Data 100, Final

Fall 2019

Name: ____

Email: ________________________________@berkeley.edu

Student ID: ____

Exam Room: ____

First and last name of student to your left: ____________________________

First and last name of student to your right: ____________________________

All work on this exam is my own ***(please sign)***: ____________________________

Instructions:

- This final exam consists of **117 points** and must be completed in the **170 minute** time period ending at **6:00PM**, unless you have accommodations supported by a DSP letter.
- Please write your initials on the top of every page.
- Note that some questions have circular bubbles to select a choice. This means that you should only **select one choice**. Other questions have boxes. This means you should **select all that apply**. When selecting your choices, you must **fully shade** in the box/circle. Check marks will likely be mis-graded.
- You may use three cheat sheets each with two sides.
- **Please show your work for computation questions as we may award partial credit.**

1 An Instructor Thinks This Is A Good Question [9 Pts.]

The average response time for a question on Piazza this semester was 11 minutes. As always, the number of questions answered by each TA is highly variable, with a few TAs going above and beyond the call of duty. Below are the number of contributions for the top four TAs (out of $20,000$ total Piazza contributions):

TA	# contributions
Daniel	2000
Suraj	1800
Mansi	700
Allen	500

Suppose we take an SRS (simple random sample) of size $n = 500$ contributions from the original $20,000$ contributions. We will also define some random variables:

- $D_i = 1$ when the i^{th} contribution in our sample is made by Daniel; else $D_i = 0$.
- $S_i = 1$ when the i^{th} contribution in our sample is made by Suraj; else $S_i = 0$.
- $M_i = 1$ when the i^{th} contribution in our sample is made by Mansi; else $M_i = 0$.
- $A_i = 1$ when the i^{th} contribution in our sample is made by Allen; else $A_i = 0$.
- $O_i = 1$ when the i^{th} contribution is made by anyone other than Daniel, Suraj, Mansi, or Allen; else, $O_i = 0$

Throughout this problem, **you may leave your answer as an unsimplified fraction**. If your answer is much more complicated than necessary, we may deduct points. Some of these problems are simple, and some are quite tricky. If you're stuck, move on and come back later.

(a) i. [1 Pt] What is $P(A_1 = 1)$?

$P(A_1 = 1)$ = □

Solution: $\frac{500}{20000} = \frac{1}{40}$

ii. [1 Pt] What is $\mathbb{E}[S_1]$?

$\mathbb{E}[S_1]$ = □

Solution: $\frac{1800}{20000} = \frac{9}{100}$

iii. [1 Pt] What is $\mathbb{E}[M_{100}]$?

$\mathbb{E}[M_{100}]$ = □

Solution: $\dfrac{700}{20000} = \dfrac{7}{200}$

iv. [1 Pt] What is Var$[D_{50}]$?

Var$[D_{50}]$ = □

Solution: $D_{50} \sim$ Bernoulli$(\frac{2000}{20000} = \frac{1}{10})$. $Var(D_{50}) = \frac{1}{10} \cdot (1 - \frac{1}{10}) = \frac{9}{100}$

v. [1 Pt] What is $D_{400} + S_{400} + A_{400} + M_{400} + O_{400}$?

$D_{400} + S_{400} + A_{400} + M_{400} + O_{400}$ = □

Solution: 1. The 400th contribution must be made by Daniel, Suraj, Mansi, or other so one of the 5 random variables must 1 and the rest are 0s.

(b) For parts b.i and b.ii, let:

- $N_D = \sum_{i=1}^{500} D_i$
- $N_S = \sum_{i=1}^{500} S_i$
- $N_M = \sum_{i=1}^{500} M_i$
- $N_A = \sum_{i=1}^{500} A_i$
- $N_O = \sum_{i=1}^{500} O_i$

i. [1 Pt] What is $\mathbb{E}[N_A]$?

$\mathbb{E}[N_A]$ = □

Solution:

$$\begin{aligned}\mathbb{E}[N_A] &= \mathbb{E}[\sum_{i=1}^{500} A_i] \\ &= \sum_{i=1}^{500} \mathbb{E}[A_i] \\ &= \sum_{i=1}^{500} \frac{500}{20000} \\ &= 500 \cdot \frac{500}{20000} \\ &= \frac{25}{2}\end{aligned}$$

ii. [1 Pt] What is Var$(N_D + N_S + N_A + N_M + N_O)$?

Var$(N_D + N_S + N_A + N_M + N_O)$ = □

Solution: 0. $N_D + N_S + N_A + N_M + N_O = 500$. $Var(500) = 0$. The variance of a constant is 0.

(c) [2 Pts] Let's consider the situation where we sample with replacement instead of taking a SRS. If we take a sample with replacement of 10 contributions, what is the probability that 3 were by Daniel, 3 were by Suraj, and 4 were by Mansi?

Probability = □

Solution: This is a multinomial distribution where $P(\text{Daniel}) = \frac{2000}{20000}$, $P(\text{Suraj}) = \frac{1800}{20000}$ and $P(\text{Mansi}) = \frac{700}{20000}$ and the number of trials $n = 10$.
Then,

$$P(3 \text{ Daniel}, 3 \text{ Suraj}, 4 \text{ Mansi}) = \binom{10}{3} \cdot \left(\frac{2000}{20000}\right)^3 \cdot \binom{7}{3} * \left(\frac{1800}{20000}\right)^3 \cdot \left(\frac{700}{20000}\right)^4$$

Note, $\binom{10}{3} \cdot \binom{7}{3} = \frac{10!}{3!3!4!}$.

2 Relative Mean Squared Error [6 Pts.]

Consider a set of points $\{x_1, x_2, ..., x_n\}$, where each $x_i \in \mathbb{R}$, and further suppose we want to determine a summary statistic c for this data. Naturally, our choice of loss function determines the optimal c.

In this problem, let's consider a new loss function $l(c) = (x - c)^2/x$. We call this loss function the **relative** squared error loss. If we compute the average over an entire dataset, we get the empirical risk function below:

$$L(c) = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - c)^2}{x_i}$$

For example, suppose our data is `[0.1, 0.2, 0.5, 0.5, 1]`, and we consider the summary statistic c = 1. The empirical risk would be:

$$\frac{1}{5}\left(\frac{(0.1-1)^2}{0.1} + \frac{(0.2-1)^2}{0.2} + \frac{(0.5-1)^2}{0.5} + \frac{(0.5-1)^2}{0.5} + \frac{(1-1)^2}{1}\right)$$

$$= \frac{(8.1 + 3.2 + 0.5 + 0.5)}{5} = 2.46$$

[6 pts] Give the summary statistic that minimizes the relative mean squared error for the data above, i.e. `[0.1, 0.2, 0.5, 0.5, 1]`. **Make sure to show your work in the space below, correct answers will not be accepted without shown work**.

