

# CS/ENGR M148 L10: Midterm Review

Sandra Batista

### **Quiz 2 on PS2 today!**

Only 15 minutes, T/F, multiple choice/select

Please bring laptop and hard copy of notes.

### **Midterm next Tuesday, 11/5!**

100 minutes. Covering lectures 1-10

Please hard copy of notes. We will scan exams.

### **For CAE accommodations:**

Please schedule your testing at CAE testing center for quizzes, midterm (100 minutes regular time), and final by 10/29/24.

**Please contact TA first for homework submission help.** I can help if TA cannot resolve.

**This week in discussion section:**

Lab on KNN classification

Project Data Check-in: Already posted on BruinLearn

**New for Project Check-ins Early:**

11am-11:50am Fridays in Boelter 5436 with our wonderful

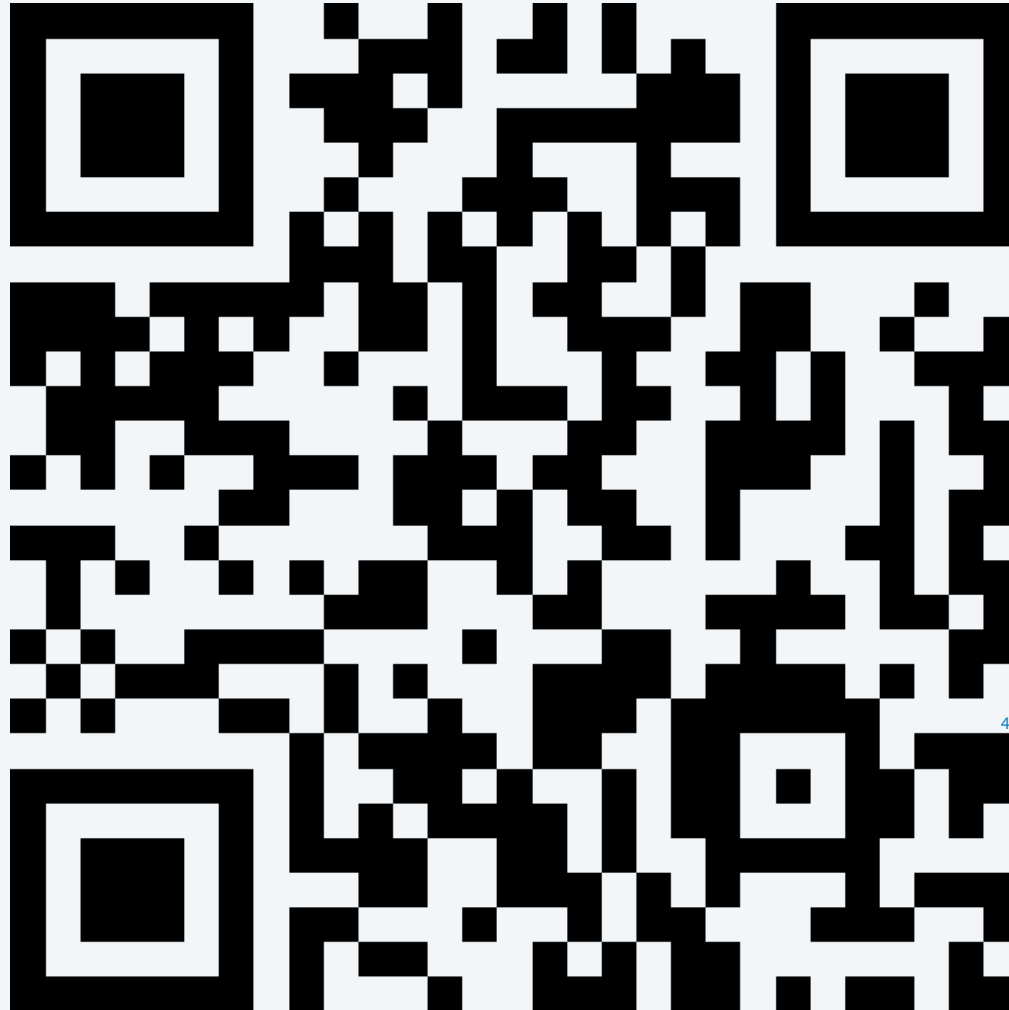
TA Yihe

LAs will be helping with project check-ins!

# Join our slido for the week...

---

<https://app.sli.do/event/vb9RXFWoKnxhYMBAnTwdgA>



# Metrics from EDA

***Know how to calculate:***

1. *Mean*
2. *Correlation*
3. *Variance*

$$SD(x) = \sqrt{\text{Var}(x)} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}.$$

