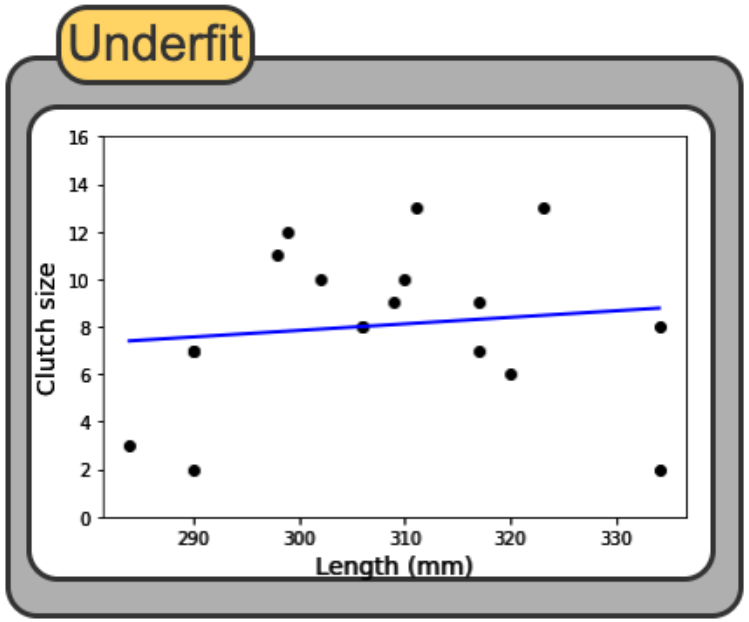
Lecture 8

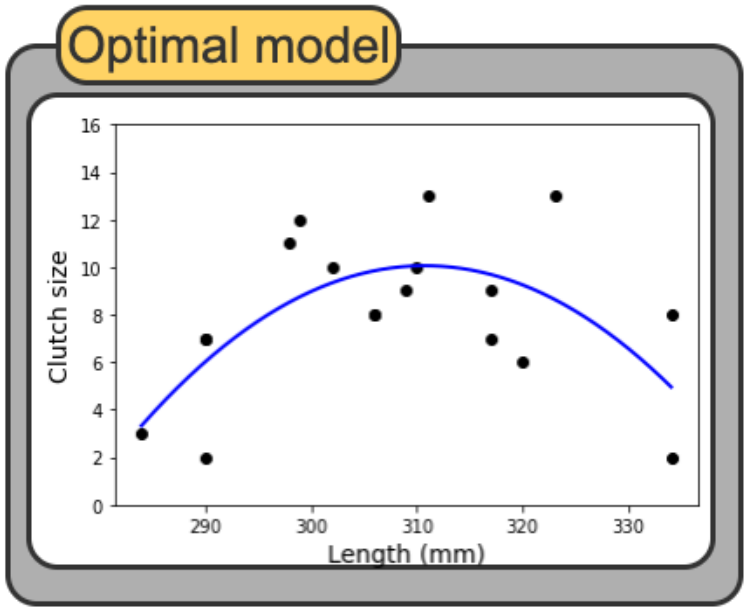
Model Performance Evaluation

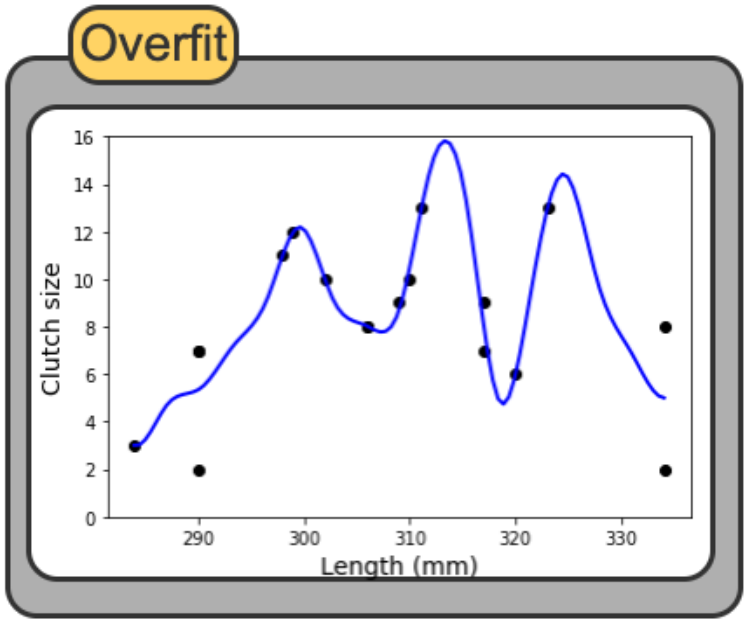
Model Specification and Complexity

- A model for an output feature y using an input feature(s) x is a function $f(x)$ so that $y = f(x)$.
- One of the most important parts of selecting a model is choosing the complexity of $f(x)$.
- Ex., would $y = \beta_0 + \beta_1 x$ be a better model than $y = \beta_0 + \beta_1 x + \beta_2 x^2$?
- Underfitting (too simple) and overfitting (too complex) models both fail to grasp data patterns.
- Model selection aims for a moderately complex model that captures trends without fitting noise, ensuring better generalization to new data.