$\hat{c}$ = □

Solution: Since the loss function is convex, we can find the optimizer of the loss function by setting the derivative with respect to c to 0 and solve for c.

$$
\begin{aligned}
\frac{dL}{dc} &= \frac{1}{n}\sum_{i=1}^{n}\frac{d}{dc}\frac{(x_i - c)^2}{x_i} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{-2(x_i - c)}{x_i} \\
&= \frac{1}{n}\sum_{i=1}^{n} -2 + \frac{2c}{x_i} \\
&= \frac{1}{n}(\sum_{i=1}^{n} -2 + \sum_{i=1}^{n}\frac{2c}{x_i}) \\
&= \frac{1}{n}(-2n + 2c\sum_{i=1}^{n}\frac{1}{x_i}) = -2 + \frac{2c}{n}\sum_{i=1}^{n}\frac{1}{x_i} = 0
\end{aligned} \tag{1}
$$

Therefore,

$$-2n + 2c\sum_{i=1}^{n}\frac{1}{x_i} = 0$$

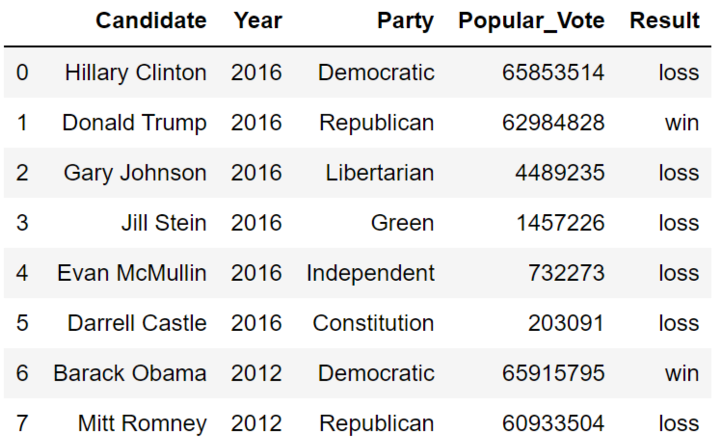
$$\hat{c} = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}}$$

Plugging in the data, we have that the optimal value of c is

$$\hat{c} = \frac{5}{\frac{1}{0.1} + \frac{1}{0.2} + \frac{1}{0.5} + \frac{1}{0.5} + \frac{1}{1}} = \frac{5}{20} = \frac{1}{4}$$

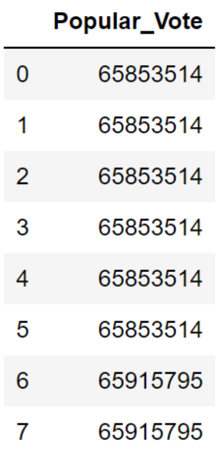
3 Election (Pandas) [12 Pts.]

You are given an DataFrame `elections` with the results of each U.S. presidential election. The first 8 rows of `elections` is shown on the left. The `max_votes` Series on the right is described later on this page.



	Candidate	Year	Party	Popular_Vote	Result
0	Hillary Clinton	2016	Democratic	65853514	loss
1	Donald Trump	2016	Republican	62984828	win
2	Gary Johnson	2016	Libertarian	4489235	loss
3	Jill Stein	2016	Green	1457226	loss
4	Evan McMullin	2016	Independent	732273	loss
5	Darrell Castle	2016	Constitution	203091	loss
6	Barack Obama	2012	Democratic	65915795	win
7	Mitt Romney	2012	Republican	60933504	loss

`elections`



	Popular_Vote
0	65853514
1	65853514
2	65853514
3	65853514
4	65853514
5	65853514
6	65915795
7	65915795

`max_votes`

(a) [3 Pts] Suppose we want to add a new column called `Popular_Result` that is equal to 'win' if the candidate won the popular vote and 'loss' if the candidate lost the popular vote. Note, this is not the same thing as the `Result` column, e.g. Donald Trump won the 2016 election but lost the popular vote, i.e. did not have the largest value for `Popular_Vote` in 2016.

To do this, we'll start by using a new pandas function we have not learned in class called `transform`. For example, the code below creates a Series called `max_votes` shown at the top right of this page.

```
max_votes = elections.groupby("Year")["Popular_Vote"].transform(max)
max_votes.to_frame().head(8) # to_frame used so that it looks nicer
```

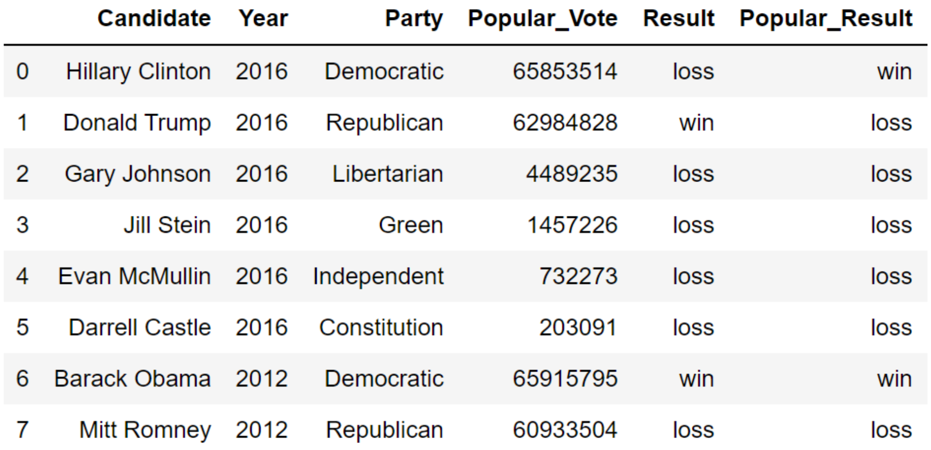
Using the `max_votes` Series, create the new `Popular_Result` column in `elections`. Your code may not use any loops. We have done the first line for you. If you're not quite sure what your goal is, we provide a picture of the result on the next page. You may not need all lines. **Hint: The `.loc` feature in pandas accepts boolean arrays for either of its arguments.**

```
elections["Popular_Result"] = "loss"

_____________________________________________________________

_____________________________________________________________

_____________________________________________________ = "win"
```

Solution:

```
elections["Popular_Result"] = "loss"
popular_winners = elections["Popular_Vote"] == max_votes
elections.loc[popular_winners, "Popular_Result"] = "win"
```

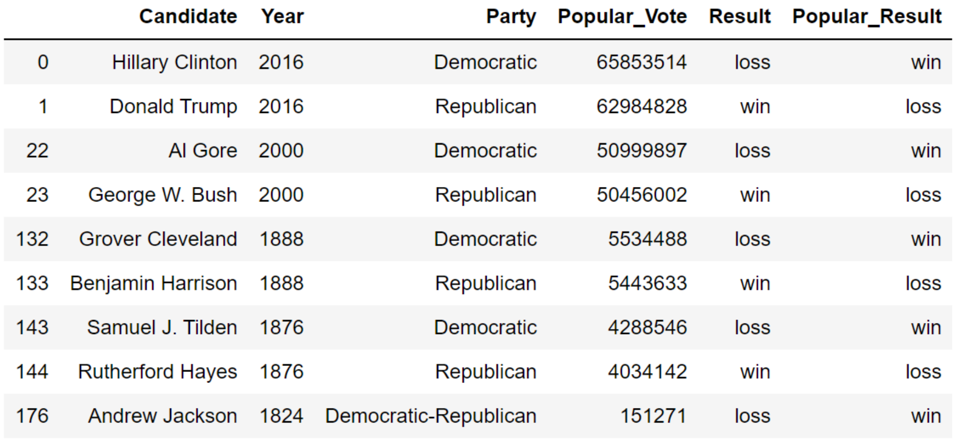
(b) [2 Pts] Below is the correct result for part a of this problem.



