Data C100/200 - Final

Spring 2026

Write your name BIG and clearly: _____

Email: _____@berkeley.edu

Student ID: _____

Name and SID of the person on your left: _____

Name and SID of the person on your right: _____

Exam Room:_____ Seat Number: _____

Instructions:

This exam consists of **112 points** spread out over **11 questions**. The exam must be completed in **170 minutes** unless you have accommodations supported by a DSP letter.

- Note that some questions have circular bubbles to select a choice. Please shade in the circle fully to mark your answer.
- **Write clearly and legibly.** We reserve the right to withhold points from answers that are very difficult to read.
- Blank answers and incorrect answers are graded identically, so it's in your best interest to answer every question.
- **You MUST write your Student ID number at the top of each page.**
- You should not use a calculator, scratch paper, or notes you own other than the reference sheets distributed at the beginning of the exam.

For all Python questions, you may assume `Pandas` has been imported as `pd`, `NumPy` as `np`, the Python RegEx library as `re`, `matplotlib.pyplot` as `plt`, and `seaborn` as `sns`.

Honor Code:

As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. I am the person whose name is on the exam, and I completed this exam in accordance with the Honor Code.

Signature: _____

This page has been intentionally left blank.

1 [9 Pts] Potpourri Island

(a) [3 Pts] Sarika uses 10-fold cross-validation (CV) to compute the CV error of a regularized model. Which of the following statements about this evaluation process is true? *Assume Sarika does not fit a final model after calculating the CV error.*

- ◯ True ◯ **False** Each data point is included in exactly 10 folds.
- ◯ True ◯ **False** Each data point is included in the validation data exactly 10 times.
- ◯ True ◯ **False** Each data point is in the training data of exactly 10 fitted models.

(b) [1 Pt] Sarika refits her original model by minimizing the following loss function:

$$L(\vec{\theta}) = \left(\|Y - X\vec{\theta}\|_2\right)^2 + \lambda g(\vec{\theta})$$

where $g(\vec{\theta})$ is a non-constant function of $\vec{\theta}$ and $\lambda \geq 0$. $\vec{\theta}$ has p elements. If the magnitude of any θ_j increases, $g(\vec{\theta})$ also increases, holding all other θ_j's constant.

Which $g(\vec{\theta})$ is best for **feature selection** (i.e., eliminating irrelevant features and keeping useful ones) and satisfies the conditions above? Select one:

◯ $\sum_{j=1}^{p} \log(\theta_j)$ ◯ $\sum_{j=1}^{p} \theta_j$ ◯ $\sum_{j=1}^{p} |\theta_j|$ ◯ $\sum_{j=1}^{p} \theta_j^2$

(c) [3 Pts] Sarika collects 3 observations. She fits an Ordinary Least Squares (OLS) model with 2 features and **no intercept**. All columns of the design matrix $\mathbb{X}$ used to fit the model are linearly independent.

Which of the following must be true?

- ◯ **True** ◯ False $\mathbb{X}^T\mathbb{X}$ is invertible.
- ◯ True ◯ **False** The sum of the residuals is 0.
- ◯ **True** ◯ False If Sarika adds one additional feature to her model, the new OLS solution will be unique. *Assume that the updated design matrix is full rank.*

(d) [2 Pts] Recall the lecture on the Cook County Assessor's Office (CCAO). Which of the following best describes the central problem with the property valuation model used by the CCAO to determine property taxes? Choose the single correct option.

- ◯ **The CCAO model undervalued expensive properties and overvalued less expensive properties, resulting in a regressive tax burden on lower-wealth homeowners.**
- ◯ The CCAO model overvalued expensive properties and undervalued less expensive properties, resulting in a regressive tax burden on lower-wealth homeowners.
- ◯ The CCAO model overvalued both expensive properties and less expensive properties, but the impact was greater on lower-wealth homeowners.
- ◯ The CCAO model undervalued both expensive properties and less expensive properties, but the impact was greater on lower-wealth homeowners.

2 [16 Pts] Readdit and Weep

Rohan recently vibecoded Readdit, a clone of Reddit where posts have to be long. Users can make posts on Readdit, and they can leave comments on other users' posts.

Rohan has two DataFrames containing information about posts and comments on Readdit:

- `posts`: Each row represents a single post on Readdit.
  - **post_id**: Unique identifier for the post. (type = String)
  - **views**: Number of views the post has received. (type = int)
- `comments`: Each row represents a single comment made on a Readdit post.
  - **comment_id**: Unique identifier for the comment. (type = int)
  - **post_id**: Identifier of the original post associated with the comment. (type = String)
  - **word_count**: Number of words **in the comment** (not the post). (type = int)

The first 5 rows of each DataFrame are shown below.

`posts`

	post_id	views
0	a	0
1	b	10
2	c	5
3	d	20
4	e	15

`comments`

	comment_id	post_id	word_count
0	1	a	30
1	2	a	20
2	3	c	5
3	4	c	20
4	5	d	15

(a) [4 Pts] The first 2 rows of `posts` and `comments` are shown again for reference:

`posts`

	post_id	views
0	a	0
1	b	10

`comments`

	comment_id	post_id	word_count
0	1	a	30
1	2	a	20

Consider the following SQL query:

```
SELECT p.post_id AS post_id, SUM(c.word_count) AS word_count
FROM posts AS p
JOIN comments AS c ON p.post_id = c.post_id
GROUP BY p.post_id
```

Write a **Pandas** expression that produces the same output as the SQL query above.

Hint: You do not need to rename columns or use `reset_index()` to receive full credit.

Note: Your code does not need to use all lines below. You cannot use more than the allotted lines to write your code. If you do not write clearly, we may deduct points.

PANDAS CODE

Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	

Solution:

```
(posts.merge(comments, on="post_id")
      .groupby("post_id")["word_count"]
      .agg("sum"))
```

(b) [12 Pts] Rohan wants to report the number of views for each post with **at least 10 thoughtful comments**. Rohan defines a comment as thoughtful if it has **at least 20 words**.

In this question, you will write **both** a SQL query and a Pandas expression that return a DataFrame or table with two columns:

- `post_id`: Unique identifier for each post
- `views`: Number of views of the post

Your dataframe should only contain posts with **at least 10 thoughtful comments**.

The first 2 rows of `posts` and `comments` are shown again for reference:

`posts`

	post_id	views
0	a	0
1	b	10

`comments`

	comment_id	post_id	word_count
0	1	a	30
1	2	a	20

Write your **SQL** query in the lines below.

Note: Your query does not need to use all lines below. You cannot use more than the allotted lines to write your query. If you do not write clearly, we may deduct points.

SQL QUERY

Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	

Solution:

```
SELECT p.post_id, FIRST(p.views) AS views
FROM posts p
JOIN comments c
ON p.post_id = c.post_id
WHERE word_count >= 20
GROUP BY post_id
HAVING COUNT(*) >= 10
```

Repeated for reference: Rohan wants to report the number of views for each post with at least 10 thoughtful comments. Rohan defines a comment as thoughtful if it has at least 20 words. Your code should return a DataFrame with two columns: `post_id` and `views`.

The first 2 rows of `posts` and `comments` are shown again for reference:

`posts`

	post_id	views
0	a	0
1	b	10

`comments`

	comment_id	post_id	word_count
0	1	a	30
1	2	a	20

Write your **Pandas code** in the lines below:

Hint: You do not need to rename columns or use `reset_index()` to receive full credit. Our solution to this question uses four `pd.DataFrame` methods: `merge`, `groupby`, `filter`, and `agg`.