Model Specification and Complexity







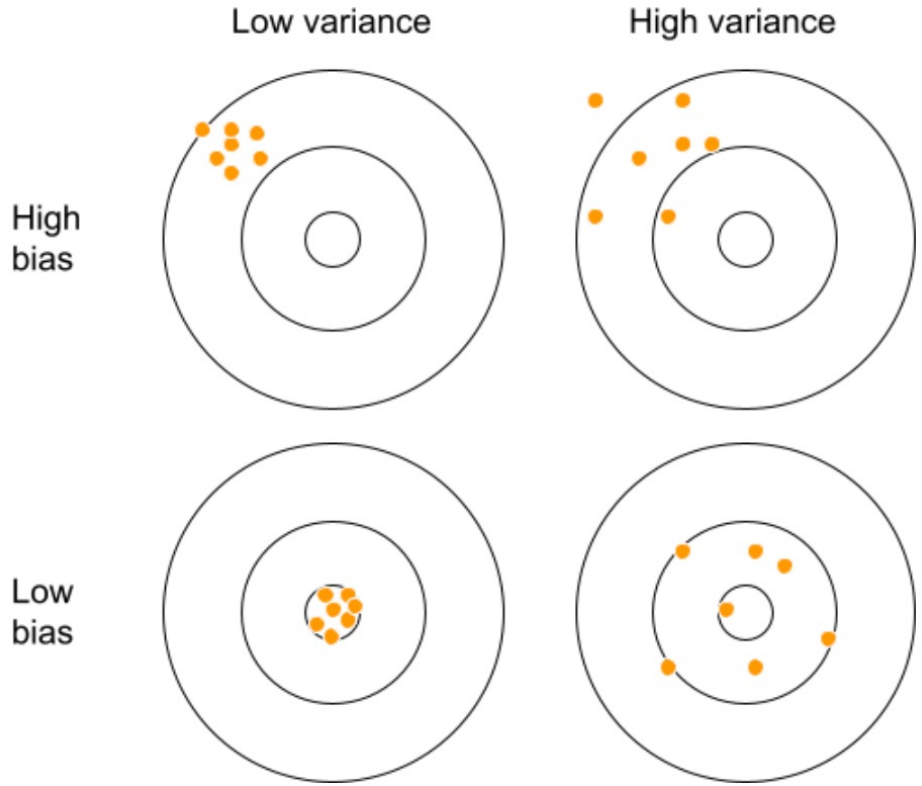
Breaking Down Error

- Total model error is the gap between observed and predicted values, split into three components:
  - **Bias**: Model assumption deviation from observed values.
  - **Variance**: Spread measure of model predictions.
  - **Irreducible error**: Inherent to situation, unavoidable.

- Metrics for Fit Assessment:
  - Metrics evaluate model fit with sample data using numeric values.
  - Complex models fit closely, lowering errors, yielding favorable metric values.