	Candidate	Year	Party	Popular_Vote	Result	Popular_Result
0	Hillary Clinton	2016	Democratic	65853514	loss	win
1	Donald Trump	2016	Republican	62984828	win	loss
2	Gary Johnson	2016	Libertarian	4489235	loss	loss
3	Jill Stein	2016	Green	1457226	loss	loss
4	Evan McMullin	2016	Independent	732273	loss	loss
5	Darrell Castle	2016	Constitution	203091	loss	loss
6	Barack Obama	2012	Democratic	65915795	win	win
7	Mitt Romney	2012	Republican	60933504	loss	loss

`elections`

Fill in the code below so that `df` is a dataframe with only candidates whose ultimate result was not the same as the popular vote, i.e.



	Candidate	Year	Party	Popular_Vote	Result	Popular_Result
0	Hillary Clinton	2016	Democratic	65853514	loss	win
1	Donald Trump	2016	Republican	62984828	win	loss
22	Al Gore	2000	Democratic	50999897	loss	win
23	George W. Bush	2000	Republican	50456002	win	loss
132	Grover Cleveland	1888	Democratic	5534488	loss	win
133	Benjamin Harrison	1888	Republican	5443633	win	loss
143	Samuel J. Tilden	1876	Democratic	4288546	loss	win
144	Rutherford Hayes	1876	Republican	4034142	win	loss
176	Andrew Jackson	1824	Democratic-Republican	151271	loss	win

`df`

You may not need all lines. Make sure to assign `df` somewhere.

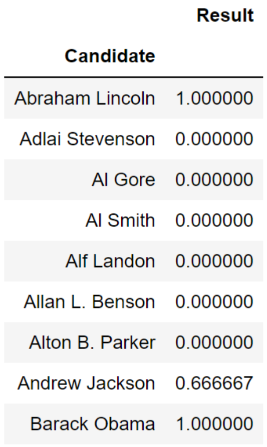
Solution:

```
df = elections[elections["Result"] != elections["Popular_Result"]]
```

alternatively,

```
df = elections.query("Result != Popular_Result")
```

(c) [4 Pts] Create a series win_fraction giving the fraction each candidate won out of all elections participated in by that candidate. For example, Andrew Jackson participated in 3 presidential elections (1824, 1828, and 1832) and won 2 of these (1828 and 1832), so his fraction is 2/3. You should use the Result column, not the Popular_Result column. For example, win_fraction.to_frame().head(9) would give us:



	Result
Candidate	
Abraham Lincoln	1.000000
Adlai Stevenson	0.000000
Al Gore	0.000000
Al Smith	0.000000
Alf Landon	0.000000
Allan L. Benson	0.000000
Alton B. Parker	0.000000
Andrew Jackson	0.666667
Barack Obama	1.000000

win_fraction

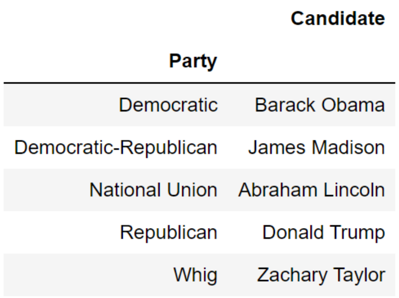
You may not use loops of any kind. You do not need to worry about the order of the candidates. You may assume that no two candidates share the same name.

```
def f(s):
    ______________________________________________
    ______________________________________________
    ______________________________________________

win_fraction = ______________________________________________
```

Solution:

```
def f(s):
    num_wins = sum(s == "win")
    num_elections = len(s)
    return num_wins / num_elections

win_fraction = elections.groupby("Candidate")["Result"].agg(f)
```

(d) [3 Pts] Create a series s that gives the name of the last candidate who successfully won office for each party. That is, s.to_frame() would give us:



	Candidate
Party	
Democratic	Barack Obama
Democratic-Republican	James Madison
National Union	Abraham Lincoln
Republican	Donald Trump
Whig	Zachary Taylor

s

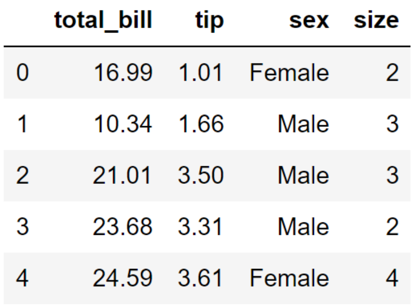
```
elections_sorted = elections.sort_values(______________)
winners_only = ________________________________________
s = winners_only._____________(____________)[_____________].___________
```

Solution:

```
elections_sorted = elections.sort_values("Year")
winners_only = elections_sorted[elections_sorted["Result"]=="win"]
winners_only.groupby("Party")["Candidate"].last()
```

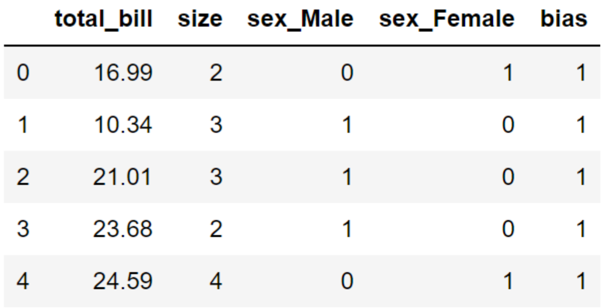
4 Regression [13 Pts.]

Recall from lab 9 the tips dataset from the seaborn library, which contains records about tips, total bills, and information about the person who paid the tip. Throughout this entire problem, assume there are a total of 20 records, though we only ever show 5. The first 5 rows of the resulting dataframe are shown below. The integer on the far left is the index, not a column of the DataFrame.



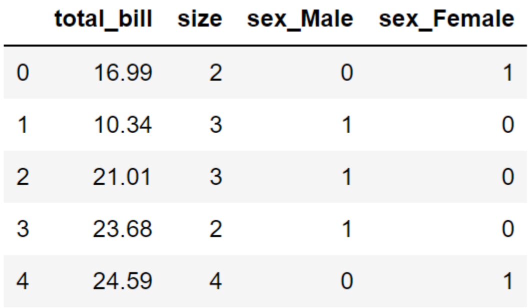
	total_bill	tip	sex	size
0	16.99	1.01	Female	2
1	10.34	1.66	Male	3
2	21.01	3.50	Male	3
3	23.68	3.31	Male	2
4	24.59	3.61	Female	4

Suppose we want to predict the tip from the other available data. Four possible design matrices $\mathbb{X}_{MFB}$, $\mathbb{X}_{MF}$, $\mathbb{X}_{FB}$, and $\mathbb{X}_F$ are given below.



