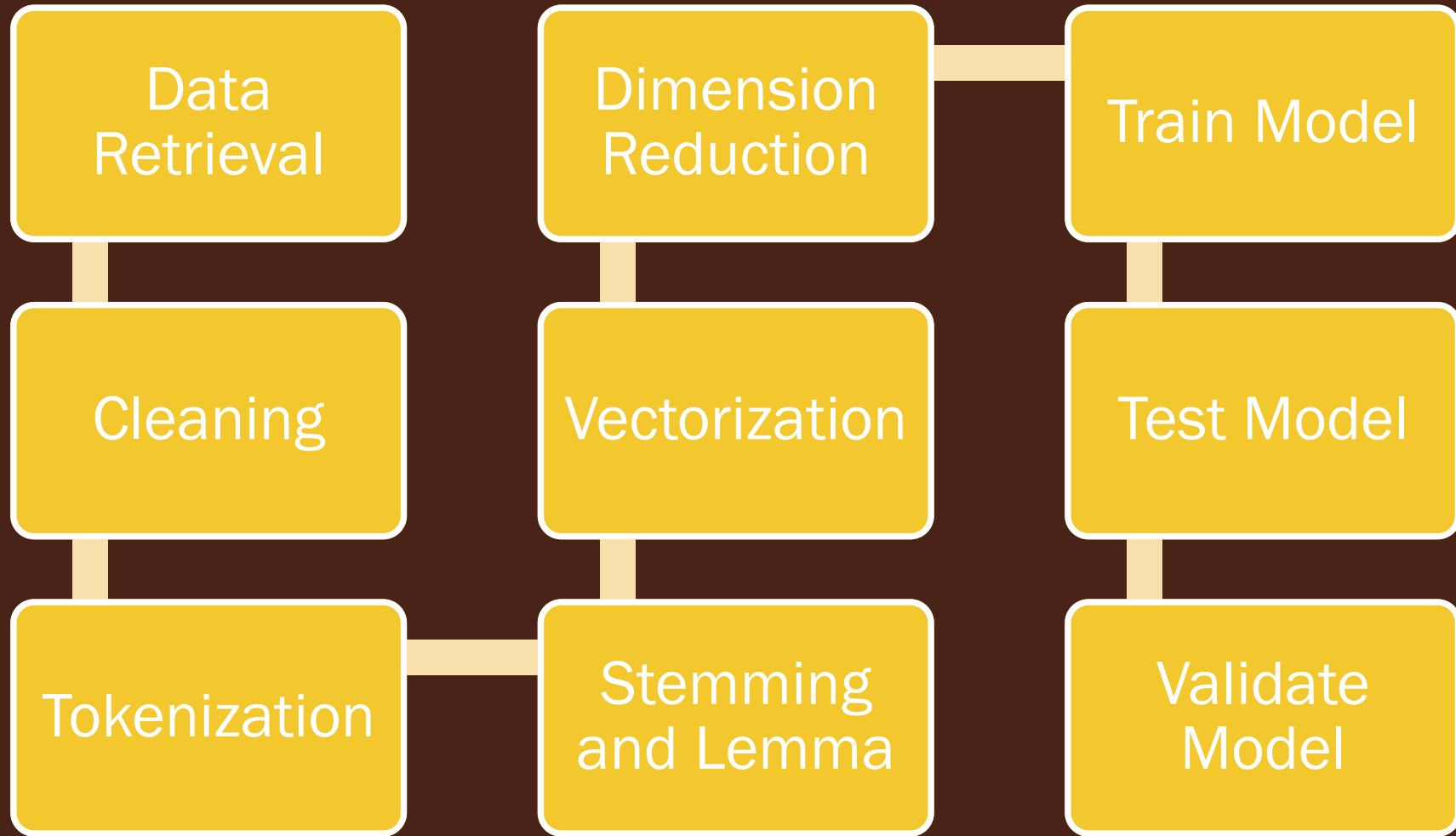