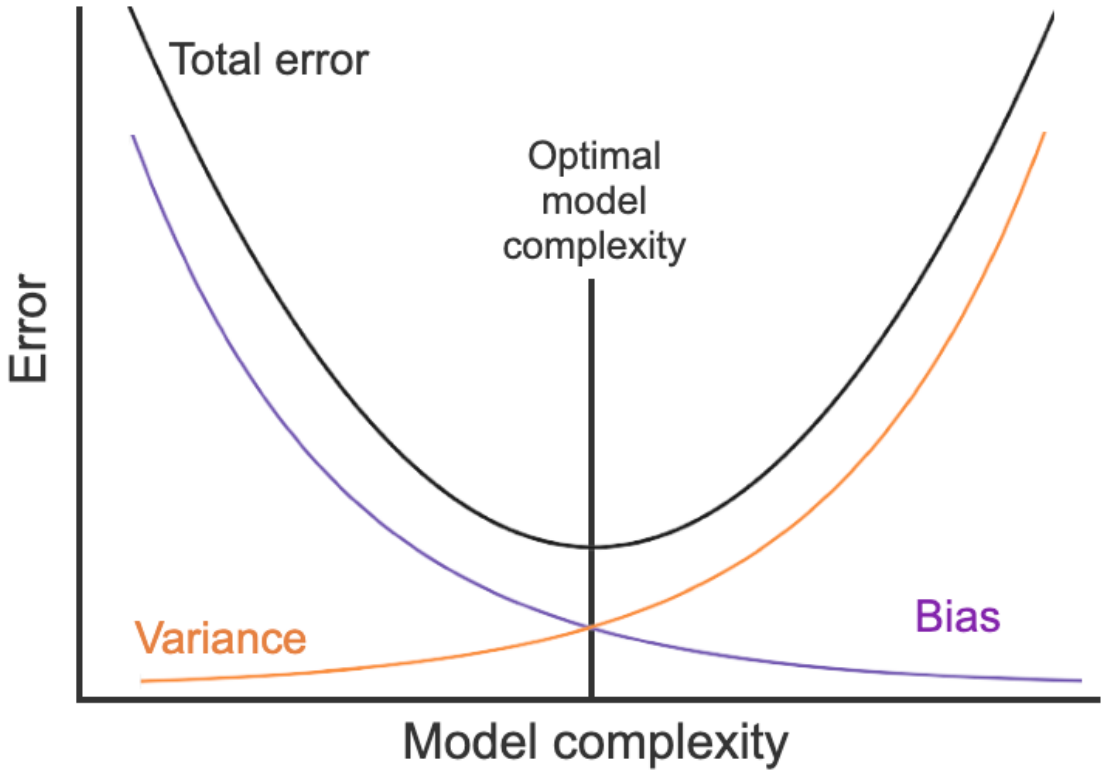
Bias vs. Variance

Modeling aims to hit a target like a bullseye. Bias measures shots' average distance from the bullseye, variance gauges their average distance from one another.



Bias-Variance Tradeoff

- As model complexity increases, variance increases.
- As model complexity increases, bias decreases.
- An optimal model minimizes total error, balancing bias and variance.



Model Training

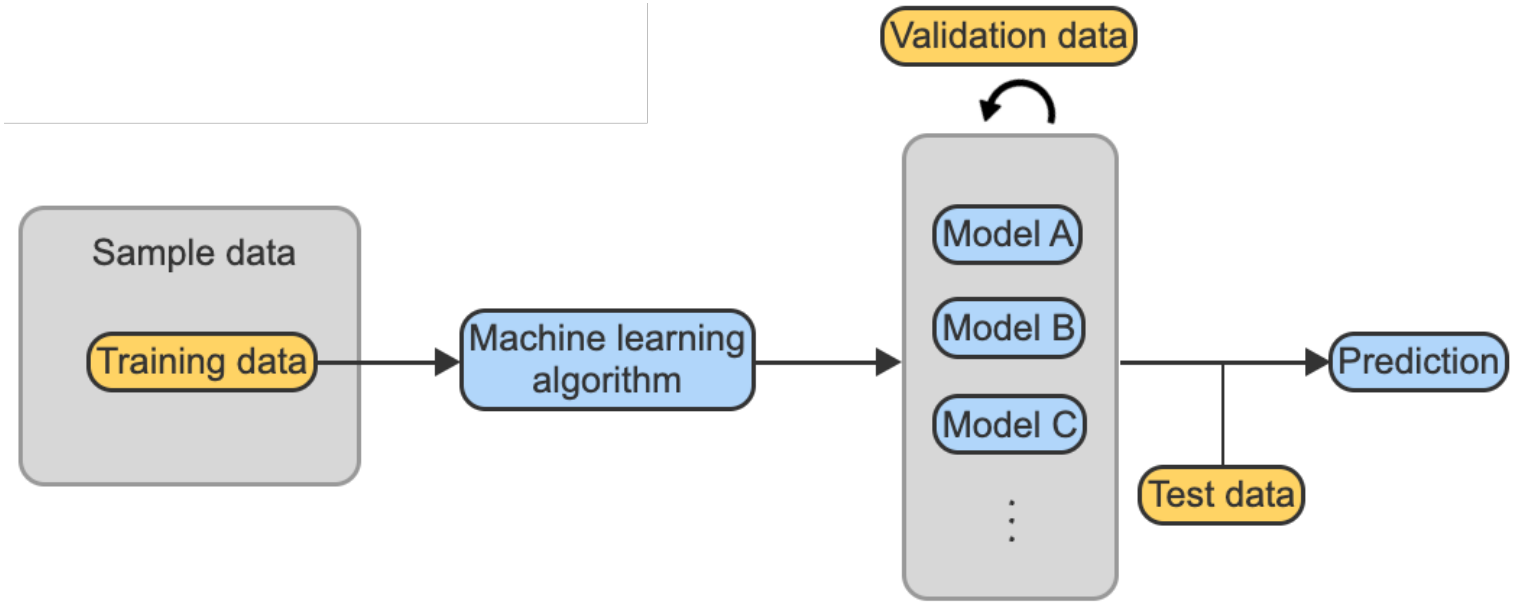
- Machine Learning Basics:
  - Machine learning creates models for predictions using data.
  - Data has labeled instances with features and output labels, or unlabeled instances.
  - Algorithms predict output based on input features or unveil feature relationships.

