Final Exam Solutions - DSC 80, Spring 2024

Instructions:

- This exam consists of 14 questions. A total of 100 points are available.
- Questions marked with (M) will be used for your midterm exam redemption.
- Write name in the top right of each page in the space provided.
- Please write neatly in the provided answer boxes. We will not grade work that appears elsewhere.
- Completely fill in bubbles and square boxes.

  ○ A bubble means that you should only **select one choice**.

  □ A square box means you should **select all that apply**.
- You may refer to two 8.5" × 11" sheets of notes of your own creation. No other resources or technology (including calculators) are permitted.
- Do not turn the page until instructed to do so.

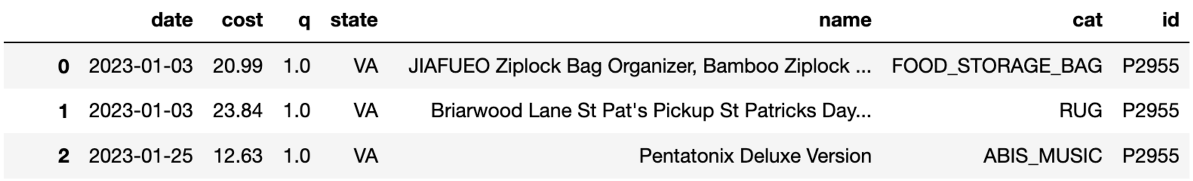
Last name	
First name	
Student ID number	
UCSD email	
Name of the person to your left	
Name of the person to your right	
All the work on this exam is my own. **(please sign)**	

This page is intentionally left blank. Feel free to use it as scratch paper.

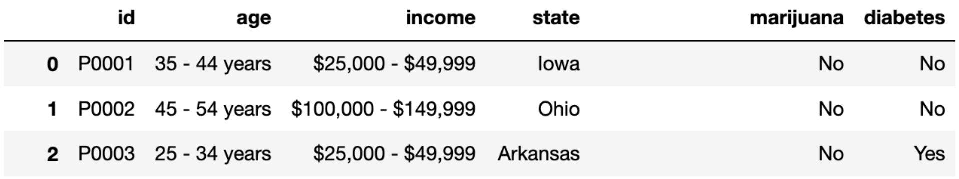
Name: ____

Question 1 . . . . *6 points*

(M) Fill in Python code below so that the last line of each code snippet evaluates to each desired result, using the `df` and `survey` DataFrames described on Page 1 of the Reference Sheet. **You may not use `for` or `while` loops in any answer for this question.** For convenience, the first few rows of `df` (top) and `survey` (bottom) are displayed below; see your Reference Sheet for the full details.



	date	cost	q	state	name	cat	id
0	2023-01-03	20.99	1.0	VA	JIAFUEO Ziplock Bag Organizer, Bamboo Ziplock ...	FOOD_STORAGE_BAG	P2955
1	2023-01-03	23.84	1.0	VA	Briarwood Lane St Pat's Pickup St Patricks Day...	RUG	P2955
2	2023-01-25	12.63	1.0	VA	Pentatonix Deluxe Version	ABIS_MUSIC	P2955



	id	age	income	state	marijuana	diabetes
0	P0001	35 - 44 years	$25,000 - $49,999	Iowa	No	No
1	P0002	45 - 54 years	$100,000 - $149,999	Ohio	No	No
2	P0003	25 - 34 years	$25,000 - $49,999	Arkansas	No	Yes

(a) (2 points) Find the participant ID of the person who made the most recent purchase in the dataset.

```
df.sort_values(_______'date'_______, ascending=True).iloc[_____-1_____, _____-1_____]
```

(b) (4 points) Create a DataFrame that compares the range of item costs for people with diabetes and people that don't have diabetes. The DataFrame should by indexed by the unique values in the `diabetes` column (Yes and No) and have one column: the range of item costs (max cost - min cost) for each group.

```
def f(x):
    return ____________________x.max() - x.min()____________________

(df.merge(survey, on='id')

.groupby(_________'diabetes'_________)[__________['cost']__________]

._________agg_________(f))
```

Question 2 . . . . *2 points*

(a) (1 point) (M) What is the most likely missingness mechanism for the `state` column in `df`?

○ **Missing by design**
○ Missing completely at random
○ Missing at random
○ Not missing at random

(b) (1 point) (M) What is the most likely missingness mechanism for the `income` column in `survey`?