$$\text{Corr}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}},$$

# The Regression Line

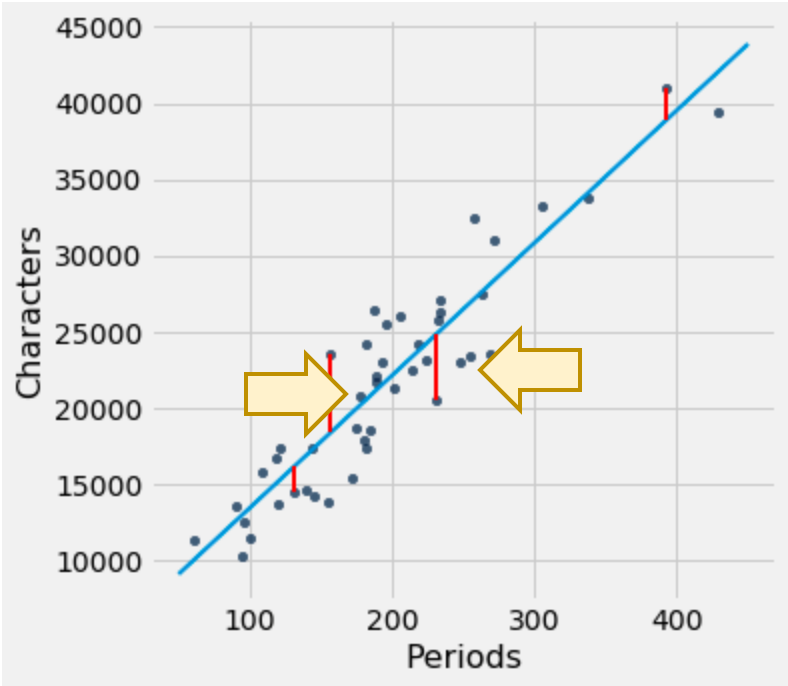
From Data 8 ([textbook](#)):

The **regression line** is the unique straight line that minimizes the **mean squared error** of estimation among all straight lines.

$$\begin{aligned} \text{intercept} &= \text{average of } y \\ &\quad - \text{slope} \times \text{average of } x \\ \text{regression estimate} &= \text{intercept} + \text{slope} \times x \end{aligned}$$

**residual**

$$= \text{observed } y - \text{regression estimate}$$



For every chapter of the novel *Little Women*, Estimate the **# of characters**  $\hat{y}$  based on the **number of periods**  $x$  in that chapter.

# The Multiple Linear Regression Model Using Matrix Notation

We can express our linear model on our entire dataset as follows:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \mathbf{X}\boldsymbol{\theta}$$

Prediction vector  
 $\mathbb{R}^n$

Design matrix  
 $\mathbb{R}^{n \times (p+1)}$

Parameter vector  
 $\mathbb{R}^{(p+1)}$

Note that our **true output** is also a vector:  
 $\mathbf{Y} \in \mathbb{R}^n$



# Least Squares Loss Function

**L2 loss function or squared loss:**

$$\frac{1}{n} \sum_{i=1}^n (\text{observed response}_i - \text{predicted response}_i)^2.$$

$$\frac{1}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2.$$



# Closed Form for Estimates

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$b_0 = \overline{\text{observed price}} - b_1 \times \overline{\text{area}} \text{ and}$$

$$b_1 = \frac{\sum_{i=1}^{10} (\text{area}_i - \overline{\text{area}})(\text{observed price}_i - \overline{\text{observed price}})}{\sum_{i=1}^{10} (\text{area}_i - \overline{\text{area}})^2},$$

$$\text{predicted price} = 61,190 + 64 \times \text{area}.$$

# Error metrics to know

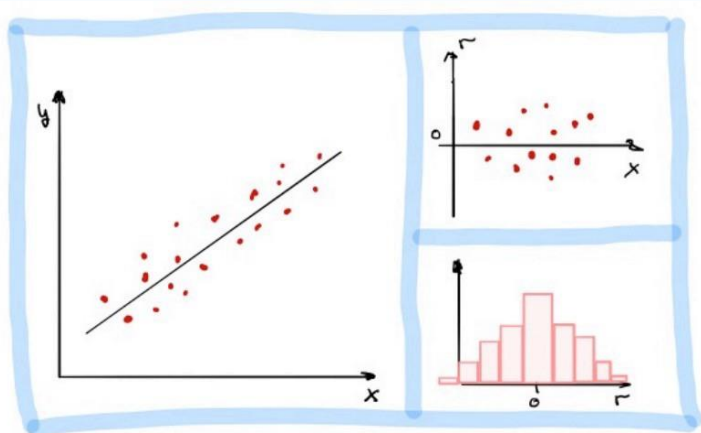
**root mean squared error, rMSE**

**mean absolute error, MAE**

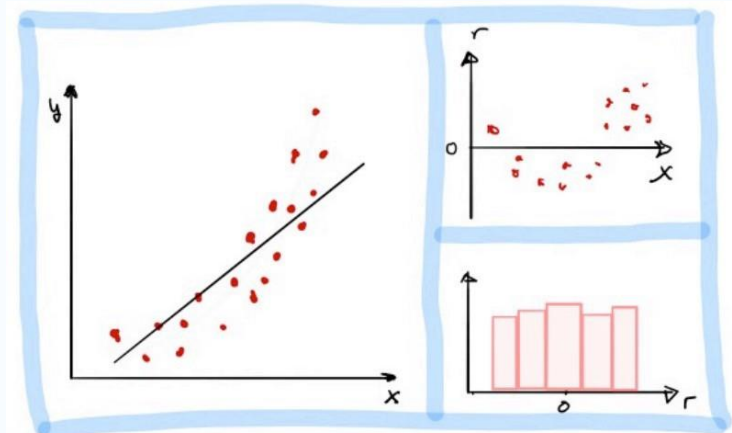
**median absolute deviation, MAD**

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

# Residual Analysis



Linear assumption is correct. There is no obvious relationship between residuals and  $x$ . Histogram of residuals is symmetric and normally distributed.



