CS/ENGR M148 L7:
Logistic Regression

Sandra Batista

Quiz on Problem Set 1 during class on 10/24/24.

Only 15 minutes. Multi-select, multiple choice, T/F questions.

<u>Please bring laptop to take quiz and hard copy of notes.</u>

For CAE accommodations:

1) We will email you with specific details as we are waiting for CAE reply.
2) Schedule your testing at CAE testing center for midterm (100 minutes regular time) by 10/29/24.

This week in discussion section:

Lab on logistic regression

Project Data Check-in: Your team will need to demonstrate a logistic regression model on your project data.

Join our slido for the week…

Today's Learning Objectives

Students will be able to:

- Use bootstrapping to compare coefficients in regression
- Review: Apply **cross validation** and **regularization** to address overfitting
- Handle **categorical variables**
- Understand **classification problems** and **use logistic regression** on real data
- Evaluate classification problems with quantitative metrics

Using multiple predictors

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - \text{predicted price}_i)^2 \\
&\quad = \frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - (b_0 + b_1\text{area}_i + b_2\text{quality}_i \\
&\qquad\qquad + b_3\text{year}_i + b_4\text{bedrooms}_i))^2.
\end{aligned}$$

[Yu, Barter 2024]

Using the model in matrix form

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

Can you extract this data from the house data?
To fit the model can you call sklearn
linear_model.LinearRegression()?

Using multiple predictors

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

How do we interpret the coefficients?

[Yu, Barter 2024]

Comparing coefficients

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

How do we compare the coefficients? Can we simply use how large some coefficients are compared to others?

[Yu, Barter 2024]

Comparing coefficients

The coefficients of different predictive features (predictor variables) are not comparable unless

- They're on the same scale

- Coefficients have been standardized to create **t-values:**

$$t_j = \frac{b_j}{SD(b_j)}.$$

[Yu, Barter 2024]

Creating t-values

1. Create N (e.g., $N = 100$ or $N = 1,000$) bootstrapped versions of the original dataset so that each of the N bootstrapped datasets has the same number of observations as the original data.

2. For each of the N bootstrapped datasets, compute an LS fit and extract the relevant coefficient value (so that you have N versions of each coefficient value).

3. Compute the SD of the N bootstrapped coefficients.

$$t_j^{\text{boot}} = \frac{b_j}{SD^{\text{boot}}(b_j)}.$$

[Yu, Barter 2024]

Bootstrap

Bootstrapping is the practice of sampling from the observed data (X,Y) in estimating statistical properties.

Bootstrap

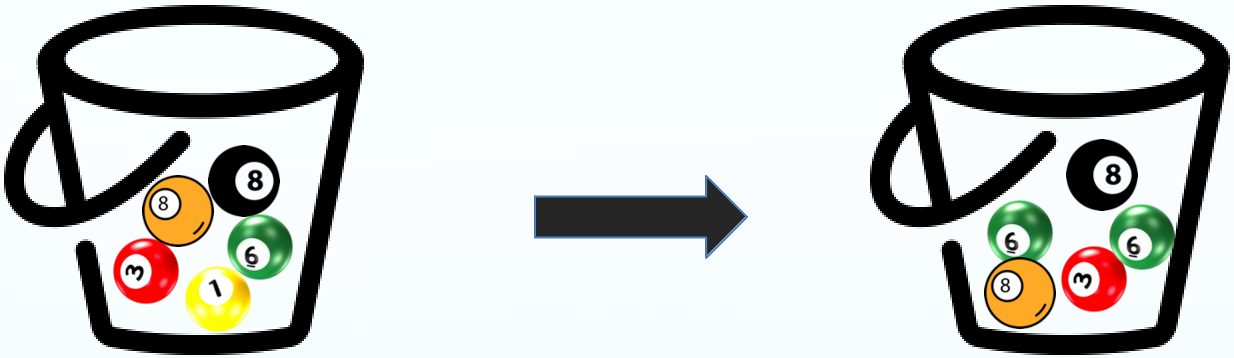
We pick a ball and replicate it and move it to the other bucket. This is ***sampling with replacement.***

Bootstrap

We then randomly pick another ball and again we replicate it. As before, we move the replicated ball to the other bucket.

Bootstrap

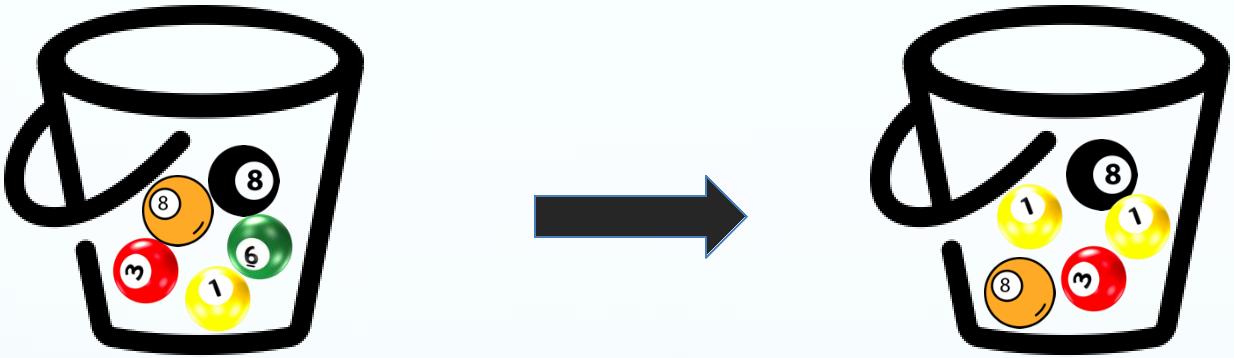
*We continue until the "other" bucket has **the same number of balls** as the original one.*