Note: Your code does not need to use all lines below. You cannot use more than the allotted lines to write your code. If you do not write clearly, we may deduct points. Feel free to use intermediate variables to organize your work, but it's not required.

PANDAS CODE

Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	

Solution:

```
thoughtful = comments[comments["word_count"] >= 20]

(posts.merge(thoughtful, on="post_id")
        .groupby("post_id")
        .filter(lambda sf:  len(sf) >= 10)
        .groupby("post_id")
        .agg({"views":"first"}))
```

3 [4 Pts] Lord of the Strings

Gisella creates a new site called BetterBoxd to store her movie reviews. Each of her reviews is stored as a string. Three example reviews are shown below:

```
review1 = "9.35 stars - Everything Everywhere All at Once (2022).
           a great theater experience!!!!"
review2 = "GOAT ((2026)) may be the GOAT. 8/10 stars"
review3 = "another rewatch (5th) of Star Wars: Episode III - Revenge of the
          Sith (2005), hasn't changed my rating (7 stars!)"
```

Gisella wants to use a regular expression to match the **star rating** from each review. For the three reviews above, the regular expression should match `9.35 stars`, `8/10 stars`, and `7 stars`, respectively.

Gisella decides the rating will always be a number followed by a space and the word "stars".

- The number may be expressed as an integer, a decimal with at least one digit after the decimal point (e.g., 5.6789), or a fraction with an integer between 1 and 9 in the numerator and the number 10 in the denominator.
- The title of a movie could contain the word "stars", but it will never contain a number followed by a space and the word "stars".

Write a regular expression that Gisella can use for this task.

You will not receive credit if your expression only works for the three reviews above.

- For example, the expression `"(9.35|8\/10|7)␣stars"` would receive **no credit**.
- The character `"␣"` represents a single space. If your regular expression contains any spaces, **you must use ␣ in your answer**.

Note: Please write your answer neatly. If there is any ambiguity in your handwriting, we will take off points. You do not need to wrap your answer with quotation marks.

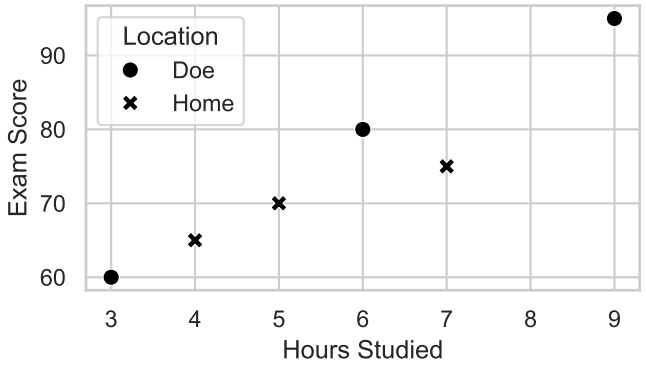
Solution: All reasonable interpretations receive full credit. For example, both of these receive full credit:

```
(1?0|[1-9](\/10|\.\d+)?)\sstars
```

```
([1-9]\/10|([1-9]+\d*|0)(\.\d+)?)\sstars
```

4 [10 Pts] FYP! (For You Plot)

(a) [4 Pts] The scatter plot below was generated using a **single call** to `sns.scatterplot()` on a pandas DataFrame called `df`. *Assume that the `df` DataFrame was passed directly to `sns.scatterplot()` with no modifications and that no additional functions are used inside of the call to `sns.scatterplot()`.*



Construct a DataFrame `df` that could have been used to generate this plot.

Fill in **all** cells in the table below, including the header row. You may not need all the rows and columns.

Put an X in any cells you do not need.

Column Name:				
Row 1				
Row 2				
Row 3				
Row 4				
Row 5				
Row 6				
Row 7				
Row 8				

Solution: The minimum required columns are `Hours Studied`, `Exam Score`, and `Location` (used as `hue`). Any additional columns are irrelevant to the plot and can be arbitrary. One valid answer:

Hours Studied	Exam Score	Location
3	60	Doe
4	65	Home
5	70	Home
6	80	Doe
7	75	Home
9	95	Doe

(b) [3 Pts] You are given a dataset containing information about **five people** and their social media engagement. Each row represents a single Instagram post, along with the number of likes that post received. A single person can have multiple rows due to multiple posts.

- **Name**: the name of the person. (type = String)
- **Post_ID**: unique identifier for the post. (type = String)
- **Date**: the date the post was made. (type = String)
- **Likes**: the number of likes received on that specific post. (type = int)

Here are the first few rows of the dataset:

	Name	Post_ID	Date	Likes
0	Dan	p001	2025-01-01	67
1	Dan	p002	2025-01-02	51
2	Rohan	p003	2025-01-02	2
3	Zara	p004	2025-01-03	64
4	Dan	p005	2025-01-04	49

Suppose you want to visualize how fast **each of the five person's cumulative number of likes across all of their posts** increases over time, separately for each person.

- For example, if Dan receives 67 likes on Day 1, 51 likes on Day 2, and 10 likes on Day 3, then Dan would have 67 cumulative likes on Day 1, $67 + 51 = 118$ cumulative likes on Day 2, and $67 + 51 + 10 = 128$ cumulative likes on Day 3.

Which of the following are appropriate for this visualization task?

◯ True ◯ **False** Single histogram (no overlay)
◯ True ◯ **False** Overlaid histograms
◯ True ◯ **False** Lineplot with one line
◯ **True** ◯ False Lineplot with multiple lines
◯ True ◯ **False** Side-by-side boxplots
◯ True ◯ **False** Overlaid KDE plots

(c) [3 Pts] *This question is unrelated to the previous part.* Suppose you want to visualize the relationship between a quantitative variable and a nominal qualitative variable with three categories. Which of the following plots are appropriate for this visualization task?

◯ True ◯ **False** Single histogram (no overlay)
◯ **True** ◯ False Overlaid histograms
◯ True ◯ **False** Lineplot with more than one line
◯ **True** ◯ False Side-by-side boxplots
◯ **True** ◯ False Overlaid KDE plots

5 [5 Pts] Red Bull Gives You Data

Sarah uses a survey to estimate the proportion of enrolled UC Berkeley undergraduates who drink caffeine at least once per week. For this question, assume that all contacted individuals complete the survey and that Sarah has access to the official UC Berkeley enrollment database.

(a) [1.5 Pts] In which of the following scenarios would Sarah's final estimate have **substantial selection bias**?

- ◯ True ◯ **False** Sarah selects a simple random sample of 1,000 enrolled undergraduates.
- ◯ True ◯ **False** Sarah selects a stratified random sample of 1,000 enrolled undergraduates, stratified by the total number of semesters each student has been enrolled at Berkeley.
- ◯ True ◯ **False** Sarah randomly selects and surveys 1,000 enrolled undergraduates who live within a mile of campus and 2,000 enrolled undergraduates who live more than a mile from campus. Sarah uses post-stratification to reweight her results, and Sarah knows the true proportion of all undergraduates who live within a mile of campus.

(b) [1.5 Pts] Sarah obtains a representative sample of 1,000 enrolled undergraduates. Sarah's current survey is a single yes or no question: "Did you drink anything containing caffeine in the last seven days?"

Which of the following would likely **increase reporting bias** in Sarah's survey?

Note: Reporting bias is often referred to as response bias.