Linear assumption is incorrect. There is an obvious relationship between residuals and  $x$ . Histogram of residuals is symmetric but not normally distributed.

*Note: For multi-regression, we plot the residuals vs predicted value since there are too many  $x$ 's and that could wash out the relationship.*

# Residual Analysis

***Residuals*** are the difference between the predicted and observed values.

*In residual analysis, we typically create two types of plots:*

- 1. Plot residuals against predictor variable or predicted value in a scatterplot*
- 2. Plot histogram of residuals*

# Regularization

**Regularization** is a technique that forces predictive algorithms to simpler solutions by adding constraints to the minimization/optimization problem.

- Adds penalty based on weights to the loss function.
  - Automated feature selection technique
  - Addresses overfitting (too many features)
  - Collinearity of features
- 
- Best practice: **Standardize variables before regularization**

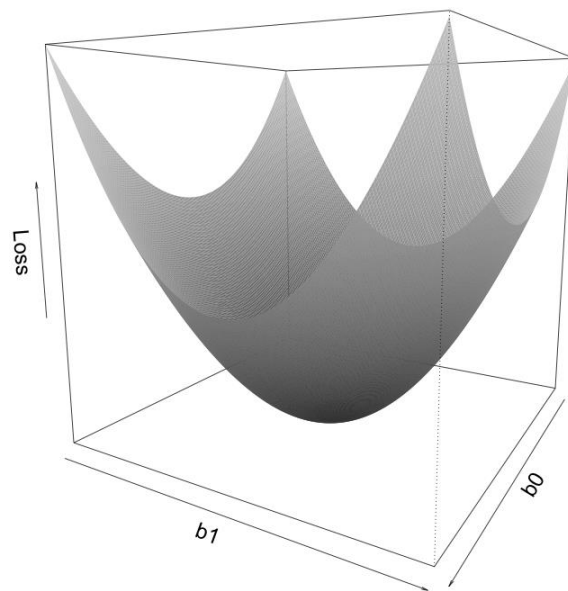
# Ridge Regression

*Find the values of  $b_0$  and  $b_1$  that make the regularized LS loss*

$$\sum_i (\text{observed price}_i - (b_0 + b_1 \text{area}_i))^2 + \lambda(b_0^2 + b_1^2)$$

*as small as possible (for some  $\lambda \geq 0$ ).*

- Quadratic (squared) L2 **penalty term** is called **L2 regularization**
- **Regularization hyperparameter is  $\lambda$**



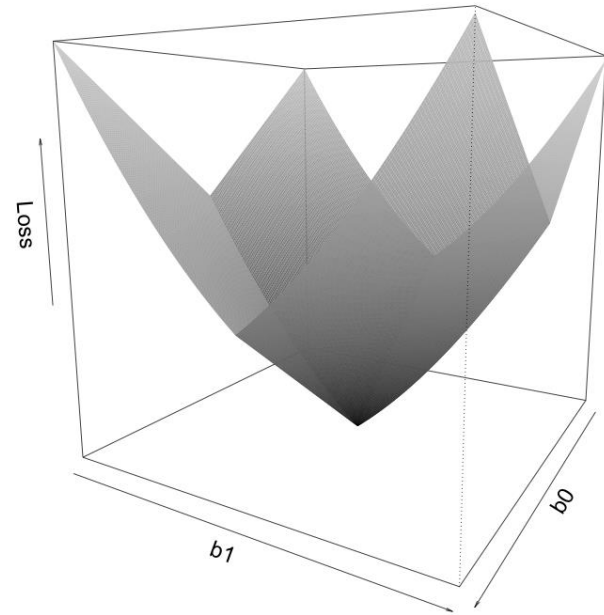
# Lasso Regression

Find the values of  $b_0$  and  $b_1$  that make the regularized LS loss

$$\sum_i (\text{observed price}_i - (b_0 + b_1 \text{area}_i))^2 + \lambda(|b_0| + |b_1|)$$

as small as possible (for some  $\lambda \geq 0$ ).

- Absolute value or L1 **penalty term** is called **L1 regularization**
- **Regularization hyperparameter is  $\lambda$**



# Your turn:

## Regularization and CV

Please review Lecture 6 Jupyter notebook for concrete examples with numbers:

Go to:

[https://colab.research.google.com/drive/12z5-caCx9wWoGzRJEuDBkHwEdGSfq\\_mA?usp=sharing](https://colab.research.google.com/drive/12z5-caCx9wWoGzRJEuDBkHwEdGSfq_mA?usp=sharing)

Save a copy to your Google Drive and keep notes there...



