Predicting Dating App Subreddit with NLP

Columbia University Data Scientist,
James Pecore

Project Agenda

I. Problem Statement & Executive Summary

II. Scraping “Tinder” and “Tinder Stories” Subreddits

III. EDA of “Tinder” and “Tinder Stories”

IV. Tokenizing/Preprocessing Concatenated Data

V. Modeling Data

VI. Limits of Models

VII. Conclusions and Recommendations

VIII. Works Sourced

I. Problem Statement

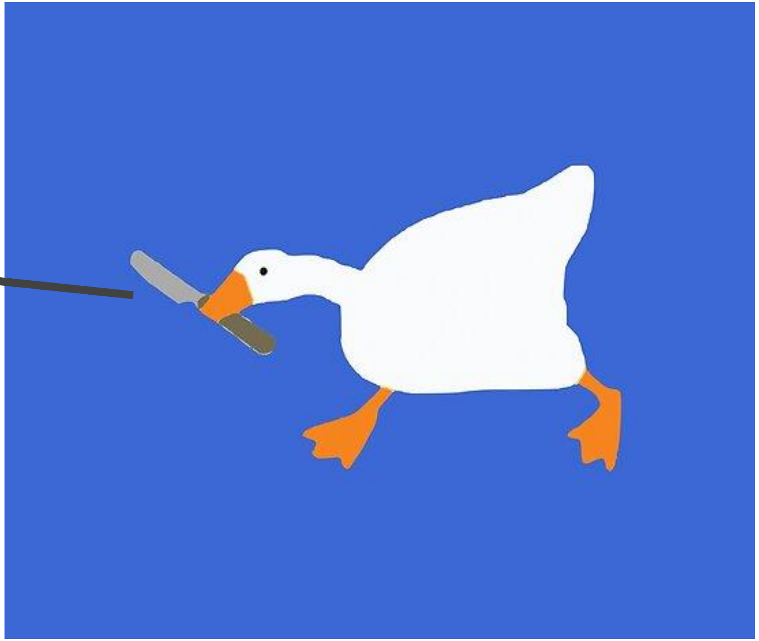
- As a representative of the Match company, good customer service is essential to our platform Tinder’s wellbeing.
- We want to examine the similarity of the subreddits “Tinder” and “Tinder Stories.”
- Can classification models accurately (< 60-80 % of the time) predict the difference between a post on “Tinder Stories” and one on “Tinder?”
- What differentiates the contents of “Tinder” and “Tinder Stories.”
- What is the most frequent verbal content that “Tinder” and “Tinder Stories” produce on a huge, statistical level?
- Finally, what can I recommend doing to investigate and reclaim control over the reputation of Tinder set by its community of users?

I. Executive Summary

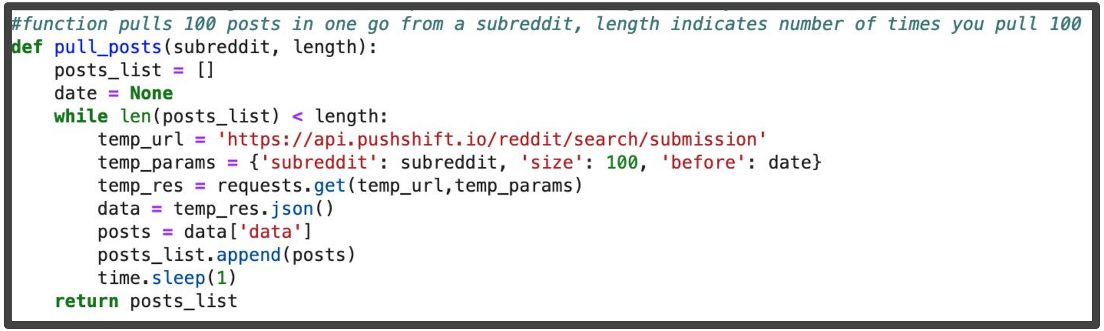
- By using NLP and many different classifiers, I created models able to differentiate these two subreddits.
- I hypothesize that heterosexual men tend to use the “Tinder” Subreddit to talk about dating women. Meanwhile, heterosexual women tend to use the “Tinder Stories” to talk about men.
- The models used can accurately predict the differences between posts on each of these subreddits between 87% and 97% of the time.
- Gendered words and preferences such as these impact our models rather significantly.
- These gendered words can inform how we build our User Interface more effectively for each gender/subreddit group.