○ Missing by design
○ Missing completely at random
○ **Missing at random**
○ Not missing at random

Question 3 . . . . *7 points*

(M) The code snippet below uses a `for` loop.

```
mystery = 0
for i in df['id'].unique():
    temp = df[df['id'] == i]
    if temp['q'].sum() > 100:
        mystery += 1
```

(a) (5 points) Rewrite the snippet without using any loops.

```
mystery = (df.groupby(____________’id’____________)

.____filter____(lambda x: ______________________x[’q’].sum() > 100______________________)

[____________’id’____________].____________nunique____________() )
```

(b) (2 points) Suppose you see the output below:

```
>>> df['id'].value_counts()
P2955    200
P3001    150
P3125    100
Name: id, Length: 3, dtype: int64
```

Fill in the blank in the sentence below with a single number.

The code without `for` loops runs approximately ______3______ times faster than the code with a `for` loop.

Question 4 . . . . *10 points*

You want to use regular expressions to extract out the number of ounces from the 5 product names below.

Index	Product Name	Expected Output
0	Adult Dog Food 18-Count, 3.5 oz Pouches	3.5
1	Gardetto’s Snack Mix, 1.75 Ounce	1.75
2	Colgate Whitening Toothpaste, 3 oz Tube	3
3	Adult Dog Food, 13.2 oz. Cans 24 Pack	13.2
4	Keratin Hair Spray 2!6 oz	6

The names are stored in a pandas Series called `names`. For each snippet below, select the indexes for all the product names that **will not** be matched correctly.

(a) (5 points) Snippet:

```
regex = r'([\d.]+) oz'
names.str.findall(regex)
```

□ 0 ■ 1 □ 2 □ 3 □ 4 □ All names will be matched correctly.

(b) (5 points) Snippet:

```
regex = r'(\d+?.\d+) oz|Ounce'
names.str.findall(regex)
```

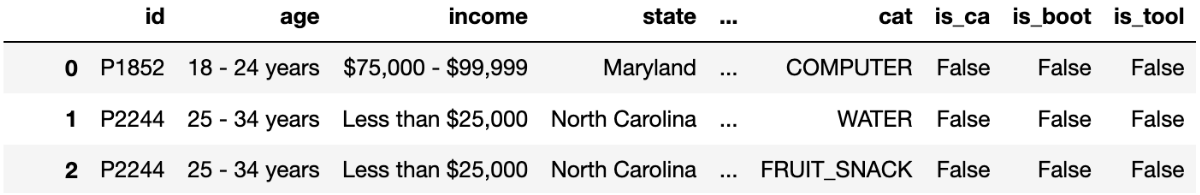
□ 0 ■ 1 ■ 2 □ 3 ■ 4 □ All names will be matched correctly.

Question 5 . . . . *4 points*

(M) Suppose you define a DataFrame `t` as follows:

```
t = (survey.merge(df, on='id', suffixes=('', '2'))
     .assign(is_ca=t['state'] == 'California',
             is_boot=t['cat'] == 'BOOT',
             is_tool=t['cat'] == 'TOOLS'))
```

The first few rows of `t` are shown below:



	id	age	income	state	...	cat	is_ca	is_boot	is_tool
0	P1852	18 - 24 years	$75,000 - $99,999	Maryland	...	COMPUTER	False	False	False
1	P2244	25 - 34 years	Less than $25,000	North Carolina	...	WATER	False	False	False
2	P2244	25 - 34 years	Less than $25,000	North Carolina	...	FRUIT_SNACK	False	False	False

For each pivot table below, state whether it is **possible** to observe Simpson's paradox without any extra information about the data.

(a) (2 points) Pivot table:

```
t.pivot_table(
    index='is_ca',
    columns='is_boot',
    values='cost',
    aggfunc='count',
)
```

○ Yes [✓ **No**] ○ Need more information to determine

(b) (2 points) Pivot table:

