Week: 12 Date: 4/22/21

|  |
| --- |
| 15-110 Recitation Week 12 |

# **Reminders**

* Check 6-1 is due tomorrow, Friday 4/23 @ 12 PM EST. Make sure to come to OH if you need help!
* Check 6-1 revisions are due next Wednesday 4/28 @ 12 PM EST.

# **Overview**

* ML Fast Facts
* Simulation (MVC)
* Working with data
* HW6 Group OH

|  |
| --- |
| Problems |

# **ML FAST FACTS**

What is the difference between classification, regression, and clustering?

Answer/Notes

What is the difference between supervised and unsupervised learning?

Answer/Notes

T/F: A common step in machine learning is training on testing data.

Answer

T/F: Naive Bayes is a classification algorithm that uses conditional probabilities to determine the class labelling of data points.

Answer

# **SIMULATIONS!**

Consider this setup from lecture:

def makeModel(data):

data["cx"] = 200

data["cy"] = 200

data["size"] = 50

data["color"] = “red”

def makeView(data, canvas):

color = data["color"]

canvas.create\_oval(data["cx"] - data["size"], data["cy"] - data["size"], data["cx"] + data["size"], data["cy"] + data["size"], fill=color)

Write a function mousePressed(data, event)that if the mouse is pressed within the circle, will move the circle to a random x and y coordinate on the canvas. Note that you can use random.randint(low, high) which generates a random number between low and high inclusive.

code

# **WORKING WITH DATA**

Your Global Business professor gives you some sample corporate data to analyze more thoroughly for homework. You realize there is way too much data to draw graphs for by hand and want to try using your newfound MatPlotLib skills to write some code to make graphs for you! Make sure to download the starter code and csv files from the 15110 website and store them in the same folder.

You will be implementing the following items:

1. First, we want to read in the corporate data from the csv file. A function readCSVFile is already provided for you, just make a call to it and save the function output in a variable *data*.
2. Write a function **departmentNameDict** that stores the information in an input list as a dictionary. The input list is a 2D list where each inner list contains 4 elements: department, department\_name, location\_id, and department\_expenses. The output dictionary should be created so that for each inner list, there is a key of that department\_name and an associated value which is the 2 element list of the corresponding lcation\_id and department\_expenses.
3. Write a function **departmentInfo** that takes in a dictionary like the one outputted above and a list of departments, and finds the mean expenses across the given departments and the most common location among the departments and prints them to the user. You may want to import a package to help you with this function!
4. Write a function **expensesGraphs**, that uses the MatPlotLib package to display 2 graphs from an input dictionary of the same form as the one created in **departmentNameDict**:
	1. The first graph should display the expenses of departments within each location in a scatterplot
	2. The second graph should only display the expenses of departments that spent over $100,000 in a bar graph

Put solution code in the .py stater file!

# **HW 6 CHECK-INS**

Individually meet with your TA(s) to check in and explain how you are feeling about HW 6 - are you on track to submit by the deadline tomorrow, do you want some help catching up, do you plan to catch up over the weekend, are you currently dealing with a specific bug, etc.