Lecture 1: Introduction to CPSC 330

Varada Kolhatkar

Learning outcomes

From this lecture, you will be able to

QR code of CPSC 330 website

- Course Jupyter book: https://ubc-cs.github.io/cpsc330-2024W1
- Course GitHub repository: https://github.com/UBC-CS/cpsc330-2024W1

Introductions

Meet your instructor

- Varada Kolhatkar [ʋərəda kɔːlɦəʈkər]
- You can call me Varada, **V**, or **Ada**.
- I am an Assistant Professor of Teaching in the Department of Computer Science.
- I did my Ph.D. in Computational Linguistics at the University of Toronto.
- I primarily teach machine learning courses in the Master of Data Science (MDS) program.
- Contact information
  - Email: kvarada@cs.ubc.ca
  - Office: ICCS 237

Meet Eva (a fictitious persona)!

Eva is among one of you. She has some experience in Python programming. She knows machine learning as a buzz word. During her recent internship, she has developed some interest and curiosity in the field. She wants to learn what is it and how to use it. She is a curious person and usually has a lot of questions!

You all

- Introduce yourself to your neighbour.
- Since we're going to spend the semester with each other, I would like to know you a bit better.
- Please fill out Getting to know you survey when you get a chance.

Asking questions during class

You are welcome to ask questions by raising your hand. There is also a reflection Google Document for this course for your questions/comments/reflections. It will be great if you can write about your takeaways, struggle points, and general comments in this document so that I'll try to address those points in the next lecture.

Activity 1: https://shorturl.at/CteOU

- Write your answers to the questions below in this Google doc: https://shorturl.at/CteOU
- What do you know about machine learning?
- What would you like to get out this course?
- Are there any particular topics or aspects of this course that you are especially excited or anxious about? Why?

What is Machine Learning (ML)?

Spam prediction

- Suppose you are given some data with labeled spam and non-spam messages

Code | Output

```
sms_df = pd.read_csv(DATA_DIR + "spam.csv", encoding="latin-1")
sms_df = sms_df.drop(columns = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"])
sms_df = sms_df.rename(columns={"v1": "target", "v2": "sms"})
train_df, test_df = train_test_split(sms_df, test_size=0.10, random_state=4
```

Traditional programming vs. ML

Let’s train a model

- There are several packages that help us perform machine learning.

```
X_train, y_train = train_df["sms"], train_df["target"]
X_test, y_test = test_df["sms"], test_df["target"]
clf = make_pipeline(CountVectorizer(max_features=5000), LogisticRegression(
clf.fit(X_train, y_train); # Training the model
```

Unseen messages

- Now use the trained model to predict targets of unseen messages:

	sms
3245	Funny fact Nobody teaches volcanoes 2 erupt, tsunamis 2 arise, hurricanes 2 sway aroundn no 1 teaches hw 2 choose a wife Natural disasters just happens
944	I sent my scores to sophas and i had to do secondary application for a few schools. I think if you are thinking of applying, do a research on cost also. Contact joke ogunrinde, her school is one m...

	sms
1044	We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6, LS15HB 150p
2484	Only if you promise your getting out as SOON as you can. And you'll text me in the morning to let me know you made it in ok.

Predicting on unseen data

The model is accurately predicting labels for the unseen text messages above!

	sms	spam_predictions
3245	Funny fact Nobody teaches volcanoes 2 erupt, tsunamis 2 arise, hurricanes 2 sway aroundn no 1 teaches hw 2 choose a wife Natural disasters just happens	ham
944	I sent my scores to sophas and i had to do secondary application for a few schools. I think if you are thinking of applying, do a research on cost also. Contact joke ogunrinde, her school is one me the less expensive ones	ham
1044	We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6, LS15HB 150p	spam
2484	Only if you promise your getting out as SOON as you can. And you'll text me in the morning to let me know you made it in ok.	ham