- Supervised Algorithm Types:
  - Classification: Predicting categorical values.
  - Regression: Predicting numerical values.

- Model Training Process:
  - Model training estimates parameters for predictions.
  - For example, linear regression's slope and intercept are estimated using incurred losses data for predicting car insurance premiums.

Model Training

- Training data is obtained from sample data.
- A machine learning algorithm fits a model or several models using training data.
- Often, data scientists use validation data to optimize the performance of models.
- Test data is used to see how well models perform compared to other models when predicting unseen data.



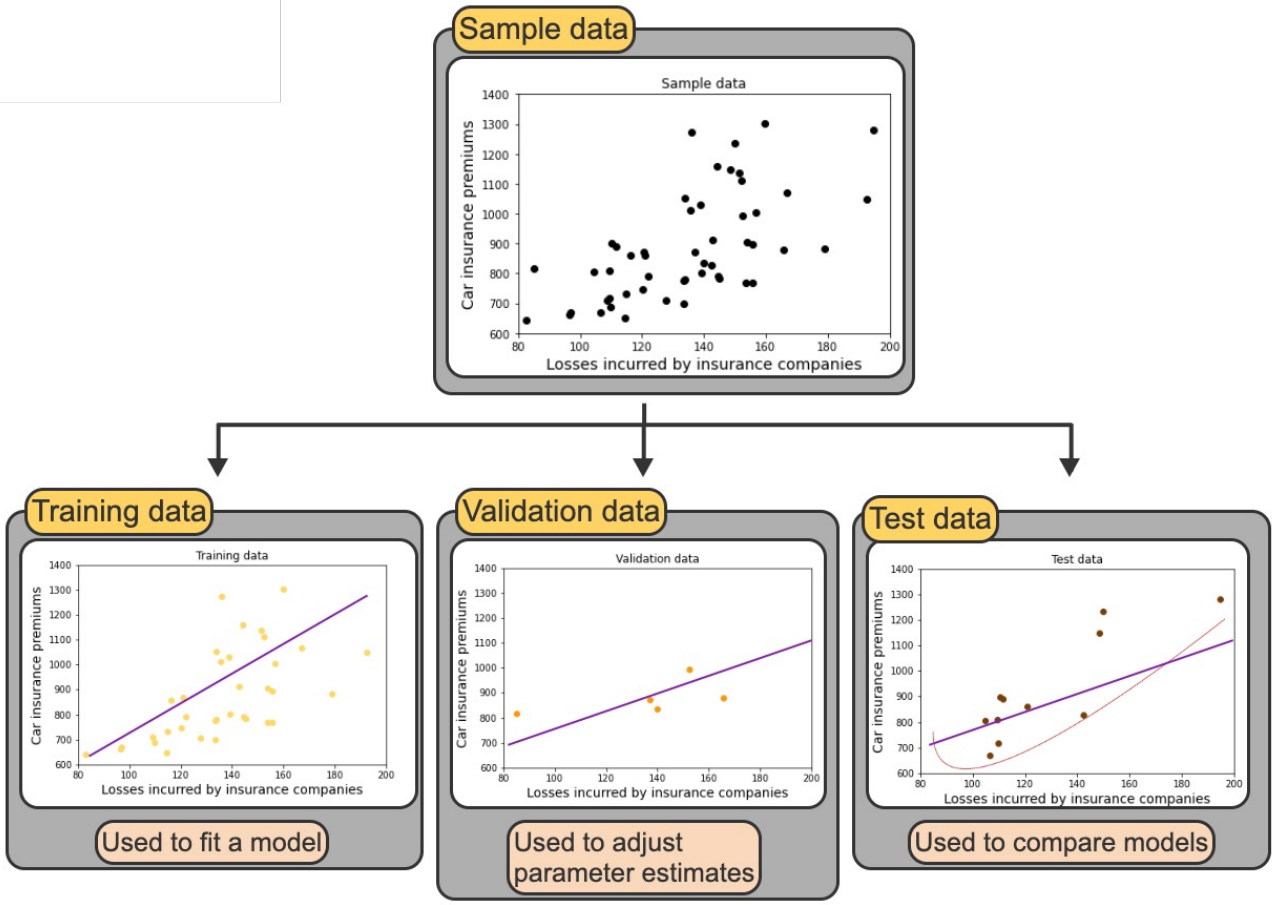
The Training-Validation-Test Split

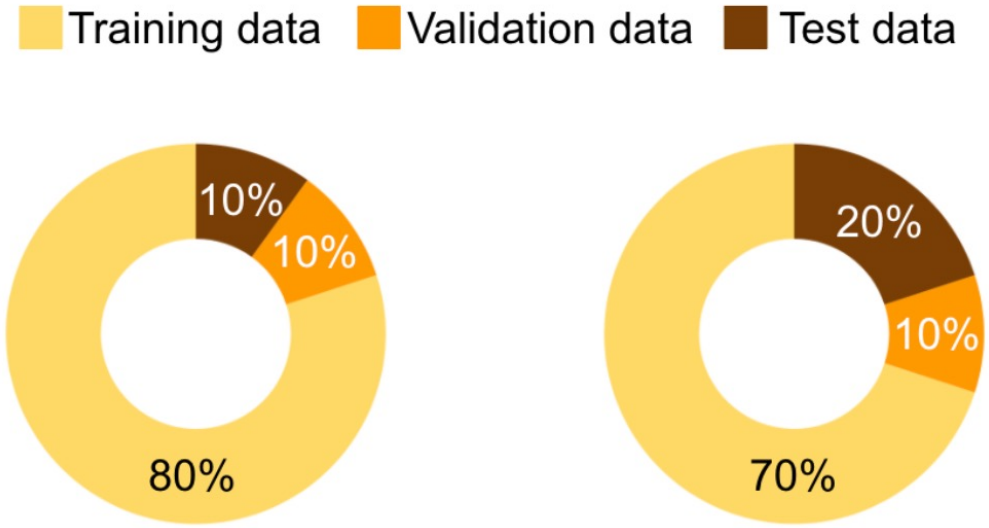
