CPSC 340/540 Tutorial 1

Winter 2024 Term 1

T1A: Tuesday 16:00-17:00;
T1C: Thursday 10:00-11:00;
Office Hour: Wednesday 15:00-16:00
Slides can be found at my personal page after T1C.

Yi (Joshua) Ren

https://joshua-ren.github.io/
renyi.joshua@gmail.com
PhD with Danica

Machine Learning:
Learning dynamics, LLM, Compositional Generalization

More helpful on theory

Less helpful on coding

Slides Credit: To various pervious TA's of this course

Topics (review for Assignment 1)

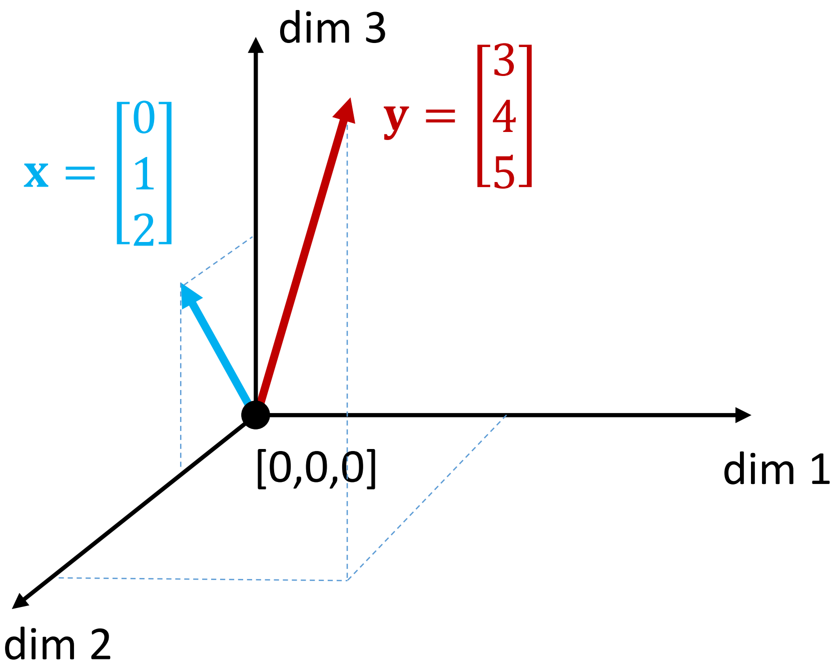
- Linear Algebra (see the world as **vectors**)
- Probability
- Calculus
- Algorithms and Data Structures
- Python

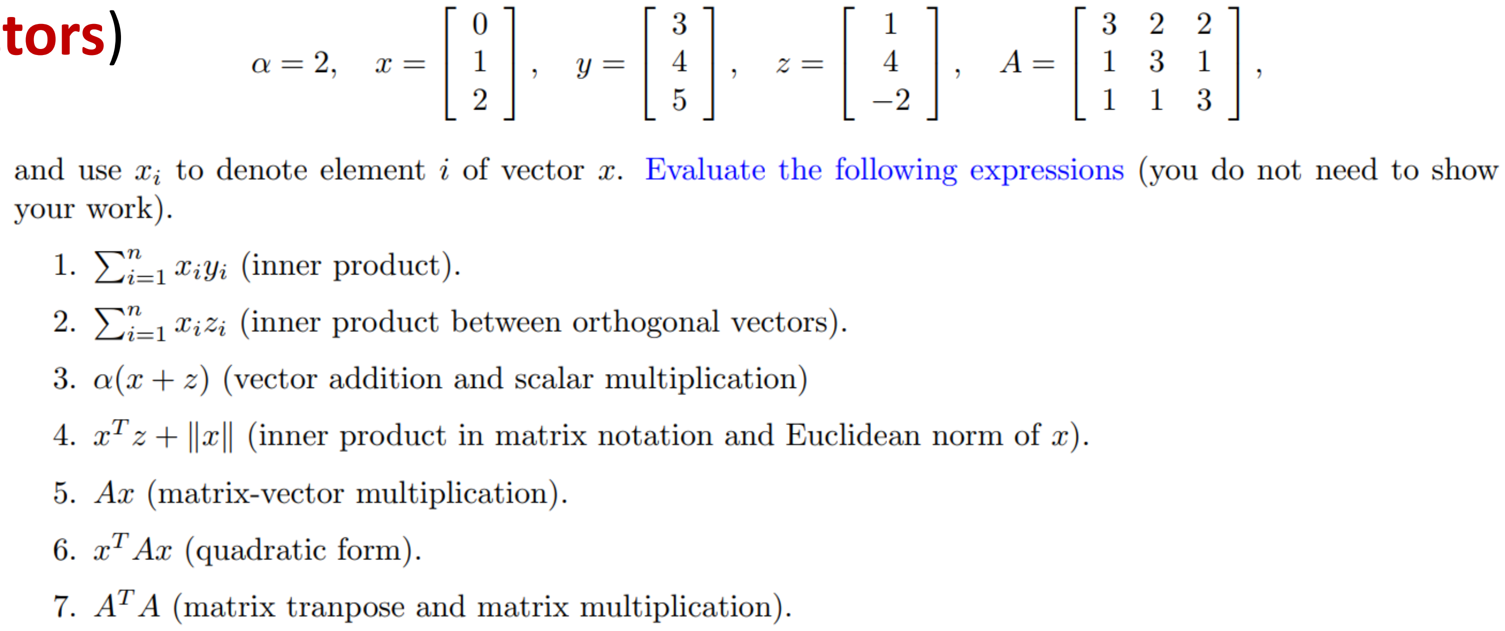
Linear Algebra (see the world as **vectors**)

- Please refer to Mark’s notes:
  https://www.cs.ubc.ca/~schmidtm/Documents/2009_Notes_LinearAlgebra.pdf
- For why we need matrix, strongly suggest:
  MIT’s opencourse for Linear Algebra (Lecture 1, W. Gibert Strang)
- For understanding matrix multiplications in 4 different ways, suggest:
  MIT’s opencourse for Linear Algebra (Lecture 3, W. Gibert Strang)

Let’s have a brief overview of some core concepts and operations.