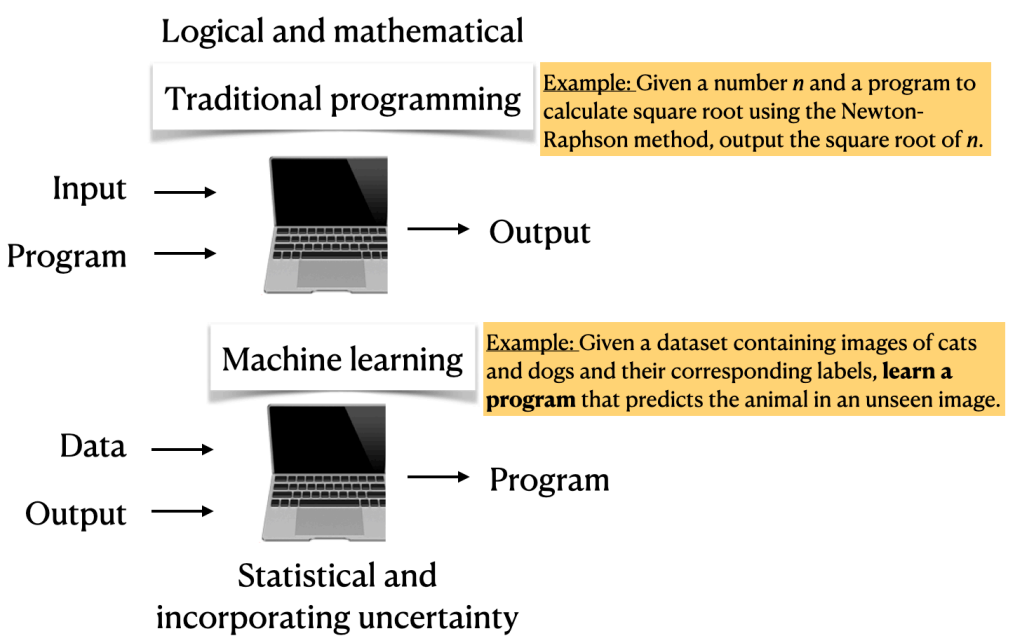
A different way to solve problems

Machine learning uses computer programs to model data. It can be used to extract hidden patterns, make predictions in new situation, or generate novel content.

> A field of study that gives computers the ability to learn without being explicitly programmed.
> – Arthur Samuel (1959)

ML vs. traditional programming

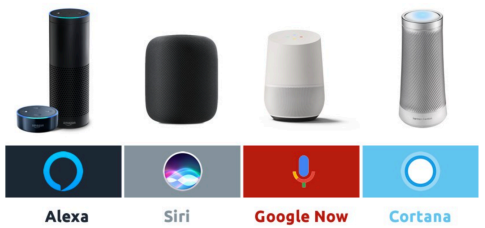
- With machine learning, you're likely to
  - Save time
  - Customize and scale products



Prevalence of ML

Let's look at some examples.

Voice assistants

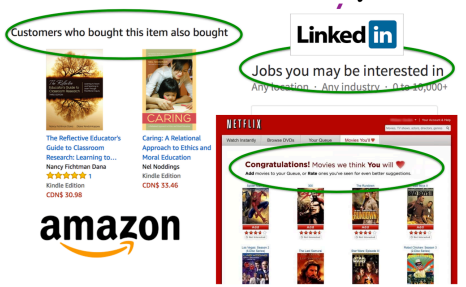


Google news

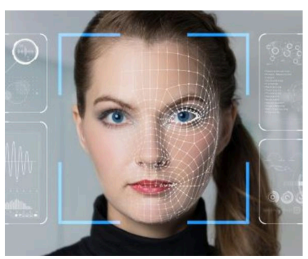
Armed man who broke into Trudeau residence charged with threatening to kill or injure PM
The Guardian · 1 hour ago

- Corey Hurren, alleged Rideau Hall intruder, threatened Trudeau: RCMP officer
  Global News · 4 hours ago
- Corey Hurren had multiple firearms, uttered threat against Trudeau, court documents allege
  CBC.ca · 2 hours ago
- Man arrested near Rideau Hall had several weapons, threatened PM Trudeau: RCMP
  CTV News · 22 minutes ago