- Models on full sample risk overfitting, performing poorly beyond. To counter, data splits into three subsets:
  - **Training** data is used to fit a model.
  - **Validation** data evaluates model, tuning parameters and selecting features.
  - **Test** data is used to evaluate final model performance and compare different models.

- **Subset Characteristics:**
  - Similar data distribution for subsets, major part in training set.

- **Dataset Description:**
  - Animation uses bad drivers dataset, car accidents and insurance data from NHTSA and NAIC1.

The Training-Validation-Test Split





Loss Functions for Regression

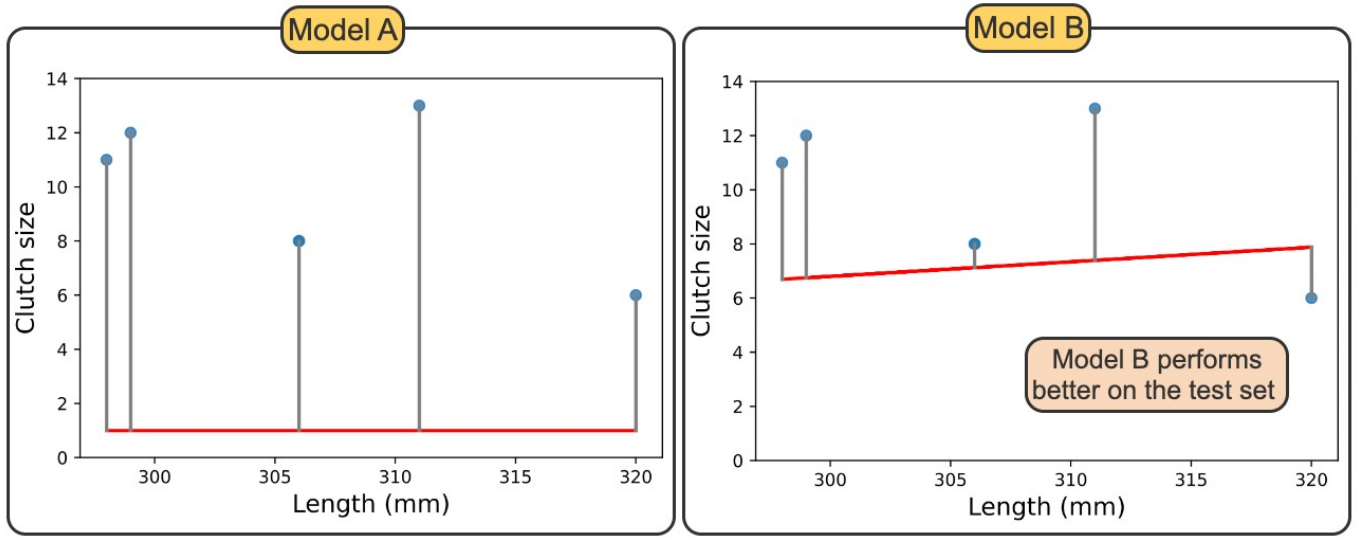
- Regression Purpose:
  - Regression aims to predict numeric values, like predicting clutch size in marine biology based on parent tortoise length.

