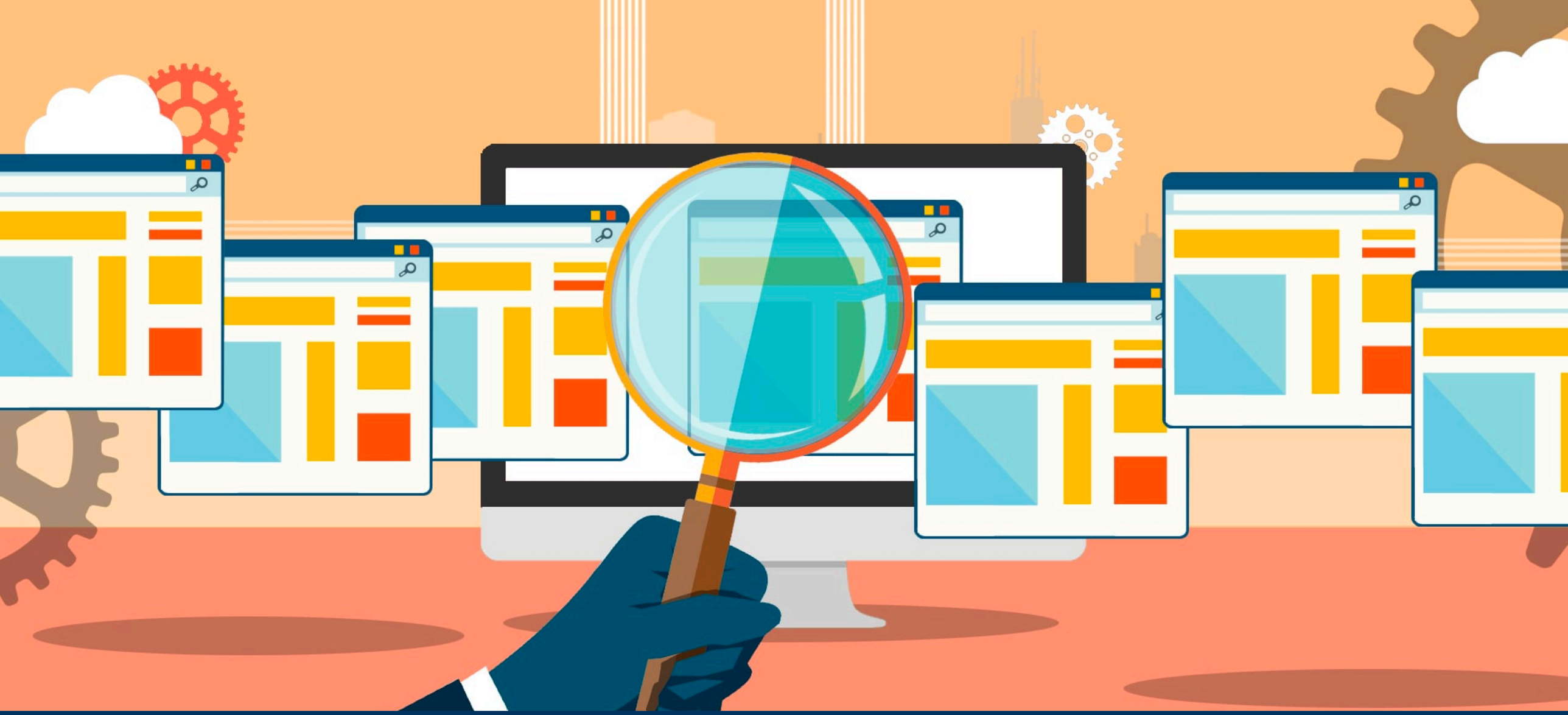