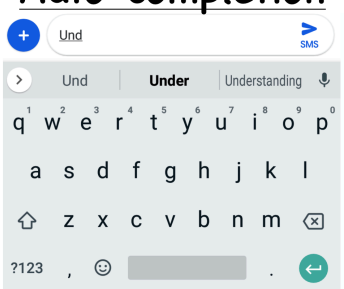
Recommendation systems



Face recognition

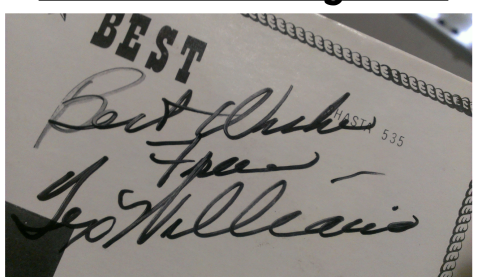


Auto-completion



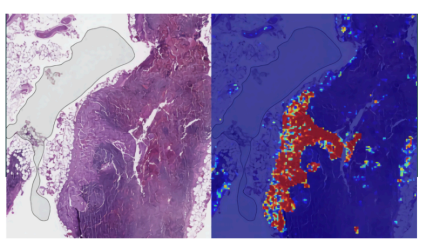
Stock market prediction

Character recognition



Self-driving car

Cancer diagnosis



Drug discovery

AlphaGo

Activity: For what type of problems ML is appropriate? (~5 mins)

Discuss with your neighbour for which of the following problems you would use machine learning

- Finding a list of prime numbers up to a limit
- Given an image, automatically identifying and labeling objects in the image
- Finding the distance between two nodes in a graph

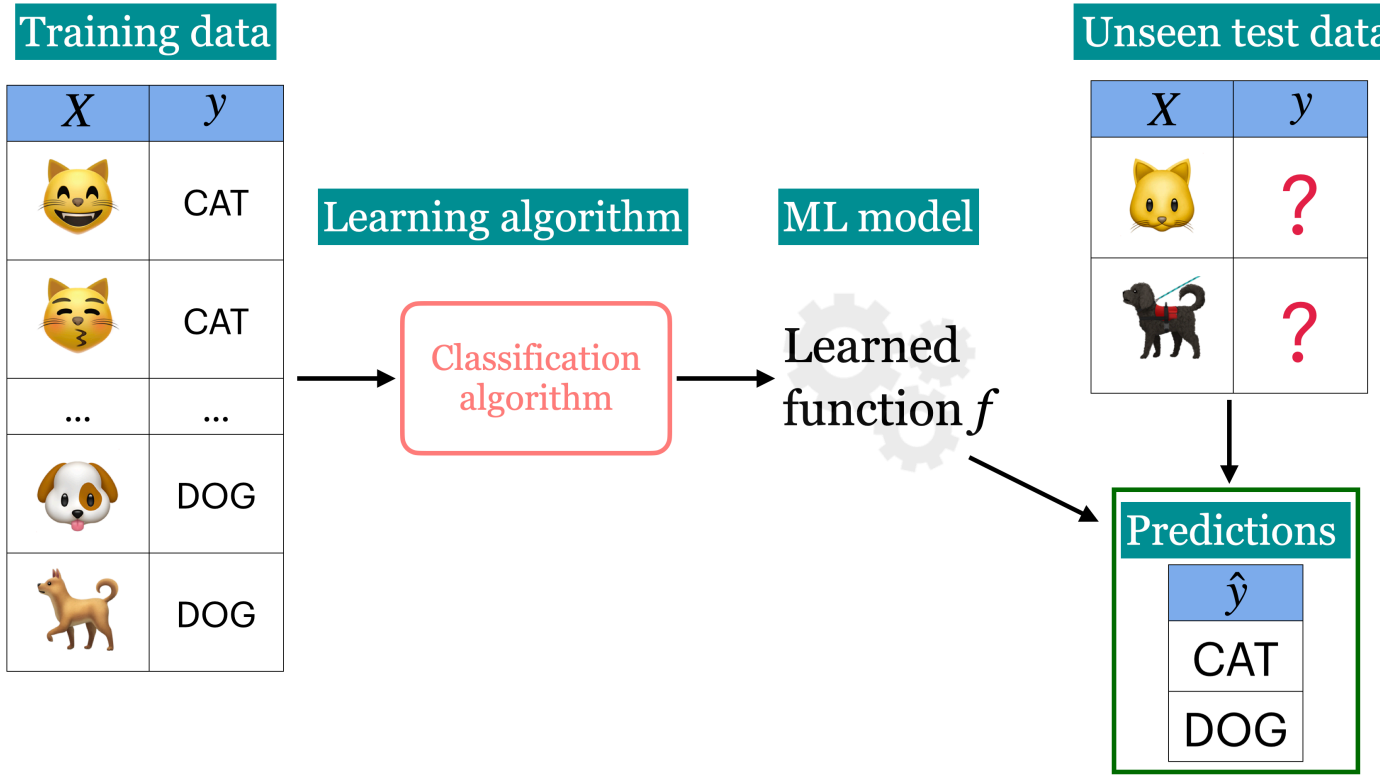
Types of machine learning

Here are some typical learning problems.