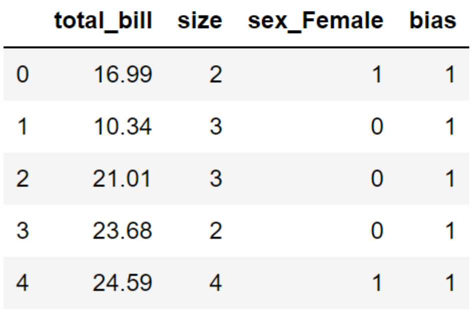
	total_bill	size	sex_Male	sex_Female	bias
0	16.99	2	0	1	1
1	10.34	3	1	0	1
2	21.01	3	1	0	1
3	23.68	2	1	0	1
4	24.59	4	0	1	1

$\mathbb{X}_{MFB}$



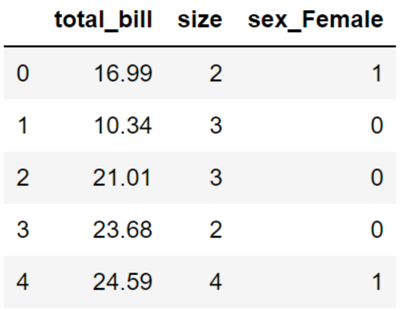
	total_bill	size	sex_Male	sex_Female
0	16.99	2	0	1
1	10.34	3	1	0
2	21.01	3	1	0
3	23.68	2	1	0
4	24.59	4	0	1

$\mathbb{X}_{MF}$



	total_bill	size	sex_Female	bias
0	16.99	2	1	1
1	10.34	3	0	1
2	21.01	3	0	1
3	23.68	2	0	1
4	24.59	4	1	1

$\mathbb{X}_{FB}$



	total_bill	size	sex_Female
0	16.99	2	1
1	10.34	3	0
2	21.01	3	0
3	23.68	2	0
4	24.59	4	1

$\mathbb{X}_F$

(a) i. [2 Pts] What is the rank of each of our four design matrices?

rank$(\mathbb{X}_{MFB}) =$ ○ 1 ○ 2 ○ 3 ● **4** ○ 5 ○ 19 ○ 20

rank$(\mathbb{X}_{MF}) =$ ○ 1 ○ 2 ○ 3 ● **4** ○ 5 ○ 19 ○ 20

rank$(\mathbb{X}_{FB}) =$ ○ 1 ○ 2 ○ 3 ● **4** ○ 5 ○ 19 ○ 20

rank$(\mathbb{X}_{F}) =$ ○ 1 ○ 2 ● **3** ○ 4 ○ 5 ○ 19 ○ 20

ii. [2 Pts] Recall that an Ordinary Least Squares (OLS) model is an unregularized linear model that minimizes the MSE for a given design matrix. Suppose we train three different unregularized OLS models on X_{MF}, X_{FB} and X_F, respectively. The resulting predictions given by each model are $\vec{\hat{y}}_{MF}$, $\vec{\hat{y}}_{FB}$, and $\vec{\hat{y}}_F$. Which of the following statements are true?

☑ $\vec{\hat{y}}_{MF} = \vec{\hat{y}}_{FB}$
☐ $\vec{\hat{y}}_{MF} = \vec{\hat{y}}_{F}$
☐ $\vec{\hat{y}}_{FB} = \vec{\hat{y}}_{F}$
☐ None of These

iii. In lecture, we said that the residuals sum to zero for an OLS model trained on a feature matrix that includes a bias term. For example, if S_{FB} is the sum of the residuals for $\vec{\hat{y}}_{FB}$, then $S_{FB} = 0$ because $\mathbb{X}_{FB}$ includes a bias term.

i. [2 Pts] Let S_{MF}, S_{FB}, and S_F be the sums of the residuals for our three models. Which of the following are true? We have omitted S_{FB} from the list below because we already gave away the answer above.

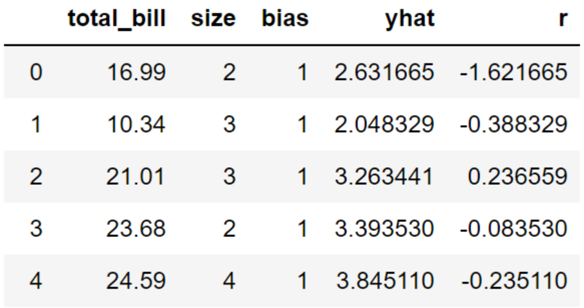
☑ $S_{MF} = 0$ ☐ $S_F = 0$ ☐ Neither of these

ii. [2 Pts] Let S^F_{MF}, S^F_{FB}, and S^F_F be the sums of the residuals for only female customers. For example, S^F_{MF} is the sum of the residuals for the 0th, 4th, etc. rows of $\mathbb{X}_{MF}$, S^F_{FB} is the sum of the residuals for the 0th, 4th, etc. rows of $\mathbb{X}_{FB}$, and similarly for S^F_F. Which of the following are true?

☑ $S^F_{MF} = 0$ ☑ $S^F_{FB} = 0$ ☑ $S^F_F = 0$ ☐ None of these

(b) Suppose we create a new design matrix $\mathbb{X}_B$ that contains only the total bill, size, and a bias term. Suppose we then fit an OLS model on $\mathbb{X}_B$, which generates predictions $\vec{\hat{y}} = [\hat{y}_0, \hat{y}_1, ..., \hat{y}_{19}] = [2.631665, 2.0483329, \dots]$ with residuals $\vec{r} = [r_0, r_1, ..., r_{19}] = [-1.621665, -0.388329, \dots]$.

Suppose we then do a very strange thing: We create a new design matrix $\mathbb{W}$ that has the columns from $\mathbb{X}_B$, as well as two new columns corresponding to $\vec{\hat{y}}$ and $\vec{r}$ from our model on $\mathbb{X}_B$. Note: You'd never ever do this, but we're asking as a way to probe your knowledge of regression. The first 5 rows of $\mathbb{W}$ are given below.



	total_bill	size	bias	yhat	r
0	16.99	2	1	2.631665	-1.621665
1	10.34	3	1	2.048329	-0.388329
2	21.01	3	1	3.263441	0.236559
3	23.68	2	1	3.393530	-0.083530
4	24.59	4	1	3.845110	-0.235110

$\mathbb{W}$

i. [2 Pts] What is the rank of $\mathbb{W}$?

○ 0 ○ 1 ○ 2 ○ 3 [✓ ○ **4**] ○ 5 ○ 10 ○ 20 ○ 40

ii. [3 Pts] Let $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$, $\hat{\beta}_5$ be optimal parameters of a linear regression model on $\mathbb{W}$, e.g. $\hat{\beta}_4$ is the weight of the `yhat` column of our data frame. Give a set of parameters that minimizes the MSE.

$\hat{\beta}_1 =$ □

$\hat{\beta}_2 =$ □

$\hat{\beta}_3 =$ □

$\hat{\beta}_4 =$ □

$\hat{\beta}_5 =$ □

Solution: $\hat{\beta}_1 = 0$, $\hat{\beta}_2 = 0$, $\hat{\beta}_3 = 0$, $\hat{\beta}_4 = 1$, $\hat{\beta}_5 = 1$.
$r = y - \hat{y}$ so $\hat{y} + r = \hat{y} + y - \hat{y} = y$. The MSE would be 0.

5 Alternate Classification Techniques [14 Pts.]

The primary technique for binary classification in our course was logistic regression, where we first calculated $P(Y = 1|\vec{x}) = \sigma(\vec{x}^T \vec{\beta})$, then applied a threshold T to compute a label (either 0 or 1). In other words, we predict $\hat{y} = f(\vec{x}) = \mathbb{I}(\sigma(\vec{x}^T \vec{\beta}) > T)$, where $\mathbb{I}$ is an indicator function (i.e. returns 1 if the argument is true, 0 otherwise).