II. Scraping “Tinder” and “Tinder Stories” Subreddits

“Let’s get scraping!”

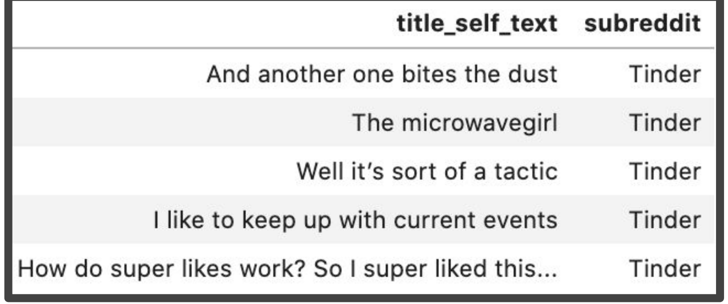


II. Scraping “Tinder” and “Tinder Stories” Subreddits



```
#function pulls 100 posts in one go from a subreddit, length indicates number of times you pull 100
def pull_posts(subreddit, length):
    posts_list = []
    date = None
    while len(posts_list) < length:
        temp_url = 'https://api.pushshift.io/reddit/search/submission'
        temp_params = {'subreddit': subreddit, 'size': 100, 'before': date}
        temp_res = requests.get(temp_url,temp_params)
        data = temp_res.json()
        posts = data['data']
        posts_list.append(posts)
        time.sleep(1)
    return posts_list
```

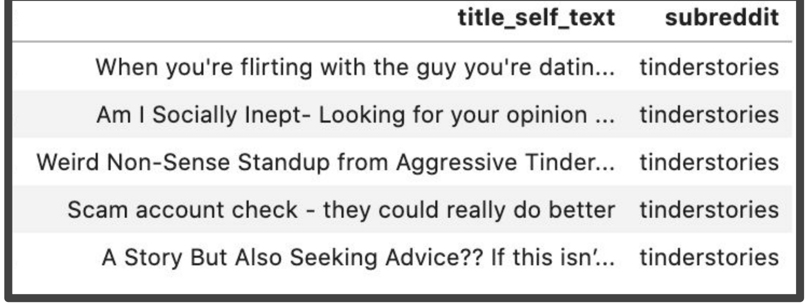
More images, less text



title_self_text	subreddit
And another one bites the dust	Tinder
The microwavegirl	Tinder
Well it’s sort of a tactic	Tinder
I like to keep up with current events	Tinder
How do super likes work? So I super liked this...	Tinder