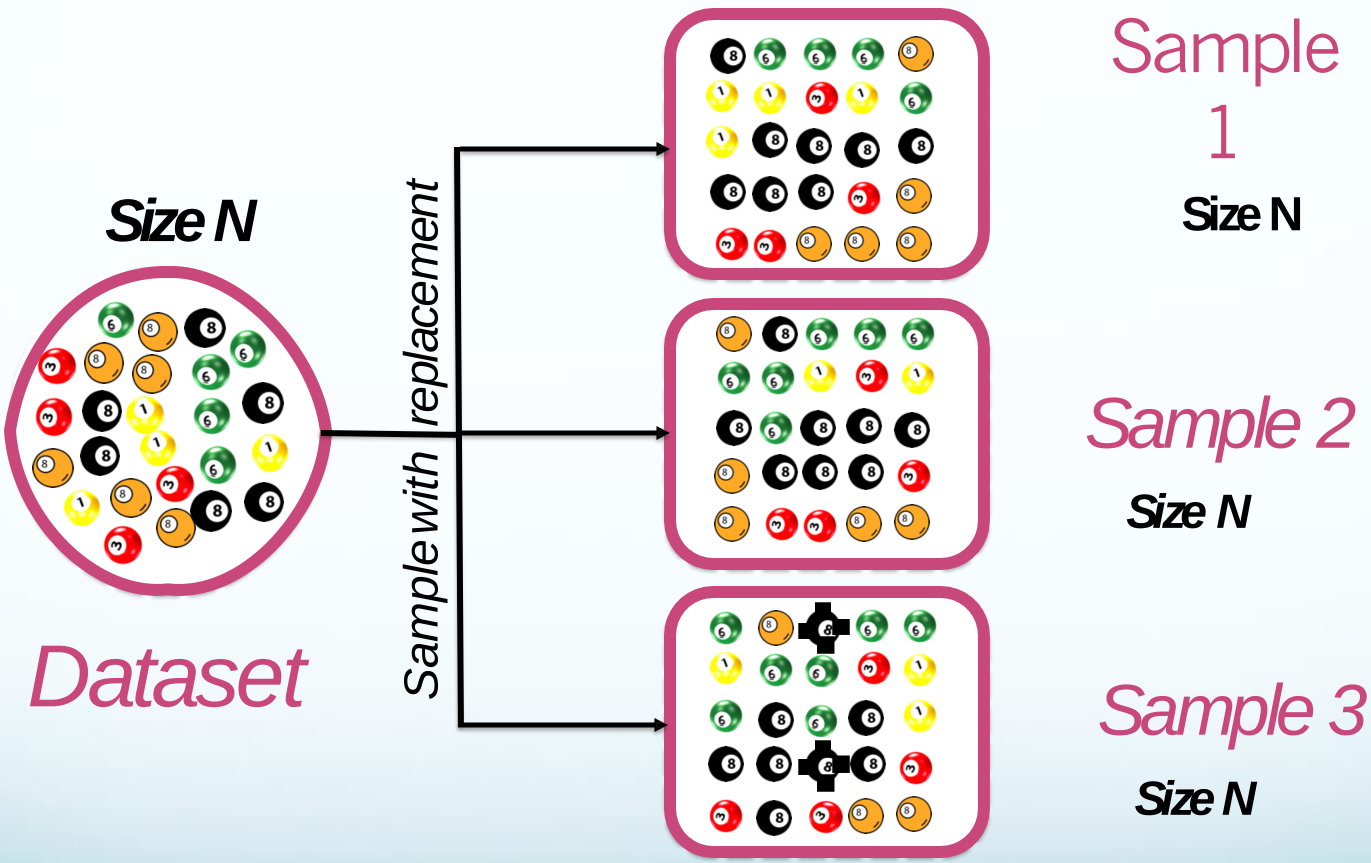
This new bucket represents a new sample

Bootstrap

We repeat the same process and acquire another sample.



Bootstrap


Size N
Dataset
Sample with replacement
Sample 1
Size N
Sample 2
Size N
Sample 3
Size N

Your turn: Comparing coefficents

Please get the Jupyter notebook

Go to:
https://colab.research.google.com/drive/1sNq6g5W_8Y-z__YigvVm0FhUPIOxOPGf?usp=sharing