We trained such a model by finding the $\vec{\beta}$ that minimizes the cross entropy loss between our predicted probabilities and the true labels.

In this problem we'll explore some variants on this idea.

(a) In this part, we'll consider various loss functions.

i. [2 Pts] Suppose our true labels are $\vec{y} = [0, 0, 1]$, our predicted probabilities of being in class 1 are $[0.1, 0.6, 0.9]$, and our threshold is $T = 0.5$. Give the total (not average) cross-entropy loss. Do not simplify your answer.

Total CE Loss = □

Solution:

$$-(\log(1 - 0.1) + \log(1 - 0.6) + \log(0.9)) = -(\log(0.9) + \log(0.4) + \log(0.9))$$

ii. [2 Pts] For the same values as above, give the total squared loss. Do not simplify your answer.

Squared Loss = □

Solution:

$$(0 - 0.1)^2 + (0 - 0.6)^2 + (1 - 0.9^2) = 0.1^2 + 0.6^2 + 0.1^2$$

iii. [2 Pts] Which of the following are valid reasons why we minimized the average cross-entropy loss rather than the average squared loss?

□ To prevent our parameters from going to infinity for linearly separable data.

□ There is no closed form solution for the average squared loss.

□ **To improve the chance that gradient descent converges to a good set of parameters.**

□ **The cross entropy loss gives a higher penalty to very wrong probabilities.**

□ None of the above

iv. [1 Pt] A third loss function we might consider is the zero-one loss, given by $L_{ZO}(y, \hat{y}) = \mathbb{I}(y \neq \hat{y})$. In other words, the loss is 1 if the label is incorrect, and 0 if it is correct. For the same values above, what is the total zero-one loss?

○ 0 ○ **1** ○ 2 ○ 3

v. [2 Pts] The zero-one loss is a function of both $\vec{\beta}$ and T. This is in contrast to the cross-entropy loss, which is only a function of $\vec{\beta}$. Let $\hat{\vec{\beta}}_{ZO}$ and $\hat{T}_{ZO}$ be parameters that minimize the zero-one loss. Which of the following are true about $\hat{\vec{\beta}}_{ZO}$ and $\hat{T}_{ZO}$?

□ **They maximize accuracy.** □ They maximize precision.
□ They maximize recall. □ None of these

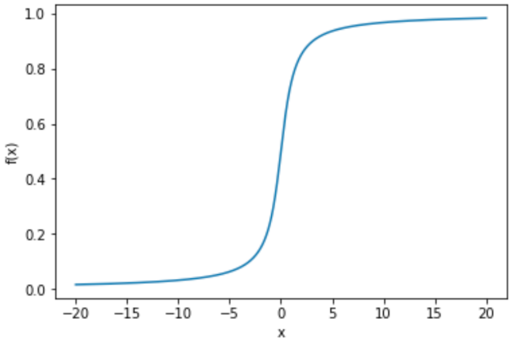
Solution: Accuracy is # correct / total, and zero-one loss is total - # correct. If you minimize the zero-one loss, you maximize the # correct. We have seen in class that something that maximizes accuracy does not necessarily maximize recall or precision.

vi. [3 Pts] A DS100 student wants to run gradient descent on the total zero-one loss to find optimal $\hat{\vec{\beta}}_{ZO}$ and $\hat{T}_{ZO}$. Give the very specific reason that this will always fail. Answer in 10 words or less. Vague answers will be given no credit.

Solution: The gradient will always be 0 so running gradient descent will never find the global minimum unless we start at it.

(b) [2 Pts] In this part, we'll consider an alternative to the logistic function.

Instead of using the logistic function as our choice of f, let's say we instead use a scaled inverse tangent function, $f(x) = \frac{1}{\pi}\tan^{-1}(x) + \frac{1}{2}$. This choice of f has the exact same tail-end behavior as $\sigma(x)$. In other words, it is always between 0 and 1. A plot of f is below:



Which of the following are true?

□ **The cross entropy loss is still well defined for all possible outputs of our model.**

□ **We are still able to construct an ROC curve and use the AUC as a metric for our classifier.**

□ **We can still compute a confusion matrix from our classifier.**

□ We can still assume that $\log \frac{P(Y=1|x)}{P(Y=0|x)}$ is linear.

□ None of the above

6 Linear Separability [4 Pts.]

Suppose we fit a logistic regression model with two features x_1, x_2, and find that with classification threshold $T = 0.75$ and $\hat{\vec{\beta}} = [\hat{\beta}_1, \hat{\beta}_2] = [2, 3]$, we achieve 100% training accuracy. Let $x_2 = mx_1 + b$ be the equation for the line that separates the two classes. Give m and b (you may leave your answers in terms of ln). Hint: You might find the following fact useful: $\sigma(ln(3)) = 0.75$.

$m =$

$b =$

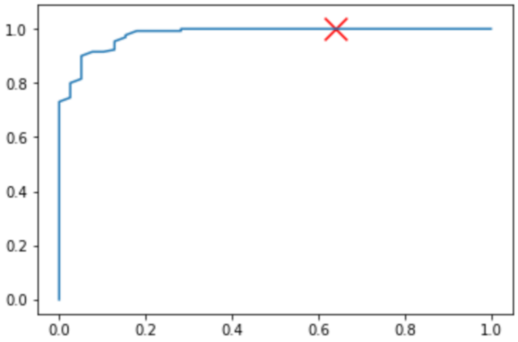
Solution:

We are given that $\sigma(2x_1 + 3x_2) = 0.75$ is a hyperplane that achieves 100% training accuracy, i.e. that linearly separates our data.

To get this in the required form, we can take the inverse sigmoid of each side, getting $2x_1 + 3x_2 = \sigma^{-1}(0.75) = ln(3)$. Rearranging, we get $x_2 = -\frac{2}{3}x_1 + \frac{ln(3)}{3}$, so $m = -\frac{2}{3}$ and $b = \frac{ln(3)}{3}$.

7 ROC Curves [5 Pts.]

Here, we present a ROC curve, with unlabelled axes.



(a) [4 Pts] Fill in the pseudocode below to generate a ROC Curve. (Ignore the ”X” above.)

Hint: You can convert a boolean array to an array of 1’s and 0’s by multiplying the array by 1:

```
>>> y
array([False, False,  True,  True], dtype=bool)
>>> 1 * y
array([0, 0, 1, 1])
```

```
predicted_probs = np.array([0.37, 0.1, ...])
y_actual = np.array([1, 0, ...])
thresholds = np.linspace(______, _______, 1000)
tprs, fprs = [], []
for t in ________________________________:
    y_pred = ________________________________
    a = np.sum((y_pred == y_actual) & (y_pred == 1))
    b = np.sum((y_pred == y_actual) & (y_pred == 0))
    c = np.sum((y_pred != y_actual) & (y_pred == 1))
    d = np.sum((y_pred != y_actual) & (y_pred == 0))
    tprs.append(________________________________)
    fprs.append(________________________________)
plt.plot(fprs, tprs)
```

Solution:

```
thresholds = np.linspace(0, 1, 1000)

for t in thresholds:
    y_pred =  1 * (pedicted_probs > t)
```

```
a = np.sum((y_pred == y_actual) & (y_pred == 1))
b = np.sum((y_pred == y_actual) & (y_pred == 0))
c = np.sum((y_pred != y_actual) & (y_pred == 1))
d = np.sum((y_pred != y_actual) & (y_pred == 0))

tprs.append(a / (a + d))
fprs.append(c / (b + c))
```

(b) [1 Pt] Which of the following classification thresholds most likely corresponds to the point marked with an ”X” above?