1000 “Tinder” Posts

More text, less images

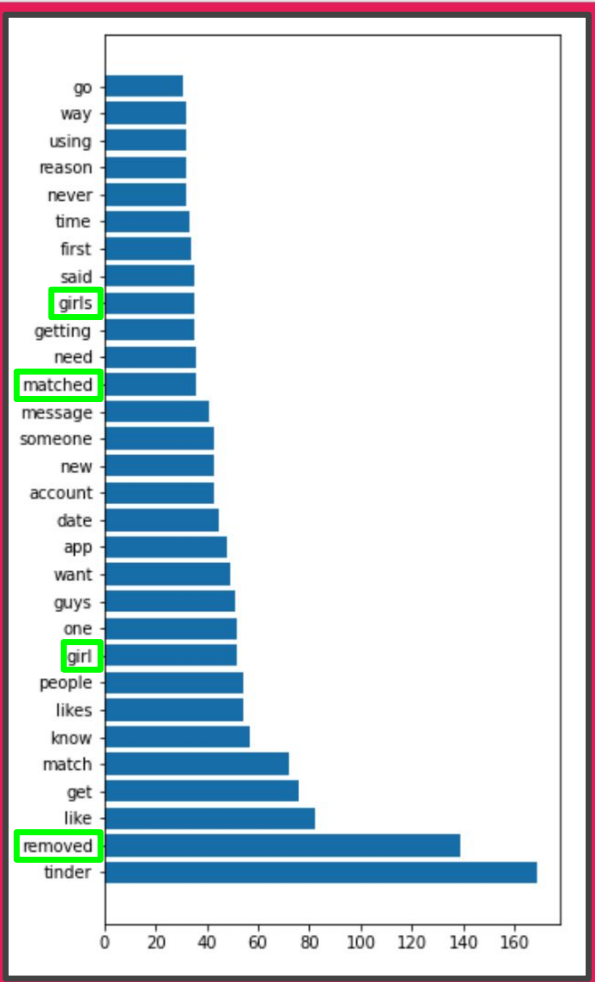


title_self_text	subreddit
When you're flirting with the guy you're datin...	tinderstories
Am I Socially Inept- Looking for your opinion ...	tinderstories
Weird Non-Sense Standup from Aggressive Tinder...	tinderstories
Scam account check - they could really do better	tinderstories
A Story But Also Seeking Advice?? If this isn’...	tinderstories