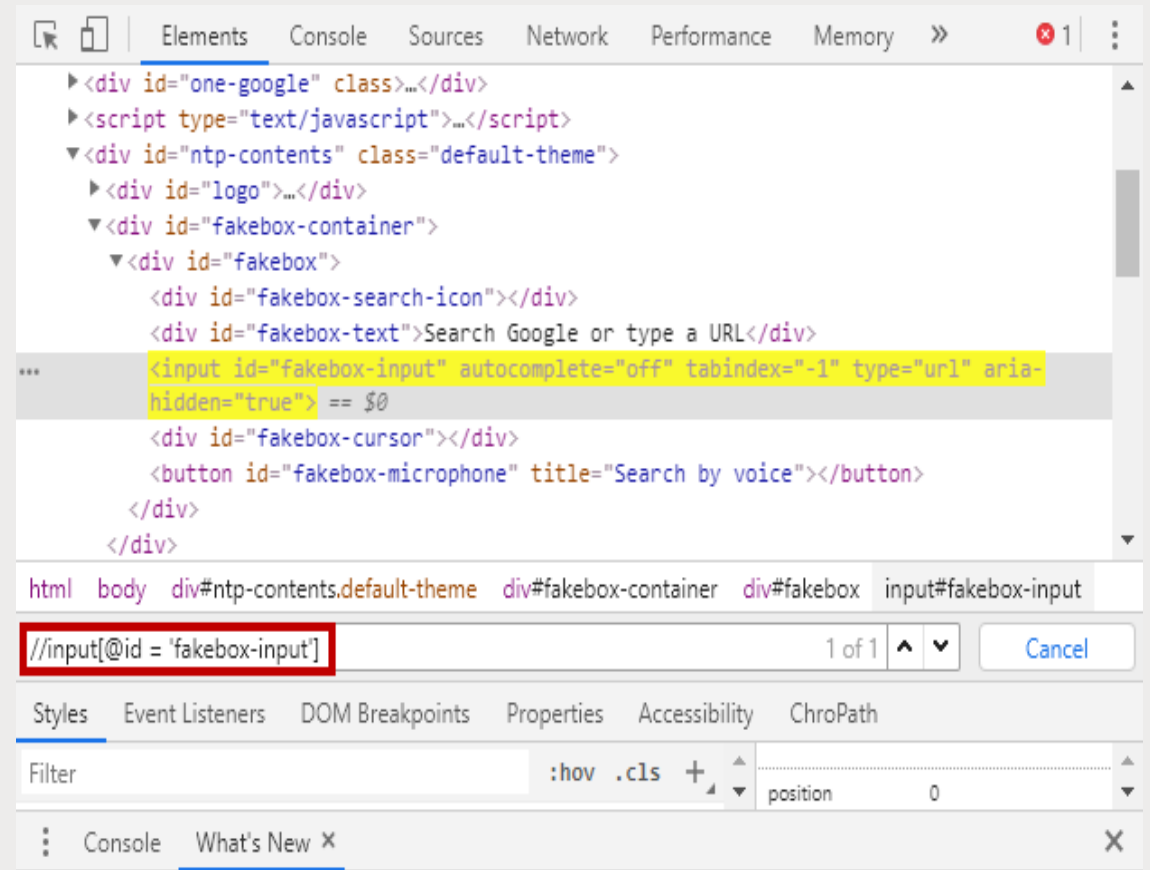