Linear Algebra (see the world as **vectors**)


dim 3
$\mathbf{y} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$
$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$
[0,0,0]
dim 1
dim 2



$$\alpha = 2, \quad x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}, \quad z = \begin{bmatrix} 1 \\ 4 \\ -2 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix},$$

and use x_i to denote element i of vector x. Evaluate the following expressions (you do not need to show your work).

1. $\sum_{i=1}^{n} x_i y_i$ (inner product).
2. $\sum_{i=1}^{n} x_i z_i$ (inner product between orthogonal vectors).
3. $\alpha(x+z)$ (vector addition and scalar multiplication)
4. $x^T z + \|x\|$ (inner product in matrix notation and Euclidean norm of x).
5. Ax (matrix-vector multiplication).
6. $x^T A x$ (quadratic form).
7. $A^T A$ (matrix tranpose and matrix multiplication).

a. $||\mathbf{x}||_2$
b. $a\mathbf{x}$
c. $\langle \mathbf{x}, \mathbf{y} \rangle, \cos(\mathbf{x}, \mathbf{y}), \langle \mathbf{x}, \mathbf{y} \rangle = 0$
d. $\mathbf{x} + \mathbf{y}; \mathbf{x} - \mathbf{y}$
e. $A\mathbf{x}$

Operation

a. Norm (4)
b. Scalar multiplication (3)
c. Inner product, direction change, orthognal (1, 2)
d. Addition, subtraction (3, parallelogram law)
e. Manipulating the vector (5, egien values, ...)

Intuition

Linear Algebra (matrix multiplication)

- Check the dimension before calculating:

$$\underbrace{\nabla_\theta \log \pi^t(\chi_o)|_{\theta^t}}_{V\times d}\underbrace{\Delta\theta^t}_{d\times 1} = \Big(\underbrace{\nabla_{\mathbf{z}} \log \pi^t(\chi_o)|_{\mathbf{z}^t}}_{V\times V}\underbrace{\nabla_\theta \mathbf{z}^t(\chi_o)|_{\theta^t}}_{V\times d}\Big)\Big(-\eta\underbrace{\nabla_\theta \mathcal{L}(\mathbf{x}_u, \mathbf{y}_u^+, \mathbf{y}_u^-)|_{\theta^t}}_{1\times d}\Big)^{\mathsf{T}}$$

$$= \underbrace{\nabla_{\mathbf{z}} \log \pi^t(\chi_o)|_{\mathbf{z}^t}}_{V\times V}\underbrace{\nabla_\theta \mathbf{z}^t(\chi_o)|_{\theta^t}}_{V\times d}\Big(\underbrace{-\eta\nabla_{[\mathbf{z}^+;\mathbf{z}^-]}\mathcal{L}|_{\mathbf{z}^t}}_{1\times 2V}\underbrace{\left[\nabla_\theta \mathbf{z}^+(\chi_u^+); \nabla_\theta \mathbf{z}^-(\chi_u^-)\right]|_{\theta^t}}_{2V\times d}\Big)^{\mathsf{T}}$$

- Step-wise calculation (quadratic form as an example)

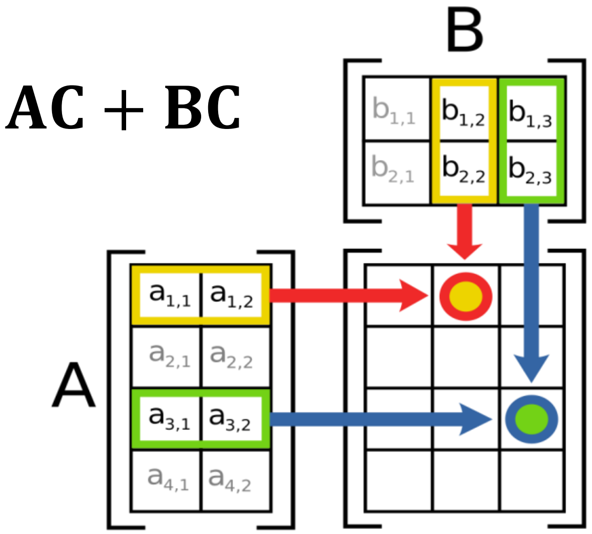
$$\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x}$$

➢ Think about its relationship to $\|\mathbf{x}\|_2^2$

- Transpose, basic laws

$$(\mathbf{ABC})^{\mathbf{T}} = \mathbf{C}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}} \qquad (\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}) \qquad (\mathbf{A}+\mathbf{B})\mathbf{C} = \mathbf{AC}+\mathbf{BC}$$

- Rule of matrix multiplication– elementwise (Understanding 1)