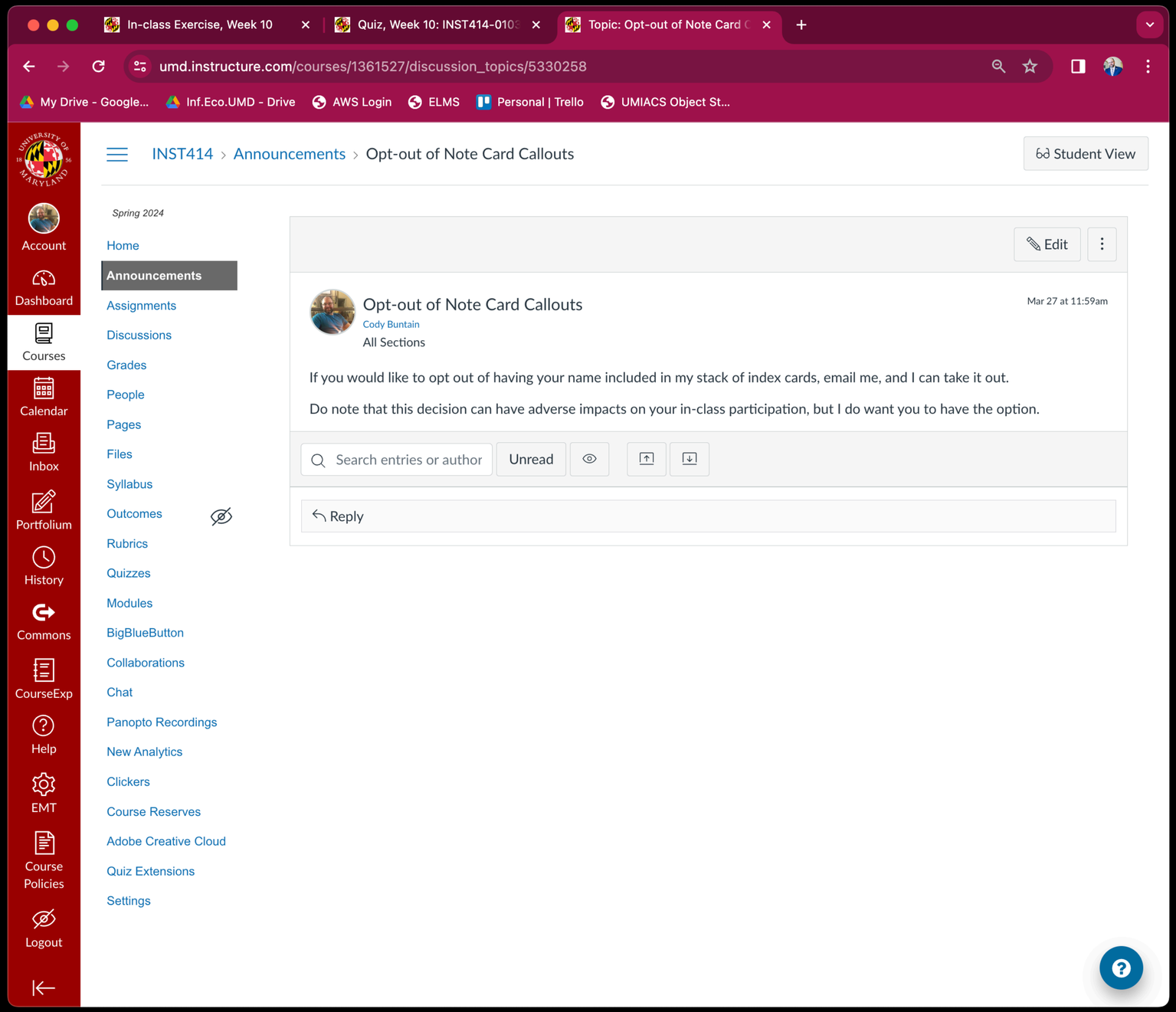


