Comparison of r/WFH and r/digitalnomad using Natural Language Processing

Aridania Gerardo

Problem Statement

Reddit is one of the largest social platforms that is unlike Twitter, Facebook or any other platform, it gets a constant flow of communities posting new discussions daily every minute. It can be intimidating to join a subreddit community that you don't know if it is right for you. By using r/WFH and r/digitalnomads, I want to see if I can create a model that could predict if I were to write a post, in which subreddit group would it be more likely to appear?

Dataset Subreddit's Analyzed

r/WFH

- "Welcome to 'WFH - Working From Home,' the subreddit dedicated to those of us who work from home, be it for yourself or a company. Learn tips and tricks to make yourself more productive, avoid distractions and generally make your experience a more positive one."
- 28.4k Members
- Created Dec 8, 2010

r/digitalnomad

- "Digital Nomads are individuals that leverage technology in order to work remotely and live an independent and nomadic lifestyle."
- 1.7m Memberse
- Created Oct 15, 2009

- Number of features: 9558
- Number of record: 9570 Unique Documents - documents~ 5000 post titles, text, and comments for first subreddit using PRAW
- Shape of Data: 9558rows × 2 columns
- Data was scraped from reddit on 07/29/2022

Data Wrangling using Reddit's API through PRAW

```
# Instantiate Reddit using PRAW.
# API Pull Set-Up of Comments with Praw
# reddit = praw.Reddit(
#     client_id=" ",
#     client_secret=" ",
#     password=" ",
#     user_agent="Comment Extraction (by u/USERNAME)",
#     username=" ",
#                     )
```

```
max_docs = 5000

CombinedSubName =  CombinedSubsList[0]
subreddit = reddit.subreddit(CombinedSubName).hot(limit=max_docs)
subreddit_text = []

for submission in subreddit:
    if len(subreddit_text) > max_docs:
        break
    subreddit_text.append(submission.title)
    subreddit_text.append(submission.selftext)
    submission.comments.replace_more(limit = max_docs)
    for comment in submission.comments.list():
        if len(subreddit_text) > max_docs:
            break
        subreddit_text.append(comment.body)
        print(len(subreddit_text))
```

EDA

In the Exploratory Data Analysis section I go through the data frames created for the post titles and comments from July 29th 2022 to the last 5,000 documents from each subreddit. I remove duplicate texts and the first column of each subreddit of where it is the introduction. All text in the data frame is unique.

Clean Data and Verify that there are ~10,000 Unique Documents

```
len(df["text"].unique())
```

```
9558
```

```
print(df['subreddit'].value_counts())
```

```
WFH             5001
digitalnomad    5001
Name: subreddit, dtype: int64
```

Data Modeling

```
pipeLR = Pipeline([
    ('vect', CountVectorizer()),
    ('model', LogisticRegression())
])

params = {
    'vect__min_df':[3,4],
    'vect__stop_words':[None ,'english'],
    'model__penalty':['l1','l2'],
    'model__C':[0.1, 1, 10]
}

gs = GridSearchCV(pipe, params, cv=5, verbose=2, n_jobs=-1)

gs.fit(X_train, y_train)

print('Best Params: ',gs.best_params_)

print('Best Estimator Score Train: ', gs.best_estimator_.score(X_train, y_train))
print('Best Estimator Score Test: ', gs.best_estimator_.score(X_test, y_test))
```

```
Best Params:  {'model__C': 0.1, 'model__penalty': 'l2', 'vect__min_df': 3, 'vect__stop_words': 'english'}
Best Estimator Score Train:  0.875242501119236
Best Estimator Score Test:  0.8118751893365647
```

The gridsearch for Logistic Regression was the most successful and best scores that did not overfit the data.

Top 10 WFH Features

r/wfh
- Work
- Business
- Serious

wfh	-7.713109
office	-5.478953
job	-4.187229
home	-3.059532
work	-2.980772
desk	-2.842666
company	-2.389099
jobs	-2.283791
working home	-2.112716
day	-2.041842

Top 10 digitalnomads Features

r/digitalnomads
- ❏ Travel
- ❏ Adventures

country	3.573275
airbnb	3.425779
month	2.863306
nomad	2.721290
places	2.641610
place	2.640761
city	2.639234
visa	2.624555
dn	2.479216
countries	2.418015

Trained Logistic Regression Classifier

```
test_post = ["office"]
test_counts = vectorizer.transform(test_post)
print(Log.predict(test_counts))
```

```
['WFH']
```

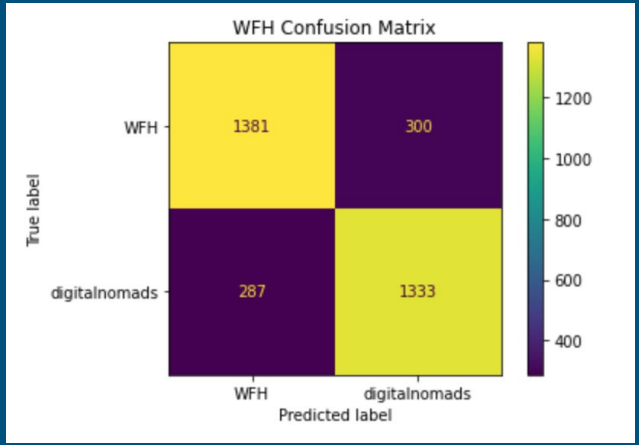
```
test_post = ["contract"]
test_counts = vectorizer.transform(test_post)
print(Log.predict(test_counts))
```

```
['digitalnomad']
```

The trained Logistic Regression Classifier used the the test string “office” and placed it as if it would originate under the subreddit r/WFH. Although, if given “contract”, it places the string if it would originate under the subreddit r/digitalnomad.