- ◯ True ◯ **False** Sarah forgets to contact all first-year students on her list of 1,000 enrolled undergraduates.
- ◯ True ◯ **False** Sarah accidentally includes some enrolled graduate students in her sample.
- ◯ True ◯ **False** Sarah's survey includes a link to a trustworthy government website where users can search for the amount of caffeine in any drink.

(c) [2 Pts] For this part, Sarah considers the **population of all people who drink coffee and/or tea**. She wants to estimate the proportion of this population that has trouble sleeping after drinking anything with caffeine. So, she draws a large sample from this population.

- 25% of people in the sample who prefer coffee have trouble sleeping after drinking anything with caffeine.
- 15% of people in the sample who prefer tea have trouble sleeping after drinking anything with caffeine.
- From previous market research, Sarah knows that 60% of this population prefers coffee, and the remaining 40% prefer tea.
- 30% of the sample prefers coffee, and 70% of the sample prefers tea.

What is the **post-stratification estimate** of the proportion of this population that has trouble sleeping after drinking anything with caffeine?

You can assume that the assumptions of post-stratification are satisfied. You can leave your answer as an unsimplified algebraic expression (e.g., $(5 + 3)/2 + 1$).

Solution: `(0.4 * 0.15) + (0.6 * 0.25)`

6 [7 Pts] Let's Get this Bread

(a) [4 Pts] Milena fits a constant model to predict how much time it will take to bake a loaf of bread. She uses the following convex loss function:

$$L(\theta) = \sum_{i=1}^{n} \left(\theta - y_i^2 \log \theta\right)$$

Hint: $\frac{d}{dy} \log(y) = \frac{1}{y}$

Derive an expression for the optimal $\hat{\theta}$ in terms of n and y_i, for $i = 1 \ldots n$. Show your work.

Solution: Taking the derivative with respect to θ and setting it equal to zero:

$$\frac{dL}{d\theta} = \sum_{i=1}^{n} \left(1 - \frac{y_i^2}{\theta}\right) = n - \frac{\sum_{i=1}^{n} y_i^2}{\theta}$$

$$0 = n - \frac{\sum_{i=1}^{n} y_i^2}{\hat{\theta}}$$

Rearranging:

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} y_i^2$$

(b) [3 Pts] Milena collects the following dataset:

x (cups of flour)	y (hours)
π	3
e	1
0	8

Using your expression for $\hat{\theta}$ from part (a), compute the optimal $\hat{\theta}$ for this dataset. Your final answer should be an algebraic expression containing only numbers (e.g., 0, 1, e or π) and no variables (e.g., n or y_i). You do not need to simplify.

To be eligible for partial credit, make sure to clearly copy your final expression for $\hat{\theta}$ from part (a) at the top of the answer box below.

Solution:

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} y_i^2 = \frac{1}{3}(3^2 + 1^2 + 8^2) = \frac{74}{3}$$

7 [10 Pts] I'm at a Total Loss

(a) [4 Pts] Consider the following loss function:

$$L(\theta_0, \theta_1) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\hat{y} = \theta_0 + \theta_1 x^2$$

Solve for the partial derivative of $L(\theta_0, \theta_1)$ with respect to θ_1. **To be eligible for partial credit, be sure to show your work.**

Write neatly. We may deduct points from answers that are hard to read.

Solution:

$$\frac{\partial L}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \sum_{i=1}^{n} \left(y_i - (\theta_0 + \theta_1 x_i^2)\right)^2$$

$$= -2 \sum_{i=1}^{n} \left(y_i - \theta_0 - \theta_1 x_i^2\right) \cdot (x_i^2)$$

(b) [6 Pts] **This question does not depend on your answer to the previous part.** Consider the gradient of an unknown loss function:

$$\nabla L(\vec{\theta}) = \begin{bmatrix} \sum_{i=1}^{n} (x_i\theta_0 + y_i\theta_1) \\ \sum_{i=1}^{n} (x_i\theta_1 + y_i\theta_0) \end{bmatrix}$$

Suppose Kevin runs stochastic gradient descent (SGD, i.e., mini-batch gradient descent with $b = 1$) using this unknown loss function and its gradient.

- The starting guess of θ_0 is 1 and the starting guess of θ_1 is 2.
- The first data point of the shuffled dataset used for SGD is $(x_1 = 3, y_1 = 4)$.
- The second data point of the shuffled dataset used for SGD is $(x_2 = 5, y_2 = 6)$.
- The learning rate α is 0.5.

What is the updated value of θ_1 after one step of stochastic gradient descent (SGD)? **To be eligible for partial credit, be sure to show your work.** If you do not write clearly, you will not be eligible for partial credit.

Solution:

$$\frac{\partial L}{\partial \theta_1} = \sum_{i=1}^{n} (x_i\theta_1 + y_i\theta_0)$$

SGD uses one datapoint selected randomly without replacement to estimate the gradient. So, at $t = 1$, our estimate of $\frac{\partial L}{\partial \theta_1}$ is:

$$x_1\theta_1^{(0)} + y_1\theta_0^{(0)} = 3 \cdot 2 + 4 \cdot 1 = 10$$

Using the gradient descent update rule:

$$\theta_1^{(1)} = \theta_1^{(0)} - \alpha \frac{\partial L}{\partial \theta_1} = 2 - 0.5 \cdot 10 = -3$$

8 [21 Pts] Petri Fied of Overfitting

Sarika and Sara conduct experiments with bacteria to try to improve yield (i.e., the total number of bacteria at the end of the experiment). They run **750 different experiments**. The data from each experiment is stored in the rows of a DataFrame called `experiments`:

- **experiment_id**: Unique ID of each experiment
- **temp**: Incubator temperature (degrees Fahrenheit)
- **population**: Initial population size
- **type**: Type of bacteria (`Tau`, `CO2`, or `N-resistant`). `Tau` is the reference level.
- **energy**: Energy density level (`low` or `high`). `low` is the reference level.
- **yield**: Final population size

Here are the first 5 rows of the data:

experiment_id	temp	population	type	energy	yield
0	98.6	1000	Tau	low	2000
1	75.4	1500	CO2	high	3000
2	89.3	900	N-resistant	low	1800
3	95.2	2000	Tau	high	4000
4	77.8	1200	CO2	low	2400

(a) [6 Pts] Sara uses Ordinary Least Squares (OLS) to predict **yield** from all other variables in the DataFrame, **excluding** `experiment_id`. She uses one-hot encoding for categorical features (`energy` and `type`) with reference levels, and includes an intercept in her model.

Fill in the dimensions of each of the following by writing an integer in each box:

(i) $\mathbb{X}$ has ☐ rows and ☐ columns.

Solution: 750×6.
Counting columns: 2 numerical features + 2 OHE columns for **type** + 1 OHE columns for **energy** + 1 bias column $= 6$ columns.

(ii) $\hat{\mathbb{Y}}$ has $\square$ rows and $\square$ columns.

Solution: 750×1.

(iii) $\hat{\theta}$ has $\square$ rows and $\square$ columns.

Solution: 6×1.

(b) [6 Pts] After fitting the model, the intercept term is $\hat{\theta}_0 = 202.6$ and $\hat{\theta}_{\text{temp}} = 1.3$. **For each of the following parts, if you do not have enough information to answer the question, write "NA".** You can leave algebraic expressions unsimplified (e.g., $(5+3)/2+1$).

(i) If Sara feeds a new observation into the model where all numerical features are 0 and all categorical features are at their **reference levels**, what does the model predict for **yield**?

Solution: 202.6.