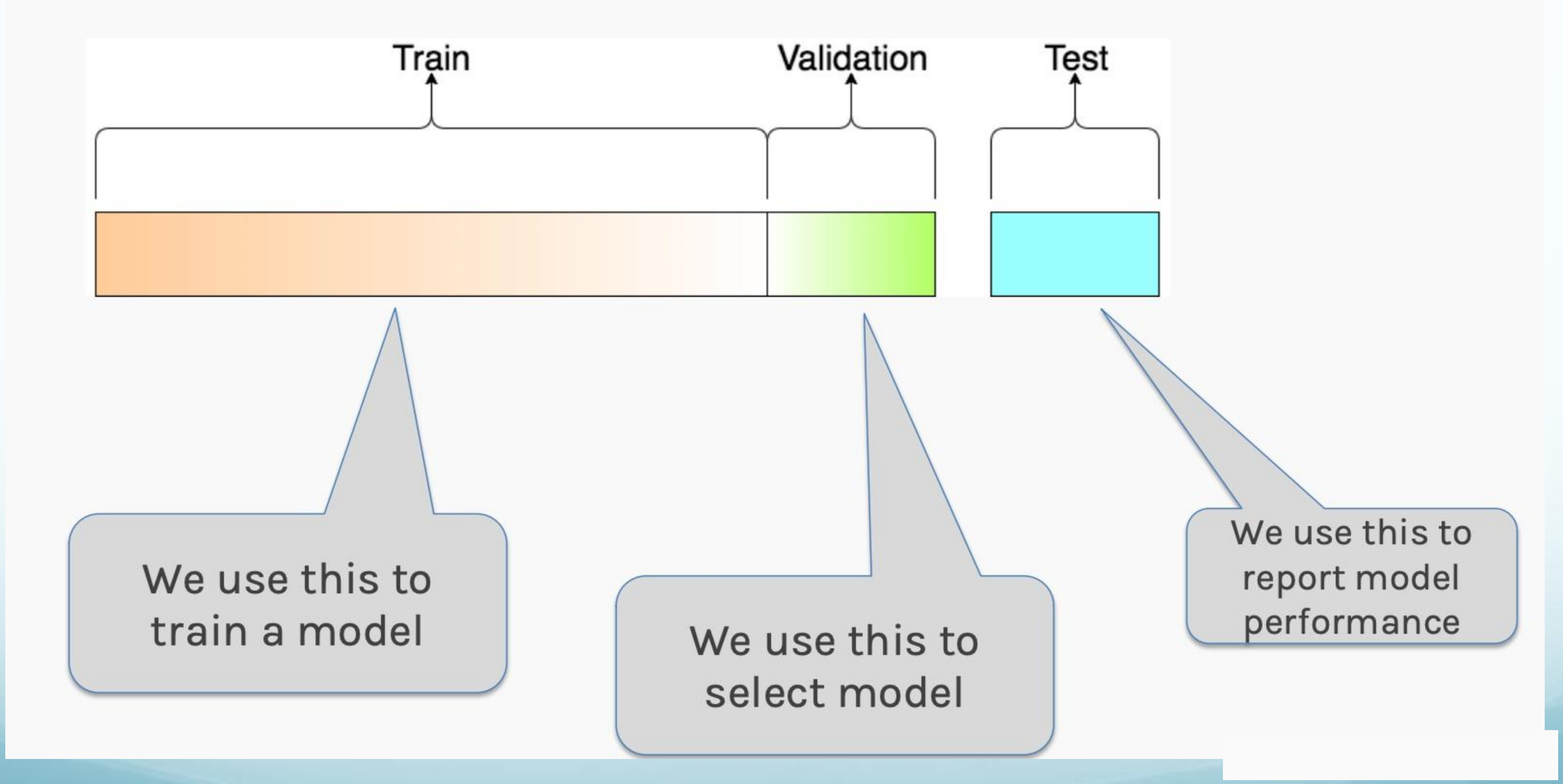
[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- ✔ Use bootstrapping to compare coefficients in regression
- ✕ Review: Apply **cross validation** and **regularization** to address overfitting
- ✕ Handle **categorical variables**
- ✕ Understand **classification problems** and **use logistic regression** on real data
- ✕ Evaluate classification problems with quantitative metrics

Train-Validation-Test



V-Fold Cross Validation

V-fold CV aims to emulate the training/validation split for evaluating results on "pseudo"-validation data. The general process is as follows:

1. Split the data into V equal-sized non-overlapping subsets, called *folds*.
2. Remove the first fold (this fold will play the role of the pseudo-validation set), and use the remaining $V-1$ folds (the pseudo-training set) to train the algorithm.
3. Use the withheld first fold (the pseudo-validation set) to evaluate the algorithm that you just trained on the other $V-1$ folds using a relevant performance measure.

V-Fold Cross Validation

4. Replace the first fold, and remove the second fold (the second fold is now the pseudo-validation set). Train the algorithm using the other $V - 1$ folds (including the previously withheld first fold). Evaluate the algorithm on the withheld second fold.

5. Repeat step 4 until each fold has been used as the pseudo-validation set, and you have V values of the performance measure for your algorithm.

6. Combine (e.g., compute the mean of) the V performance measure values, each computed on a withheld fold.

Leave-One-Out

In practice:
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

Regularization

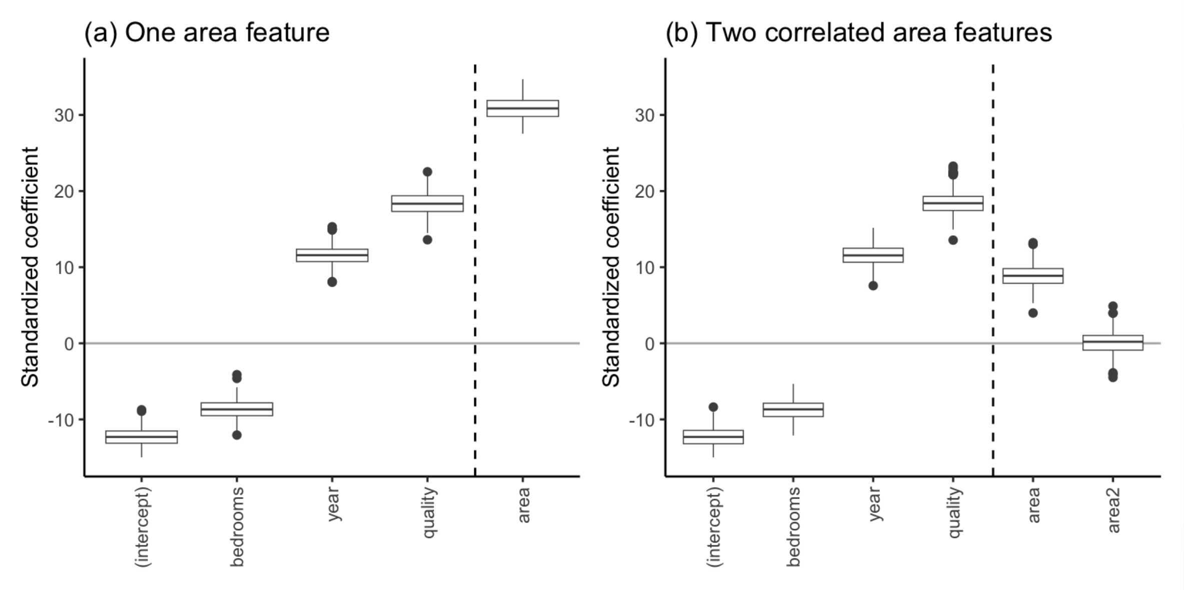
Regularization is a technique that forces predictive algorithms to simpler solutions by adding constraints to the minimization/optimization problem.