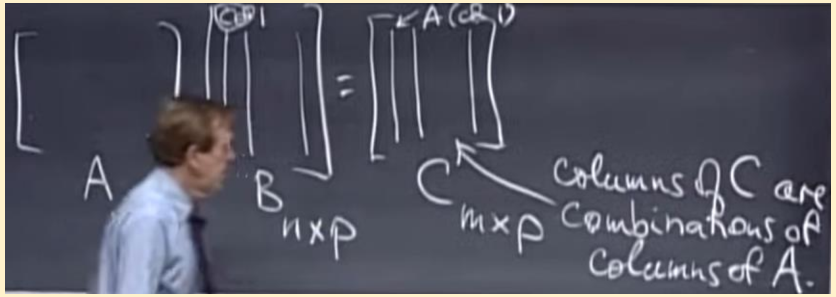


Linear Algebra (matrix multiplication – 3 more useful understandings)

- Column combination (Understanding 2) [White borad drawing]

$$\mathbf{A}\mathbf{b}_i = \mathbf{c}_i$$ Linear combination of columns in $\mathbf{A}$, weighted by x_i

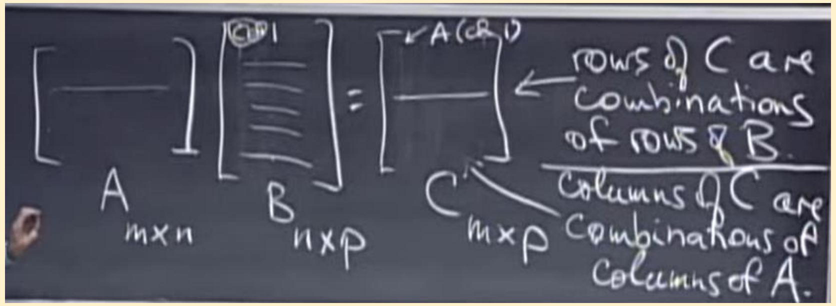
$$\mathbf{AB} = \mathbf{C}$$ Stacking the combined columns from $\mathbf{A}$



- Row combination (Understanding 3) [White borad drawing]

$$\mathbf{a}^j\mathbf{B} = \mathbf{c}_j^T$$ Linear combination of rows in $\mathbf{B}$, weighted by x_i

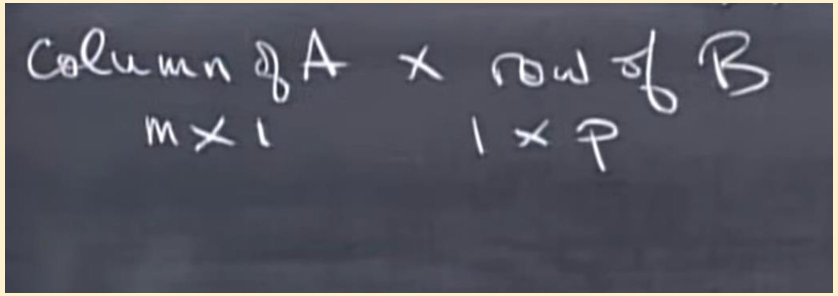
$$\mathbf{AB} = \mathbf{C}$$ Stacking the combined rows from $\mathbf{B}$



rows of C are combinations of rows of B.
A
m×n
B
n×p
C
m×p
Columns of C are Combinations of Columns of A.

- Rank-1 matrix combination (Understanding 4) [White borad drawing]

$$\mathbf{a}_j \cdot \mathbf{b}^i = \boldsymbol{M}_{ij} \qquad \mathbf{AB} = \sum \boldsymbol{M}_{ij}$$



Please refer to MIT's opencourse for Linear Algebra (Lecture 3, W. Gibert Strang) for more details.

Probability (basic rules)

Use Venn diagram to understand (related to set theory):

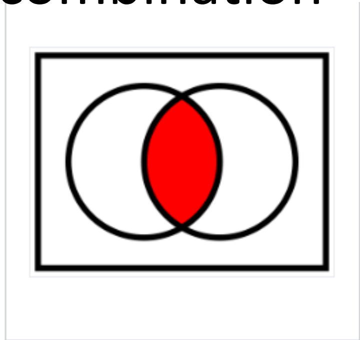
- Event combination
- Conditional distribution
- Entropy, mutual information

Use formulas to understand:

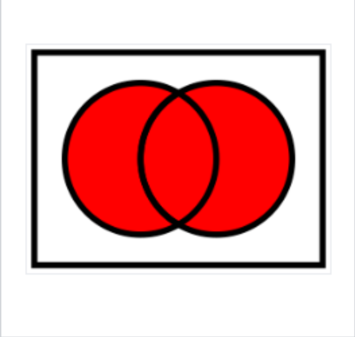
- Independence (a bit counter-intuitive on Venn)
- Bayes
- Marginal distribution
- Total probability (use Venn to find indep. events)

Probability (Veen Diagram)