# Overfitting

*Overfitting* occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

*Ways to address:*

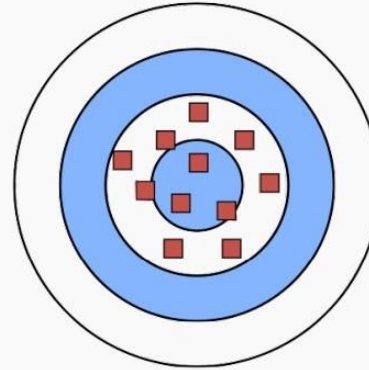
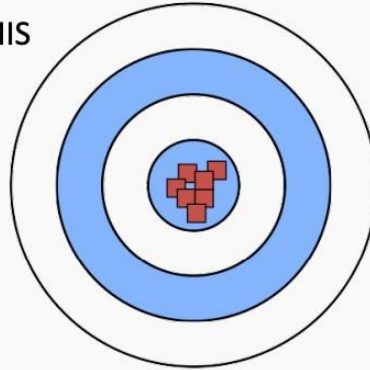
- 1. Model selection: Limiting the number of parameters in model*
- 2. Using more validation data sets*

**Low Variance**  
(Precise)

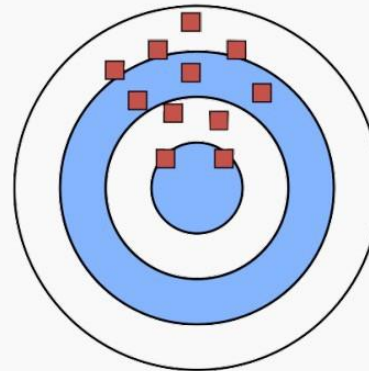
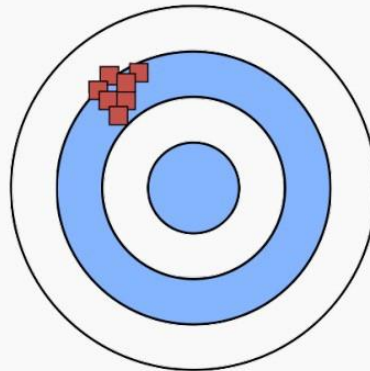
**High Variance**  
(Not Precise)

WE WANT THIS

**Low Bias**  
(Accurate)



**High Bias**  
(Not Accurate)



Nobody cares

# Cross Validation

Create **multiple** validation sets and average the validation performance In practice

Separate data into  $V$  non-overlapping folds. Leave fold out. Train on rest of data. Get metrics from fold and average metrics across folds.

5-fold CV (i.e.,  $V=5$ ) is common for small datasets up to a few thousand data points),

10-fold ( $V=10$ ) CV is common for larger datasets.

**Leave-one-out cross-validation:** Each data point is a fold, so  $V = n$  where  $n$  is the number of

# Model Selection

*What are some methods you learned for model selection?*

# Classification

A binary response is often referred to as the **class label** of the observation.

**Classification problems:** Prediction problems with binary responses that involve *classifying* each observation as belonging to one of the two classes.

(It is possible to have more than 2 classes...)

# Categorical Variables

**Categorical variables** are variables that do not have numerical measurements (e.g. neighborhood in Ames housing data)

Categorical variables can be **ordinal** if categories can be sorted.

Categorical variables can be **nominal** if categories do not have specific order.

Categorical variables can be made converted to numeric values (e.g. one-hot encoding)

# Logistic regression

We apply a **logistic** transformation to the equation to get valid probabilities from the predictor

**Logistic regression** uses a *logistic* linear combination to predict the *probability* of a class label (success).

# Log Odds or Logit Function

The log odds (logit function) corresponds to the logarithm of the odds ratio:

$$\log \left( \frac{p}{1-p} \right).$$

The log odds is an unbounded continuous number.

*We apply the logit function to the probability, so it equals a linear combination of predictors:*

$$\log \left( \frac{p}{1-p} \right) = b_0 + b_1 \times \text{product-related duration}$$



# Logistic Function

*We invert the logit function and solve for the probability to get the **logistic function**:*

Logistic regression computes binary response probability predictions,  $p$ , based on the logistic-transformed linear combination:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}},$$

where  $x$  is a relevant predictive feature.

*Logistic regression uses this function to compute values for Parameters.*

# What is the loss function?

***Logistic Loss function*** to minimize.

*Make probabilities small for 1 class  
and close to 0 for 0 class*

$$\sum_{i \text{ in pos class}} (-\log p_i) + \sum_{i \text{ in neg class}} (-\log(1 - p_i)).$$

*No nice closed form. Can use techniques  
such as Maximum Likelihood Estimation  
(MLE). Regularization can be used.*