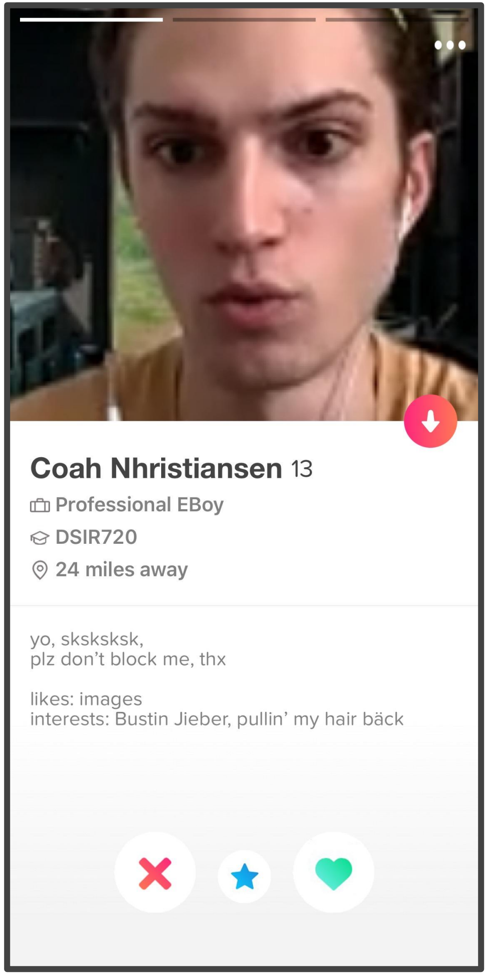
1000 “Tinder Stories” Posts

III. Exploratory Data Analysis of “Tinder” and “Tinder Stories

I. CountVectorizer
II. Stopword Cleaning
III. Analysis of Most Frequent Words
IV. Concatenate Dataframes



Popular "Tinder" Subreddit (M) Words



Coah Nhristiansen 13

Professional EBoy

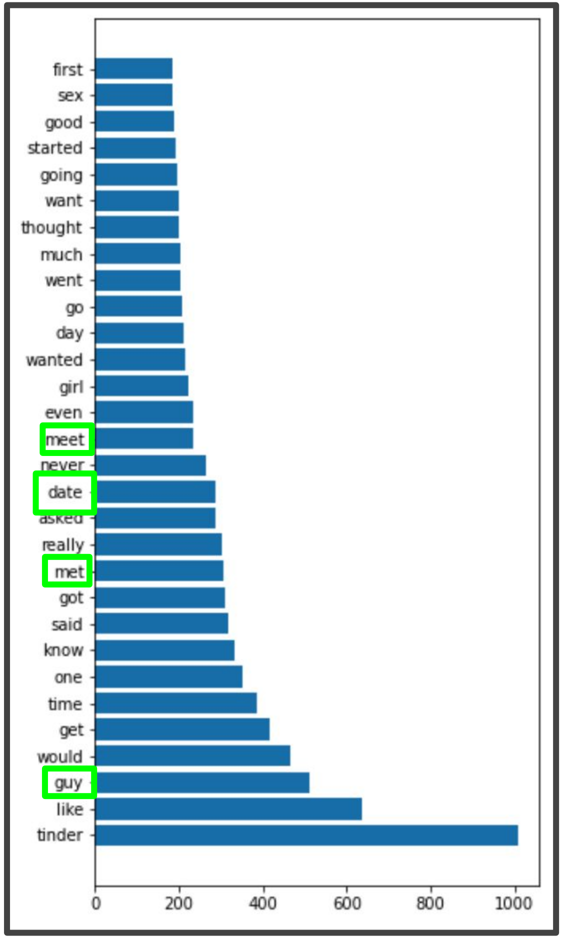
DSIR720

24 miles away

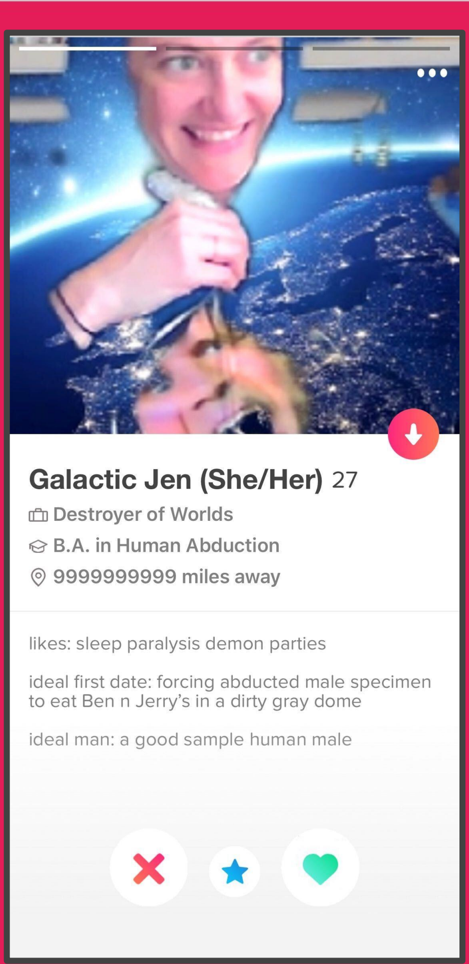
yo, sksksksk,