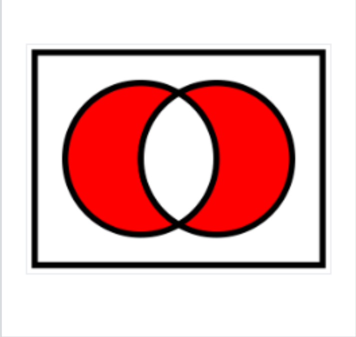
- Event combination



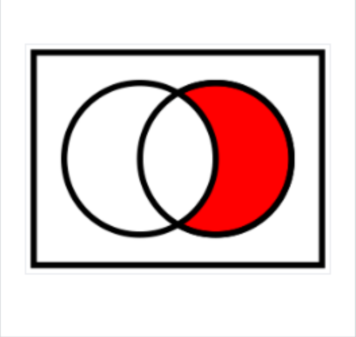
Intersection of two sets $A \cap B$



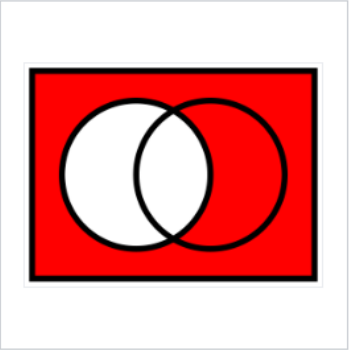
Union of two sets $A \cup B$



Symmetric difference of two sets $A \triangle B$



Relative complement of *A* (left) in *B* (right) $A^c \cap B = B \setminus A$



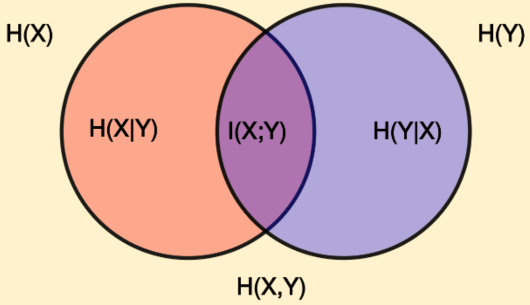
Absolute complement of A in U $A^c = U \setminus A$

- Conditional distribution

P(A) = P(A|U)

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- Mutual information



From: https://en.wikipedia.org/wiki/Venn_diagram

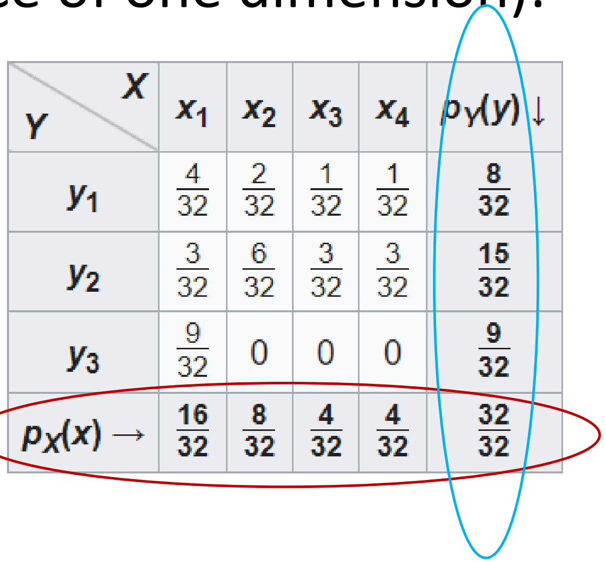
Probability (formulas)

X is salary, Y is age. P(X,Y) can tell us the correlation.
P(X) and P(Y) usually tells us some statists facts of the data

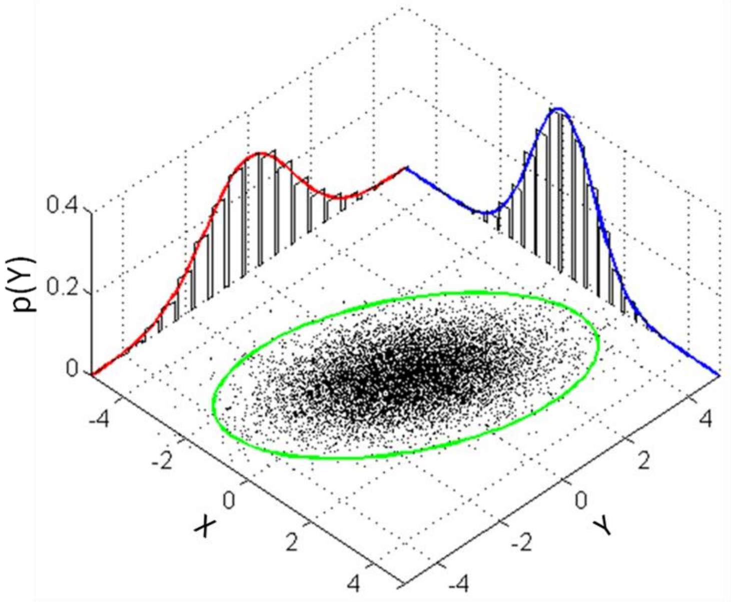
- Marginal (ignore the influence of one dimension):

$$P(X) = \sum_{Y_i} P(X, Y_i)$$

$$P(Y) = \sum_{X_i} P(X_i, Y)$$

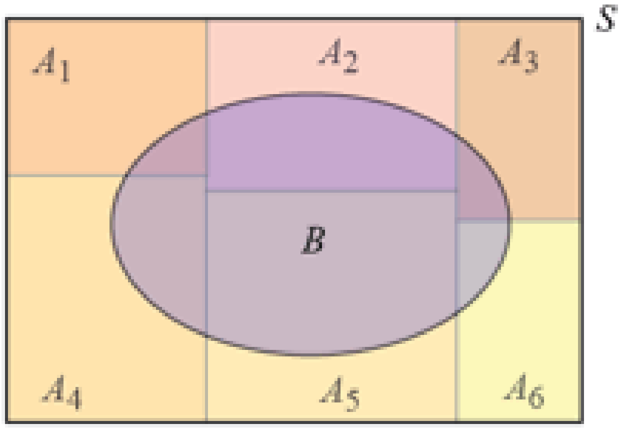


Y \ X	x_1	x_2	x_3	x_4	$p_Y(y)$ ↓
y_1	$\frac{4}{32}$	$\frac{2}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{8}{32}$
y_2	$\frac{3}{32}$	$\frac{6}{32}$	$\frac{3}{32}$	$\frac{3}{32}$	$\frac{15}{32}$
y_3	$\frac{9}{32}$	0	0	0	$\frac{9}{32}$
$p_X(x)$ →	$\frac{16}{32}$	$\frac{8}{32}$	$\frac{4}{32}$	$\frac{4}{32}$	$\frac{32}{32}$



- Total probability (case-by-case):

$$P(B) = P(B \cap A_1) + \cdots P(B \cap A_6)$$
$$= P(B|A_1)P(A_1) + \cdots P(B|A_6)P(A_6)$$



B is the average satisfication level for CPSC 340's tutorial sessions.
A1~A6 are T1A,..., T1G; P(A1) is how many students in T1A.