- **Supervised learning** (Gmail spam filtering)
  - Training a model from input data and its corresponding targets to predict targets for new examples.
- Unsupervised learning (Google News)
  - Training a model to find patterns in a dataset, typically an unlabeled dataset.
- Reinforcement learning (AlphaGo)
  - A family of algorithms for finding suitable actions to take in a given situation in order to maximize a reward.
- Recommendation systems (Amazon item recommendation system)
  - Predict the “rating” or “preference” a user would give to

What is supervised learning?

- Training data comprises a set of observations (X) and their corresponding targets (y).
- We wish to find a model function f that relates X to y.
- We use the model function to predict targets of new examples.


Training data
X
y
CAT
CAT
...
...
DOG
DOG
Learning algorithm
Classification algorithm
ML model
Learned function f
Unseen test data
X
y
?
?
Predictions
ŷ
CAT
DOG

🤔 Eva's questions

At this point, Eva is wondering about many questions.

- How are we exactly “learning” whether a message is spam and ham?
- Are we expected to get correct predictions for all possible messages? How does it predict the label for a message it has not seen before?
- What if the model mis-labels an unseen example? For instance, what if the model incorrectly predicts a non-spam as a spam? What would be the consequences?
- How do we measure the success or failure of spam identification?
- If you want to use this model in the wild, how do you know how reliable it is?
- Would it be useful to know how confident the model is about

Predicting labels of a given image

- We can also use machine learning to predict labels of given images using a technique called **transfer learning**.

Predicting housing prices

Suppose we want to predict housing prices given a number of attributes associated with houses. The target here is **continuous** and not **discrete**.

target	bedrooms	bathrooms	sqft_living	sqft_lot	f
509000.0	2	1.50	1930	3521	1
675000.0	5	2.75	2570	12906	1
420000.0	3	1.00	1150	5120	1
680000.0	8	2.75	2530	4800	1
357823.0	3	1.50	1240	9196	1

Building a regression model

```
1 from lightgbm.sklearn import LGBMRegressor
2
3 X_train, y_train = train_df.drop(columns= ["target"]), train_df["target"]
4 X_test, y_test = test_df.drop(columns= ["target"]), train_df["target"]
5
6 model = LGBMRegressor()
7 model.fit(X_train, y_train);
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testi
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2333
[LightGBM] [Info] Number of data points in the train set: 17290, number of used
[LightGBM] [Info] Start training from score 539762.702545
```