(ii) Sara considers Experiment A, which will use a temperature of 100 degrees. The model predicts a yield of 920 for Experiment A. Sara also considers Experiment B, which will have the same value of all features as Experiment A, except Experiment B will be run at 105 degrees. What does the model predict for the yield of Experiment B?

Solution: Holding all other features constant, a one degree Fahrenheit increase in `temperature` is associated with an increase of 1.3 in the predicted final population size (`yield`). So, the answer is $920 + 1.3 \cdot 5 = 926.5$.

(iii) Sara compares two observations where all variables other than temperature and yield are the same. In the first experiment, the temperature is 80 degrees and the actual yield is 813 bacteria. If the actual yield of the second experiment is 814.3, what was its temperature?

Solution: NA. We know nothing about the true values of features even if we know true outcomes.

(c) [9 Pts] Sara and Sarika consider three changes to the OLS model from the previous parts:

- **Scenario A:** Add a new feature called `sunray_exposure` to the model and refit. *Assume all features remain linearly independent.*
- **Scenario B:** Apply L1 (LASSO) regularization to the model with $\lambda > 0$. *Hint: Think about possible shapes of the diamond-shaped constraint of LASSO.*
- **Scenario C:** Remove all features from the model. Fit the model with just an intercept term (i.e., a constant model).

Note: Each of the changes above is considered separately. For example, Scenario B is applied to the original model, not to the model in Scenario A.

In the table below, record all possible effects of each scenario on four measures of the original OLS model: (model bias)^2, model variance, MSE on the training data, and MSE on held-out test data.

- For each box in the table, select Increase, Decrease, and/or Stay the same. **At least one option will always apply**.
- The effect(s) you select **do not have to be guaranteed**, they just **have to be possible**.
- For each row, assume there are no changes to the model except for the proposed change, that the data-generating process does not change, and that the model fitting process always converges.
- Two checkboxes have been removed as out-of-scope corner cases. This is not a typo!

Change to Model	**Model Bias^2**	**Model Variance**	**Training MSE**	**Test MSE**
Scenario A	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease	☐ Increase ☐ Decrease ☐ Stay the same
Scenario B	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same
Scenario C	☐ Increase ☐ Decrease	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same	☐ Increase ☐ Decrease ☐ Stay the same

Solution:

Change to Model	**Model Bias^2**	**Model Variance**	**Training MSE**	**Test MSE**
Scenario A	✗ Increase ✓ Decrease ✓ Stay the same	✓ Increase ✗ Decrease ✗ Stay the same	✗ Increase ✓ Decrease	✓ Increase ✓ Decrease ✓ Stay the same
Scenario B	✓ Increase ✗ Decrease ✓ Stay the same	✗ Increase ✓ Decrease ✓ Stay the same	✓ Increase ✗ Decrease ✓ Stay the same	✓ Increase ✓ Decrease ✓ Stay the same
Scenario C	✓ Increase ✗ Decrease	✗ Increase ✓ Decrease ✗ Stay the same	✓ Increase ✗ Decrease	✓ Increase ✓ Decrease ✓ Stay the same

LASSO with a low enough penalty may result in the original model solution. Recall the geometry of LASSO — if the diamond is large enough, it will contain the original OLS solution.

9 [6 Pts] TAs and the TSA

The check-in counters at San Francisco International Airport (SFO) decide to hire some Data 100 TAs to help streamline their process for weighing luggage.

Cristina and Willy model the weights of small bags and large bags as random draws from two different distributions. The random variable X_i represents the weight of a single small bag i, and the random variable Y_j represents the weight of a single large bag j. X_i and Y_j are independent from one another.

- The weight of each **small bag** X_i is i.i.d. with mean $8\,\text{kg}$ and variance $0.2\,\text{kg}^2$.
- The weight of each **large bag** Y_j is i.i.d. with an unknown mean and an unknown variance.

Cristina and Willy suggest that the check-in counters always measure the weight of **16 bags** at a time. Every group of 16 bags is called a **"baglet"**.

- In every baglet, there are always **10 small bags** and **6 large bags**.
- Cristina and Willy use the random variable Z to represent the **total weight** of a single baglet. So, $Z = \sum_{i=1}^{10} X_i + \sum_{j=1}^{6} Y_j$.
- The mean of Z is $200\,\text{kg}$ and the variance of Z is $27\,\text{kg}^2$

(a) [3 Pts] Compute $\mathbb{E}[Y_j]$. To be eligible for partial credit, show your work and write neatly! You can leave algebraic expressions unsimplified (e.g., $(5+3)/2+1$).

Solution: Since Z is the total weight of all bags:

$$\mathbb{E}[Z] = 10\,\mathbb{E}[X] + 6\,\mathbb{E}[Y]$$

$$200 = 10(8) + 6\,\mathbb{E}[Y] = 80 + 6\,\mathbb{E}[Y]$$

$$6\,\mathbb{E}[Y] = 120 \implies \mathbb{E}[Y] = 20\ \text{kg}$$

(b) [3 Pts] Compute $\text{Var}(Y_j)$. To be eligible for partial credit, show your work and write neatly! You can leave algebraic expressions unsimplified (e.g., $(5+3)/2+1$).

Here is some relevant information repeated for reference:

- The weight of each **small bag** X_i is i.i.d. with mean $8\,\text{kg}$ and variance $0.2\,\text{kg}^2$.
- The weight of each **large bag** Y_j is i.i.d. with an unknown mean and an unknown variance.
- $Z = \sum_{i=1}^{10} X_i + \sum_{j=1}^{6} Y_j$. The mean of Z is $200\,\text{kg}$ and the variance of Z is $27\,\text{kg}^2$

Solution: Since the bag weights are i.i.d., the variance of the total weight is:

$$\text{Var}(Z) = 10\,\text{Var}(X) + 6\,\text{Var}(Y)$$

$$27 = 10(0.2) + 6\,\text{Var}(Y) = 2 + 6\,\text{Var}(Y)$$

$$6\,\text{Var}(Y) = 25 \implies \text{Var}(Y) = 25/6$$

10 [12 Pts] Labubu Regression

James and Dan collect Pop Mart's Labubus. They want to build a classifier that predicts whether someone wants to buy a Labubu (**Class 1**) or not (**Class 0**).

(a) [2 Pts] To help in James and Dan's efforts, Henry builds a preliminary logistic regression model with an intercept and three features: `age` (in years), `money` (in dollars \$), and `is_female`. These three features appear in the design matrix $\mathbb{X}$ in the same order they are written.

After fitting the model, he gets the following parameter vector:

$$\vec{\theta} = \begin{bmatrix} 1 & -2 & 1 & 3 \end{bmatrix}^\top$$

Suppose that a particular person is 21 years old, has \$40, and identifies as female. Compute the predicted probability $\hat{p}$ that this person wants to buy a Labubu. Your final answer can be an unsimplified algebraic expression (e.g., $(5+3)/2+1$).

Solution:

$$\hat{p} = \frac{1}{1+e^{-(\theta_0+\theta_1(21)+\theta_2(40)+\theta_3(1))}} = \frac{1}{1+e^{-(1-2(21)+1(40)+3(1))}} = \frac{1}{1+e^{-2}} = \sigma(2)$$

(b) [5 Pts] After analyzing Henry's model, Skyla builds a separate model that estimates the probabilities below, paired with their true and predicted outcomes for five data points:

$\hat{P}(Y = 1 \mid \mathbf{x})$	Y	$\hat{Y}$
0.05	1	0
0.35	0	0
0.45	0	1
0.50	1	1
0.90	1	1

(i) Skyla forgot what threshold she used to generate her predictions in the table above. What is the range of possible thresholds that are consistent with the predictions above?