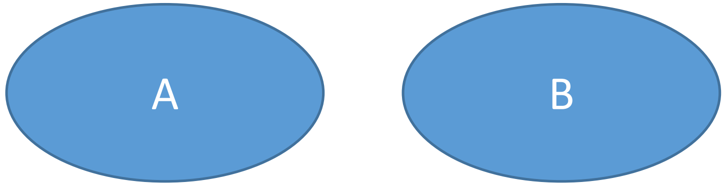
Probability (formulas)

- Independence:

$$A \perp B \Rightarrow P(A|B) = P(A)$$

$$P(A \cap B) = P(A)P(B)$$

Quick question: where A and B are independent from the following Venn graph?



- Bayesian rule (why we need it?):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Hard to know

Medical knowledge

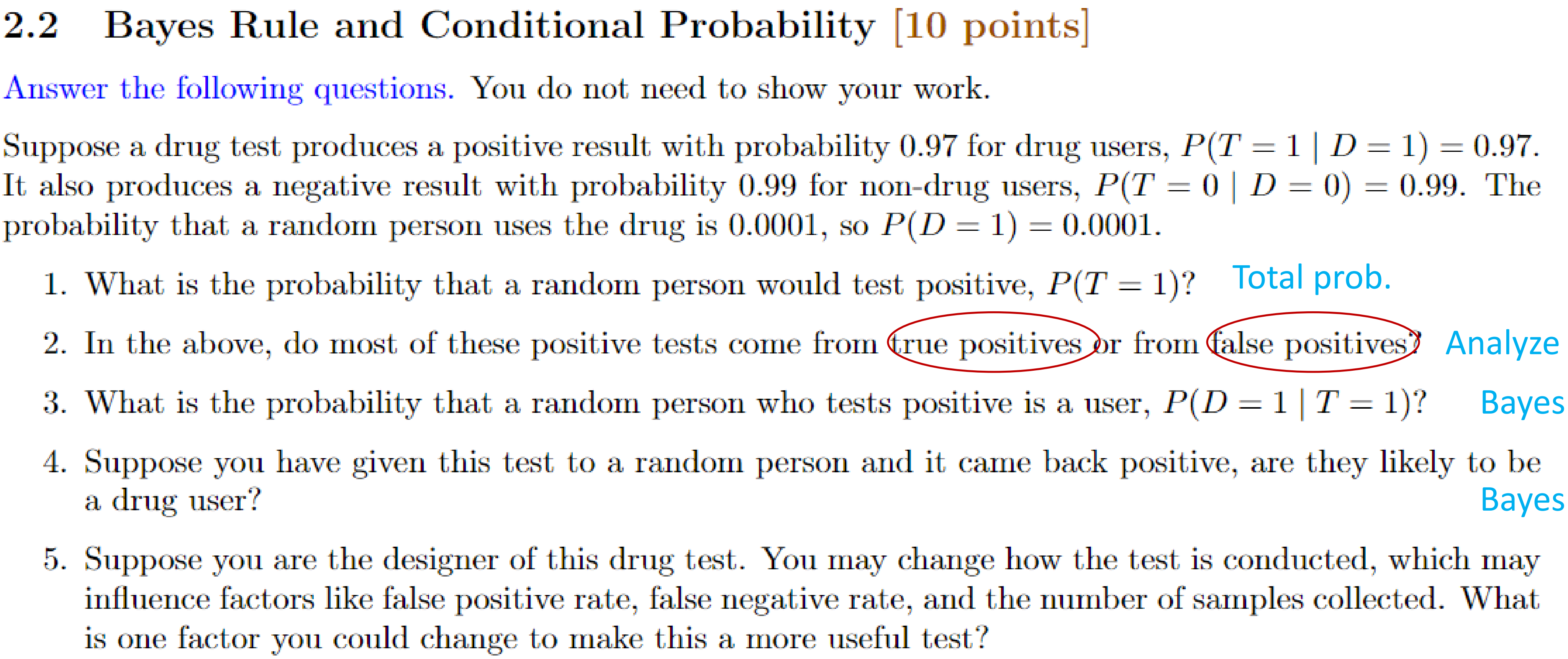
Statistics

$$P(\text{Cancer}|\text{Symptoms}) = \frac{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer})}{P(\text{Symptoms})}$$

Think about when P(Symptom) is common or rare.

Statistics

Probability (putting them together for Assignment 1-2.2)



2.2 Bayes Rule and Conditional Probability [10 points]

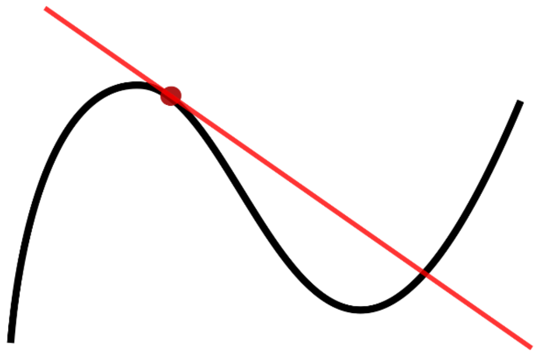
Answer the following questions. You do not need to show your work.