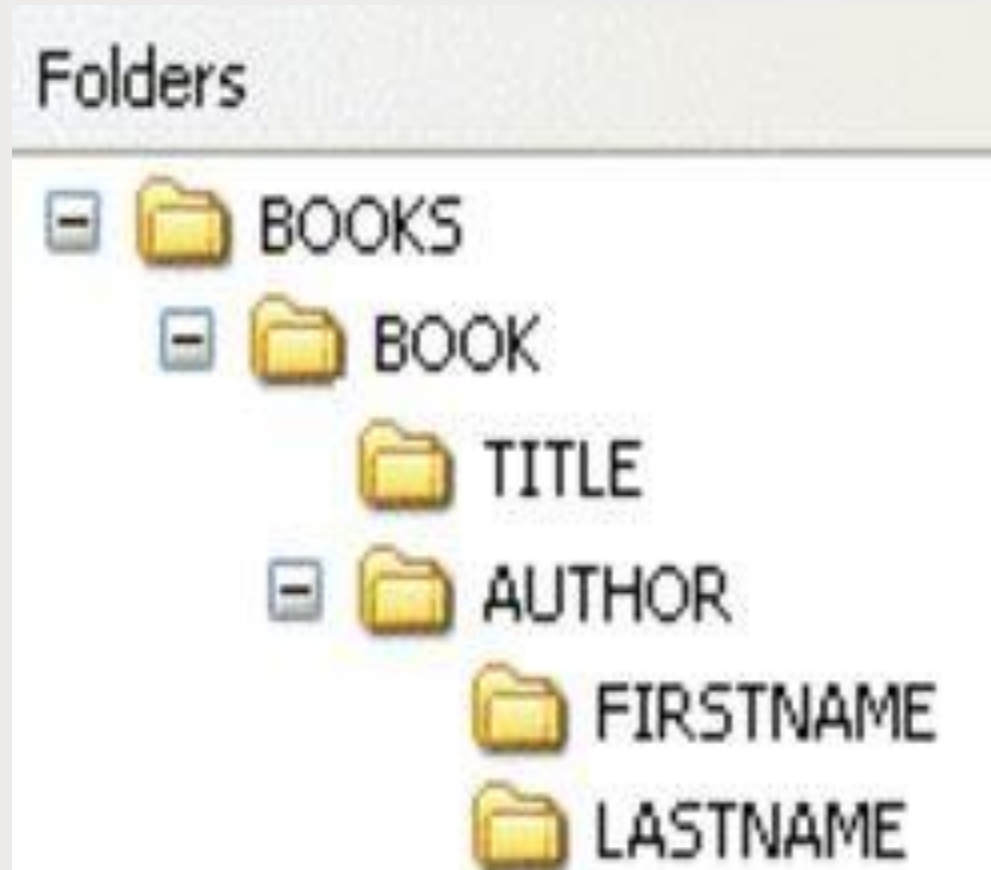