○ **0.1** ○ 0.65 ○ 0.9 ○ 1.0

8 PCA [7 Pts.]

(a) Consider the matrix X below.

$$X = \begin{bmatrix} 0 & 2 & -1 \\ 0 & 2 & -2 \\ 1 & 1 & -3 \\ 1 & 1 & -4 \\ 2 & 0 & -5 \end{bmatrix}$$

Suppose we decompose X using PCA into $X = U\Sigma V^T$. Let r x c be the dimensions of V^T.

i. [1 Pt] What is r?

○ 0 ○ 1 ○ 2 ○ **3** ○ 4 ○ 5 ○ None of these

ii. [1 Pt] What is c?

○ 0 ○ 1 ○ 2 ○ **3** ○ 4 ○ 5 ○ None of these

(b) [3 Pts] Let P be the principal component matrix of X. That is, $P = U\Sigma$.
Suppose we now decompose the principal component matrix P into its principal components, giving us $P = U_P \Sigma_P V_P^T$. What is V_P^T?

$V_P^T =$

Solution:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

One can arrive at this answer by equating $U\Sigma = U_P \Sigma_P V_P^T$ and noticing that this implies $V_P^T = I$, but **that isn't very interesting or useful**.

Instead, we can recognize that V^T is a rotation matrix that rotates our original data X so that it is axis-aligned. In other words, $P = U\Sigma$ is the data rotated such that the greatest variance occurs along the x axis.

If we perform PCA again, we're basically trying to rotate P so that it is axis-aligned. However, it is already axis-aligned so the rotation matrix V_P^T should do nothing. The matrix which does nothing is the identity matrix. The dimensions are 3 x 3 for the same reasons as in part a.

Note, it was not enough to state I as the solution, as that leaves the dimensions ambiguous. We had many students who gave identity matrices that had the wrong dimensions.

(c) Consider the statement: ”When we created 2D PCA scatter plots in this class, we were usually plotting the first 2 ___________ of ____________”?

i. [1 Pt] For the first blank, what is the appropriate word?

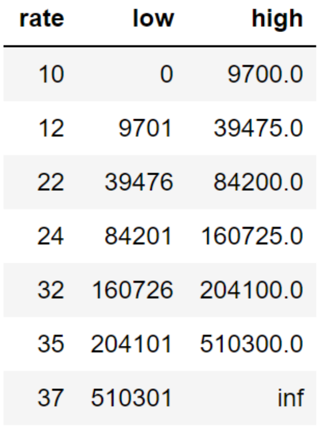
○ rows ○ **columns**

ii. [1 Pt] For the second blank, what is the appropriate object?

○ X ○ U ○ Σ ○ V^T ○ $U\Sigma$ ○ ΣV^T ○ $U\Sigma V^T$

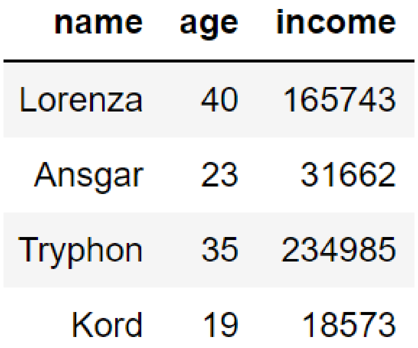
9 SQL [10 Pts.]

(a) [5 Pts] In this problem, we have the two tables below, named brackets and names respectively. The left table is a list of U.S. tax brackets, e.g. a person's first $9700 of income is taxed at 10%, income between $9701 and $39475 is taxed at 12%, etc. The right table is a list of people, their ages, and incomes.



rate	low	high
10	0	9700.0
12	9701	39475.0
22	39476	84200.0
24	84201	160725.0
32	160726	204100.0
35	204101	510300.0
37	510301	inf

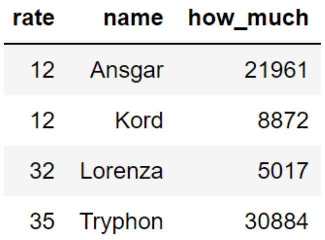
brackets



name	age	income
Lorenza	40	165743
Ansgar	23	31662
Tryphon	35	234985
Kord	19	18573

names

Give a SQL query that results in the table below, except that the order of your rows may be different. Here, the rate column represents the highest tax bracket at which their income is taxed. For example, Lorenza earns $165,743, so her highest income is taxed at the 32% rate. The how_much column says how much of the person's income is taxed at this rate, e.g. $5,017 of Lorenza's income is taxed at 32% since her income exceeds the low of the 32% bracket of $160,726 by $5,017. Your output should have the same column names as the example below.



rate	name	how_much
12	Ansgar	21961
12	Kord	8872
32	Lorenza	5017
35	Tryphon	30884

SELECT ____

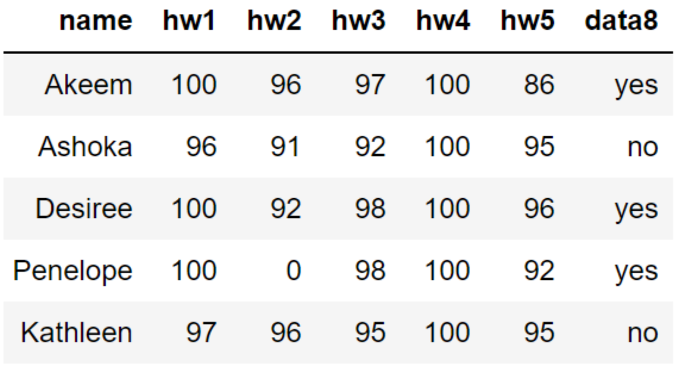
FROM ____

WHERE ______________________ AND ______________________;

Solution:

```
SELECT rate, name, and income-low AS how_much
FROM brackets, names
WHERE income >= low AND income <= high;
```

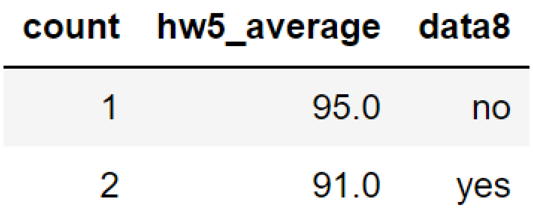
(b) [5 Pts] For this problem, we have the `ds100_grades` table below.



name	hw1	hw2	hw3	hw4	hw5	data8
Akeem	100	96	97	100	86	yes
Ashoka	96	91	92	100	95	no
Desiree	100	92	98	100	96	yes
Penelope	100	0	98	100	92	yes
Kathleen	97	96	95	100	95	no

`ds100_grades`

Suppose we want to generate the table below.



count	hw5_average	data8
1	95.0	no
2	91.0	yes

The table above provides the average grade on HW5 for students with ”ee” in their name, separated into two groups: those who have taken Data 8 and those who have not. For example, Akeem and Desiree both have ”ee” in their names, and have taken Data 8. The average of their scores is 91. Kathleen has an ”ee” in her name, but has not taken Data 8. Since she is the only person in the table who has not taken Data 8, the average is just her score of 95. Penelope and Ashoka do not have ”ee” in their names, so their data will not get included in the table. Each table includes the count, the HW5 average, and whether the row corresponds to students who took Data 8 or not. Give a query below that generates this table. The order of your rows does not matter. Your output should have the same column names as the example below.