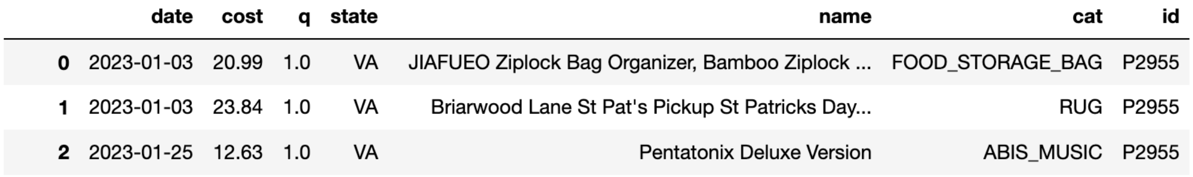
```
t.pivot_table(
    index='is_ca',
    columns='is_tool',
    values='cost',
    aggfunc='mean',
)
```

[✓ **Yes**] ○ No ○ Need more information to determine

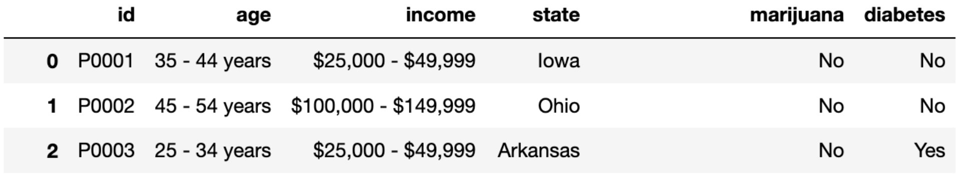
Name: ______________________________

Question 6 . . . . *9 points*

(M) For each hypothesis test below, select the **one** correct procedure to simulate a single sample under the null hypothesis, and select the **one** test statistic that can be used for the hypothesis test among the choices given. For convenience, the first few rows of `df` (top) and `survey` (bottom) are displayed below; see your Reference Sheet for the full details.



	date	cost	q	state	name	cat	id
0	2023-01-03	20.99	1.0	VA	JIAFUEO Ziplock Bag Organizer, Bamboo Ziplock ...	FOOD_STORAGE_BAG	P2955
1	2023-01-03	23.84	1.0	VA	Briarwood Lane St Pat's Pickup St Patricks Day...	RUG	P2955
2	2023-01-25	12.63	1.0	VA	Pentatonix Deluxe Version	ABIS_MUSIC	P2955



	id	age	income	state	marijuana	diabetes
0	P0001	35 - 44 years	$25,000 - $49,999	Iowa	No	No
1	P0002	45 - 54 years	$100,000 - $149,999	Ohio	No	No
2	P0003	25 - 34 years	$25,000 - $49,999	Arkansas	No	Yes

(a) (3 points) Null: Every purchase is equally likely to happen in all 50 states.
Alternative: At least one state is more likely to have purchases than others.

Simulation procedure:

○ **`np.random.multinomial(len(df), [1/50] * 50)`**
○ `np.random.multinomial(len(survey), [1/50] * 50)`
○ `np.random.multinomial(len(df), [1/2] * 2)`
○ `np.random.permutation(df['state'])`

Test statistic:

○ Difference in means
○ Absolute difference in means
○ **Total variation distance**
○ K-S test statistic

(b) (3 points) Null: The income distribution of people who smoke marijuana is the same as the income distribution for people who don't smoke marijuana.
Alternative: The income distributions are different.

Simulation procedure:

○ `np.random.multinomial(len(survey), [1/50] * 50)`
○ `np.random.multinomial(len(survey), [1/2] * 2)`
○ **`np.random.permutation(survey['income'])`**

Test statistic:

○ Difference in means
○ Absolute difference in means
○ **Total variation distance**
○ K-S test statistic

(c) (3 points) Null: The distribution of prices for items with missing categories is the same as the distribution of prices for items with recorded categories.
Alternative: Items with missing categories are more expensive than items with with recorded categories.

Simulation procedure:

○ `np.random.multinomial(len(df), [1/50] * 50)`
○ `np.random.multinomial(len(df), [1/2] * 2)`
○ **`np.random.permutation(df['cost'])`**

Test statistic:

○ **Difference in means**
○ Absolute difference in means
○ Total variation distance
○ K-S test statistic

Question 7 . . . . ***6 points***

(M) Suppose that `df` doesn't have any missing data in the `cost` column. Sam accidentally loses values from the `cost` column and values are more likely to be missing for `states` with expensive purchases. Sam's data is stored in a DataFrame called `missing`.

To recover the missing values, Sam applies the imputation methods below to the `cost` column in `missing`, then recalculates the mean of the `cost` column. For each imputation method, choose whether the new mean will be lower (-), higher (+), exactly the same (=), or approximately the same (≈) as the original mean of the `cost` column in `df` (the data without any missing observations).