PREDICTION OF SEARCH QUERIES USING VIDEO COMMENTS

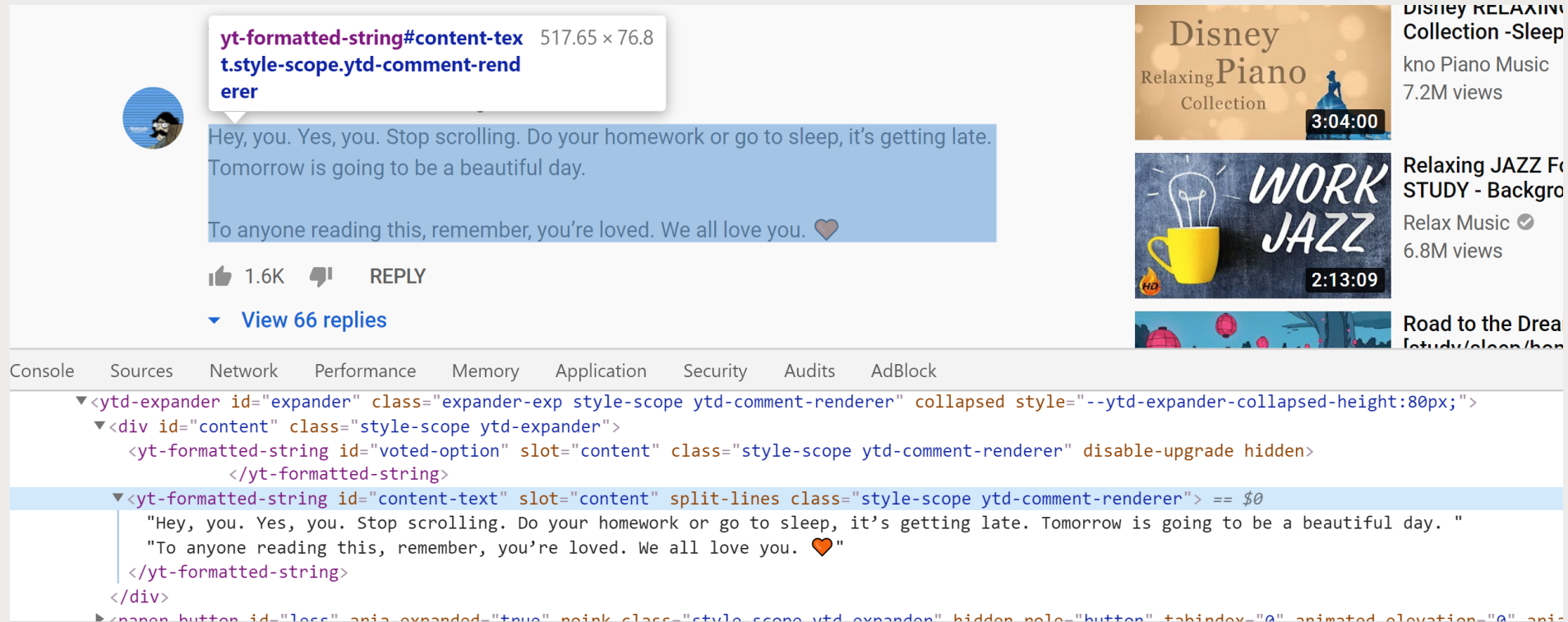
Lemon Lin Reimer



xpaths in HTML code



Data collection through selenium



The screenshot shows a YouTube comment interface. A comment by a user with a profile picture of a person with glasses and a blue background says: "Hey, you. Yes, you. Stop scrolling. Do your homework or go to sleep, it's getting late. Tomorrow is going to be a beautiful day. To anyone reading this, remember, you're loved. We all love you. ❤️". The comment has 1.6K likes and a "REPLY" button. Below the comment is a "View 66 replies" link. To the right of the comment are three video thumbnails: "Disney Relaxing Piano Collection - Sleep", "Relaxing JAZZ For STUDY - Background Music", and "Road to the Dream". Below the comment and videos is a browser's developer console showing the DOM structure. The DOM structure is as follows:

```
<ytd-expander id="expander" class="expander-exp style-scope ytd-comment-renderer" collapsed style="--ytd-expander-collapsed-height:80px;">
  <div id="content" class="style-scope ytd-expander">
    <ytd-formatted-string id="voted-option" slot="content" class="style-scope ytd-comment-renderer" disable-upgrade hidden>
      </ytd-formatted-string>
    <ytd-formatted-string id="content-text" slot="content" split-lines class="style-scope ytd-comment-renderer"> == $0
      "Hey, you. Yes, you. Stop scrolling. Do your homework or go to sleep, it's getting late. Tomorrow is going to be a beautiful day. "
      "To anyone reading this, remember, you're loved. We all love you. ❤️"
    </ytd-formatted-string>
  </div>
  <button id="less" aria-expanded="true" role="button" class="style-scope ytd-expander" hidden="" tabindex="0" animated="">
```

```
self.driver = webdriver.Chrome(chrome_options = Options().add_argument("--headless"))
self.driver.get("https://www.youtube.com/")
comment = self.driver.find_element(s)_by_xpath('//*[@id="content-text"]')
text = comment.text
slot_name = comment.get_attribute("slot")
```