- 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**

Collinearity Example

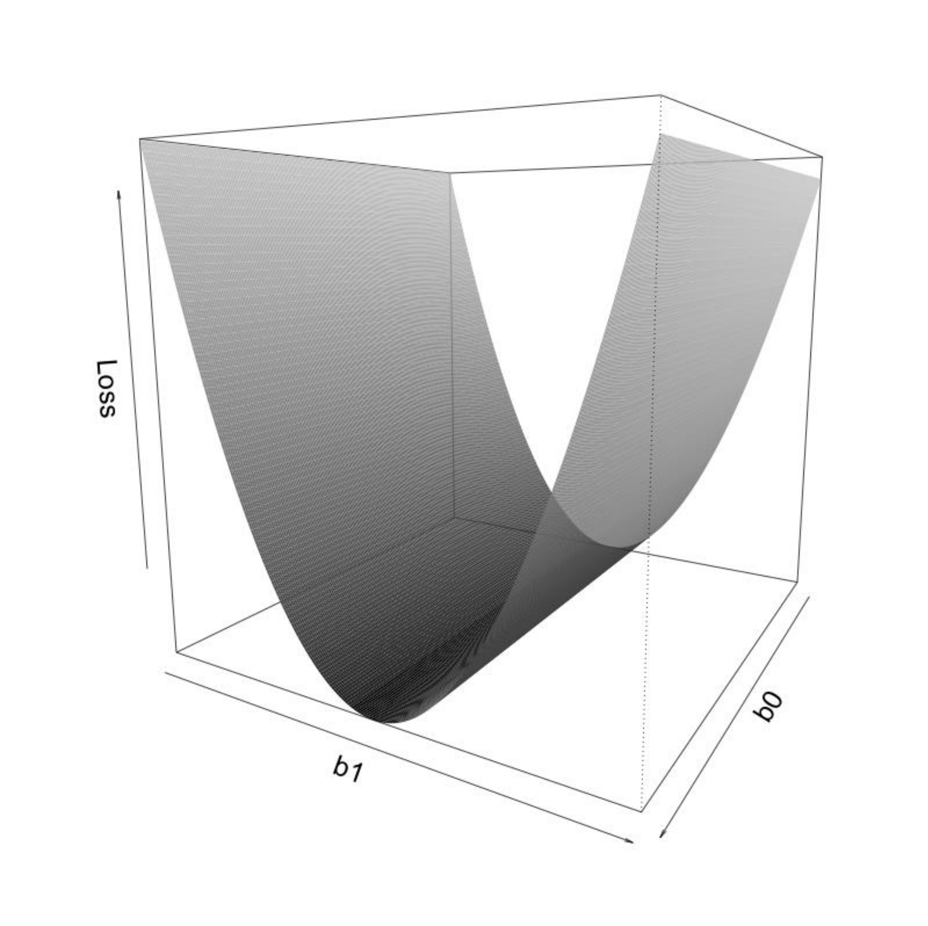
$$\text{predicted price} = b_0 + b_1\text{year} + b_2\text{bedrooms} + b_3\text{quality} + b_4\textbf{area}.$$



$$\text{predicted price} = b_0 + b_1\text{year} + b_2\text{bedrooms} + b_3\text{quality} + b_4\textbf{area} + b_5\textbf{area2}.$$

How do we optimize this?