(a) (2 points) `missing['cost'].fillna(missing['cost'].mean())`

[✓ ○ -] ○ + ○ = ○ ≈

(b) (2 points)

```
def mystery(s):
    return s.fillna(s.mean())
missing.groupby('state')['cost'].transform(mystery).mean()
```

○ - ○ + ○ = [✓ ○ ≈]

(c) (2 points)

```
def mystery2(s):
    s = s.copy()
    n = s.isna().sum()
    fill_values = np.random.choice(s.dropna(), n)
    s[s.isna()] = fill_values
    return s

missing.groupby('state')['cost'].transform(mystery2).mean()
```

○ - ○ + ○ = [✓ ○ ≈]

Question 8 . . . . *8 points*

Suppose you are trying to scrape album names from a website. The website has an HTML table structured as follows:

```
<table><thead>
  <tr>
    <th>Name</th> <th>Price</th> <th>Number of Reviews</th>
  </tr></thead>
<tbody>
  <tr class=”row”>
    <td>Radical Optimism</td> <td>25</td> <td>10000</td>
  </tr>
  <tr class=”row”>
    <td>Hit Me Hard and Soft</td> <td>30</td> <td>12000</td>
  </tr>
  <tr class=”row”>
    <td>SOS</td> <td>18</td> <td>30000</td>
  </tr>
  <!-- 997 <tr> elements omitted -->
</tbody>
</table>
```

Notice that the `<tbody>` tag contains 1000 `<tr>` elements, but only the first three are shown above. Suppose that you've read the HTML table above into a BeautifulSoup object called `soup`. Fill in the code below so that the `albums` variable contains a list of all the album names with (strictly) more than 15,000 reviews.

```
albums = []
for tag in soup.find_all(___(a)___):
    reviews = int(___(b)___)
    if reviews > 15000:
        album = ___(c)___
        albums.append(album)
```

(a) (2 points) What should go in blank (a)?

> **Solution:**
> `class_=”row”`

(b) (3 points) What should go in blank (b)?

> **Solution:**
> `tag.find_all('td')[2].text`

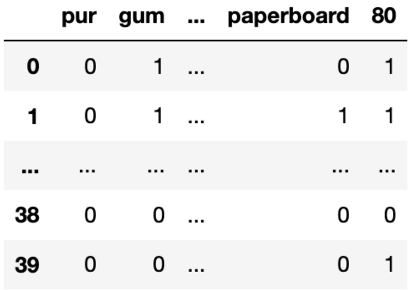
(c) (3 points) What should go in blank (c)?

> **Solution:**
> `tag.find('td').text`

Name: ____________________________________

Question 9 . . . . *7 points*

You create a table called `gums` that only contains the chewing gum purchases of `df`, then you create a bag-of-words matrix called `bow` from the `name` column of `gums`. The `bow` matrix is stored as a DataFrame shown below:



	pur	gum	...	paperboard	80
0	0	1	...	0	1
1	0	1	...	1	1
...	...	...	...	...	...
38	0	0	...	0	0
39	0	0	...	0	1

You also have the following outputs:

```
>>> bow_df.sum(axis=0)     >>> bow_df.sum(axis=1)     >>> bow_df[0, 'pur']
pur              5         0      21                  0
gum             41         1      22
sugar            2         2      22                  >>> (bow_df['paperboard'] > 0).sum()
                ..                ..                  20
90               4         37     22
paperboard      22         38     10                  >>> bow_df['gum'].sum()
80              20         39     17                  41
Length: 139                Length: 40
```

For each question below, write your answer as an unsimplified math expression (no need to simplify fractions or logarithms) in the space provided, or select "Need more information" if there is not enough information provided to answer the question.

(a) (2 points) What is the TF-IDF for the word `pur` in document 0?

Solution: 0

○ Need more information

(b) (2 points) What is the TF-IDF for the word `gum` in document 0?

Solution:

○ **Need more information**

(c) (3 points) What is the TF-IDF for the word `paperboard` in document 1?