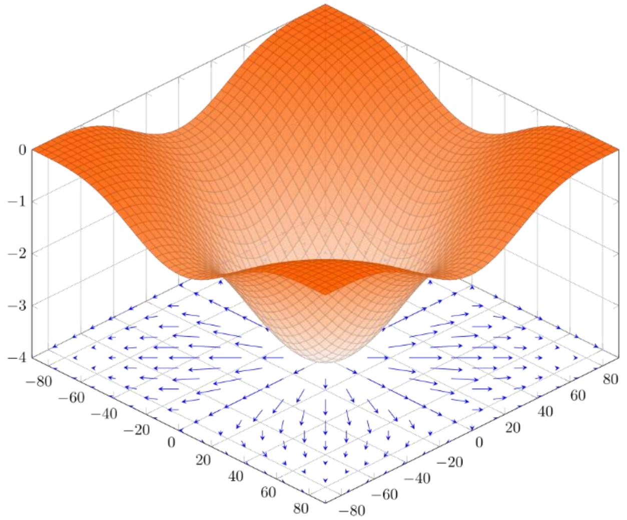
Suppose a drug test produces a positive result with probability 0.97 for drug users, $P(T = 1 \mid D = 1) = 0.97$. It also produces a negative result with probability 0.99 for non-drug users, $P(T = 0 \mid D = 0) = 0.99$. The probability that a random person uses the drug is 0.0001, so $P(D = 1) = 0.0001$.

1. What is the probability that a random person would test positive, $P(T = 1)$? Total prob.
2. In the above, do most of these positive tests come from true positives or from false positives? Analyze the decomposed form
3. What is the probability that a random person who tests positive is a user, $P(D = 1 \mid T = 1)$? Bayesian
4. Suppose you have given this test to a random person and it came back positive, are they likely to be a drug user? Bayesian
5. Suppose you are the designer of this drug test. You may change how the test is conducted, which may influence factors like false positive rate, false negative rate, and the number of samples collected. What is one factor you could change to make this a more useful test?

Calculus (to calculate something)

- Derivatives and Gradients





$$f = x_1^2 + x_2; \quad \frac{\partial f}{\partial x_1} = 2x_1; \quad \frac{\partial f}{\partial x_2} = 1$$

$$f = x^2$$

$$f' = 2x$$

$$\nabla f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 1 \end{bmatrix}$$

Calculus (to calculate something)

- Gradient calculation of matrix form **(be very careful!)**

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{a}^{\mathbf{T}}\mathbf{x} \qquad\qquad f(x) = ax$$

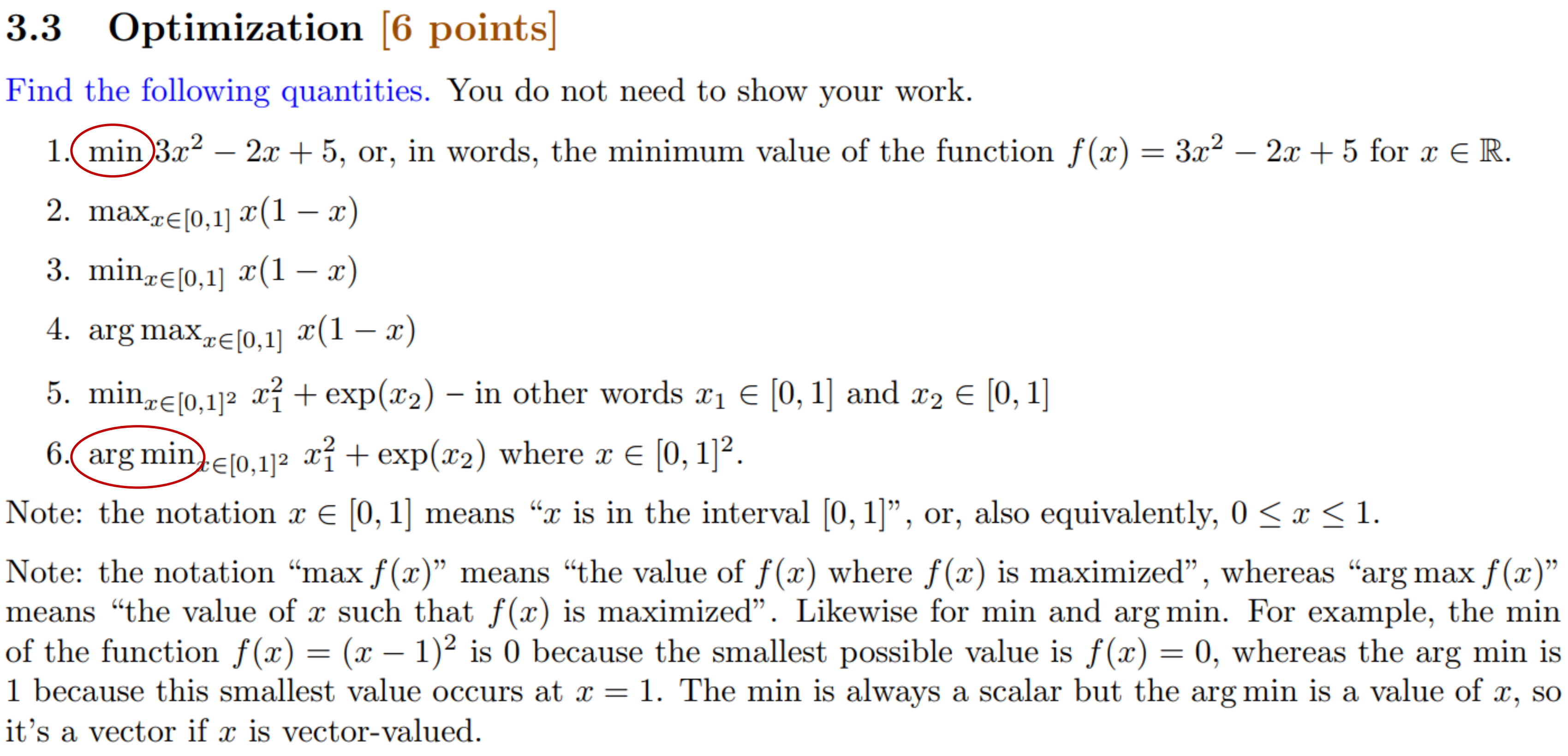
$$\boldsymbol{f}(\mathbf{x}) = \mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x} \qquad\qquad f(x) = ax^2$$