# Interpreting coefficients

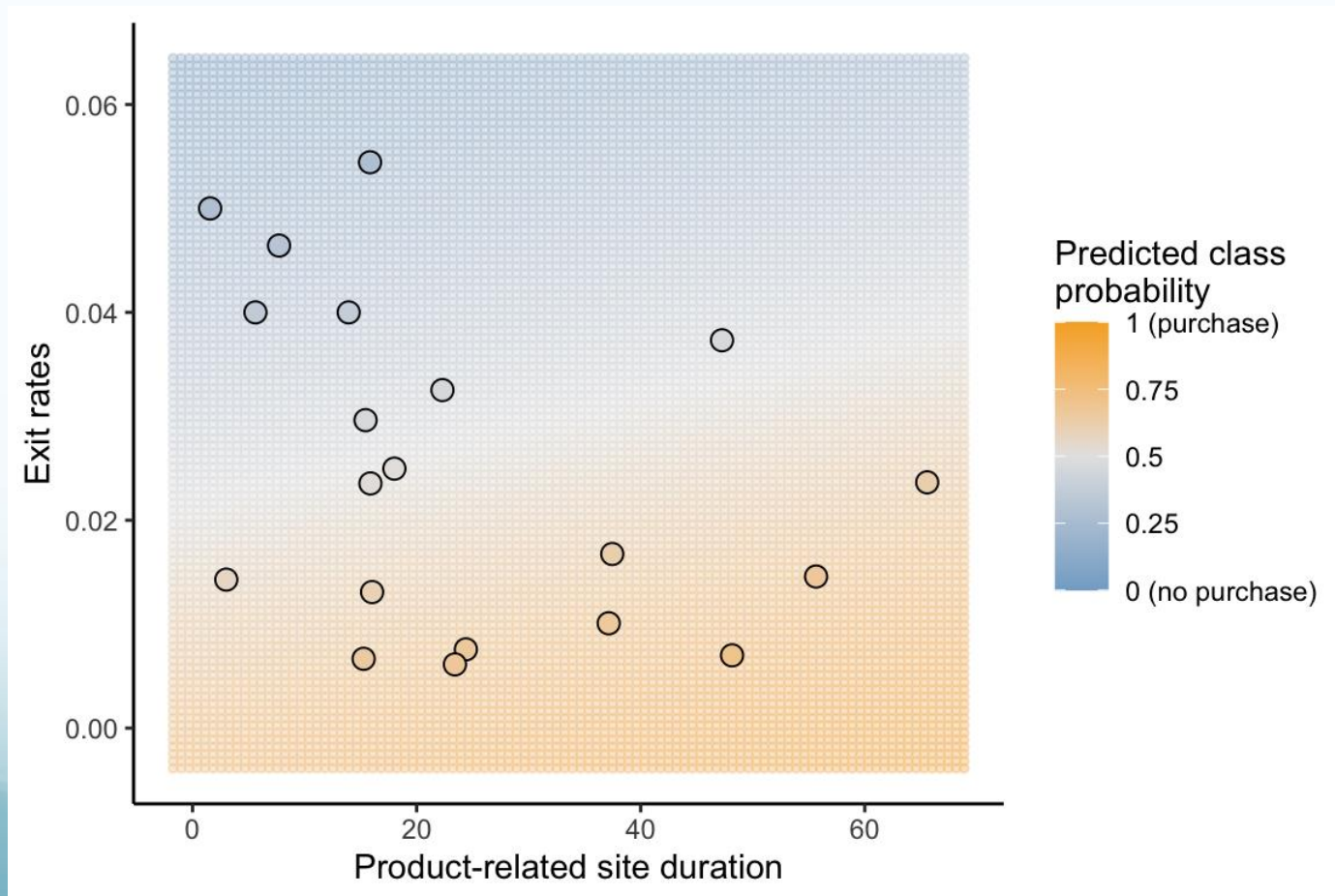
*Consider logistic regression on UCI shopping data:*

$$\log \left( \frac{p}{1-p} \right) = b_0 + b_1 \text{product-related duration} + b_2 \text{exit rates},$$

*What do coefficients mean?*

# What are decision boundaries?

*Decision boundaries are surface where classification changes from positive (1 for purchase) to negative (0 for no purchase) :*

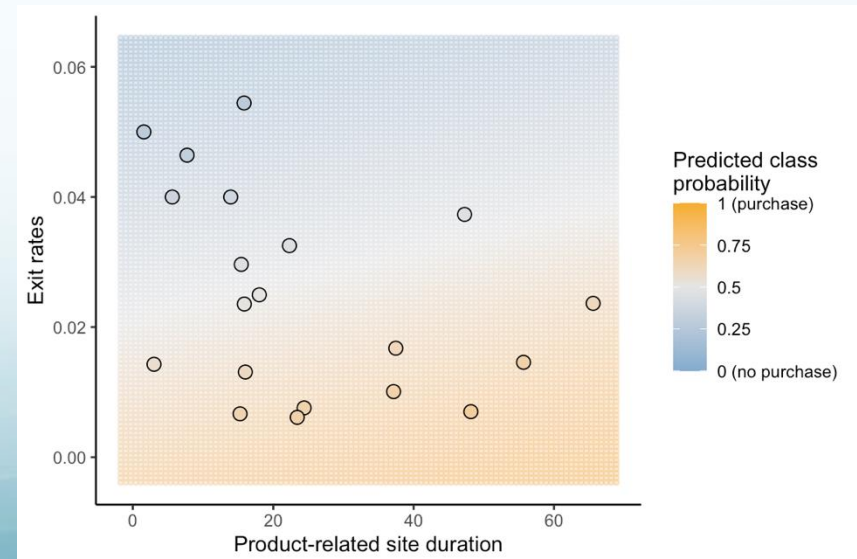


# What are decision boundaries?

*Decision boundaries are surface where classification changes from positive (1 for purchase) to negative (0 for no purchase) :*





$$\log \left( \frac{p}{1-p} \right) = b_0 + b_1 \text{product-related duration} + b_2 \text{exit rates},$$

*What would this decision boundary be when  $p=.5$ ?*



# Confusion matrix

The **confusion matrix** is a 2-by-2 table that cross-tabulates the predicted and observed binary response.