The threshold must be strictly greater than [ ] and less than or equal to [ ].

Solution: $(0.35, 0.45]$, as any threshold in this range produces exactly the $\hat{Y}$ values shown above.

(ii) What would have been the precision of the classifier on these five data points if Skyla had instead used a threshold of 0.55?

Solution:

$$\frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{1}{1 + 0} = 1$$

(iii) *This question is unrelated to the previous parts.* If the threshold of a binary classification model is **decreased**, which of the following are possible?

◯ **True** ◯ False Precision could increase

◯ **True** ◯ False Recall could increase
◯ **True** ◯ False The number of false positives could increase
◯ **True** ◯ False Accuracy could increase
◯ True ◯ **False** Area under the ROC curve could increase

Solution: Lowering the threshold causes more points to be predicted positive. This can only increase (or maintain) TP and FP counts, so recall ($\frac{TP}{TP+FN}$) and the number of false positives both increase or stay the same. Precision can decrease (more FPs in denominator). Accuracy can go either way.

(c) [5 Pts] James evaluates Skyla's model at four candidate thresholds.

(i) For each threshold in the table on the right, fill in the count of FPs and count of FNs. The table on the left refers to the same five points from the previous part.

$\hat{P}(Y = 1 \mid \mathbf{x})$	Y
0.05	1
0.35	0
0.45	0
0.50	1
0.90	1

Threshold	**FP**	**FN**
0.10		
0.40		
0.60		
0.95		

Solution:

Threshold T	**FP**	**FN**
0.10	2	1
0.40	1	1
0.60	0	2
0.95	0	3

(ii) James shares the model's predictions with a Labubu store. The store pays a penalty of \$3 for every false positive and \$1 for every false negative. Of the four thresholds from part (i), which one minimizes the store's total penalty?

$$\text{Cost}(T) = 3 \cdot \text{FP} + 1 \cdot \text{FN}$$

Solution: Computing cost at each threshold:

T	FP	FN	Cost
0.10	2	1	$3(2) + 1(1) = 7$
0.40	1	1	$3(1) + 1(1) = 4$
0.60	0	2	$3(0) + 1(2) = 2$
0.95	0	3	$3(0) + 1(3) = 3$

The threshold $T = 0.60$ minimizes the cost with a total cost of $\boxed{2}$.

(iii) The store's cost function above penalizes false positives more heavily than false negatives. Which one of the following business scenarios best explains why the store would use this cost function? *Recall that Class 1 denotes a person who wants to buy a Labubu, and Class 0 denotes a person who does not want to buy a Labubu.*

- ◯ The store would prefer to order too many Labubus than order too few, since unsold Labubus are cheap to store for the future.
- ◯ **The store would rather order too few Labubus than order too many, since ordering too many Labubus is expensive.**
- ◯ The store considers the cost of ordering too many Labubus and ordering too few Labubus to be equal.
- ◯ The store does not care about the number of false positives or false negatives.

Solution: A higher penalty on FP means the model is penalized more for predicting someone will buy a Labubu when they won't. This corresponds to a store that would rather risk running out (FN) than over-order and be stuck with excess inventory (FP).

11 [12 Pts] The Summer I Saw PCA and Clusters

(a) [4 Pts] Consider the following dataset with three features x_1, x_2, x_3 and three observations with IDs a, b, c:

ID	x_1	x_2	x_3
a	10	20	30
b	30	20	10
c	20	10	30

Here are the approximate principal component vectors for the dataset above:

$$\text{PC}_1 = \begin{bmatrix} -0.6 \\ -0.1 \\ 0.8 \end{bmatrix} \quad \text{PC}_2 = \begin{bmatrix} -0.5 \\ 0.8 \\ -0.3 \end{bmatrix} \quad \text{PC}_3 = \begin{bmatrix} 0.6 \\ 0.6 \\ 0.6 \end{bmatrix}$$

What is the approximate value of the first latent feature for the observation with ID=b? To be eligible for partial credit, show your work. You can leave your answer as an unsimplified algebraic expression (e.g., $(5+3)/2+1$).

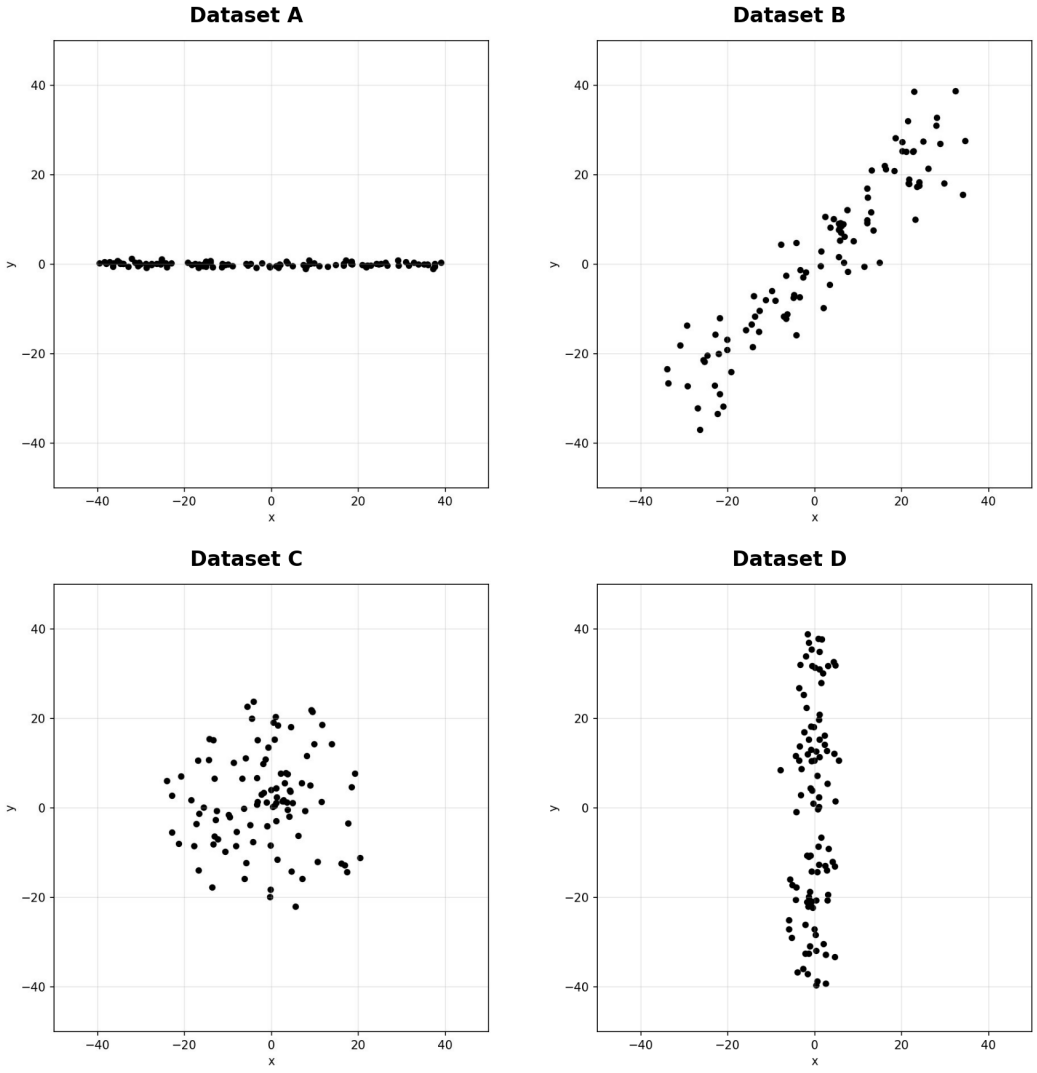
Solution:

$$(-0.6)(30) + (-0.1)(20) + (0.8)(10)$$

(b) [2 Pts] Consider this ratio:

$$\frac{\text{variance of the 1st latent feature}}{\text{variance of the 2nd latent feature}}$$

For which of the following datasets is this ratio the **smallest**?