- Chain rule (**very common in machine learning**)

$$\boldsymbol{f}(\mathbf{x}) = \log(\mathbf{a}^{\mathbf{T}}\mathbf{x}) \qquad\qquad \frac{df_1}{dx} = \frac{df_1}{df_2}\frac{df_2}{df_3}\cdots\frac{df_n}{dx}.$$

$$[\mathcal{G}^t_{\text{DPO}}]_l = \frac{\partial \mathcal{L}_{\text{DPO}}}{\partial a}\frac{\partial a}{\partial b}\nabla_{\pi} b|_{\pi_{\theta^t}}\nabla_{\mathbf{z}_l}\pi^t|_{\mathbf{z}_l^t}$$

Calculus (to calculate something)



3.3 Optimization [6 points]

Find the following quantities. You do not need to show your work.

1. $\min 3x^2 - 2x + 5$, or, in words, the minimum value of the function $f(x) = 3x^2 - 2x + 5$ for $x \in \mathbb{R}$.
2. $\max_{x\in[0,1]} x(1-x)$
3. $\min_{x\in[0,1]} x(1-x)$
4. $\arg\max_{x\in[0,1]} x(1-x)$
5. $\min_{x\in[0,1]^2} x_1^2 + \exp(x_2)$ – in other words $x_1 \in [0,1]$ and $x_2 \in [0,1]$
6. $\arg\min_{x\in[0,1]^2} x_1^2 + \exp(x_2)$ where $x \in [0,1]^2$.

Note: the notation $x \in [0,1]$ means "x is in the interval $[0,1]$", or, also equivalently, $0 \le x \le 1$.

Note: the notation "$\max f(x)$" means "the value of $f(x)$ where $f(x)$ is maximized", whereas "$\arg\max f(x)$" means "the value of x such that $f(x)$ is maximized". Likewise for min and arg min. For example, the min of the function $f(x) = (x-1)^2$ is 0 because the smallest possible value is $f(x) = 0$, whereas the arg min is 1 because this smallest value occurs at $x = 1$. The min is always a scalar but the arg min is a value of x, so it's a vector if x is vector-valued.

What's the relationships between gradients and optimization?

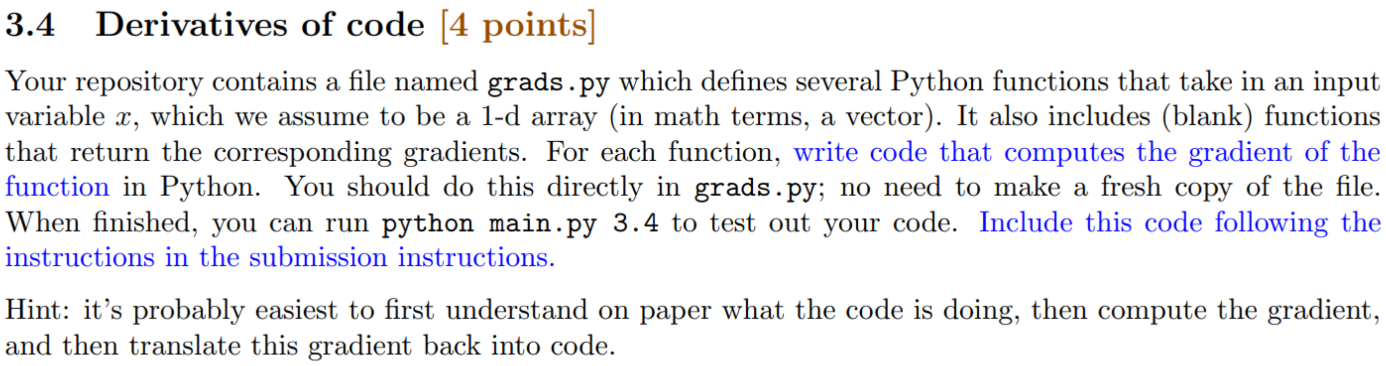
- Q1: for a convex function, when optimum is reached?
- Q2: for a non-convex function, what does it mean by satisfying the same requirement in Q1?
- Q3: start from an arbitrary point at the function, how to find a optimum?

Calculus (to calculate something)

- Drawing the curves out are always helpful

https://www.geogebra.org/calculator
https://www.wolframalpha.com/
Use matplotlib and numpy

- For assignment 3.4, nothing specical, practice using python



3.4 Derivatives of code [4 points]

Your repository contains a file named `grads.py` which defines several Python functions that take in an input variable x, which we assume to be a 1-d array (in math terms, a vector). It also includes (blank) functions that return the corresponding gradients. For each function, write code that computes the gradient of the function in Python. You should do this directly in `grads.py`; no need to make a fresh copy of the file. When finished, you can run `python main.py 3.4` to test out your code. Include this code following the instructions in the submission instructions.