Solution:

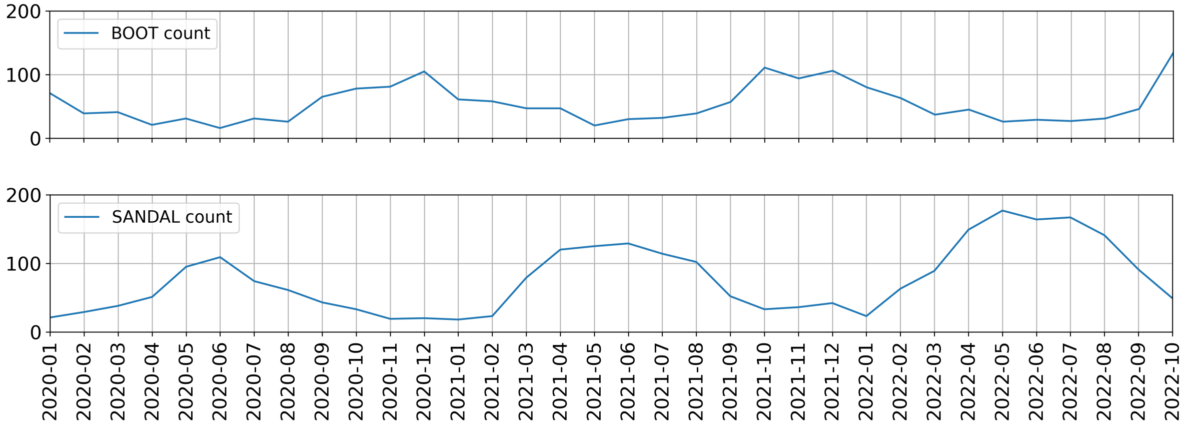
$$\frac{1}{22}\log\left(\frac{40}{20}\right) = \frac{1}{22}$$

○ Need more information

Name: ______________________________

Question 10 . . . . ***17 points***

The two plots below show the total number of boots (top) and sandals (bottom) purchased per month in the `df` table. Assume that there is one data point per month.



For each of the following regression models, use the visualizations shown above to select the value that is *closest* to the fitted model weights. If it is not possible to determine the model weight, select "Not enough info". For the models below: the notation `boot` refers to the number of boots sold; `sandal` refers to the number of sandals sold; `summer=1` is a column with value 1 if the month is between March (03) and August (08), inclusive; and `winter=1` is a column with value 1 if the month is between September (09) and February (02), inclusive.

(a) (2 points) `boot` $= w_0$

w_0: ○ 0 [✓ **50**] ○ 100 ○ Not enough info

(b) (4 points) `boot` $= w_0 + w_1 \cdot$ `sandal`

w_0: ○ -100 ○ -1 ○ 0 ○ 1 [✓ **100**] ○ Not enough info

w_1: ○ -100 [✓ **-1**] ○ 1 ○ 100 ○ Not enough info

(c) (4 points) `boot` $= w_0 + w_1 \cdot ($`summer=1`$)$

w_0: ○ -100 ○ -1 ○ 0 ○ 1 [✓ **100**] ○ Not enough info

w_1: [✓ **-80**] ○ -1 ○ 0 ○ 1 ○ 80 ○ Not enough info

(d) (4 points) `sandal` $= w_0 + w_1 \cdot ($`summer=1`$)$

w_0: ○ -20 ○ -1 ○ 0 ○ 1 [✓ **20**] ○ Not enough info

w_1: ○ -80 ○ -1 ○ 0 ○ 1 [✓ **80**] ○ Not enough info

(e) (3 points) `sandal` $= w_0 + w_1 \cdot ($`summer=1`$) + w_2 \cdot ($`winter=1`$)$

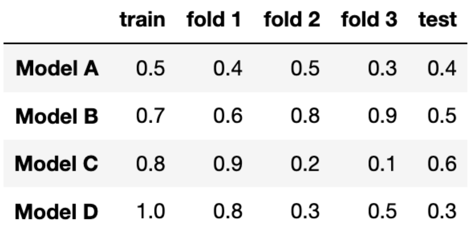
w_0: ○ -20 ○ -1 ○ 0 ○ 1 ○ 20 [✓ **Not enough info**]

w_1: ○ -80 ○ -1 ○ 0 ○ 1 ○ 80 [✓ **Not enough info**]

w_2: ○ -80 ○ -1 ○ 0 ○ 1 ○ 80 [✓ **Not enough info**]

Question 11 . . . . *9 points*

Suppose you fit four different models to predict whether someone has an income greater than $100,000 a year using their purchase history. You split the data into a training and test set and use 3-fold cross-validation. The table below shows all the calculated accuracies for each model (higher accuracy is better).