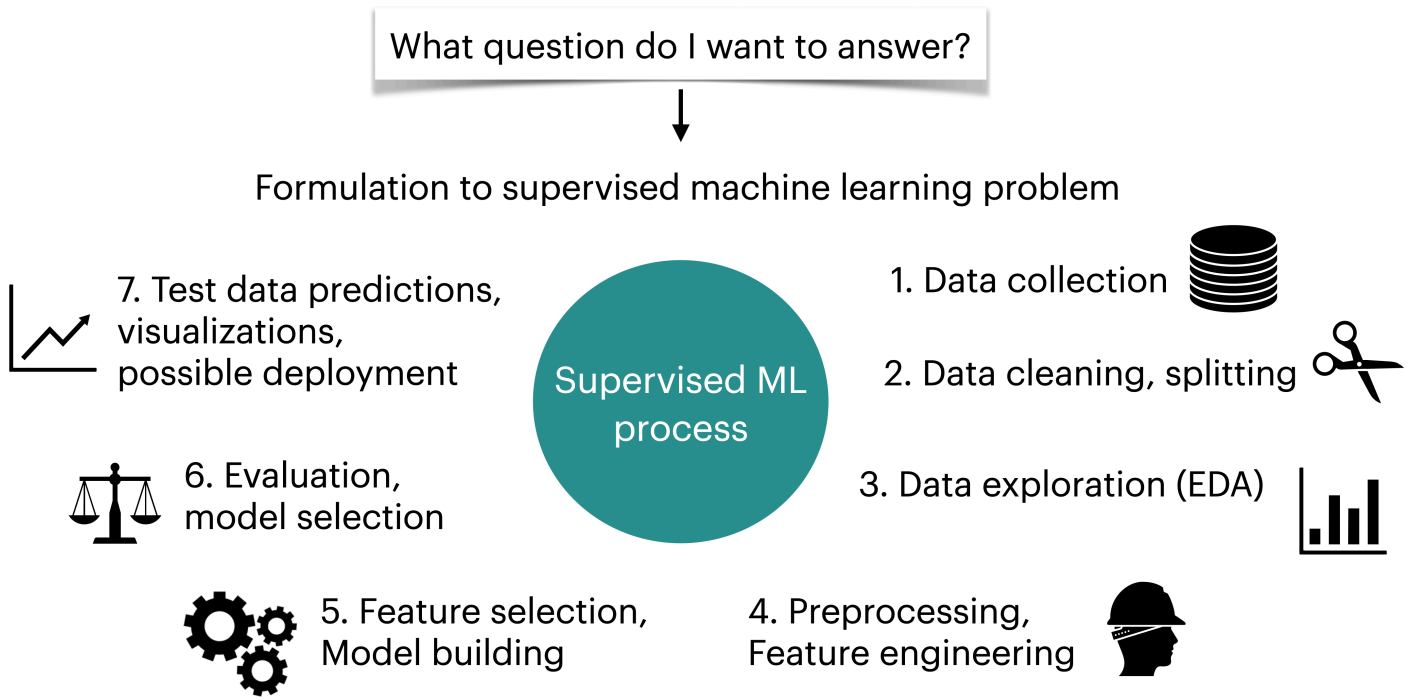
Predicting prices of unseen houses

Predicted_target	bedrooms	bathrooms	sqft_living	sq
345831.740542	4	2.25	2130	80
601042.018745	3	2.50	2210	76
311310.186024	4	1.50	1800	95
597555.592401	3	2.50	1580	13

We are predicting continuous values here as apposed to discrete values in `spam` vs. `ham` example.

Machine learning workflow

Supervised machine learning is quite flexible; it can be used on a variety of problems and different kinds of data. Here is a typical workflow of a supervised machine learning systems.



We will build machine learning pipelines in this course, focusing on some of the steps above.

? ? Questions for you

iClicker cloud join link: https://join.iclicker.com/VYFJ

Select all of the following statements which are True (iClicker)

- a. Predicting spam is an example of machine learning.
- b. Predicting housing prices is not an example of machine learning.
- c. For problems such as spelling correction, translation, face recognition, spam identification, if you are a domain expert, it's usually faster and scalable to come up with a robust set of rules manually rather than building a machine learning model.
- d. If you are asked to write a program to find all prime numbers up to a limit, it is better to implement one of the algorithms for doing so rather than using machine learning.
- e. Google News is likely be using machine learning to

Surveys

- Please complete the “Getting to know you” survey on Canvas.
- Also, please complete the anonymous restaurant survey on Qualtrics here.
  - We will try to analyze this data set in the coming weeks.

About this course

Important

Course website: https://github.com/UBC-CS/cpsc330-2024W1 is the most important link. Please read everything on this GitHub page!

Important

Make sure you go through the syllabus thoroughly and complete the syllabus quiz before Monday, Sept 19th at 11:59pm.

CPSC 330 vs. 340

Read https://github.com/UBC-CS/cpsc330-2024W1/blob/main/docs/330_vs_340.md which explains the difference between two courses.

TLDR:

- 340: how do ML models work?
- 330: how do I use ML models?
- CPSC 340 has many prerequisites.
- CPSC 340 goes deeper but has a more narrow scope.
- I think CPSC 330 will be more useful if you just plan to apply basic ML.

Registration, waitlist and prerequisites

Important

Please go through this document carefully before contacting your instructors about these issues. Even then, we are very unlikely to be able to help with registration, waitlist or prerequisite issues.