- Loss Function and Metric:
  - Loss function measures prediction-observation difference; lower values indicate better model performance.
  - Regression metric is the loss function's observed value for a fitted model.

- Loss Function Utility:
  - Loss functions are valuable for model creation with training data, generalization to test/unseen data, and model performance comparison.

Common Loss Functions for Regression

Loss function	Description
Mean squared error	Average of the squared difference between observed and predicted values
Root mean squared error	Square root of the mean square error
Mean absolute error	Absolute value of the difference between actual and predicted values



Mean Squared Error and Root Mean Squared Error

- Mean squared error (MSE) is the average squared difference between observed and predicted values, calculated using the formula:

$$MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$$

- Mean squared error (MSE), measured in squared y units, is sensitive to outliers and reflects model variance under certain assumptions.

- Root mean squared error (RMSE) is the square root of average squared differences between observed and predicted values, calculated using the formula:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$$

- RMSE shares y's units, comparing models, with lower values indicating better fit.

Mean Absolute Error

- MAE is the average of absolute observed-predicted differences, found using the formula:

$$MAE = \frac{1}{n}\sum|y_i - \hat{y}_i|$$

- Mean absolute error (MAE) shares y's units, less affected by outliers. Like (root) mean squared error, smaller MAE means closer model fit.

- However, MAE lacks some math properties of (root) mean squared error, so it's used less.

Loss Functions for Classification

- Classification predicts categories, e.g., tumor malignancy.
- Loss functions penalize misclassifications, even for uncertain correct predictions.
- Perfect classifier has 0 loss; worsening as value increases shows performance.
- Metric is observed loss value for fitted model.

Absolute Error

- Logistic regression predicts probabilities for outcomes.
  - If probability surpasses a threshold, outcome is positive (class 1); if not, it's negative (class 0).
  - When forecasting tumor malignancy at radius 14.05 mm with a 0.5 threshold, a 0.286 probability implies benign classification (class 0), indicating uncertainty due to nonzero probability.

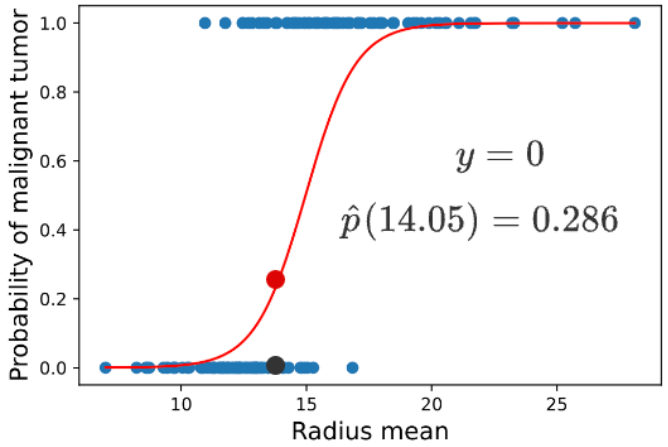
- Absolute loss quantifies uncertainty's impact, expressed as:

$$L_{abs}(y, \hat{p}) = |y - \hat{p}|$$

- Where y is observed class and $\hat{p}$ is predicted probability. Logistic model's absolute loss is the average of these across all instances.

Absolute Loss of the Breast Cancer Logistic Regression.

Logistic regression $\hat{p}$



Absolute loss for a benign tumor with a radius mean of 14.05 mm

$$L_{\text{abs}}(0, 0.286) = |0 - 0.286| = 0.286$$

Overall absolute loss

$$(L_{\text{abs}}(0, \hat{p}(11.25)) + \ldots + L_{\text{abs}}(0, \hat{p}(14.05)) + \ldots + L_{\text{abs}}(1, \hat{p}(19.95)))/\#\text{instances}$$

$$= (0.0239 + \ldots + 0.286 + \ldots + 0.982)/171$$

$$= 0.183$$

Log Loss

- Likelihood assesses model-data consistency; higher values mean better fit. Due to small values, log-likelihood is preferred. Higher likelihood yields smaller log loss.

- The **log loss** is the negative log-likelihood of a probability predicted by a logistic model. For an instance, the log loss is given by:

$$L_{log}(y, \hat{p}) = -(ylog(\hat{p}) + (1 - y)\log(1 - \hat{p}))$$

- Where $\hat{p}$ is predicted probability of positive outcome and y is observed class. Closer $\hat{p}$ to 0/1, instance's log loss is near 0.