	train	fold 1	fold 2	fold 3	test
Model A	0.5	0.4	0.5	0.3	0.4
Model B	0.7	0.6	0.8	0.9	0.5
Model C	0.8	0.9	0.2	0.1	0.6
Model D	1.0	0.8	0.3	0.5	0.3

(a) (2 points) Which model has the lowest model bias?
○ Model A ○ Model B ○ Model C ○ **Model D**

(b) (2 points) Which model most severely underfits the data?
○ **Model A** ○ Model B ○ Model C ○ Model D

(c) (2 points) Which model most severely overfits the data?
○ Model A ○ Model B ○ Model C ○ **Model D**

(d) (3 points) Which model should you pick overall?
○ Model A ○ **Model B** ○ Model C ○ Model D

Question 12 . . . . *8 points*

Suppose you fit a decision tree to the training set below, using the features `x0` and `x1` to predict the outcome `y`.

x0	x1	y
0	0	0
0	1	0
0	2	1
1	0	0
1	1	1
2	0	1
2	1	1
3	0	1

Write the first four splitting rules that are created by the decision tree when fitting this training set (using weighted entropy). Assume that the tree is constructed in a depth-first order. If two candidate splits have the same weighted entropy, choose the one that splits on `x0`.

(a) The first splitting rule is: `___(i)___ <= ___(ii)___`

i. (1 point) What goes in blank (i)?
○ **`x0`** ○ `x1`

ii. (1 point) What goes in blank (ii)?
○ `0` ○ **`1`** ○ `2` ○ `3`

(b) The second splitting rule is: `___(i)___ <= ___(ii)___`

i. (1 point) What goes in blank (i)?
○ `x0` ○ **`x1`**

ii. (1 point) What goes in blank (ii)?
○ **`0`** ○ `1` ○ `2` ○ `3`

(c) The third splitting rule is: `___(i)___ <= ___(ii)___`

i. (1 point) What goes in blank (i)?
○ **`x0`** ○ `x1`

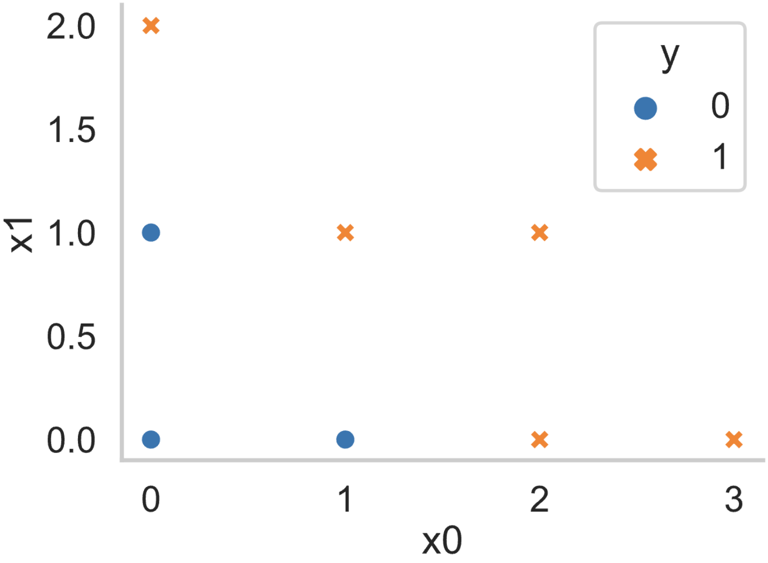
ii. (1 point) What goes in blank (ii)?
○ **`0`** ○ `1` ○ `2` ○ `3`

(d) The fourth splitting rule is: `___(i)___ <= ___(ii)___`

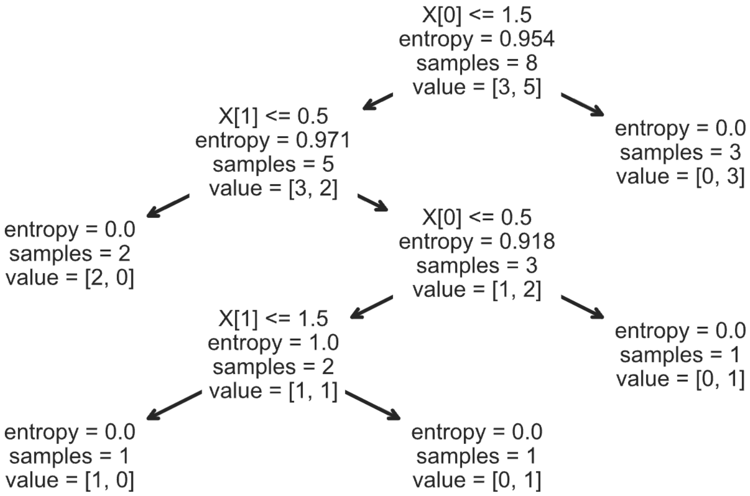
i. (1 point) What goes in blank (i)?
○ `x0` ○ **`x1`**