Hint: it's probably easiest to first understand on paper what the code is doing, then compute the gradient, and then translate this gradient back into code.

- For pytorch, **autograd**. Very convinent, but usually makes u forget how to calculate the gradient manually ...

- For theoretical analysis, always very important to be good at it!

Python

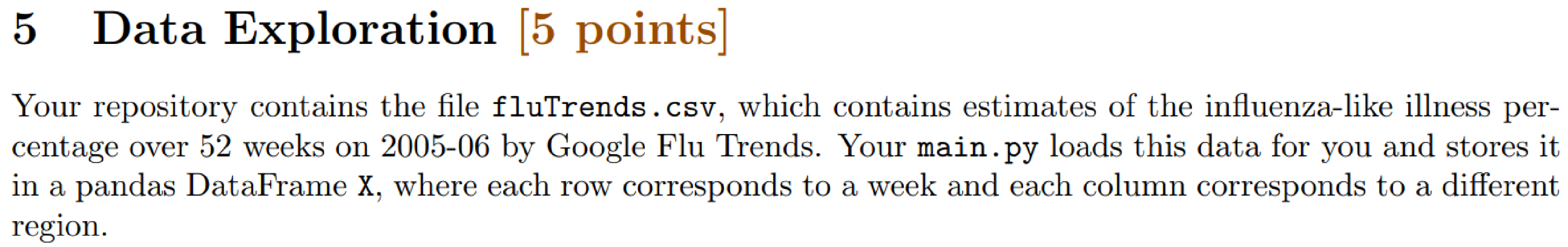
Some resources

- https://cs231n.github.io/python-numpy-tutorial/
- Tutorial 1 python notebooks (two attached with slides)

Suggested packages

- Numpy (good for matrix manipulation)
- Matplotlib (good for visualizing results)
- Pandas (good for tabular data management)
- Scipy (good for basic machine learning problems)
- Pytorch (good for neural networks)
  - Torchvision
  - Transformer
  - Many other good tools for DNN

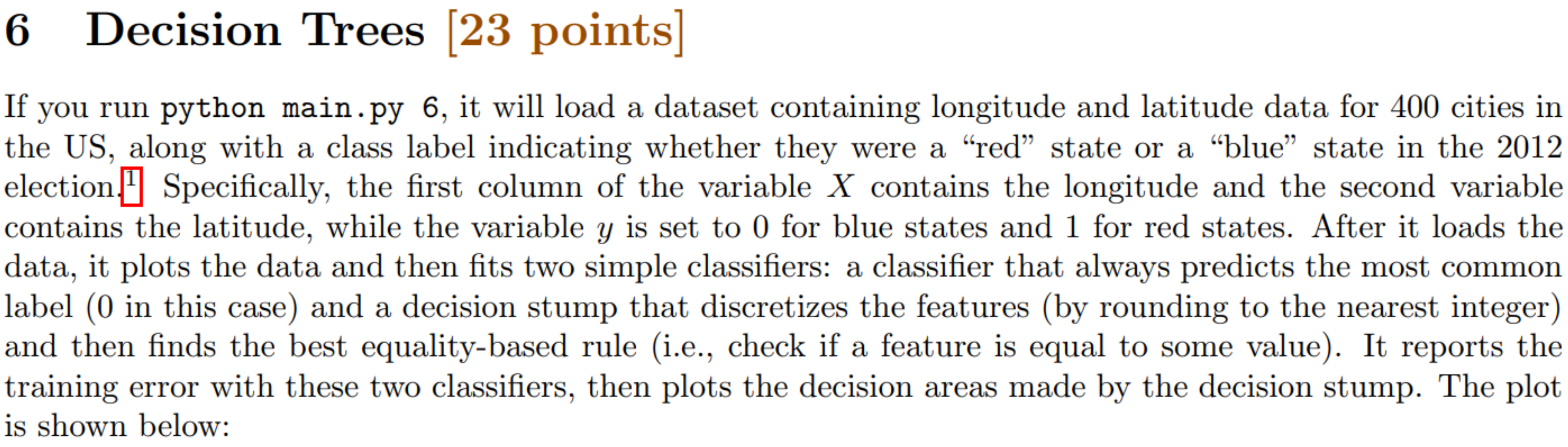
- We can discuss together if you want



5 Data Exploration [5 points]

Your repository contains the file `fluTrends.csv`, which contains estimates of the influenza-like illness percentage over 52 weeks on 2005-06 by Google Flu Trends. Your `main.py` loads this data for you and stores it in a pandas DataFrame `X`, where each row corresponds to a week and each column corresponds to a different region.

- Not quite familar with decision tree, but we can learn together



6 Decision Trees [23 points]

If you run `python main.py 6`, it will load a dataset containing longitude and latitude data for 400 cities in the US, along with a class label indicating whether they were a “red” state or a “blue” state in the 2012 election.[1] Specifically, the first column of the variable X contains the longitude and the second variable contains the latitude, while the variable y is set to 0 for blue states and 1 for red states. After it loads the data, it plots the data and then fits two simple classifiers: a classifier that always predicts the most common label (0 in this case) and a decision stump that discretizes the features (by rounding to the nearest integer) and then finds the best equality-based rule (i.e., check if a feature is equal to some value). It reports the training error with these two classifiers, then plots the decision areas made by the decision stump. The plot is shown below:

Thanks for your time!
Questions?