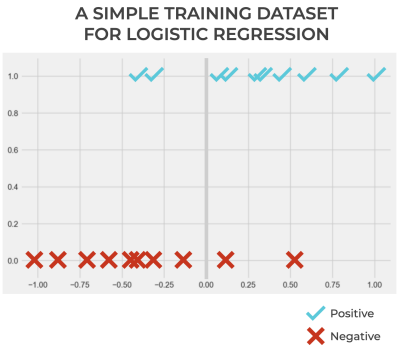
- This trade-off shows in comparing:
  - **TPR (True Positive Rate)**: Proportion of actual positives correctly identified
  - **FPR (False Positive Rate)**: Proportion of actual negatives incorrectly identified

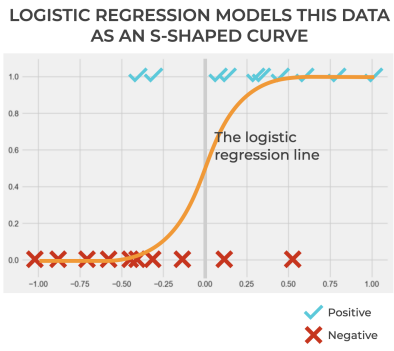
$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

- Can we always improve TPR without hurting FPR, or vice versa?

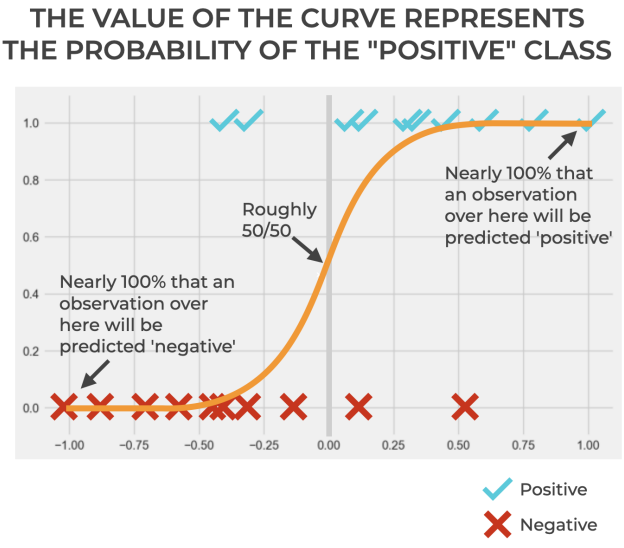
Type I and II Errors Trade-off

- Let's use Logistic Regression to visualize the trade-off



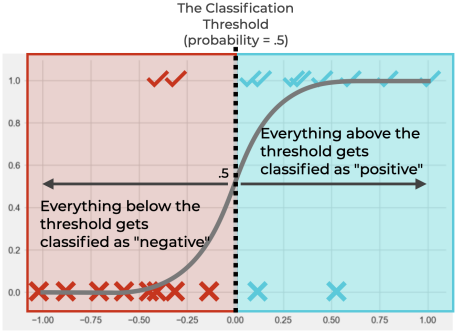


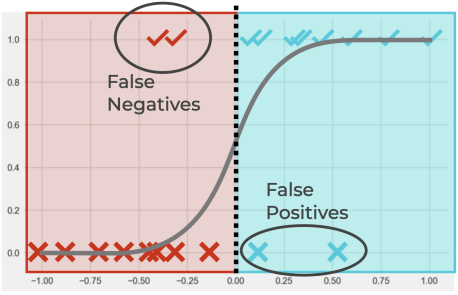
Type I and II Errors Trade-off



Type I and II Errors Trade-off

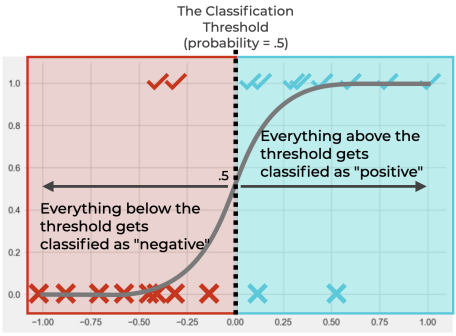
- Threshold determines how predicted probabilities map into classifications