SELECT ____

FROM ____

WHERE ____

____;

Solution:

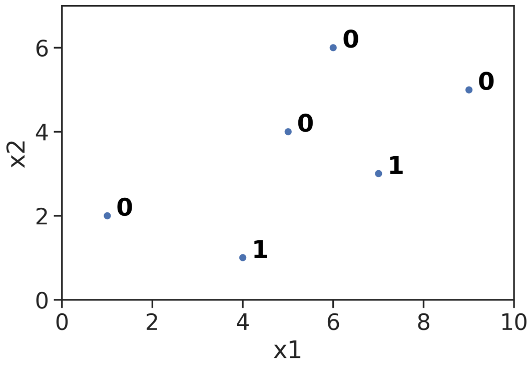
```
SELECT COUNT(*) AS count, AVG(hw5) AS hw5_average, data8
FROM ds100_grades
WHERE name LIKE '%ee%'
GROUP BY data8;
```

10 Decision Trees [8 Pts.]

Suppose we are trying to train a decision tree model for a binary classification task. We denote the two classes as **0** (the negative class) and **1** (the positive class) respectively. Our input data consists of 6 sample points and 2 features x_1 and x_2.

The data is given in the table below, and is also plotted for your convenience on the right.

x_1	x_2	Label (y)
1	2	0
4	1	1
5	4	0
6	6	0
7	3	1
9	5	0



(a) [2 Pts] What is the entropy at the root of the tree? Do not simplify your answer.

entropy =

Solution: Proportion 0 = $\frac{4}{6} = \frac{2}{3}$.
Proportion 1 = $\frac{2}{6} = \frac{1}{3}$.
Entropy = $-(\frac{2}{3} \log \frac{2}{3} + \frac{1}{3} \log \frac{1}{3})$.

(b) [3 Pts] Suppose we split the root note with a rule of the form $x_i \geq \beta$, where i could be either 1 or 2. Which of the following rules minimizes the weighted entropy of the two resulting child nodes?

○ $x_1 \geq 3$ ○ $x_1 \geq 4.5$ ○ $x_1 \geq 8.5$ [✓ ○ $x_2 \geq 3.5$] ○ $x_2 \geq 4.5$

(c) [3 Pts] Now, suppose we split the root note with a different rule of the form below:

$$x_1 \geq \beta_1 \text{ and } x_2 \leq \beta_2,$$

where β_1, β_2 are the thresholds we choose for splitting. Give a β_1 and β_2 value that minimizes the entropy of the two resulting child nodes of the root.

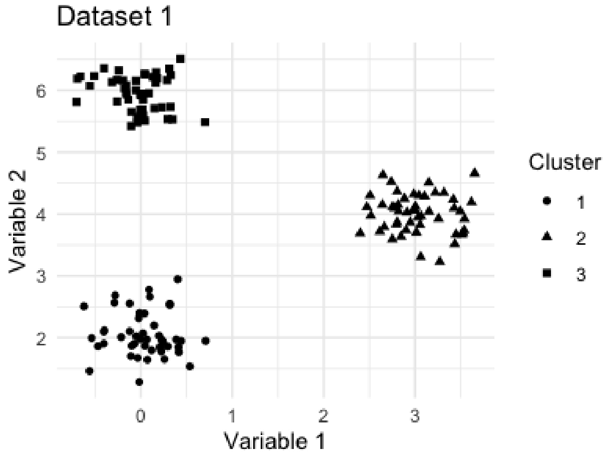
β_1 =

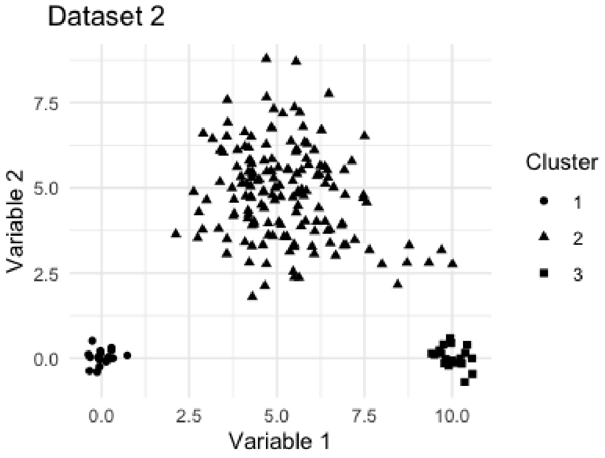
β_2 =

Solution: $\beta_1 = 4, \beta_2 = 3$. Any solution that contains only the two points labeled 1 is sufficient.

11 Clustering [7 Pts.]

(a) The two figures below show two datasets clustered into three clusters each. **For each dataset, state whether the given clustering could have been generated by the K-means and Max-agglomerative clustering algorithms**. By max-agglomerative we mean the exact algorithm discussed in class, where the distance between two clusters is given by the maximum distance between any two points in those clusters.





Note: There are no hidden overplotted cluster markers. For example, there's no need to look closely at all the triangles to see if there is a square or circle hidden somewhere.

i. [2 Pts] Dataset 1:

□ **K-means** □ **Max-agglomerative** □ None of these

ii. [2 Pts] Dataset 2:

□ K-means □ Max-agglomerative □ **None of these**

Solution: This problem was more difficult than we had intended. The problem was scored entirely based on your answer to the K-means part of the problem, i.e. you were not penalized (or rewarded) for getting max-agglomerative correct as actually proving the correct answer is extremely difficult without a computer.

(b) For each of the following statements, say whether the statement is true or false.

i. [1 Pt] If we run K-Means clustering three times, and the generated labels are exactly equal all three times, then the locations of the generated cluster centers are also exactly equal all three times.

○ **True** ○ False

ii. [1 Pt] Assuming no two points have the same distance, the cluster labels computed by K-means are always the same for a given dataset.

○ True ○ **False**

iii. [1 Pt] Assuming no two points have the same distance, the cluster labels computed by Max-agglomerative clustering are always the same for a given dataset.

○ **True** ○ False

12 Potpourri [16 Pts.]

(a) [1 Pt] Suppose we train an OLS model to predict a person's salary from their age and get β_1 as the coefficient. Suppose we then train another OLS model to a predict a person's salary from both their age and number of years of education and get parameters γ_1 and γ_2, respectively. For these two models $\beta_1 = \gamma_1$.

○ Always True ○ **Sometimes True** ○ Never True

(b) [1 Pt] Suppose we train a ridge regression model with non-zero hyperparameter λ to predict a person's salary from their age and get β_1 as the coefficient. Suppose we then train another ridge regression model using the same non-zero hyperparameter λ to predict a person's salary from both their age and number of years of education and get parameters γ_1 and γ_2, respectively. For these two models $\beta_1 = \gamma_1$.