plz don't block me, thx

likes: images
interests: Bustin Jieber, pullin' my hair bäck

Example "Tinder" Subreddit User



Popular “Tinder Stories” Subreddit (F) Words



Galactic Jen (She/Her) 27

Destroyer of Worlds

B.A. in Human Abduction

9999999999 miles away

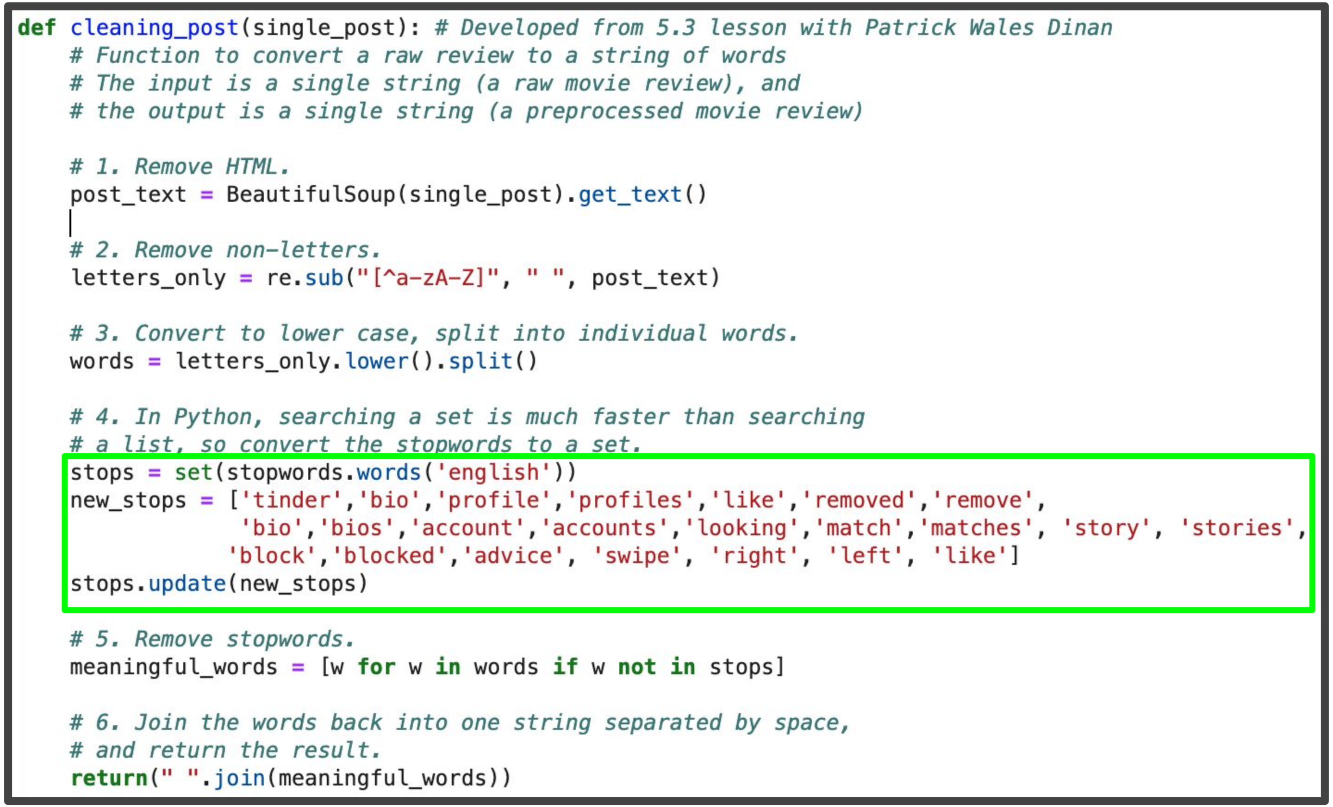
likes: sleep paralysis demon parties

ideal first date: forcing abducted male specimen to eat Ben n Jerry’s in a dirty gray dome