Probabilities, Pt 2

INST414 - Data Science Techniques

Quiz


In-class Exercise, Week 10
Quiz, Week 10: INST414-010
Topic: Opt-out of Note Card
umd.instructure.com/courses/1361527/discussion_topics/5330258
My Drive - Google...
Inf.Eco.UMD - Drive
AWS Login
ELMS
Personal | Trello
UMIACS Object St...
Account
Dashboard
Courses
Calendar
Inbox
Portfolium
History
Commons
CourseExp
Help
EMT
Course Policies
Logout
INST414 › Announcements › Opt-out of Note Card Callouts
Student View
Spring 2024
Home
Announcements
Assignments
Discussions
Grades
People
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Modules
BigBlueButton
Collaborations
Chat
Panopto Recordings
New Analytics
Clickers
Course Reserves
Adobe Creative Cloud
Quiz Extensions
Settings
Edit
Opt-out of Note Card Callouts
Mar 27 at 11:59am
Cody Buntain
All Sections
If you would like to opt out of having your name included in my stack of index cards, email me, and I can take it out.
Do note that this decision can have adverse impacts on your in-class participation, but I do want you to have the option.
Search entries or author
Unread
Reply

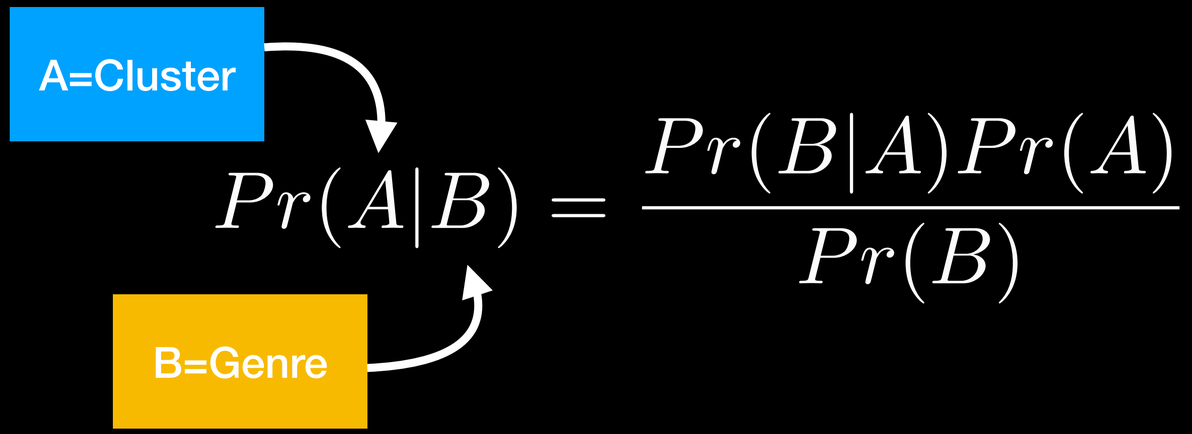
This Module's Learning Objectives

Part 2

Define Bayes' Theorem

Use Bayes' Theorem to identify most likely cluster

What is the **most likely** movie **cluster** given a particular **genre**?



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

$$Pr(Cluster = X|Genre = Y) = \frac{Pr(Genre = Y|Cluster = X)Pr(Cluster = X)}{Pr(Genre = Y)}$$

$$= \frac{Pr(Genre = Y, Cluster = X)}{Pr(Genre = Y)}$$

What is the **most likely** movie **cluster** given a particular **genre**?

But first, what do we know?

Pr(Cluster=X) = ?

Pr(Cluster=X) = (# of movies in cluster X) / (# of movies)

Pr(Genre=Y) = ?

Pr(Genre=Y) = (# of movies w/ genre=Y) / (# of movies)

What is the **most likely** movie **cluster** given a particular **genre**?

But first, what do we know?

Pr(Genre=Y | Cluster=X) = ?

= (# of movies in Cluster X w/ genre =Y) /
(# of movies in Cluster X)

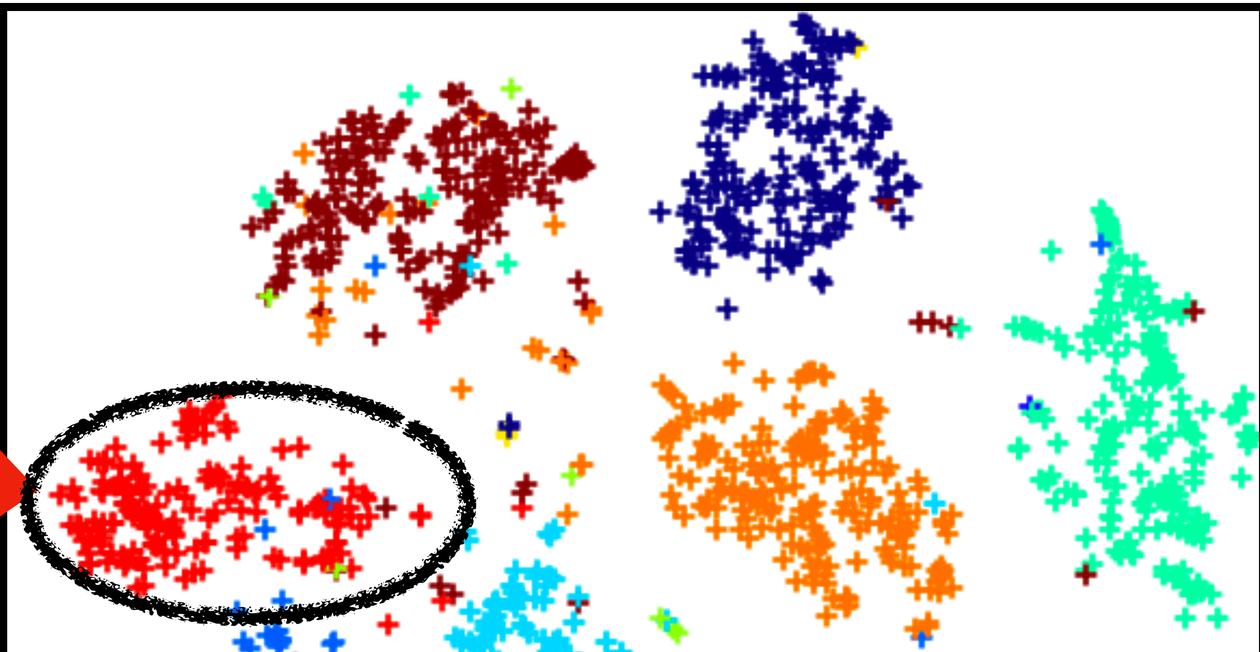
What is the **most likely** movie **cluster** given a particular **genre**?