$$predicted\ price = b\_0 + b\_1 \times area.$$



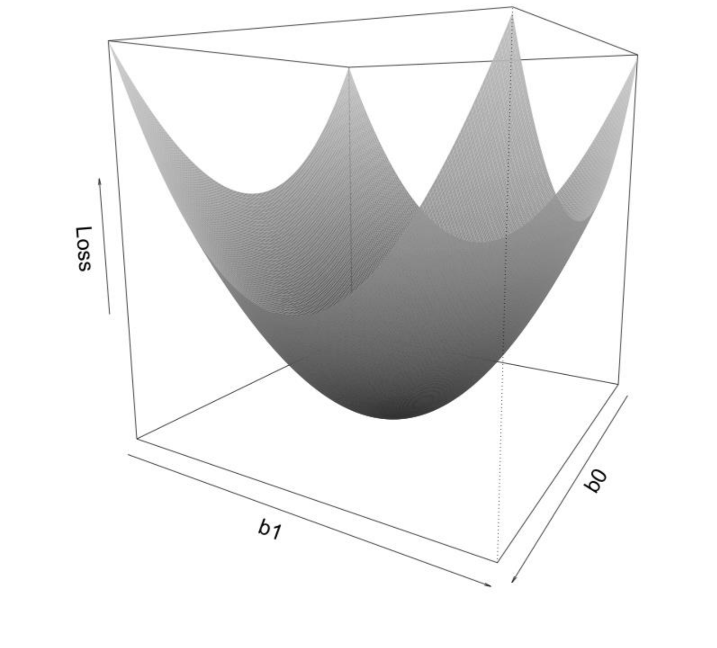
Ridge Regression

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 ƛ**



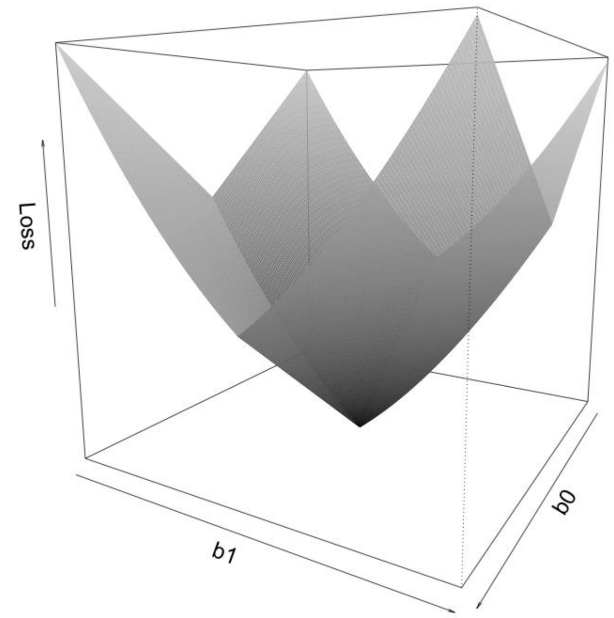
Lasso Regression

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 λ**



Lasso: least absolute shrinkage and selection operator

How to choose hyperparameters?

- We can use **cross validation!**

1. Decide on a range of potential values for the penalty term, λ. Note that many software implementations will do this for you.
2. Split the data into V (e.g., $V = 10$) nonoverlapping folds of approximately the same size.
3. Remove the first fold (this fold will play the role of the pseudo-validation set), and use the remaining $V - 1$ folds (which will play the role of the pseudo-training set) to train the regularized LS fit using each value of λ.
4. Calculate the error (e.g., mean squared error (MSE)) for each of the regularized LS fits—one for each λ—using the first withheld CV-fold pseudo-validation set.

How to choose hyperparameters?

- We can use **cross validation!**

5. Replace the withheld first fold and now remove the second fold (the second fold will now play the role of the pseudo-validation set). Use the remaining $V - 1$ folds to train the algorithm for each value of λ. Evaluate the fits using the withheld second fold (e.g., using MSE).

6. Repeat this process until all the V folds have been used as the withheld validation set, resulting in V measurements of the algorithm's performance for each λ.

7. For each value of λ, calculate the average of the V errors. The average of the V errors is called the *CV error*.

8. Select the λ that had the lowest CV error, or that you judge to be the best (e.g., taking stability into consideration).

Your turn: Regularization and CV

Please get the Jupyter notebook

Go to:
https://colab.research.google.com/drive/1sNq6g5W_8Y-z__YigvVm0FhUPIOxOPGf?usp=sharing

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

[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- ✓ Use bootstrapping to compare coefficients in regression
- ✓ Review: Apply **cross validation** and **regularization** to address overfitting
- ✗ Handle **categorical variables**
- ✗ Understand **classification problems** and **use logistic regression** on real data
- ✗ Evaluate classification problems with quantitative metrics

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)

[Yu, Barter 2024]

Categorical Variables Example

```python
import pandas as pd
shirts = pd.DataFrame([
                    ['green', 'M', 10.2, 'long'],
                    ['red', 'L', 13.5, 'short'],
                    ['blue', 'XL', 14.5, 'long']])
shirts.columns = ['color', 'size', 'price', 'sleeve length']
print(shirts)
```

```
   color size  price sleeve length
0  green    M   10.2          long
1    red    L   13.5         short
2   blue   XL   14.5          long
```

[Raschka 2022]

Using Dictionary

```
[13] size_mapping = {'M': 1, 'L': 2, 'XL': 3}
     shirts['size'] = shirts['size'].map(size_mapping)
     print(shirts)
```

```
    color  size  price sleeve length
0   green     1   10.2          long
1     red     2   13.5         short
2    blue     3   14.5          long
```

[Raschka 2022]

Using Label Encoder

```
[14] from sklearn.preprocessing import LabelEncoder
     sleeve_le = LabelEncoder()
     shirts['sleeve length'] = sleeve_le.fit_transform(shirts['sleeve length'].values)
     print(shirts)
```

```
   color  size  price  sleeve length
0  green     1   10.2              0
1    red     2   13.5              1
2   blue     3   14.5              0
```

[Raschka 2022]

One-Hot Encoding

```
pd.get_dummies(shirts,drop_first=True, dtype=int)
```

	size	price	sleeve length	color_green	color_red
0	1	10.2	0	1	0
1	2	13.5	1	0	1
2	3	14.5	0	0	0

[Raschka 2022]

Today's Learning Objectives

Students will be able to:

- ✔ Use bootstrapping to compare coefficients in regression
- ✔ Review: Apply **cross validation** and **regularization** to address overfitting
- ✔ Handle **categorical variables**
- ✕ Understand **classification problems** and **use logistic regression** on real data
- ✕ Evaluate classification problems with quantitative metrics

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…)

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…)

Logistic regression

As an example we'll consider UCI shopping data set in notebook, let's predict purchase made or not.