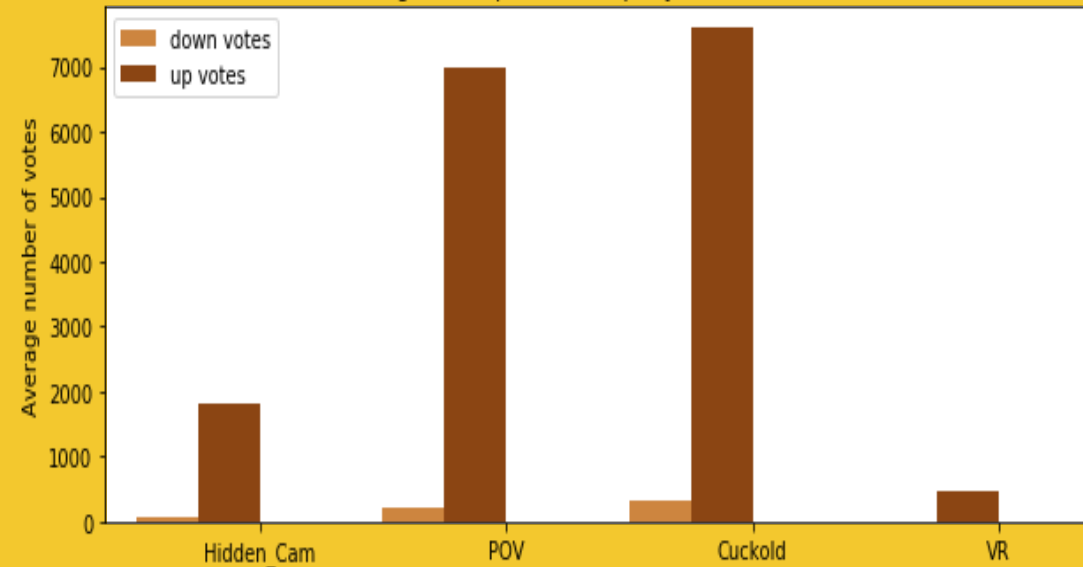
VARIABLES EXTRACTED INTO DATA FRAME

- Query string used to search for videos
- Link to video (36 per query)
- Number of upvotes and number of downvotes per video
- Number of views per video
- Raw comments + cleaned + lemmatized + stemmed
- Gender and sexuality of commenter
- Link to user profile of commenter
- Date of comment posted in UTC