But first, what do we know?

Pr(Genre=Y | Cluster=X) = ?

= (# of movies in Cluster X w/ genre =Y) /
(# of movies in Cluster X)

Pretend this “red” cluster is the whole dataset



What is the **most likely** movie **cluster** given a particular **genre**?

But first, what do we know?

Pr(Cluster=X), Pr(Genre=Y), Pr(Genre=Y | Cluster=X)

$$Pr(Genre = Y, Cluster = X) = Pr(Genre = Y|Cluster = X) \cdot Pr(Cluster = X)$$
$$= Pr(Cluster = X|Genre = Y) \cdot Pr(Genre = Y)$$

Global probability of a movie being BOTH Y *and* X

Can change the order and maintain equivalence

What is the **most likely** movie **cluster** given a particular **genre**?

But first, what do we know?

Pr(Cluster=X), Pr(Genre=Y), Pr(Genre=Y | Cluster=X)

$$Pr(Genre = Y, Cluster = X) = Pr(Genre = Y|Cluster = X) \cdot Pr(Cluster = X)$$
$$= Pr(Genre = Y) \cdot Pr(Cluster = X)$$
$$\neq Pr(Genre = Y|Cluster = X)$$

Global probability of a movie being BOTH Y *and* X

Product of Prs for both events *if they're independent*

What is the **most likely** movie **cluster** given a particular **genre**?