Performance of Algorithm with a Confusion Matrix

The confusion matrix shows here that digitalnomads is a much larger subreddit group but both digitalnomads and WFH are predicted the same amount.

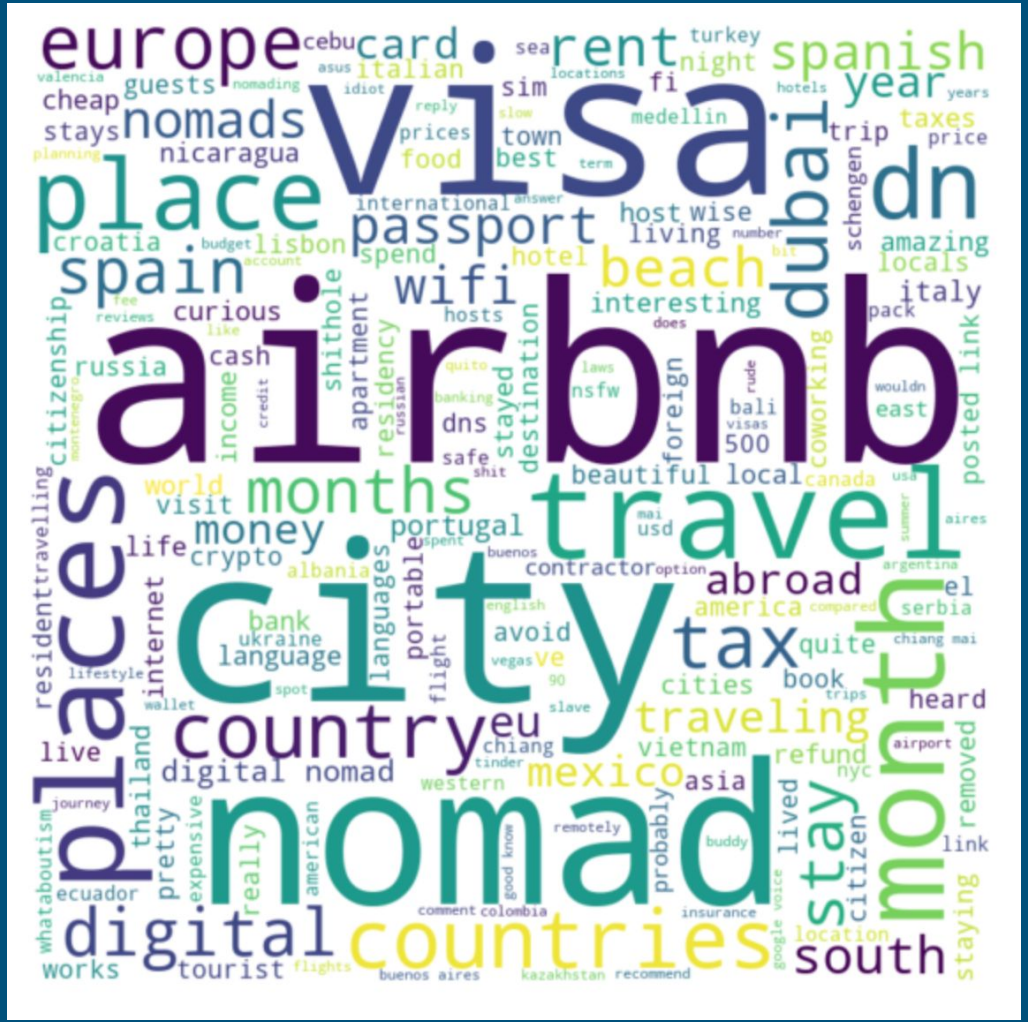



```
print(classification_report(y_test, preds))
```

	precision	recall	f1-score	support
WFH	0.80	0.85	0.82	1681
digitalnomad	0.83	0.77	0.80	1620
accuracy			0.81	3301
macro avg	0.81	0.81	0.81	3301
weighted avg	0.81	0.81	0.81	3301

Data Visualization

If we observe the word clouds , we can see that the wordclouds from r/digitalnomads follow a pattern that we would be able to distinguish which one is more adventurous with the work from home lifestyle while the other is more serious of working from their literal home. The word correlation airbnb, visa, city, and nomad being the most prominent for example.



Summary

Using Natural Language Processing methods,I was able to analyze both subreddit's r/WFH and r/digitalnomads to train a Natural Language Processing model to identify what subreddit a test string is more likely to originate from (subreddit group). With further effort, I could observe more specific subreddits or users and look at sentiment analysis and add a view of how individual users change over time their subreddit digital fingerprint.

A good next step could be to use PRAW to gather posts from more groups such as r/WFH, r/workfromhome, and r/digitalnomands and analyze the intensity of the interaction connections between these three subreddits and visualize the promising interconnections as the connections extend to outer groups.