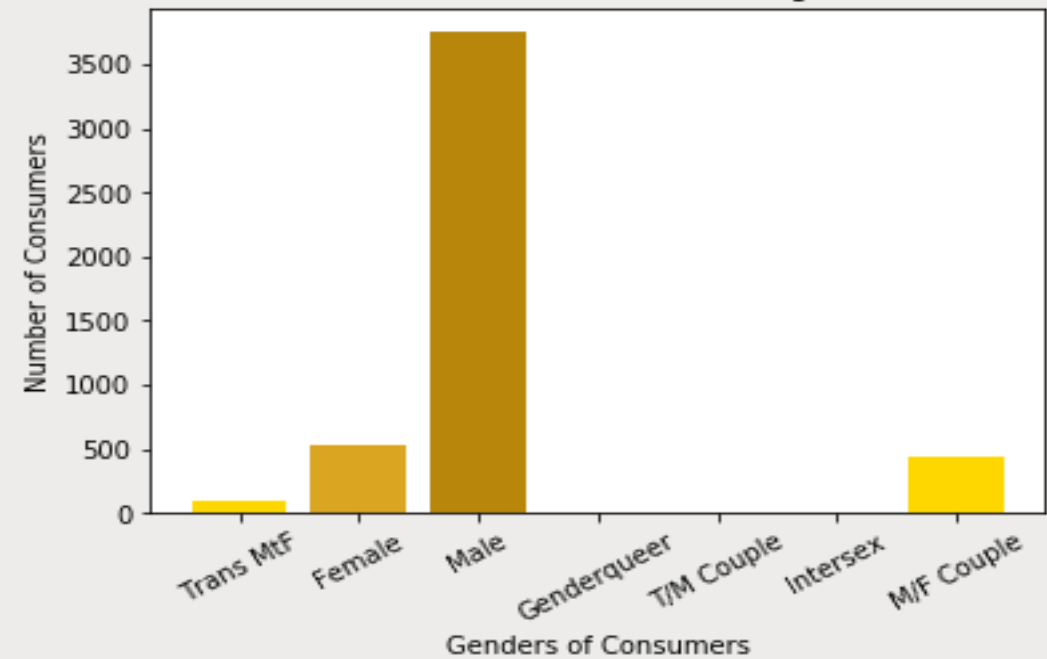
Description of the variables

Graphics were produced using the matplotlib package.

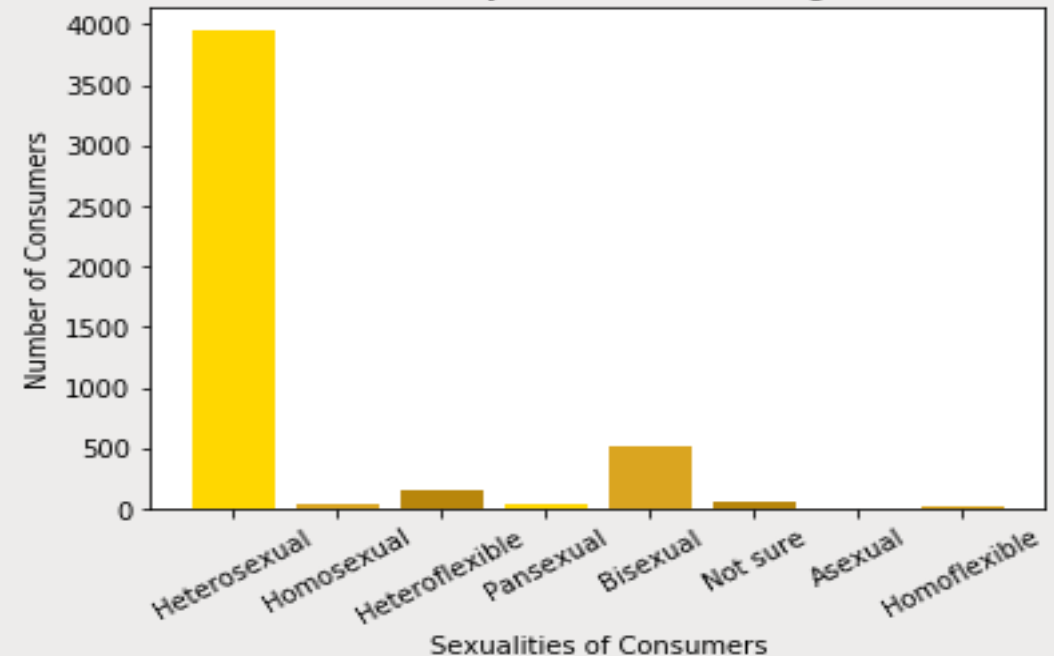
Average votes per search query across 36 videos




Bar Plot of Gender Distribution Among Consumers




Bar Plot of Sexuality Distribution Among Consumers





Data Collection: Selenium (Chrome driver)
Cleaning: re
Vectorization: tfidfVectorizer
Tokenization: NLTK
Stemming/Lemma: sklearn
Dimensionality Reduction: PCA
Grid Search: sklearn
Classifier: RandomForestClassifier



Validation in progress!



CONCLUSION

- ❖ This was really fun!
- ❖ Can potentially use this data for thesis
- ❖ Learned to develop class objects and call them from a top file
- ❖ Tackled plenty of bugs and glitches
- ❖ Played with a new package and some HTML with selenium
- ❖ Need to work on developing classification models
- ❖ Want to learn more about (un)supervised learning
- ❖ Unsure how to label data without expert labeling

ANY QUESTIONS?