Multiple ways to arrive to the same destination…

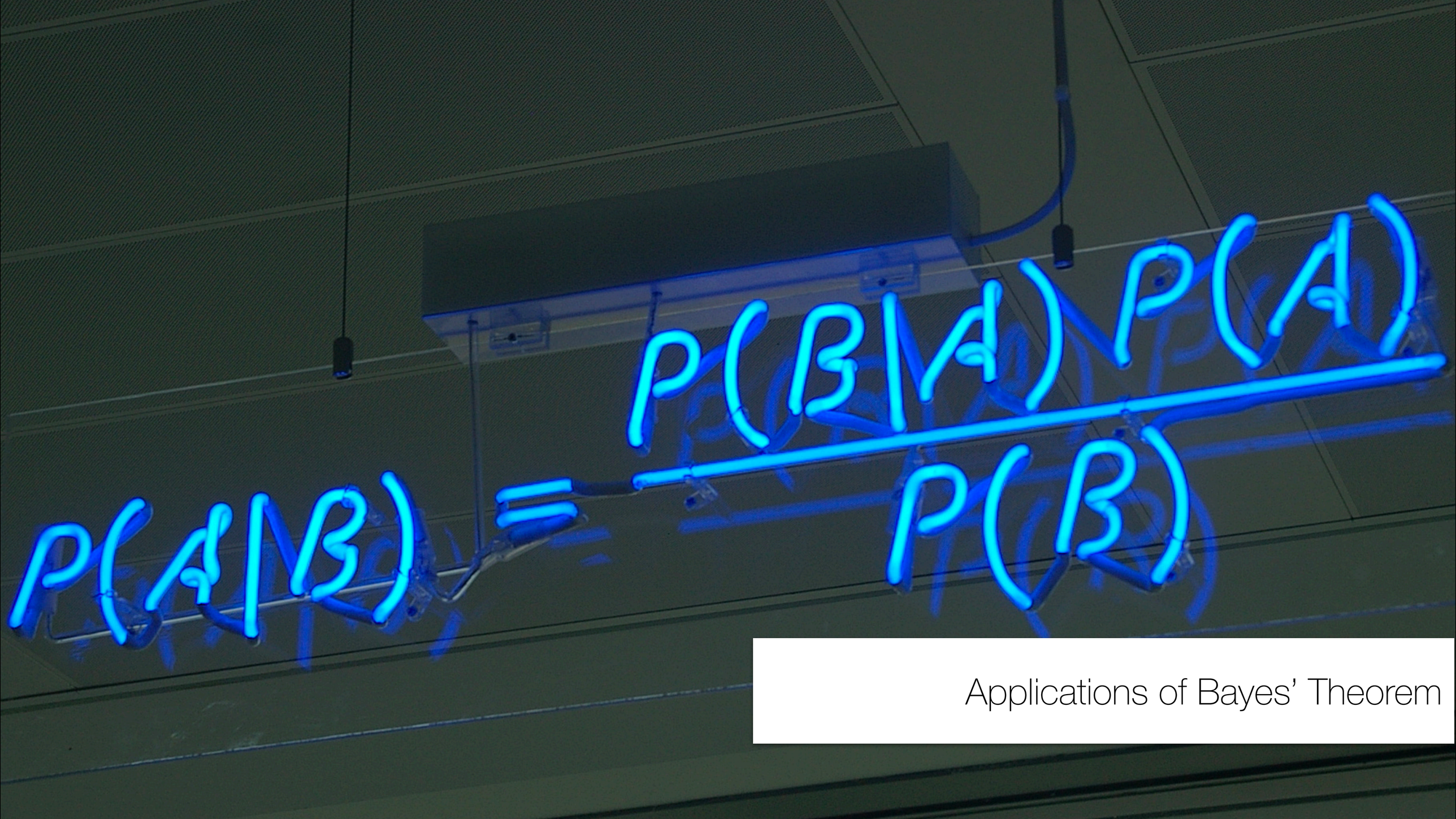
$$Pr(Genre = Y) = \frac{\#\text{of movies with Genre Y}}{\#\text{of movies}}$$

"Marginalizing out" the Clusters

$$= \sum_{X \in Clusters} Pr(Genre = Y, Cluster = X)$$

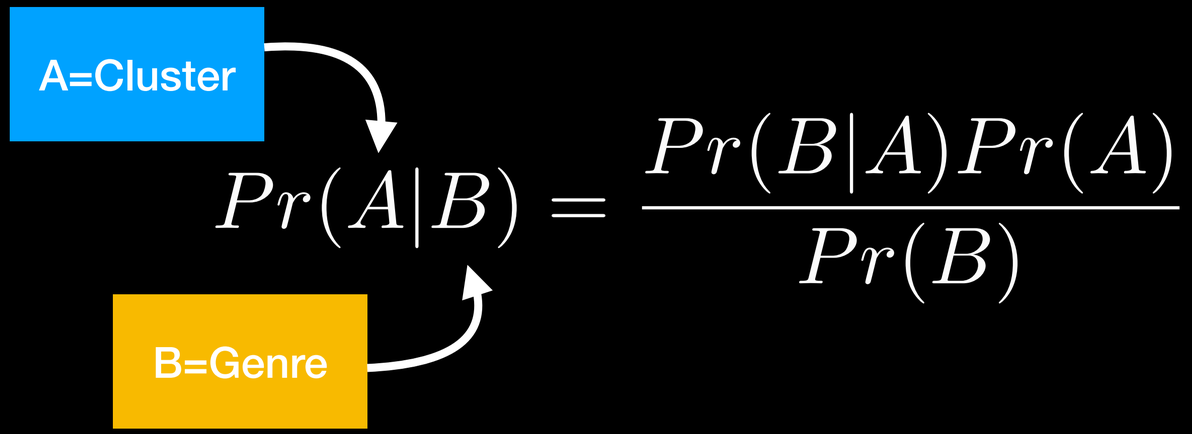
$$= \sum_{X \in Clusters} Pr(Genre = Y | Cluster = X) Pr(Cluster = X)$$

$$\neq \sum_{X \in Clusters} Pr(Genre = Y | Cluster = X)$$



Applications of Bayes' Theorem

What is the **most likely** movie **cluster** given a particular **genre**?