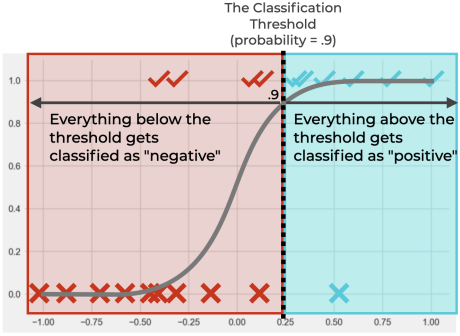




Type I and II Errors Trade-off

- Changing Threshold shifts the Trade-off

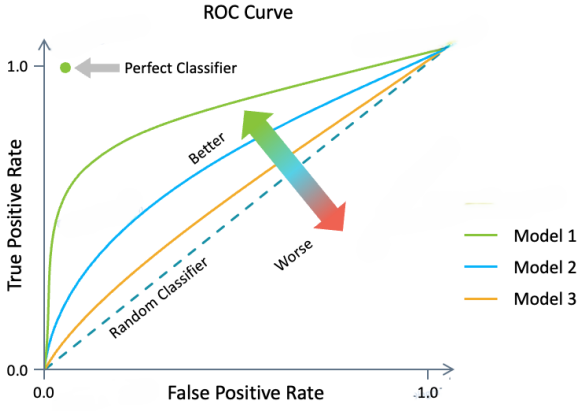




- Think about this trade-off for classifying spam emails vs. diagnosing a patient
- Can we find a performance metric for ALL thresholds?

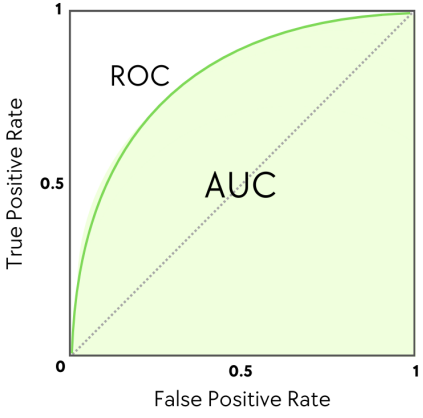
ROC Curve and AUC

- **ROC Curve** (Receiver Operating Characteristic): Plots the trade-off between TPR and FPR for different thresholds



ROC Curve and AUC

- **AUC** (Area Under Curve): metrics to compare models

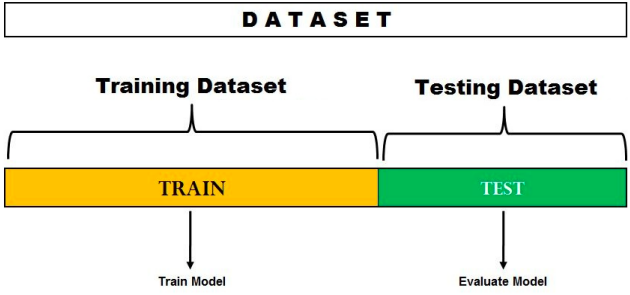


Overview: Topic 5

Recall: Model Performance Evaluation

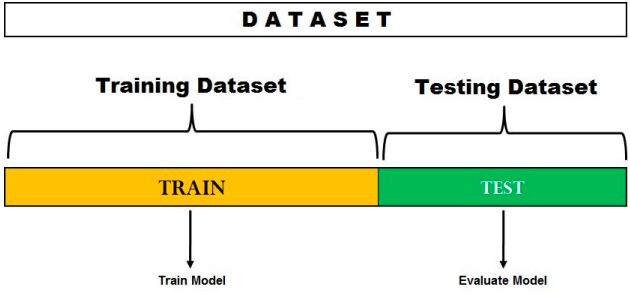
- In ML, we partition data into training and testing datasets
- Training dataset to train ML model; Testing dataset to evaluate model by

$$\text{MSE}(X) = \mathbb{E}\left[(Y - \hat{f}(X))^2\right]$$



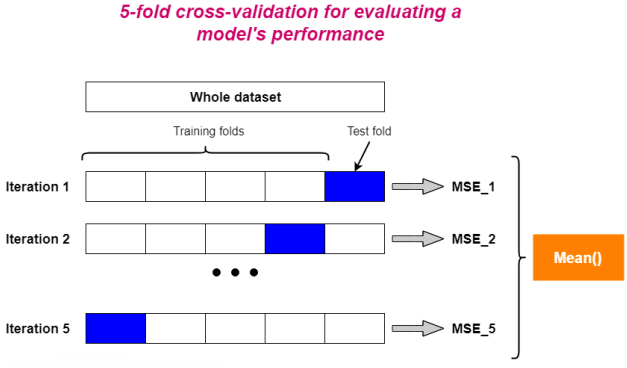
3 limitations of this approach

- Test set might not be representative: biased performance evaluation
- Model performance can vary significantly based on how the dataset is split
- Part of the dataset is exclusively used for testing and cannot contribute to training