Lecture format

- In person lectures T/Th.
- Sometimes there will be videos to watch before lecture. You will find the list of pre-watch videos in the schedule on the course webpage.
- We will also try to work on some questions and exercises together during the class.
- All materials will be posted in this GitHub repository.
- Weekly tutorials will be **office hour format** run by the TAs and are **completely optional**.
  - You do not need to be registered in a tutorial.
  - You can attend whatever tutorials or office hours your want, regardless of in which/whether you're registered.

Home work assignments

- First homework assignment is due **this coming Tuesday**, September 10, midnight. This is a relatively straightforward assignment on Python. If you struggle with this assignment then that could be a sign that you will struggle later on in the course.
- You must do the first two homework assignments on your own.

Exams

- We'll have two self-scheduled midterms and one final in Computer-based Testing Facility (CBTF).

Course calendar

Here is our course Calendar. Make sure you check it on a regular basis:

https://htmlpreview.github.io/?https://github.com/UBC-CS/cpsc330-2024W1/blob/main/docs/calendar.html

Course structure

- Introduction
  - Week 1
- Part I: ML fundamentals, preprocessing, midterm 1
  - Weeks 2, 3, 4, 5, 6, 7, 8
- Part II: Unsupervised learning, transfer learning, common special cases, midterm 1
  - Weeks 8, 9, 10, 11, 12
- Part III: Communication and ethics
  - ML skills are not beneficial if you can't use them responsibly and communicate your results. In this module we'll talk about these aspects.
  - Weeks 13, 14

Code of conduct

- Our main forum for getting help will be Piazza.

> **Important**
>
> Please read this entire document about asking for help. **TLDR:** Be nice.

Homework format: Jupyter notebooks

Jupyter notebooks

- Notebooks contain a mix of code, code output, markdown-formatted text (including LaTeX equations), and more.
- When you open a Jupyter notebook in one of these apps, the document is “live”, meaning you can run the code.

For example:

```
1  1 + 1
```

```
2
```

```
1  x = [1, 2, 3]
2  x[0] = 9999
3  x
```

```
[9999, 2, 3]
```

Jupyter

- By default, Jupyter prints out the result of the last line of code, so you don't need as many `print` statements.
- In addition to the "live" notebooks, Jupyter notebooks can be statically rendered in the web browser, e.g. this.
  - This can be convenient for quick read-only access, without needing to launch the Jupyter notebook/lab application.
  - But you need to launch the app properly to interact with the notebooks.

Lecture notes

- All the lectures from last year are available here.
- We cannot promise anything will stay the same from last year to this year, so read them in advance at your own risk.
- A “finalized” version will be pushed to GitHub and the Jupyter book right before each class.
- Each instructor will have slightly adapted versions of notes to present slides during lectures.
- You will find the link to these slides in our repository: https://github.com/UBC-CS/cpsc330-2024W1/tree/main/lectures/102-Varada-lectures

Grades

- The grading breakdown is here.
- The policy on challenging grades is here.

Setting up your computer for the course

Recommended browser and tools

- You can install Chrome here.
- You can install Firefox here.

In this course, we will primarily be using `Python` , `git`, `GitHub`, `Canvas`, `Gradescope`, `Piazza`, and `PrairieLearn`.

Course `conda` environment

- Follow the setup instructions here to create a course `conda` environment on your computer.
- If you do not have your computer with you, you can partner up with someone and set up your own computer later.

Python requirements/resources

We will primarily use Python in this course.

Here is the basic Python knowledge you'll need for the course:

- Basic Python programming
- Numpy
- Pandas
- Basic matplotlib
- Sparse matrices

Homework 1 is all about Python.

> **Note**

We do not have time to teach all the Python we need but you can find some useful Python resources here.

Checklist for you before the next class

- [ ] Are you able to access course Canvas shell?
- [ ] Are you able to access course Piazza?
- [ ] Are you able to access Gradescope? (If not, refer to the Gradescope Student Guide.)
- [ ] Are you able to access iClicker Cloud for this course?
- [ ] Did you follow the setup instructions here to create a course conda environment on your computer?
- [ ] Did you complete the “Getting to know you” survey on Canvas?
- [ ] Did you complete the anonymous restaurant survey on Qualtrics?
- [ ] Are you almost finished or at least started with