		Predicted	
		positive	negative
Observed	positive	 # true pos	 # false neg
	negative	 # false pos	 # true neg

# Evaluating Classification

## Prediction accuracy

$$\text{prediction accuracy} = \frac{(\text{number of correct predictions})}{n}$$

## Prediction error

$$\text{prediction error} = \frac{(\text{number of incorrect predictions})}{n}$$

# Sensitivity and Specificity

The **true positive rate** or “**sensitivity**” or “**recall**”

$$\begin{aligned}\text{true positive rate} &= \frac{(\text{number of correctly predicted positive class obs})}{(\text{number of positive class observations})} \\ &= \frac{(\text{number of true positives})}{(\text{number of positive class observations})}\end{aligned}$$

The **true negative rate** (often called “**specificity**”) is the proportion of negative class observations whose class is correctly predicted

False positive rate = 1-specificity

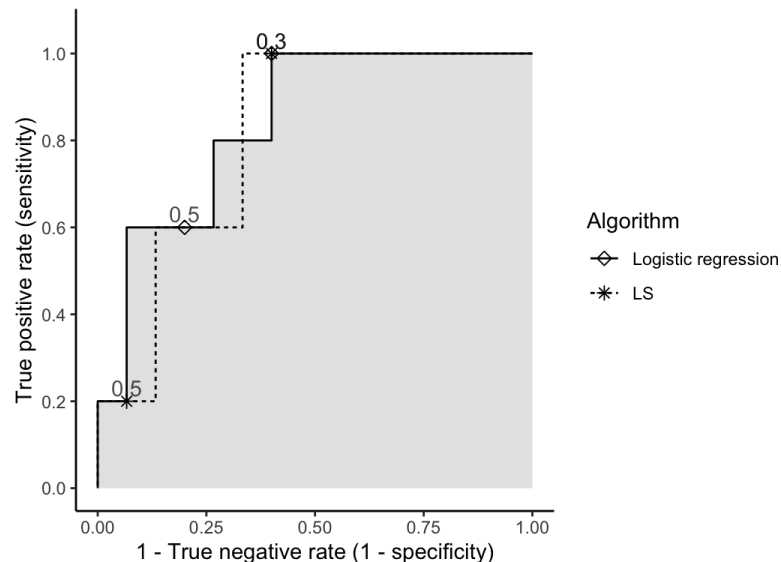
Tradeoff between sensitivity and specificity



# ROC Curves

**Receiver Operating Characteristics (ROC) curve** plot true positive rate against (1- true negative rate) or false positive rate for various thresholds to compare models and algorithms.

**Area under the curve (AUC)** quantifies predictive potential of algorithm by computing the literal area under the ROC curve.



# Your turn...

Compute the prediction accuracy, prediction error, true positive rate, and true negative rate for the following confusion matrix for 394 samples. (It is fine to leave fractions)

	<b>Predicted positive</b>	<b>Predicted negative</b>
Observed positive	122	8
Observed negative	59	205

# Minkowski distance

The *Minkowski distance* between two instances **a** and **b** in a feature space with  $m$  descriptive features is:

$$\text{Minkowski}(\mathbf{a}, \mathbf{b}) = \left( \sum_{i=1}^m \text{abs}(\mathbf{a}[i] - \mathbf{b}[i])^p \right)^{\frac{1}{p}}$$

where different values of the parameter  $p$  result in different distance metrics

- The Minkowski distance with  $p = 1$  is the Manhattan distance and with  $p = 2$  is the Euclidean distance.
- The larger the value of  $p$  the more emphasis is placed on the features with large differences in values