- Logistic model's log loss is average of these across all instances.

False Positives and False Negatives

- Binary classifier predicts positive/negative outcome.

- Predictions can be correct or incorrect. Combinations summarized below:
  - A **true positive** (*TP*) is an outcome that was correctly identified as positive.
  - A **true negative** (*TN*) is an outcome that was correctly identified as negative.
  - A **false positive** (*FP*) is an outcome that was identified as positive but was actually negative.
  - A **false negative** (*FN*) is an outcome that was identified as negative but was actually positive.

- **Confusion matrix** summarizes predicted vs. actual values. For binary classifiers, it's a 2x2 table.

Accuracy, Precision, and Recall

- **Accuracy**: Accuracy measures overall correct predictions out of all predictions, calculated as:

$$\text{Accuracy} = \frac{\#Correctly\ Predicted}{\#Total} = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Precision**: Precision quantifies correct positive predictions among all positive predictions, calculated as:

$$\text{Precision} = \frac{\#True\ Positive}{\#Predicted\ Positive} = \frac{TP}{TP+FP}$$

- **Recall** (Sensitivity): Recall gauges correct positive predictions among all actual positives, calculated as:

$$\text{Recall} = \frac{\#True\ Positive}{\#Actual\ Positive} = \frac{TP}{TP+FN}$$

Precision-Recall Tradeoff

- Precision is the proportion of correct positive predictions among all predicted positives.
- Recall (Sensitivity) is the proportion of correct positive predictions among all actual positives.
- There's a trade-off: increasing precision often decreases recall and vice versa, impacting classifier performance.

Receiver Operating Characteristic (ROC) Curve

- An ROC curve assesses binary class separation across probability cutoffs by plotting true positive rate against false positive rate.

- The AUC, area under the ROC curve, compares performance of two binary models.

- Larger AUC indicates better binary class prediction.

Cross Validation

- **Cross-Validation for Better Performance Assessment:**
  - Splitting data into training, validation, and test sets might exclude key instances from training.
  - Resampling methods, like cross-validation, provide improved model performance assessment.

- **k-Fold Cross-Validation:**
  - Common approach, divides sample data into groups (folds).
  - Model trained/validated repeatedly using different fold combinations.

- **Animated Example with Linear Model:**
  - Animation illustrates 5-fold cross-validation evaluating linear model's performance using "bad drivers" dataset.
  - Dataset includes car accidents, insurance premiums data from NHTSA and NAIC.

Cross Validation

- **k-Fold Cross-Validation and Bias-Variance:**
  - Value of *k* in *k*-fold cross-validation involves bias-variance tradeoff.
  - Larger *k* means smaller validation sets, more variability; smaller *k* leads to increased bias.
- **Computational Impact and Model Complexity:**
  - Larger *k* requires more models trained on larger training folds, increasing computation.
  - For complex models or large datasets, a smaller *k* might be needed.
- **Balancing Factors:**
  - Choosing *k* depends on balancing bias, variance, computational requirements, and dataset/model complexity.

Types of Cross-Validation Methods

- Cross-validation includes various resampling setups beyond k-fold cross-validation. It's categorized in two main ways:

- An **exhaustive cross-validation** uses every possible way to divide the sample into training and validation sets of desired sizes.
  - (k = #instances in the training data -1)

- A **non-exhaustive cross-validation** does not use every possible way to divide the sample into training and validation sets of desired sizes.
  - k ≠ #instances in the training data -1

Bootstrapping

- When population information is missing, **bootstrapping** simulates sampling by drawing replacements from existing samples.