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

$$Pr(Cluster = X|Genre = Y) = \frac{Pr(Genre = Y|Cluster = X)Pr(Cluster = X)}{Pr(Genre = Y)}$$

$$= \frac{Pr(Genre = Y, Cluster = X)}{Pr(Genre = Y)}$$

You can do this…

$$Pr(Cluster = X|Genre = Y) = \frac{Pr(Genre = Y|Cluster = X)Pr(Cluster = X)}{Pr(Genre = Y)}$$

$$Pr(Cluster = X|Genre = Y) \cdot Pr(Genre = Y) = Pr(Genre = Y|Cluster = X) \cdot Pr(Cluster = X)$$

Because this…

$$=$$

$$Pr(Cluster = X, Genre = Y)$$

What is the **most likely** movie **cluster** given a particular **genre**?

```
In [60]: movie_cluster_df["cluster"].value_counts() / movie_cluster_df.shape[0]

Out[60]: 6     0.154074
         0     0.150194
         15    0.085063
         13    0.082687
         2     0.072890
         12    0.071096
         1     0.066731
         3     0.060136
         14    0.043307
         8     0.037536
         10    0.036906
         4     0.031765
         11    0.031038
         7     0.030795
         5     0.027158
         9     0.018623
         Name: cluster, dtype: float64
```

What is the **most likely** movie **cluster** given a **genre = Sport**?

Do any of the following directly answer this question?

Pr(Cluster=X)

Pr(Genre=Y)

Pr(Genre=Y | Cluster=X)

So what quantity would directly answer this question?

Pr(???)

Into which cluster will a Sports movie go?

$$Pr(Cluster = X | Genre = Sport)$$

$$= \frac{Pr(Genre = Sport | Cluster = X) Pr(Cluster = X)}{Pr(Genre = Sport)}$$

$$\text{argmax}_{X \in Clusters} Pr(Cluster = X | Genre = Sport)$$

Probability of a specific cluster given Sport genre...

Really want this:

$$\text{argmax}_{X \in Clusters} Pr(Cluster = X | Genre = Sport)$$

Really want this:

Maximize Over:

$$= \frac{Pr(Genre = Sport | Cluster = 1) Pr(Cluster = 1)}{Pr(Genre = Sport)}$$

$$= \frac{Pr(Genre = Sport | Cluster = 2) Pr(Cluster = 2)}{Pr(Genre = Sport)}$$

...

$$= \frac{Pr(Genre = Sport | Cluster = C_n) Pr(Cluster = C_n)}{Pr(Genre = Sport)}$$

Each value has the same denominator

Given some genre, we can determine most probable cluster

Can think of this process as a supervised classification task

Movies' cluster assignments are labels

"Maximize pr of cluster label given a genre"
is our supervised model

Today's Exercises

Exercise 1: Finding the most likely cluster given a particular genre

Exercise 2: Finding the most likely cluster given a particular actor

In-class Exercise, Week 9

Published | Edit

Exercise 1.Finding the most likely cluster given a particular genre (30 minutes)

Scaffolding Code Available here: 01-ClusterProbabilities.Scaffolding.ipynb

1. Download movie_to_cluster.csv . This file contains all of the movies in the database and the cluster they were sorted into.
2. Load the csv into Python as a dataframe
3. Set your target genre as "Sci-Fi"
4. Use a for loop to iterate through the clusters. For each one, calculate the probability that a new movie will fit in that cluster given that it is a Sci-Fi movie. In other words calculate P(Cluster | Sci-Fi)
5. You should get the following output:
6.
```
Pr[Cluster 00 | Sci-Fi]:         0.17345971563981044
Pr[Cluster 01 | Sci-Fi]:         0.040758293838862564
Pr[Cluster 02 | Sci-Fi]:         0.20379146919431282
Pr[Cluster 03 | Sci-Fi]:         0.02464454976303318
Pr[Cluster 04 | Sci-Fi]:         0.05308056872037915
Pr[Cluster 05 | Sci-Fi]:         0.0
Pr[Cluster 06 | Sci-Fi]:         0.061611374407582936
Pr[Cluster 07 | Sci-Fi]:         0.0
Pr[Cluster 08 | Sci-Fi]:         0.008530805687203791
Pr[Cluster 09 | Sci-Fi]:         0.021800947867298578
Pr[Cluster 10 | Sci-Fi]:         0.0009478672985781991
Pr[Cluster 11 | Sci-Fi]:         0.047393364928909956
Pr[Cluster 12 | Sci-Fi]:         0.18862559241706164
Pr[Cluster 13 | Sci-Fi]:         0.013270142180094787
Pr[Cluster 14 | Sci-Fi]:         0.03033175355450237
Pr[Cluster 15 | Sci-Fi]:         0.13175355450236967
```
7. Repeat steps 3 and 4 for the genres "Fantasy" and "Adventure"