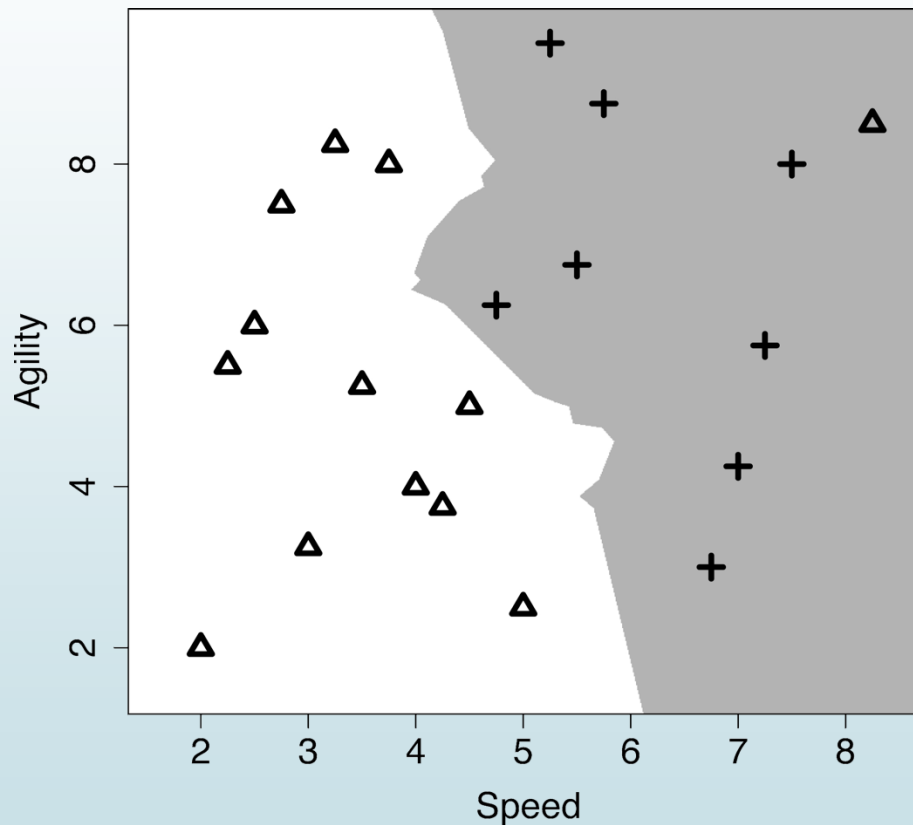
## kNN (k nearest neighbor)

1. As in the general problem of classification, we have a set of data points for which we know the correct class labels.
2. When we get a new data point, we compare it to each of our existing data points and find similarity.
3. Take the most similar  $k$  data points ( $k$  nearest neighbors).
4. From these  $k$  data points, take the majority vote of their labels. The winning label is the label/class of the new datapoint.

**Choice of  $k$  will affect classification and is hyperparameter.**

# Decision Boundaries

***Decision boundaries** are the surfaces separating classes in classification.*



***Figure:** The decision boundary using majority classification of the nearest 3 neighbors.*

# Classification and Regression Algorithm (CART)

*The **Classification and Regression Algorithm (CART)** aims to split data to minimize the variance in child nodes.*

*‘Variance’ is different for continuous and binary variables.*

*Important observation: Nodes can be a mix of different types of classes of observations, but want to minimize this.*

# Variance Split for Continuous Response

*The **variance for split** is a weighted variance of the left and right child nodes.*

$$\text{Variance measure for split} = \frac{n_{\text{left}}}{n_{\text{parent}}} \text{Var}_{\text{left}} + \frac{n_{\text{right}}}{n_{\text{parent}}} \text{Var}_{\text{right}}$$

*Variance decreases in child nodes.*

# How is Gini impurity related to Bernoulli Variable Variance?

*Let  $p_1$  be probability of success (positive class)*

*And  $p_0$  probability of failure or (negative class)*

$$\text{Gini impurity} = 1 - p_1^2 - p_0^2 = 2p_0p_1,$$



# Gini impurity for split

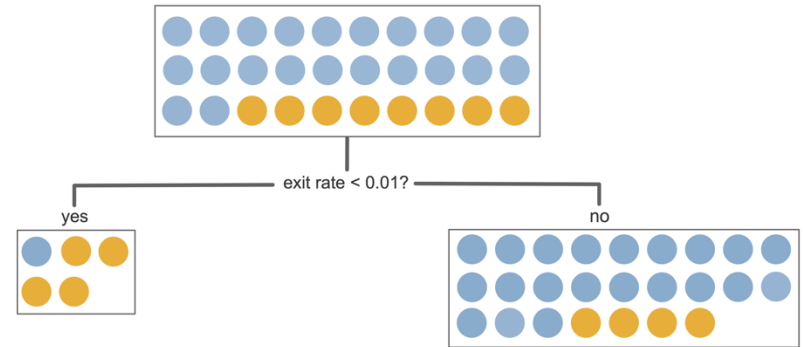
**Gini impurity for split** is weighted average of Gini impurity for left and right child nodes.

$$\text{Gini impurity for split} = \frac{n_{\text{left}}}{n_{\text{parent}}} \text{Gini}_{\text{left}} + \frac{n_{\text{right}}}{n_{\text{parent}}} \text{Gini}_{\text{right}}$$

Gini impurity decreases for child nodes

# Gini impurity calculation from last lecture..

*Calculate the Gini impurity of the left and right nodes*



# Your turn...

Compute the Gini impurity for the following potential split of a set of 30 observations (8 of which are in the positive class and 22 of which are in the negative class). (Fine to leave fractions)

The *left child node* has 7 positive class observations and 10 negative class observations, and the *right child node* has 1 positive class observation and 12 negative class observations

# CART summary

*Start with a top-level parent node that contains all the training data observations. Then:*

1. Conduct a **greedy search** for the “best” split of the parent node by identifying splits, calculating variance metric, and select split to minimize variance metric.
2. Implement the best split identified in the previous step to create two child nodes.
3. For each resulting child node, repeat steps 1 and 2 until stopping criteria. A child node that is not split further is called a “**leaf node**” or “**terminal node**”.
4. Continue until all child nodes are **leaf nodes**.
5. Generate predictions using average continuous response or positive class proportion.