ideal man: a good sample human male

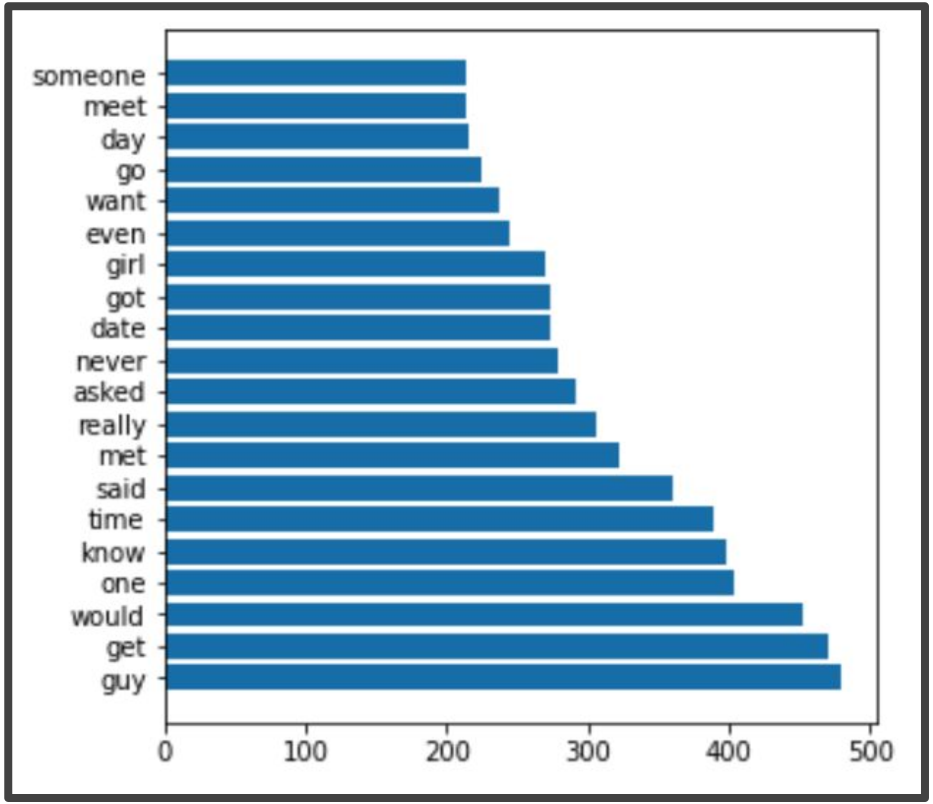
Example “Tinder Stories” Subreddit User

IV. Tokenizing and Preprocessing Data



```
def cleaning_post(single_post): # Developed from 5.3 lesson with Patrick Wales Dinan
    # Function to convert a raw review to a string of words
    # The input is a single string (a raw movie review), and
    # the output is a single string (a preprocessed movie review)

    # 1. Remove HTML.
    post_text = BeautifulSoup(single_post).get_text()
    
    # 2. Remove non-letters.
    letters_only = re.sub("[^a-zA-Z]", " ", post_text)

    # 3. Convert to lower case, split into individual words.
    words = letters_only.lower().split()

    # 4. In Python, searching a set is much faster than searching
    # a list, so convert the stopwords to a set.
    stops = set(stopwords.words('english'))
    new_stops = ['tinder','bio','profile','profiles','like','removed','remove',
                  'bio','bios','account','accounts','looking','match','matches', 'story', 'stories',
                 'block','blocked','advice', 'swipe', 'right', 'left', 'like']
    stops.update(new_stops)

    # 5. Remove stopwords.
    meaningful_words = [w for w in words if w not in stops]

    # 6. Join the words back into one string separated by space,
    # and return the result.
    return(" ".join(meaningful_words))
```