○ Always True ○ **Sometimes True** ○ Never True

(c) [1 Pt] If we get 100% training accuracy with a logistic regression model, then the data is linearly separable.

○ **Always True** ○ Sometimes True ○ Never True

(d) [1 Pt] If we get 100% training accuracy with a decision tree model, then the data is linearly separable.

○ Always True ○ **Sometimes True** ○ Never True

(e) [1 Pt] Increasing the hyperparameter λ in a ridge regression model decreases the average loss.

○ Always True ○ Sometimes True ○ **Never True**

(f) [1 Pt] Let MSE_1 be the training MSE for an unregularized OLS model trained on $\mathbb{X}_1$. Let MSE_2 be the training MSE for an unregularized OLS model trained on $\mathbb{X}_2$, where $\mathbb{X}_2$ is just $\mathbb{X}_1$ with one new linearly independent column. If $MSE_1 > 0$, then $MSE_2 < MSE_1$.

○ Always True ○ **Sometimes True** ○ Never True

Solution: This one is pretty tricky. While adding another column definitely never makes the loss worse (because we're simply increasing the size of the span of our columns a.k.a. observations), it is possible that the MSE stays the same. As a trivial counterexample:

For example suppose the y vector we're trying to predict is the column vector [1, 0, 0] and we want to predict this from a single feature [0, 1, 0]. The closest point on the span of [0, 1, 0] is [0, 0, 0] which has MSE 1/3. Suppose we add a linearly independent feature so our design matrix is now:

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The span of these two (linearly independent) columns is now a plane orthogonal to y. The closest point in this plane is still [0, 0, 0] which has MSE of 1/3.

(g) [1 Pt] When using regularization on a linear regression model, you should center and scale the quantitative non-bias columns of your design matrix.

○ **Always True** ○ Sometimes True ○ Never True

(h) [3 Pts] Suppose you have the following .xml file:

```
<catalog>
    <class>
        <name>DS 100</name>
        <semester>Fall 2019</semester>
        <professor>Josh Hug</professor>
        <professor>Deb Nolan</professor>
    </class>
    <class>
        <name>CS 61B</name>
        <semester>Spring 2019</semester>
        <professor>Josh Hug</professor>
    </class>
    <professor>Fernando Perez</professor>
</catalog>
```

Which of the following XPath queries will return only the strings ”Josh Hug” and ”Deb Nolan” (can have multiple of each)? There is at least one correct answer.

□ `//professor/text()`

□ **`//professor/../class/professor/text()`**

□ `//class/professor/../class/professor/text()`

□ **`//semester/../professor/text()`**

□ **`/catalog/class[name/text()="DS 100"]/professor/text()`**

□ `/catalog/class/name[text()="DS 100"]/professor/text()`

(i) [3 Pts] Consider the regular expression `\d\w{2,5}d+[hug+]$`

Which of the following strings match this regular expression? At least one of these is correct.

□ **`123445dg`**

□ `1234dddhug`

□ `61bdug`

□ **`61bdg`**

□ `61bdugggg`

□ `1hello234gg`

(j) [3 Pts] Consider the string `61bdugggg`

Which of these regular expressions match the entire string? At least one of these is correct.

□ `\dbug*|\w*`

□ `[61b]+\d{1,3}[a-z]*`

□ `\d{2}b+[ds100][hug]*`

□ `.*g$`

□ `61bdugggg`

□ `61[b|d]{1}ug+`

13 HCE [6 Pts.]

In this problem, we will ask a somewhat sensitive and complex real world problem. We will be lenient in grading this problem, but we want you to provide an opinion and try to defend it. You will not be penalized for unpopular or ”politically incorrect” opinions. Joke answers will receive no credit. **If there is something unclear in the problem description, write your assumption.**

In a hypothetical course, all submitted work is automatically reviewed for cheating by plagiarism detection software. However, some students also have the entirety of their subjected to an intensive manual review at the end of the semester. It is not possible to manually review all student’s work due to the large number of students in the course.

One approach is to randomly select students for manual review. An alternate approach is to use a model to try to target students who are more likely to plagiarize. For example, a student who has all perfect scores on assignments but very poor midterm grades might warrant manual review.

Suppose you build a logistic regression model to classify students with one of two labels: ”investigate” or ”do not investigate”. Students who are given the ”investigate” label have all of their work carefully reviewed by a teaching assistant (TA) for evidence of cheating. Students who are given the ”do not investigate” label are not manually reviewed at all.

The model uses as features the full text of all of the student’s electronically submitted work, grades on each problem for all assignments and exams, submission times for electronically submitted work, and the full text of all the student’s Piazza posts. The model works by generating a plagiarism probability for each student. Students with a plagiarism probability above a certain threshold will be assigned the ”investigate” label. The model is trained on a dataset collected during previous semesters of the course, where each student has a true label corresponding to whether or not the student was caught plagiarizing.

(a) [3 Pts] Below, describe at least one benefit and at least one downside of using such a logistic regression model compared to the randomized approach.

Solution: Any answer that addresses the prompt should be given full credit. Example answer:

A benefit is that the TAs may have a higher precision with this model - of all the students the TAs look at, more of them may have actually plagiarized than if the students were just randomly chosen. In turn, this causes more students to be caught for plagiarism.

A downside is that the model is trained on previous students who are labeled as having plagiarized or not plagiarized. In the past, a student who may have actually plagiarized might have not been caught, and so the model is unable to detect these ”smart” plagiarizers. This means the model may be likely to classify students who plagiarize in ways that were previously undetected as having not plagiarized.

(b) [3 Pts] Suppose we add a demographic factor to our design matrix, specifically whether the student is international or not. Suppose that after training, the coefficient related to the international feature is non-zero. Is it ethical to include this feature in your model? Why or why not?

Solution: Any answer that addresses the prompt should be given full credit. Example answer:

It is unethical to include this feature in the model because it discriminates against students based on something they cannot control. Even though this only increases the odds their work will be reviewed, there may be false positives during the review process.

Note: Some students took offense at the existence of this question on the exam, some quite gravely. The concern expressed was that the inclusion of this question on the exam reinforces stereotypes against international students. We believe that as data scientists you will face challenging moral questions like these, and we believe that engaging with a difficult question despite the discomforts is worth it. In real world situations, models often make conclusions that seem unjust, e.g. the Amazon applicant review algorithm that we discussed in lecture that specifically lowered the scores of female job applicants.

Models do not exist in a vacuum. They are trained on real world data, and the predictions they generate are used in many different ways. This problem gave you a chance to explore these broader context issues. Many of you highlighted the fact that the conclusions of the algorithm were only used as the first step in a thorough code review process. Others objected to the fact that the algorithm discriminates based on characteristics that a student cannot control. All of these and more were angles worth pursuing.

It is worth noting that we did not say whether the coefficient was positive or negative.

14 1729 [0 Pts.]

(a) [0 Pts] What is the height difference between Josh Hug and Suraj Rampure? (Make sure to specify units.)

Height difference = □

Solution: 1 foot.

(b) [0 Pts] What should Josh name his new kid (assume female if you want a gender specific name)?

Name = □

Solution: Taco Bell on Durant