We've provided you clusters of movies

raw.githubusercontent.com/cbuntain/umd.inst414/main/Module05/movie_to_cluster.csv

Update

```
movie_id,cluster
tt0035423,1
tt0088751,4
tt0096056,14
tt0113092,12
tt0116391,8
tt0117461,6
tt0117743,6
tt0118652,6
tt0118852,2
tt0118926,4
tt0119004,2
tt0119231,13
tt0119273,10
tt0120166,2
tt0120202,2
tt0120467,13
tt0120630,8
tt0120667,4
tt0120679,14
tt0120681,2
tt0120698,10
tt0120737,14
tt0120753,3
tt0120755,4
tt0120804,3
tt0120824,5
tt0120870,2
tt0120903,14
tt0120912,1
tt0120913,1
tt0120917,8
tt0121164,2
tt0121569,2
tt0121765,1
tt0121766,12
tt0122161,10
tt0122247,0
tt0122459,3
tt0123003,2
tt0123581,7
tt0124889,0
tt0125022,3
tt0126029,2
tt0127349,10
tt0129095,2
tt0130623,0
tt0131597,9
tt0131704,12
tt0132245,2
tt0132885,0
tt0132910,2
tt0133152,1
tt0133240,3
tt0133752,6
tt0134084,1
tt0134630,1
```

```
        print("\t\t", m_id, movie_data_map[m_id]["movie"])
```

```
Cluster: 0 Size: 3097
        Top Genres:
                Comedy [0.6581]
                Family [0.1611]
                Adventure [0.1531]
                Animation [0.1505]
                Crime [0.0846]
        Movie Sample:
                tt12484058 Die by the Sword
                tt2225002 Pirate Jenny
                tt13496286 Little Bookworms 3
                tt5936438 Star Paws
                tt2328630 Walking to Linas
                tt5679402 Stagecoach: The Texas Jack Story
                tt1870425 Mac & Devin Go to High School
                tt0337711 Rugrats Go Wild
                tt9379530 Max Bishop
                tt2990140 The Christmas Chronicles
Cluster: 1 Size: 1376
        Top Genres:
                Drama [1.0000]
                Thriller [1.0000]
                Crime [0.2427]
                Mystery [0.1199]
                Action [0.1090]
        Movie Sample:
                tt2873282 Red Sparrow
                tt0406754 Evilenko
                tt0780516 Flawless
                tt5834362 What Death Leaves Behind
                tt1389139 When the Bough Breaks
                tt4694518 Domain
                tt0469066 Descansos
                tt0388837 The Circle
                tt0396960 Clean
                tt5605730 Cabaret
Cluster: 2 Size: 1503
        Top Genres:
                Action [1.0000]
                Adventure [0.4351]
                Comedy [0.3426]
                Crime [0.2116]
```

Account
Dashboard
Courses
Calendar
Inbox
Commons
CourseExp
Help
EMT

Fall 2023

Home
Announcements
Assignments
Discussions
Grades
People
Pages
...abrics
Quizzes
Modules
BigBlueButton
Collaborations
Chat
Panopto Recordings
New Analytics
Clickers

In-class Exercise, Week 9

Published | Edit

Exercise 1.Finding the most likely cluster given a particular genre (30 minutes)