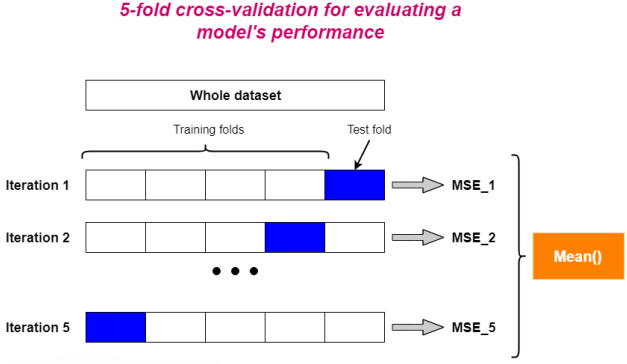
K-fold Cross-Validation

- Split the data into K equally sized folds.
- Train on $K - 1$ folds, validate on the remaining fold.
- Repeat for all K folds and average the results.



3 limitations of this approach: solved

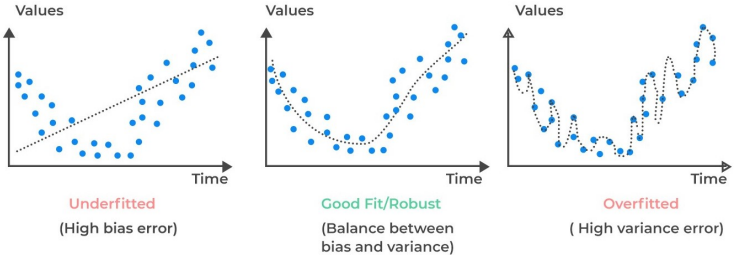
- Test datasets become representative
- Average performance across multiple splits: reducing the impact of a "lucky" or "unlucky" test set
- All data are used for both training and testing: better use of limited data



Overview: Topic 5

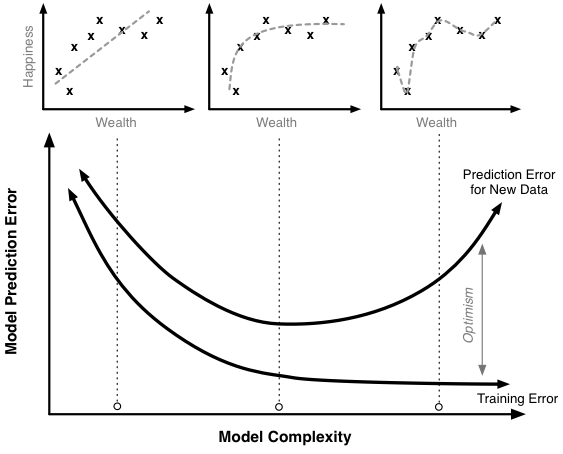
Recall: Under vs. Over-fitting

- Also called Bias and Variance Trade-off
- **Bias**: Difference between true model $f(X)$ and the estimated model $\hat{f}(X)$
- **Variance**: Variability of model prediction whenever new data comes in



- High Bias: model too simple - high error on both training and test data
- High Variance: model too complex - low error on training; high error on test

Recall: Under vs. Over-fitting

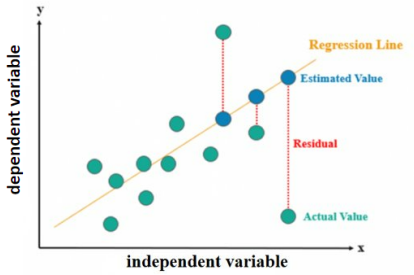


- Under vs. Over-fitting: which one is a **bigger** problem in Machine Learning?

Regularization: Combating Over-fitting

- **Goal**: Prevent over-fitting by penalizing (over)complexity
- **Key Idea**:
  - Add a penalty term to the loss function to penalize complexity
  - Prevent the model from learning overly complex relationships that are specific to the training data but not generalizable
- **Two Common Regularization Techniques**:
  - **Lasso (L1 penalty)**: Encourages sparsity by setting some coefficients to zero
  - **Ridge (L2 penalty)**: Shrinks coefficients towards zero without eliminating them

Let's Use the Simplest Model to Illustrate (Linear Regression)



- We want to have $\hat{\beta}_0$ and $\hat{\beta}_1$ such that e^2 is as small as possible across data points:

$$\text{Minimize } MSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Introducing complexity:

$$\text{Minimize } MSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i} - \beta_2 x_{2i})^2$$