◯ Dataset A

◯ Dataset B

◯ **Dataset C**

◯ Dataset D

◯ Not enough information to determine

(c) [2 Pts] For each statement below, select True, False, or Not Enough Information to Determine.

(i) Consider a dataset with two unique data points. K-means clustering with $K = 1$ will always result in the same cluster assignments for each point regardless of the location of the initial cluster centers.

- ◯ **True**
- ◯ False
- ◯ Not enough information to determine

(ii) Consider a dataset with three unique data points. K-means clustering with $K = 2$ will always result in the same cluster assignments for each point regardless of the location of the initial cluster centers.

- ◯ True
- ◯ **False**
- ◯ Not enough information to determine

(iii) Suppose K-means clustering is applied to a fixed dataset. If the number of clusters is increased, the sum of squared distances from each centroid to its assigned data points will always stay the same or increase.

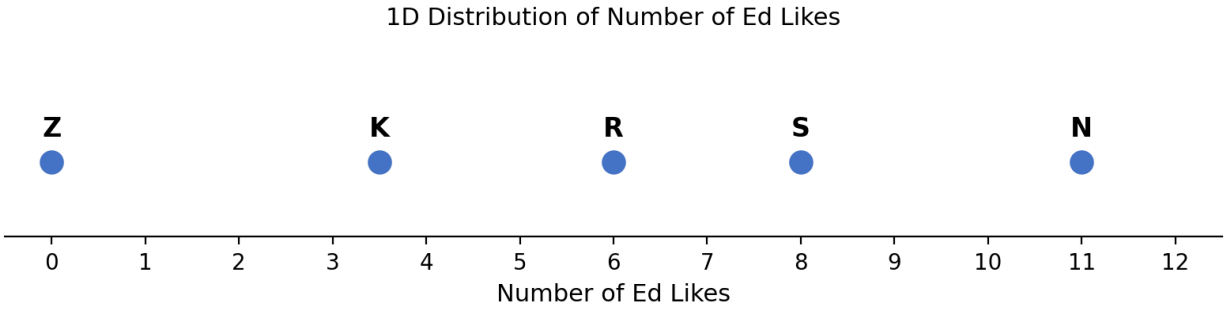
- ◯ True
- ◯ **False**
- ◯ Not enough information to determine

(iv) Suppose K-means clustering is applied to a fixed dataset. If the number of clusters is increased, the average of silhouette scores across all datapoints will always stay the same or increase.

- ◯ True
- ◯ **False**
- ◯ Not enough information to determine

(d) [4 Pts] Eli clusters course staff members by the average number of Ed Likes they received over the past few months.

	name	number of Ed likes
0	Zara	0
1	Keryssa	3.5
2	Rohan	6
3	Sarah	8
4	Neil	11



Write the steps of agglomerative clustering with **complete linkage** for this dataset, starting from when all points are their own individual clusters, and ending when all the points make one giant cluster. To help you, we have provided a template for you to fill in. We have filled in the first step and part of the last step for you. In the case of a tie, answers resulting from any tie-breaking scheme will be accepted.

1. {R} merges with {S} to form cluster {R,S}

2. {Z} merges with {K} to form cluster {Z,K}

3. {R,S} merges with {N} to form cluster {R,S,N}

4. {Z,K} merges with {R,S,N} to form cluster {Z,K,R,S,N}

Please state any relevant assumptions in the box below (Optional).

You are done with the final- Congratulations!

Draw your favorite DATA 100/200 memory so far!

Spring 2026 Data C100/C200 Final Reference Sheet

Pandas

Suppose `df` is a DataFrame; `s` is a Series. `import pandas as pd`

Function	Description
`df.shape`	Returns a tuple containing the number of rows and columns, in that order
`df.index`	Returns the index (row labels) of `df` as an Index object
`df[col]`	Returns the column labeled `col` from `df` as a Series
`df.index[i]`	Returns the row label at position `i` from `df`'s index
`df[[col1, col2]]`	Returns a DataFrame containing the columns labeled `col1` and `col2`
`s.idxmax()`	Returns the index label of the first occurrence of the maximum value in Series `s`
`s.astype(dtype)`	Returns a Series casted to the specified type `dtype`
`s.loc[rows] / df.loc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their index values
`s.iloc[rows] / df.iloc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their positions
`s.isnull() / df.isnull()`	Returns boolean Series/DataFrame identifying missing values
`s.fillna(value) / df.fillna(value)`	Returns a Series/DataFrame where missing values are replaced by `value`
`s.isin(values) / df.isin(values)`	Returns a Series/DataFrame of booleans indicating if each element is in `values`.
`df.drop(labels, axis)`	Returns a DataFrame without the rows or columns named `labels` along `axis` (either 0 or 1)
`df.rename(index=None, columns=None)`	Returns a DataFrame with renamed columns from a dictionary `index` and/or `columns`
`df.sort_values(by, ascending=True)`	Returns a DataFrame where rows are sorted by the values in columns `by`
`s.sort_values(ascending=True)`	Returns a sorted Series
`s.unique()`	Returns a NumPy array of the unique values of `s` in the order that they appear
`s.value_counts()`	Returns the number of times each unique value appears in a Series
`pd.merge(left, right, how='inner', left_on=col1, right_on=col2)`	Returns a DataFrame joining left and right on columns labeled col1 and col2; the join is of type inner
`left.merge(right, left_on=col1, right_on=col2)`	Returns a DataFrame joining `left` and `right` on columns labeled `col1` and `col2`
`df.pivot_table(values=None, index=None, columns=None, aggfunc='mean', fill_value=None)`	Returns a DataFrame pivot table where columns are unique values from `columns` (column name or list), and rows are unique values from `index` (column name or list); cells are collected `values` using `aggfunc`. If `values` is not provided, cells are collected for each remaining column with multi-level column indexing.
`df.set_index(col)`	Returns a DataFrame that uses the values in the column labeled `col` as the row index
`df.reset_index()`	Returns a DataFrame that has row index 0, 1, etc., and adds the current index as a column

Let `grouped = df.groupby(by)` where `by` can be a column label or a list of labels

Function	Description
`grouped.count()`	Return a DataFrame containing the size of each group, excluding missing values
`grouped.size()`	Return a Series containing size of each group, including missing values
`grouped.mean()`/`.min()`/`.max()`	Return a Series/DataFrame containing mean/min/max of each group for each column, excluding missing values
`grouped.head(n)`/`.tail(n)`	Return a Series/DataFrame containing first/last `n` entries of each group for each column, excluding missing values
`grouped.filter(f)` `grouped.agg(f)`	Filters or aggregates using the given function `f`

Function	Description
`s.str.len()`	Returns a Series containing length of each string
`s.str[a:b]`	Returns a Series where each element is a slice of the corresponding string indexed from `a` (inclusive, optional) to `b` (non-inclusive, optional)
`s.str.lower()`/`s.str.upper()`	Returns a Series of lowercase/uppercase versions of each string

Function	Description
`s.str.replace(pat, repl, regex=False)`	Returns a Series that replaces occurences of substrings matching `pat` with string `repl`. When `regex=False`, `pat` is treated as a literal string; when `regex=True`, `pat` is treated as a RegEx pattern.
`s.str.contains(pat)`	Returns a boolean Series indicating if a substring matching the regex `pat` is contained in each string
`s.str.extract(pat)`	Returns a DataFrame of the first subsequence of each string that matches the regex `pat`. If `pat` contains one group, then only the substring matching the group is extracted
`s.str.split(pat=" ")`	Splits the strings in `s` at the delimiter `pat` (defaults to a whitespace). Returns a Series of lists, where each list contains strings of the characters before and after the split.