Scaffolding Code Available here: 01-ClusterProbabilities.Scaffolding.ipynb

1. Download movie_to_cluster.csv . This file contains all of the movies in the database and the cluster they were sorted into.
2. Load the csv into Python as a dataframe
3. Set your target genre as "Sci-Fi"
4. Use a for loop to iterate through the clusters. For each one, calculate the probability that a new movie will fit in that cluster given that it is a Sci-Fi movie. In other words calculate P(Cluster | Sci-Fi)
5. You should get the following output:
6.
```
Pr[Cluster 00 | Sci-Fi]:         0.17345971563981044
Pr[Cluster 01 | Sci-Fi]:         0.040758293838862564
Pr[Cluster 02 | Sci-Fi]:         0.20379146919431282
Pr[Cluster 03 | Sci-Fi]:         0.02464454976303318
Pr[Cluster 04 | Sci-Fi]:         0.05308056872037915
Pr[Cluster 05 | Sci-Fi]:         0.0
Pr[Cluster 06 | Sci-Fi]:         0.061611374407582936
Pr[Cluster 07 | Sci-Fi]:         0.0
Pr[Cluster 08 | Sci-Fi]:         0.008530805687203791
Pr[Cluster 09 | Sci-Fi]:         0.021800947867298578
Pr[Cluster 10 | Sci-Fi]:         0.0009478672985781991
Pr[Cluster 11 | Sci-Fi]:         0.047393364928909956
Pr[Cluster 12 | Sci-Fi]:         0.18862559241706164
Pr[Cluster 13 | Sci-Fi]:         0.013270142180094787
Pr[Cluster 14 | Sci-Fi]:         0.03033175355450237
Pr[Cluster 15 | Sci-Fi]:         0.13175355450236967
```
7. Repeat steps 3 and 4 for the genres "Fantasy" and "Adventure"

And some scaffolding code

Cody Buntain Module 5 examples and scaffolding

Latest commit 236270d 26 minutes ago History

0 contributors

453 lines (453 sloc) | 12.8 KB

Raw Blame

Probabilities and k-Means Clustering

Using the IMDB data, construct a feature matrix, and apply `k-Means` to the data to extract clusters.

We then inspect various aspects of probability associated with these clusterings.

In [ ]:

```
import json

import pandas as pd
import numpy as np
```

In [ ]:

```

```

In [ ]:

```
actor_name_map = {}
movie_actor_map = {}
actor_genre_map = {}


with open("../data/imdb_movies_2000to2022.prolific.json", "r") as in_file:
    for line in in_file:

        # Read the movie on this line and parse its json
        this_movie = json.loads(line)

        # Add all actors to the id->name map
        for actor_id,actor_name in this_movie['actors']:
            actor_name_map[actor_id] = actor_name

        # For each actor, add this movie's genres to that actor's list
        for actor_id,actor_name in this_movie['actors']:
            this_actors_genres = actor_genre_map.get(actor_id, {})
```

Exercise 2. Finding the most likely cluster given a particular actor (Optional)

This exercise is slightly different. Here you are trying to find the most likely cluster but now it is based on the movie starring a particular actor, instead f being in a certain genre

1. Download movie_to_cluster.csv . This file contains all of the movies in the database and the cluster they were sorted into.
2. Load the csv into Python as a dataframe
3. Set your target actor as Nicolas Cage (actor ID: nm0000115)
4. Use a for loop to iterate through the clusters. For each one, calculate the probability that a new movie will fit in that cluster given that it features Nicolas Cage. In other words calculate P(Cluster | nm0000115)
5. You should get the following output:
6. ```
   Pr[Cluster 00 | nm0000115]:      0.0819672131147541
   Pr[Cluster 01 | nm0000115]:      0.13114754098360656
   Pr[Cluster 02 | nm0000115]:      0.1639344262295082
   Pr[Cluster 03 | nm0000115]:      0.09836065573770492
   Pr[Cluster 04 | nm0000115]:      0.01639344262295082
   Pr[Cluster 05 | nm0000115]:      0.0
   Pr[Cluster 06 | nm0000115]:      0.03278688524590164
   Pr[Cluster 07 | nm0000115]:      0.0
   Pr[Cluster 08 | nm0000115]:      0.0
   Pr[Cluster 09 | nm0000115]:      0.04918032786885246
   Pr[Cluster 10 | nm0000115]:      0.0
   Pr[Cluster 11 | nm0000115]:      0.03278688524590164
   Pr[Cluster 12 | nm0000115]:      0.03278688524590164
   Pr[Cluster 13 | nm0000115]:      0.06557377049180328
   Pr[Cluster 14 | nm0000115]:      0.01639344262295082
   Pr[Cluster 15 | nm0000115]:      0.27868852459016397
   ```
7. Repeat steps 3 and 4 for the actors Keanu Reeves (actor ID: nm0000206) and Tom Hiddleston (actor ID: nm1089991)

Points 100

Submitting a file upload

Due	For	Available from	Until
   ```

What questions do you have?

Prof. Cody Buntain | @codybuntain | cbuntain@umd.edu
Director, Information Ecosystems Lab