ii. (1 point) What goes in blank (ii)?
○ `0` ○ **`1`** ○ `2` ○ `3`

Solution: Here is a plot of the training data:

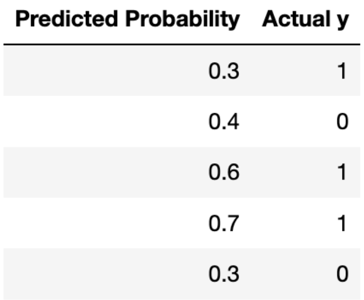


Here is a plot of the fitted tree (from scikit-learn):



Question 13 . . . . **7 *points***

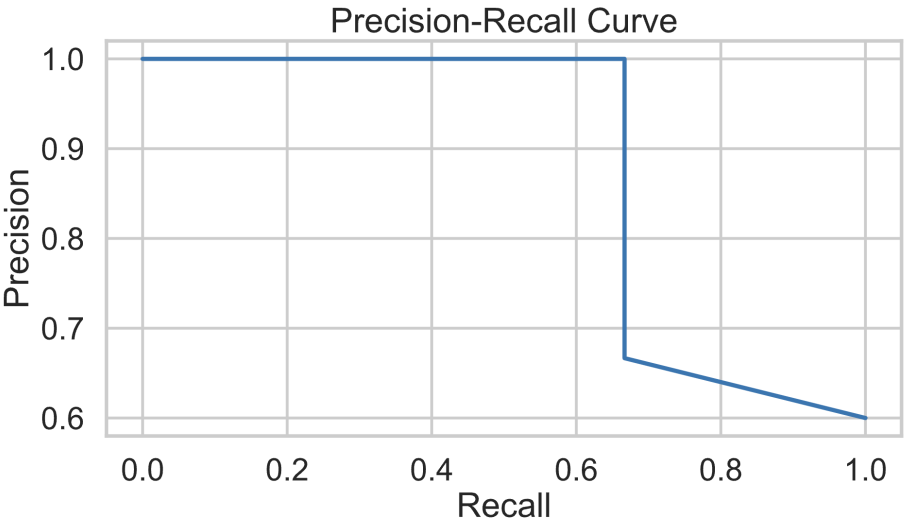
Suppose you fit a logistic regression classifier. The classifier's predictions on a test set of 5 points are shown below, along with the actual labels.



Predicted Probability	Actual y
0.3	1
0.4	0
0.6	1
0.7	1
0.3	0

Recall that for logistic regression, we must also choose a threshold τ to convert the predicted probabilities to predicted labels. For this question, assume that $0 < \tau < 1$. For this question, precision is undefined when the classifier doesn't make any positive predictions (since $\frac{0}{0}$ is undefined). For each question, show your work and draw a box around your final answer in the space provided. Each of your final answers should be a single number.

Solution: The easiest way to solve this problem is to plot the precision-recall curve, which looks like this:



(a) (2 points) What is the **lowest** possible precision for any threshold τ?

Solution: The lowest precision happens when τ is less than 0.3. In this case, the classifier predicts all points are 1, which gives a precision of $\frac{3}{5}$.

(b) (2 points) What is the **lowest** possible recall for any threshold τ?

Solution: The lowest recall happens when τ is greater than 0.7. In this case, the classifier predicts all points are 0, which gives a recall of 0.

(c) (3 points) What is the **highest** possible recall if the classifier achieves a precision of 1?

Solution: If precision is 1, the threshold must be greater than 0.4. Of these thresholds, the recall is greatest when the threshold is between 0.4 and 0.6. In this case, the recall is $\frac{2}{3}$.

Name: ______________________________

Question 14 . . . . **_0 points_**

Optional: Draw a Picture About UCSD Data Science (or use this page for scratch work)