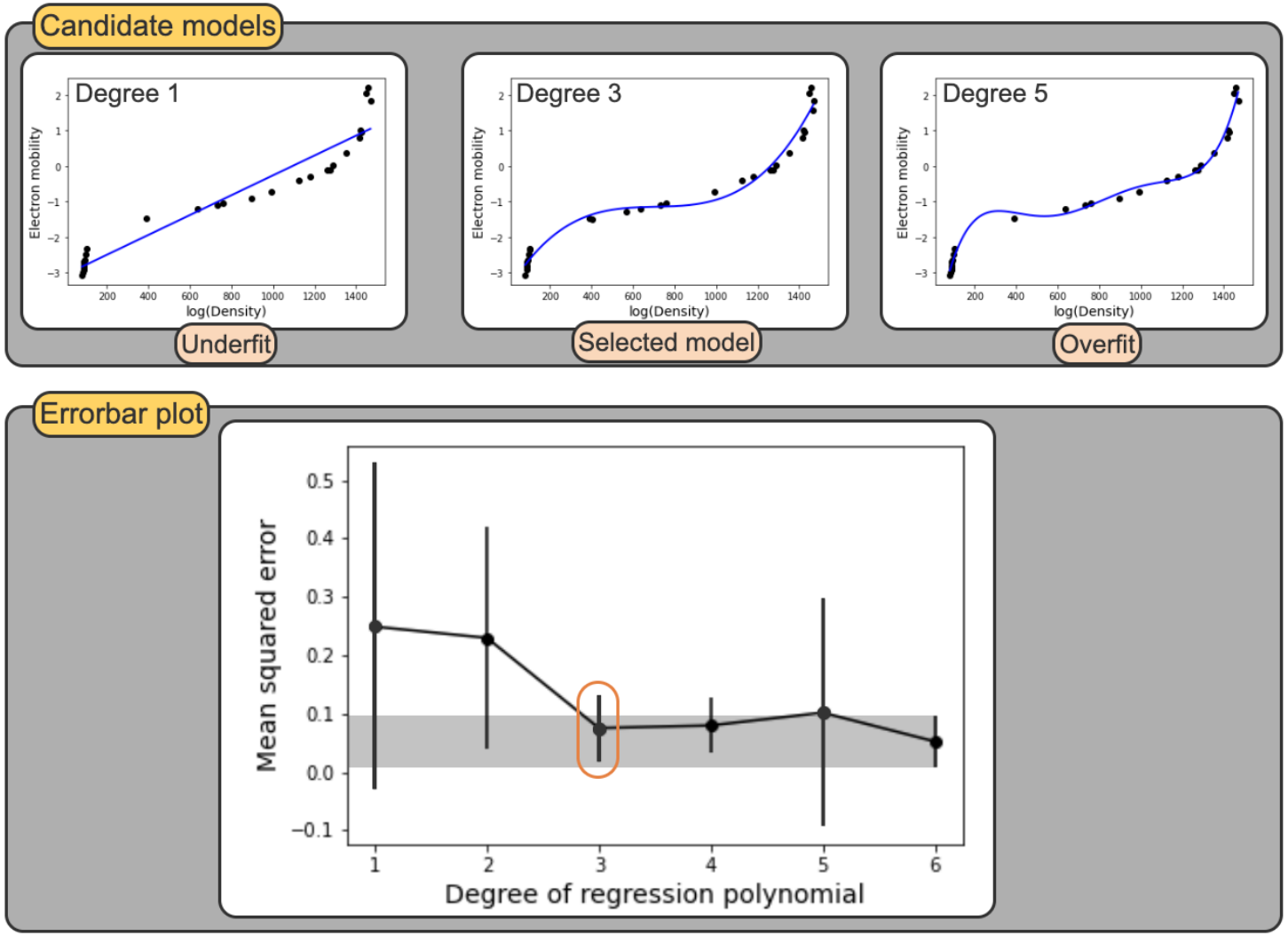
- It assesses parameter estimation through simulated samples and their statistic distribution, enabling interval estimation.

- For instance, it can create confidence intervals for proportions like incumbent candidate voters using representative exit poll interviews.

Bootstrap Method of Model Evaluation

Step	Description
Step 1: Draw bootstrap samples.	A specified number of bootstrap samples of a specified size are drawn with replacement from the existing sample.
Step 2: Generate out-of-bag samples.	Since bootstrap samples are drawn with replacement, not all instances are selected. Instances not selected form a corresponding "out-of-bag" sample.
Step 3: Train models and calculate errors.	Models are trained using the bootstrap samples. For each model, error is calculated using the corresponding out-of-bag sample as validation data.
Step 4: Examine the errors.	The characteristics of the distribution of errors can be used to evaluate the model's performance. Ex: If the distribution of errors indicates the model has a small error variance, and the model is correctly specified, then the model is considered to perform well.

Model Selection



Bootstrapping

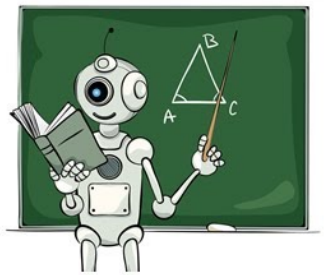
- **Model Selection Beyond Metrics:**
  - Models can be chosen based on other numerical indicators of quality.

- **Information Criterion Approach:**
  - Select model with lowest information criterion, balancing data fit and model complexity.
  - Common criteria: Akaike's (AIC) and Bayesian (BIC).

- **Adjusted R-squared Method:**
  - Opt for model with highest adjusted R-squared (R^2_{adj}).
  - Adjusts R-squared by considering feature count.

Case Studies

Home Prices



Supervised Learning

Next Lecture

Supervised Learning