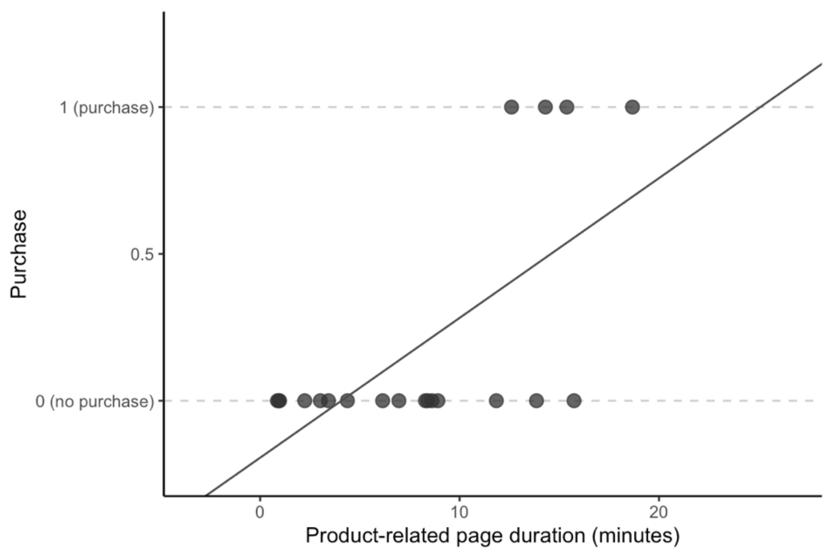
With binary response variable linear combinations of variable and least squares won't work:

$$\text{predicted purchase} = -0.194 + 0.048 \times \text{product-related duration},$$

Logistic regression

As an example we'll consider UCI shopping data set in notebook, let's predict purchase made or not.

With binary response variable linear combinations of variable and least squares won't work:



$$\text{predicted purchase} = -0.194 + 0.048 \times \text{product-related duration},$$

Logistic regression

Rather than trying to predict binary response variable, we try to predict continuous **probability of binary variable…**

$$\text{predicted purchase } probability = b_0 + b_1 \times \text{product-related duration.}$$

But something is still wrong, what?

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).

Odds Ratio

The odds (odds ratio) corresponds to the probability of a "success," p, divided by the probability of a "failure," $1 - p$:

$$\frac{p}{1-p}.$$

The odds ratio is bounded between 0 and ∞.

Example: what are odds of raining today if probability of rain is 75%?

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:***

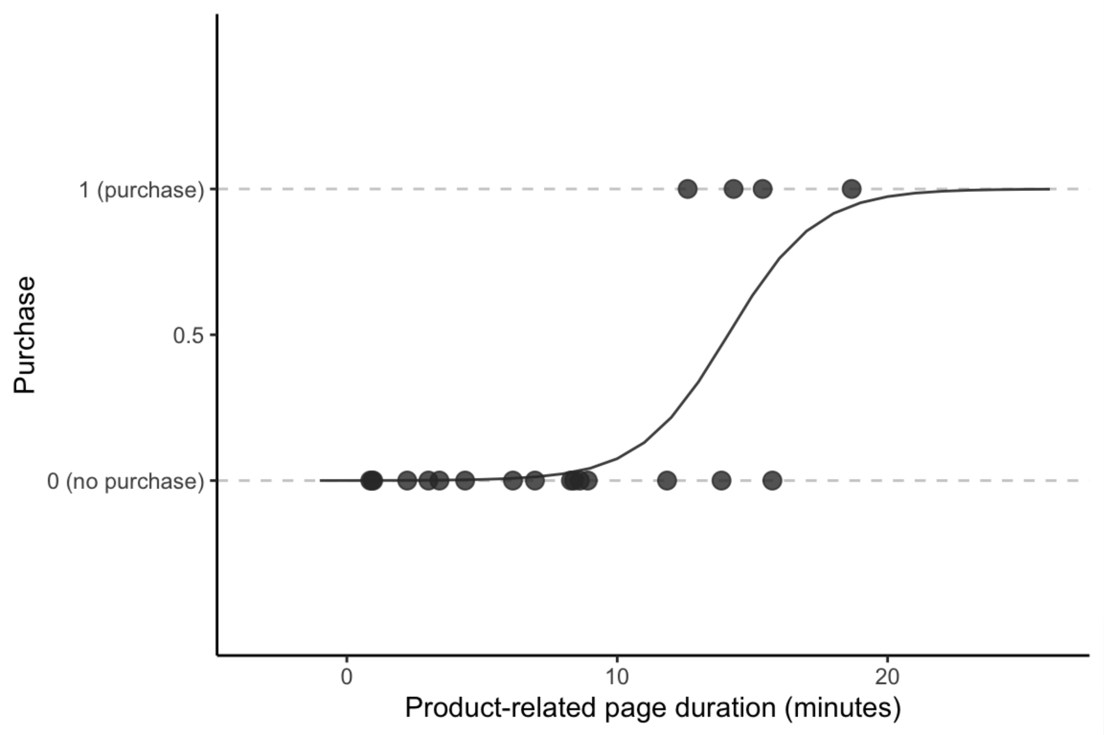
Logistic regression computes binary response probability predictions, p, based on the logistic-transformed linear combination:

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

where x is a relevant predictive feature.

Logistic regression uses this function to compute values for Parameters.