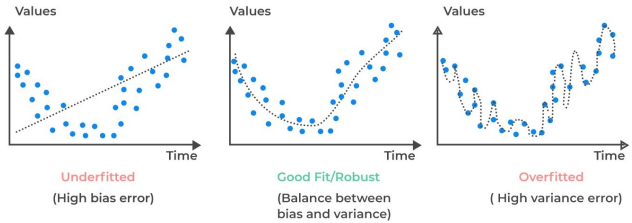
Lasso Regression

- **Loss Function**:

$$\text{Minimize: } \frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}))^2 + \lambda(|\beta_1| + |\beta_2|)$$

  where:
  - β_1, β_2: Coefficients for predictors x_1 and x_2
  - λ: Regularization/Tuning parameter controlling penalty strength
- **Effect of λ**:
  - Small λ: Minimal penalty; behaves like ordinary least squares (OLS)
  - Large λ: Shrinks some coefficients (β_1, β_2) to exactly zero
- **What is does**:
  - Prevents overfitting by eliminating irrelevant predictors

Lasso: Why does shrinkage work?



- In the language of Bias-Variance Tradeoff:
  - Large βs can lead to low bias but high variance models that perform well on training data but fail to generalize to unseen data
  - Smaller βs increase bias slightly (by simplifying the model), but significantly reduce variance, leading to better generalization

Lasso vs. Ridge Regression: Generalization

- **Lasso (L1)**:

$$\text{Minimize: } \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- **But how do we choose λ?**

High-Dimensional Data

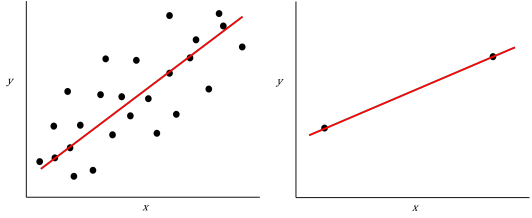
- **What is High-Dimensional Data?**
  - Data with a large # of features (predictors) relative to # of observations (e.g., $p > n$)

High-Dimensional Data

- **Why is High-Dimensional Data Common Today?**
  - **Finance**: Availability of thousands of economic, financial, and sentiment data from global markets ($>$ # of companies)
  - **Marketing**: Proliferation of thousands of consumer data from online platforms, social media, and e-commerce ($>$ # of consumers)

High-Dimensional Data

- **Challenges of High-Dimensional Data:**
  - **Overfitting**: Models become over-complex and capture noise instead of true patterns (left: n>p; right: n=p)



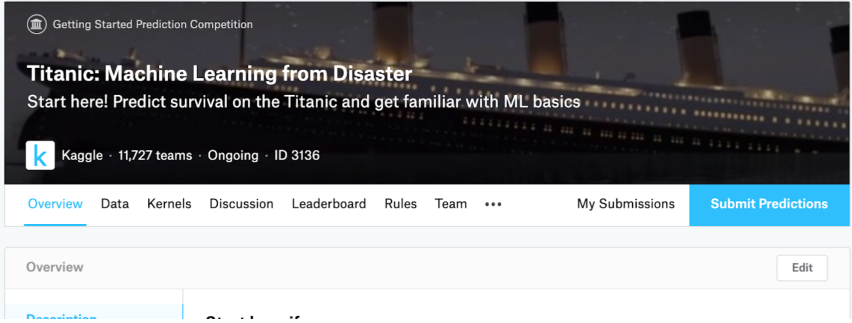
  - **Interpretability**: Large numbers of features make models difficult to interpret

High-Dimensional Data

- **How Regularization Methods Help:**
  - Select only the most relevant predictors by setting others to zero and reduce model complexity (e.g., Lasso)
  - Improve model generalization and stability in high-dimensional settings

Overview: Topic 5

Kaggle Titanic Competition



Kaggle Titanic Competition

- `train.csv` contains details of a subset of the passengers on board (891) and importantly, reveals whether they survived or not, aka the "ground truth"
- `test.csv` contains similar information but does not disclose the "ground truth" for the other 418 passengers
- Task: Train ML models using the training data (`train.csv`), predict who survived in the testing data (`test.csv`)

- **Survived**: Target variable (1 = Survived, 0 = Did Not Survive)
- **Pclass**: Passenger class (1 = First, 2 = Second, 3 = Third)
- **Sex**: Gender of the passenger (male, female)
- **Age**: Age of the passenger *(may have missing values)*
- **SibSp**: Number of siblings/spouses aboard
- **Parch**: Number of parents/children aboard
- **Fare**: Ticket price
- **Embarked**: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

Assignment for Next Week

- Lab Assignment 5: Titanic Competition at **Kaggle Titanic Tutorial**