Visualization

Matplotlib: `x` and `y` are sequences of values. `import matplotlib.pyplot as plt`

Function	Description
`plt.plot(x, y)`	Creates a line plot of `x` against `y`
`plt.scatter(x, y)`	Creates a scatter plot of `x` against `y`
`plt.hist(x, bins=None)`	Creates a histogram of `x`; `bins` can be an integer or a sequence
`plt.bar(x, height)`	Creates a bar plot of categories `x` and corresponding heights `height`

Seaborn: `x` and `y` are column names in a DataFrame `data`. `import seaborn as sns`

Function	Description
`sns.countplot(data=None, x=None)`	Create a barplot of value counts of variable `x` from `data`
`sns.histplot(data=None, x=None, stat='count', kde=False)` `sns.displot(data=None, x=None, kind='hist', rug=False)`	Creates a histogram of `x` from `data`, where bin statistics `stat` is one of `'count'`, `'frequency'`, `'probability'`, `'percent'`, and `'density'`; optionally overlay a kernel density estimator. `displot` is similar but can optionally overlay a rug plot and/or a KDE plot
`sns.rugplot(data=None, x=None)`	Adds a rug plot on the x-axis of variable `x` from `data`
`sns.boxplot(data=None, x=None, y=None)` `sns.violinplot(data=None, x=None, y=None)`	Create a boxplot of a numeric feature (e.g., `y`), optionally factoring by a category (e.g., `x`), from `data`. `violinplot` is similar but also draws a kernel density estimator of the numeric feature
`sns.scatterplot(data=None, x=None, y=None)`	Create a scatterplot of `x` versus `y` from `data`
`sns.lmplot(data=None, x=None, y=None, fit_reg=True)`	Create a scatterplot of `x` versus `y` from `data`, and by default overlay a least-squares regression line
`sns.jointplot(data=None, x=None, y=None, kind='scatter')`	Combine a bivariate scatterplot of `x` versus `y` from `data`, with univariate density plots of each variable overlaid on the axes; `kind` determines the visualization type for the distribution plot, can be `scatter`, `kde` or `hist`

Regular Expressions

Operator	Description	Operator	Description
`.`	Matches any character except `\n`	`*`	Matches preceding character/group zero or more times
`\`	Escapes metacharacters	`?`	Matches preceding character/group zero or one times
`|`	Matches expression on either side of expression; has lowest priority of any operator	`+`	Matches preceding character/group one or more times
`\d`, `\w`, `\s`	Predefined character group of digits (0-9), alphanumerics (a-z, A-Z, 0-9, and underscore), or whitespace, respectively	`^`, `$`	Matches the beginning and end of the line, respectively
`\D`, `\W`, `\S`	Inverse sets of `\d`, `\w`, `\s`, respectively	`( )`	Capturing group used to create a sub-expression
`{m}`	Matches preceding character/group exactly `m` times	`[ ]`	Character class used to match any of the specified characters or range (e.g. `[abcde]` is equivalent to `[a-e]`)
`{m, n}`	Matches preceding character/group at least `m` times and at most `n` times. If either `m` or `n` are omitted, set lower/upper bounds to 0 and ∞, respectively	`[^ ]`	Invert character class; e.g. `[^a-c]` matches all characters except `a`, `b`, `c`

Modified lecture example for capture groups:

```
import re
 lines = '169.237.46.168 - - [26/Jan/2014:10:47:58 -0800] "GET ... HTTP/1.1"'
 re.findall(r'\[\d+\/(\w+)\/\d+:\d+:\d+:\d+ .+\]', lines) # returns ['Jan']
```

Function	Description
`re.match(pattern, string)`	Returns a match if zero or more characters at beginning of `string` matches `pattern`, else None
`re.search(pattern, string)`	Returns a match if zero or more characters anywhere in `string` matches `pattern`, else None
`re.findall(pattern, string)`	Returns a list of all non-overlapping matches of `pattern` in `string` (if none, returns empty list)
`re.sub(pattern, repl, string)`	Returns `string` after replacing all occurrences of `pattern` with `repl`

Modeling

Concept	Formula	Concept	Formula
Variance, σ_x^2	$\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$	Correlation r	$r=\frac{1}{n}\sum_{i=1}^{n}\frac{x_i-\bar{x}}{\sigma_x}\frac{y_i-\bar{y}}{\sigma_y}$
L_1 loss	$L_1(y,\hat{y})=\mid y-\hat{y}\mid$	Linear regression estimate of y	$\hat{y}=\theta_0+\theta_1 x$
L_2 loss	$L_2(y,\hat{y})=(y-\hat{y})^2$	Least squares linear regression	$\hat{\theta}_0=\bar{y}-\hat{\theta}_1\bar{x}$ $\quad$ $\hat{\theta}_1=r\frac{\sigma_y}{\sigma_x}$
Empirical risk with loss L	$R(\theta)=\frac{1}{n}\sum_{i=1}^{n}L(y_i,\hat{y}_i)$		

Multiple Linear Regression Formulas

Concept	Formula	Concept	Formula
Mean squared error	$R(\theta)=\frac{1}{n}\|\|\mathbb{Y}-\mathbb{X}\theta\|\|_2^2$	Normal equation	$\mathbb{X}^T\mathbb{X}\hat{\theta}=\mathbb{X}^T\mathbb{Y}$
Least squares estimate, if $\mathbb{X}$ is full rank	$\hat{\theta}=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$	Multiple R^2 (coefficient of determination)	$R^2=\frac{\text{variance of fitted values}}{\text{variance of } y}$
Ridge Regression L2 Regularization	$\frac{1}{n}\|\|\mathbb{Y}-\mathbb{X}\theta\|\|_2^2+\lambda\|\|\theta\|\|_2^2$	Squared L2 Norm of $\theta\in\mathbb{R}^d$	$\|\|\theta\|\|_2^2=\sum_{j=1}^{d}\theta_j^2$
Ridge regression estimate (closed form)	$\hat{\theta}_{\text{ridge}}=(\mathbb{X}^T\mathbb{X}+n\lambda I)^{-1}\mathbb{X}^T\mathbb{Y}$	L1 Norm of $\theta\in\mathbb{R}^d$	$\|\|\theta\|\|_1=\sum_{j=1}^{d}\|\theta_j\|$
LASSO Regression L1 Regularization	$\frac{1}{n}\|\|\mathbb{Y}-\mathbb{X}\theta\|\|_2^2+\lambda\|\|\theta\|\|_1$		

Scikit-Learn

Package: `sklearn.linear_model`

Linear Regression	Logistic Regression	Function(s)	Description
✓	-	`LinearRegression(fit_intercept=True)`	Returns an ordinary least squares Linear Regression model.
-	✓	`LogisticRegression( fit_intercept=True, penalty='l2', C=1.0)`	Returns a Logistic Regression model. Hyperparameter C is inverse of regularization parameter, C = 1/λ.
✓	-	`LassoCV()`, `RidgeCV()`	Returns a Lasso (L1 Regularization) or Ridge (L2 regularization) linear model, respectively, and picks the best model by cross validation.