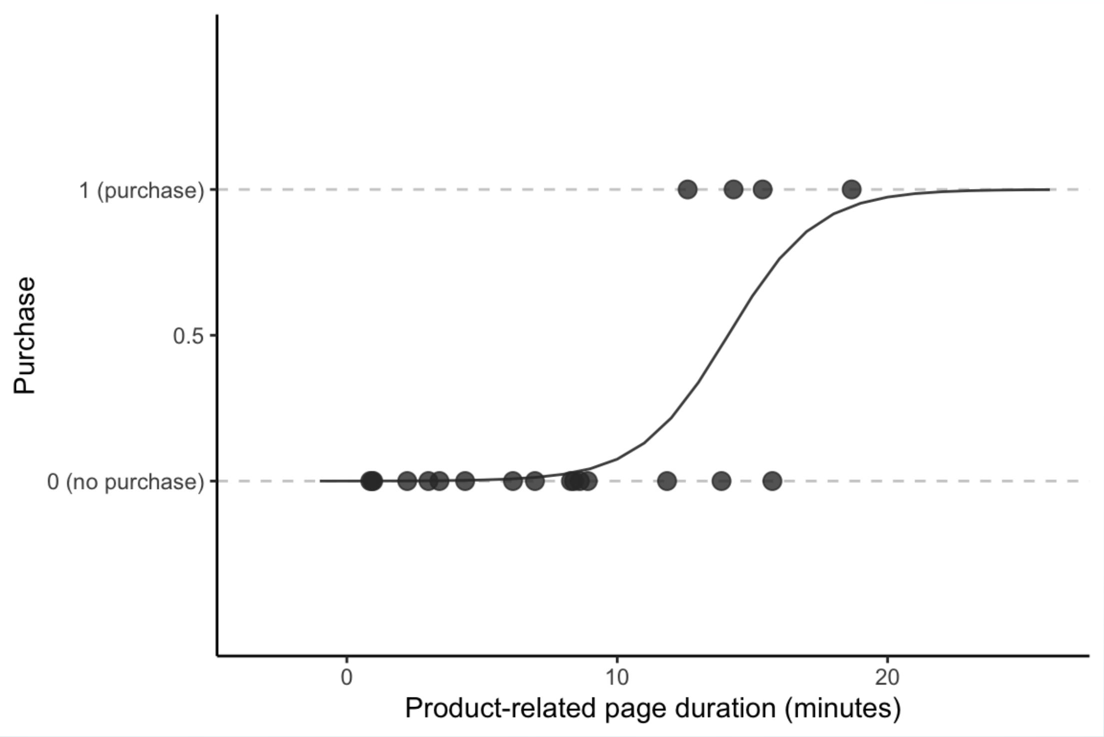
Logistic Function



$$\text{predicted purchase probability} = \frac{1}{1 + e^{-(-8.639 + 0.613\text{product-related duration})}}$$

How to get predictions?

If p greater than or equal to threshold (.5) predict 1 (or purchase)
Otherwise 0 (no purchase)



$$\text{predicted purchase probability} = \frac{1}{1 + e^{-(-8.639+0.613\text{product-related duration})}}$$

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)

Today's Learning Objectives

Students will be able to:

- ✓ Use bootstrapping to compare coefficients in regression
- ✓ Review: Apply **cross validation** and **regularization** to address overfitting
- ✓ Handle **categorical variables**
- ✓ Understand **classification problems** and **use logistic regression** on real data
- ✗ Evaluate classification problems with quantitative metrics

Evaluating Classification

Prediction accuracy is proportion of observations for which the binary predicted response label matches the observed response label.

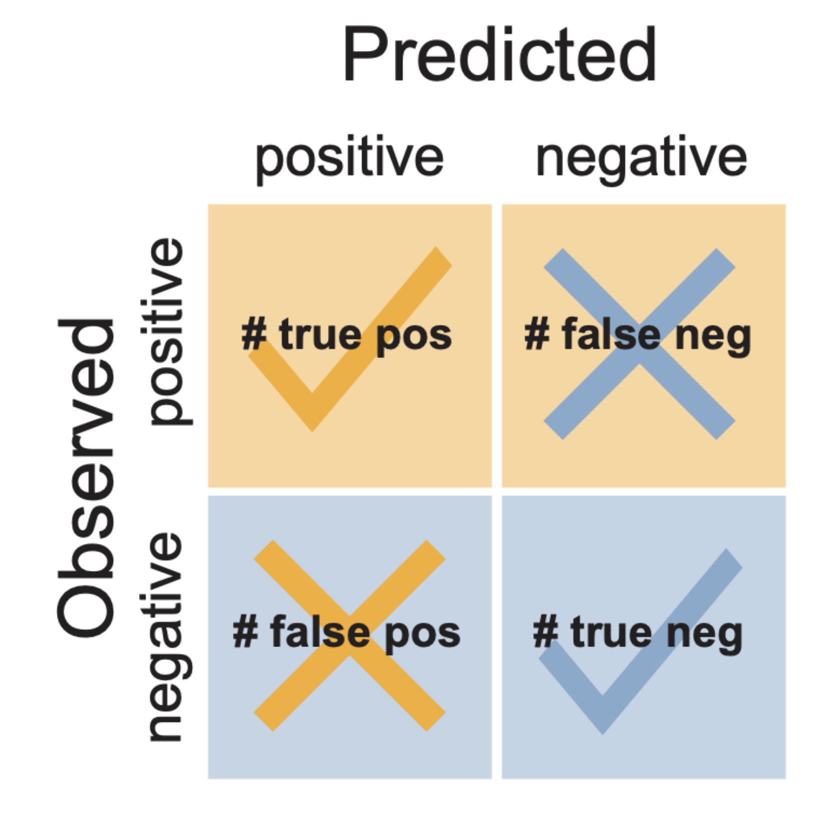
$$\text{prediction accuracy} = \frac{\text{(number of correct predictions)}}{n}$$

Prediction error corresponds to the proportion of observations for which the binary predicted response label is *different* from the observed response label.