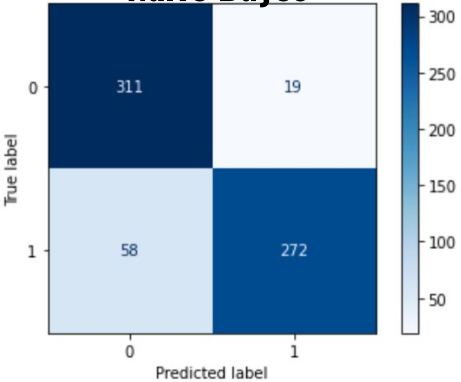
IV. Tokenizing and Preprocessing Data



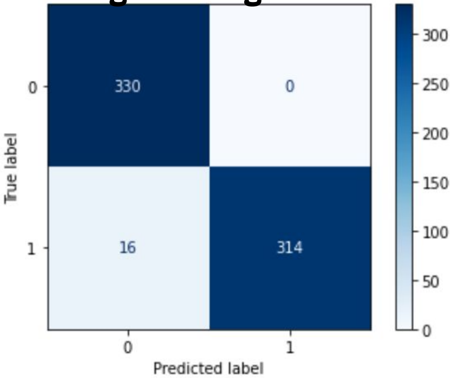
#BeautifulSoup

V. Modeling Data

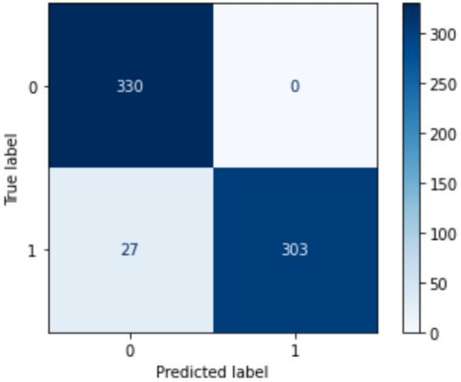
Naive Bayes



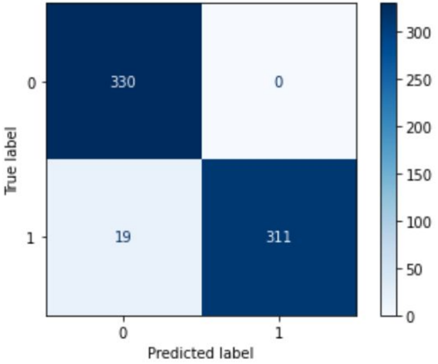
Logistic Regression

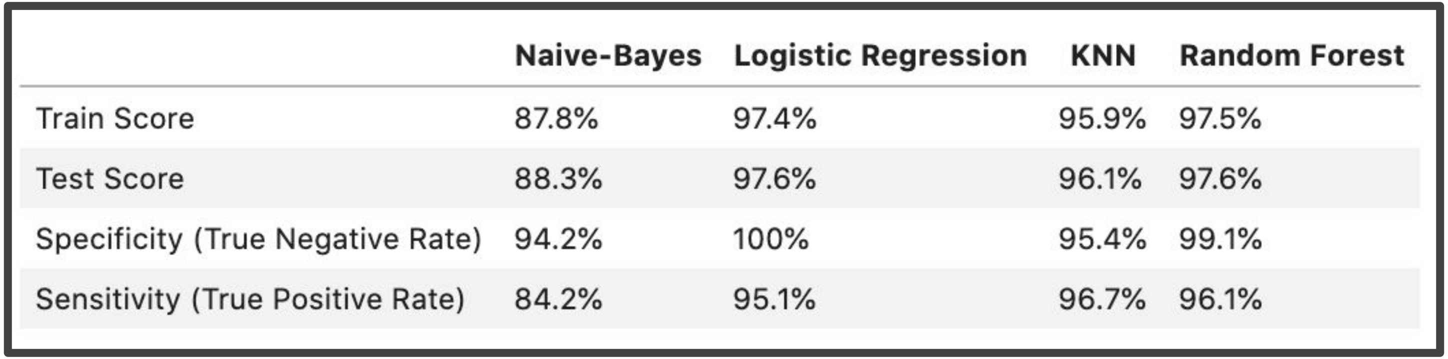


KNN



Random Forest





	Naive-Bayes	Logistic Regression	KNN	Random Forest
Train Score	87.8%	97.4%	95.9%	97.5%
Test Score	88.3%	97.6%	96.1%	97.6%
Specificity (True Negative Rate)	94.2%	100%	95.4%	99.1%
Sensitivity (True Positive Rate)	84.2%	95.1%	96.7%	96.1%

my Naive-Bayes scores:

VI. Limits of Models

- Although we can try to infer a gender-based difference between “Tinder” and “Tinder Stories” Subreddits (gendered language, images vs. text),
    - Further modeling and hypothesis testing needed to confirm
- These models do not take the entire subreddits, only 1000 posts
    - The full dataset may change our results
    - Language and culture both change over time (esp. slang)

Models tell us where to look, not how to go about things

Models are a starting point for how to revolutionize Tinder User Experience, not the endgoal.

VII. Conclusions and Recommendations

By investigating each subreddit with gender in mind, Tinder programmers can adjust Tinder to appeal more to each predominant gender's (subreddits) concerns.

- The second most common word in the male-dominated subreddit "Tinder" is "removed,"
    - Male Tinder users are scared of being “removed” or “blocked.”
    - So, we can have Tinder include more guidelines for good chatting behavior to prevent blocking issues.
- A very common word in the female-dominated subreddit "Tinder Stories" is "date."
    - We can include a bio question on our app detailing "Cool First, Second, and Third Date Ideas?"
    - Both genders can express which dating environment is personally ideal for them.

VII. Conclusions and Recommendations

- Despite the extreme similarities of the subreddits for Tinder and Tinder Stories, classification modeling is a form of supervised machine learning powerful enough to differentiate between these posts with a high degree of accuracy.
- Due to its analytical interpretability of which words are most popular, classification modeling for natural language processing will let us guide our company into the future with even more stable business decisions based on statistical fact.

VIII. Works Sourced

- General Assembly Data Science Immersive 2020
- Pushshift's API
- https://youtu.be/AcrjEWsMi_E
- https://www.reddit.com/r/Tinder/
- https://www.reddit.com/r/tinderstories/