Package: `sklearn.linear_model`			
Linear Regression	**Logistic Regression**	**Function(s)**	**Description**
✓	✓	`model.fit(X, y)`	Fits the scikit-learn `model` to the provided `X` and `y`.
✓	✓	`model.predict(X)`	Returns predictions for the `X` passed in according to the fitted `model`.
✓	✓	`model.predict_proba(X)`	Returns predicted probabilities for `X` according to the fitted `model`. If binary classes, will return probabilities for both class 0 and 1.
✓	✓	`model.coef_`	Estimated coefficients for the linear model, excluding the intercept.
✓	✓	`model.intercept_`	Bias/intercept term of the linear model. Set to 0.0 if `fit_intercept=False`.

Package: `sklearn.model_selection`		
	Function	**Description**
	`train_test_split(*arrays, test_size=0.2)`	Returns two random subsets of each array passed in, with 0.8 of the array in the first subset and 0.2 in the second subset.

Probability

Let X have a discrete probability distribution $P(X = x)$. X has expectation $\mathbb{E}[X] = \sum_x xP(X = x)$ over all possible values x, variance $\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$, and standard deviation $\text{SD}(X) = \sqrt{\text{Var}(X)}$.

Notes	**Property of Expectation**	**Property of Variance**
X is a random variable.	$\mathbb{E}[X] = \sum_x xP(X = x)$	$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = E[X^2] - (E[X])^2$
X is a random variable, $a, b \in \mathbb{R}$ are scalars.	$\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$	$\text{Var}(aX + b) = a^2\text{Var}(X)$
X, Y are random variables.	$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$	$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$
X is a Bernoulli random variable that takes the value 1 with probability p, and 0 otherwise.	$\mathbb{E}[X] = p$	$\text{Var}(X) = p(1 - p)$

Parameter Estimation and Gradient Descent Update Rule

Parameter Estimation

Suppose for each individual with fixed input x, we observe a random response $Y = g(x) + \epsilon$, where g is the true relationship and ϵ is random noise with zero mean and variance σ^2.

For a new individual with fixed input x, define our random prediction $\hat{Y}(x)$ based on a model fit to our observed sample $(\mathbb{X}, \mathbb{Y})$. The model risk is the mean squared prediction error between Y and $\hat{Y}(x)$: $\mathbb{E}[(Y - \hat{Y}(x))^2] = \sigma^2 + \Big(\mathbb{E}[\hat{Y}(x)] - g(x)\Big)^2 + \text{Var}(\hat{Y}(x))$.

Suppose that input x has p features and the true relationship g is linear with parameter $\theta \in \mathbb{R}^{p+1}$. Then $Y = f(x) = \theta_0 + \sum_{j=1}^{p} \theta_j x_j + \epsilon$ and $\hat{Y} = \hat{f}(x)$ for an estimate $\hat{\theta}$ fit to the observed sample $(\mathbb{X}, \mathbb{Y})$.

Gradient Descent

For a learning rate α, the gradient update step is:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta L(\theta^{(t)})$$

where $\nabla_\theta L(\theta^{(t)})$ is the partial derivative/gradient of L with respect to θ, evaluated at $\theta^{(t)}$.

SQL

SQL syntax:

```
SELECT [DISTINCT]
      {* | expr [[AS] c_alias]
      {, expr [[AS] c_alias] ...}}
FROM tableref {, tableref}
[[INNER | LEFT | RIGHT | FULL | CROSS] JOIN table_name
      ON qualification_list]
[WHERE search_condition]
[GROUP BY colname {, colname...}]
[HAVING search_condition]
[ORDER BY column_list]
[LIMIT number]
[OFFSET number of rows];
```

Strings in SQL should use single quotes. Column names that contain a space should use double quotes.

Syntax	Description
`SELECT column_expression_list`	List is comma-separated. Column expressions may include aggregation functions (`MAX`, `FIRST`, `COUNT`, `AVG`, etc). `AS` renames columns. `DISTINCT` selects only unique rows.
`FROM s INNER JOIN t ON cond`	Inner join tables `s` and `t` using `cond` to filter rows; the `INNER` keyword is optional.
`FROM s LEFT JOIN t ON cond`	Left outer join of tables `s` and `t` using `cond` to filter rows.
`FROM s RIGHT JOIN t ON cond`	Right outer join of tables `s` and `t` using `cond` to filter rows.
`FROM s FULL JOIN t ON cond`	Full outer join of tables `s` and `t`.
`FROM s CROSS JOIN t`	Explicit cross join of tables `s` and `t`; Equivalent to `FROM s, t`.
`FROM s, t`	Cross join of tables `s` and `t`: all pairs of a row from `s` and a row from `t`.
`WHERE a IN cons_list`	Select rows for which the value in column `a` is among the values in a `cons_list`.
`ORDER BY RANDOM() LIMIT n`	Draw a simple random sample of `n` rows.
`ORDER BY a, b DESC`	Order by column `a` (ascending by default), then `b` (descending).
`CASE WHEN pred THEN cons ELSE alt END`	Evaluates to `cons` if `pred` is true and `alt` otherwise. Multiple `WHEN`/`THEN` pairs can be included, and `ELSE` is optional.
`WHERE s.a LIKE 'p'`	Matches each entry in the column `a` of table `s` to the text pattern `p`. The wildcard `%` matches at least zero characters.
`LIMIT number`	Keep only the first `number` rows in the return result.
`OFFSET number`	Skip the first `number` rows in the return result.

Aggregation Functions

Aggregation functions operate on a set of rows and return a single value. They are typically used alongside `GROUP BY`.

Function	Description
`SUM(expr)`	Returns the sum of all non-null values of `expr` across the group.
`AVG(expr)`	Returns the arithmetic mean of all non-null values of `expr` across the group.
`FIRST(expr)`	Returns the value of `expr` from the first row in the group, as determined by the current sort order.
`COUNT(expr)`	Returns the number of non-null values of `expr` across the group. Use `COUNT(*)` to count all rows including nulls.
`MAX(expr)`	Returns the maximum value of `expr` across the group.

Logistic Regression and Classification

Logistic Regression Model: For input feature vector x, $\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$, where $\sigma(z) = 1/(1 + e^{-z})$. For a single datapoint, define cross-entropy loss as $-[y \log(p) + (1 - y) \log(1 - p)]$, where p is the probability that the response is 1.

An ROC curve has the false positive rate (FPR) on the x-axis and true positive rate (TPR) on the y-axis.

Classification Performance

Suppose you predict n datapoints.

Metric	Formula
Accuracy	$\frac{TP+TN}{n}$
Precision	$\frac{TP}{TP+FP}$
Recall, True Positive Rate (TPR)	$\frac{TP}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
F1 Score	$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Clustering

A datapoint's **silhouette score** S is defined as $S = (B - A)/\max(A, B)$, where A is the mean distance to other points in its cluster, and B is the mean distance to points in its closest cluster.