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

Confusion matrix

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



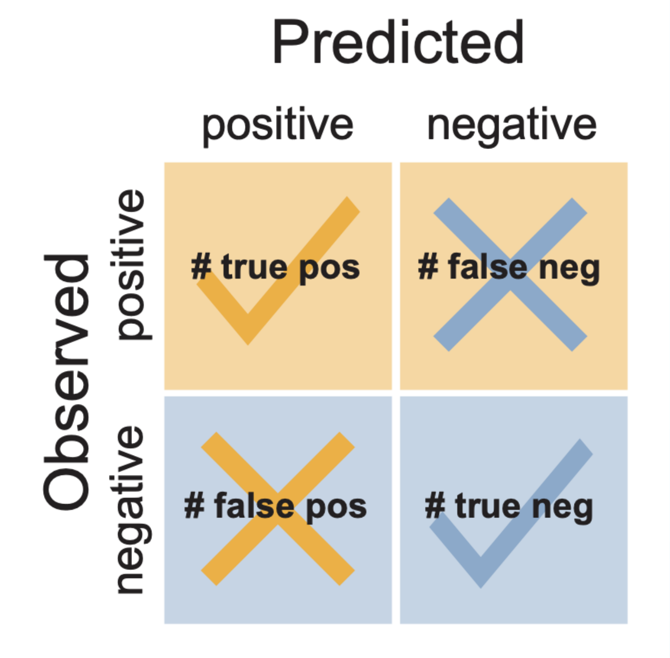
Confusion matrix

What is a **true positive?**

What is a **true negative?**

What is a **false positive?**

What is a **false negative?**



Sensitivity or True Positive Rate

The **true positive rate** (often called "**sensitivity**" or "**recall**") is the proportion of positive class observations whose class is correctly predicted.

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

	Predicted purchase	Predicted no purchase
Observed purchase	3	2
Observed no purchase	3	12

Specificity or True Negative Rate

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

	Predicted purchase	Predicted no purchase
Observed purchase	3	2
Observed no purchase	3	12

False Positive Rate

The **false positive rate** is the proportion of negative observations *incorrectly* predicted to be positive.

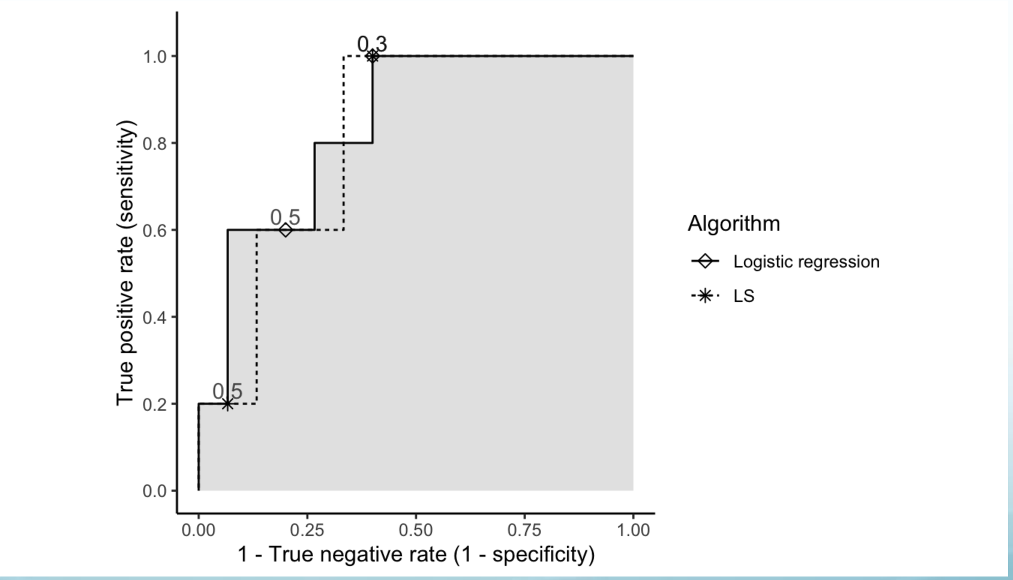
How does this relate to the true negative rate?

	Predicted purchase	Predicted no purchase
Observed purchase	3	2
Observed no purchase	3	12

ROC Curves

Receiver Operating Characteristics (ROC) curve plot true positive rate against true negative 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: Logistic Regression

Please get the Jupyter notebook for logistic regression on shopping data:

Go to:
https://colab.research.google.com/drive/1wazvX6RQGUYRMJEK46tRgHqMyJokTdFC?usp=sharing

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

[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- ✓ Use bootstrapping to compare coefficients in regression
- ✓ Review: Apply **cross validation** and **regularization** to address overfitting
- ✓ Handle **categorical variables**
- ✓ Understand **classification problems** and **use logistic regression** on real data
- ✓ Evaluate classification problems with quantitative metrics

Citations:

Yu, B., & Barter, R. L. (2024). Veridical data science: The practice of responsible data analysis and decision making. The MIT Press.

Shah. C. (2020) A hands-on introduction to data science. Cambridge University Press.

Sebastian **Raschka**, Yuxi (Hayden) **Liu**, and Vahid Mirjalili. **Machine Learning** with PyTorch and Scikit-Learn. Packt Publishing, **2022**.