# Random Forests

**Random Forest (RF)** algorithm is an **ensemble** algorithm that combines many decision trees:

- Training each tree using a different random bootstrap sample of the training data.
- Considering a different random subset of the predictor variables for each node split
- Predictions are aggregates of the decisions from trees

# A word on feature importance

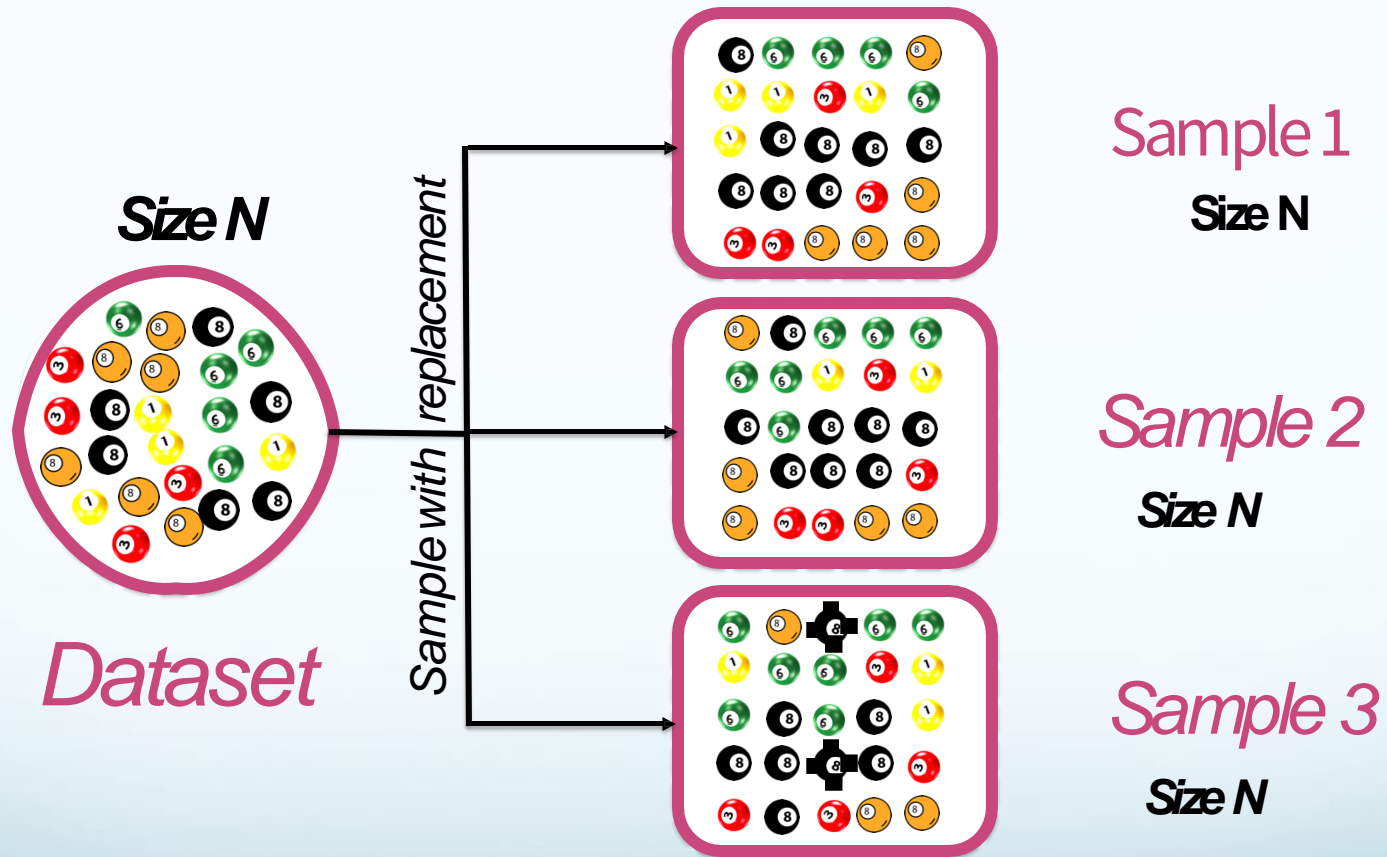
Sometimes we want to learn about features in a model not just the predictions.

Regression and Logistic regression had **parameters** that we were estimating.

Those **parameters** were the coefficients of the predictors in the models.

To use coefficients to compare features, we must **standardize the features** before fitting the model or **standardize the coefficients.**

# Bootstrap



# A word on feature importance

Notice that with decision trees and random forests, we are not learning any **parameters** for the models: **non-parametric supervised learning**

Very important: Decision Trees and Random Forest are **invariant to monotonic transformations** of the underlying variables, (i.e. logarithmic, square-root transformations or standardization do not affect results)

However, we may still want to learn what features are helping us in predictions.