Download Datasets

```
!wget -q -cO - https://zenodo.org/records/3941387/files/teaching_2018_features_tfidf_256.
!wget -q -cO -  https://zenodo.org/records/3941387/files/teaching_2019_features_tfidf_256
!wget -q -cO -  https://zenodo.org/records/3941387/files/mentalhealth_2018_features_tfidf
```

Import packages

```
import pandas as pd
from collections import Counter
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.decomposition import KernelPCA, PCA
```

Load data

```
teaching_2018 = pd.read_csv('teaching_2018.csv')
teaching_2019 = pd.read_csv('teaching_2019.csv')
mental = pd.read_csv('mental.csv')
```

```
teaching_2018#.head(2)
```

	subreddit	author	date	post	automated_readability_i
0	teaching	wdead	2018/01/01	Looking for BIG digital timers for my 7th grad...	6.20
1	teaching	tiger-bulldog1318	2018/01/02	School turnaround from the teacher’s perspecti...	7.19
2	teaching	PMcommenter	2018/01/02	Where on high school & college campuses ca...	6.96
3	teaching	amylizzie12	2018/01/02	What have you used to keep your students busy ...	7.13
4	teaching	ravensandcrowsohmy	2018/01/02	Flexible, Free Vocabulary App/Website with a F...	7.27
...	...	...	...	...	
544	teaching	Markenheimer15	2018/04/20	Looking for advice from someone who has switch...	6.40
545	teaching	ahoradevoar	2018/04/20	[UK] Another job interview with a teaching rec...	3.25
546	teaching	Amandamc2315	2018/04/20	Need Advice? I noticed a lot of people post he...	7.29
547	teaching	notwutiwantd	2018/04/20	Oh, you only want singles? OK! I posted this o	4.52

```
teaching_2019.head(2)
```

	subreddit	author	date	post	automated_readability_index	coleman_l
0	teaching	Nakatsukasa	2019/01/01	First time teaching class of 9-13 Hello, I'm c...	9.250539	
1	teaching	zdnewcomb	2019/01/01	How do I get my 6th graders to care about poet...	4.613466	

2 rows × 350 columns

```
mental.head(2)
```

	subreddit	author	date	post	automated_readability_index	cole
0	mentalhealth	LilUziVertsAutotune	2018/01/01	Any idea what this is? So I came here for awns...	1.198856	
1	mentalhealth	Kyzzen	2018/01/01	Advice, please. I've been getting so easily en...	3.750551	

2 rows × 350 columns

Combine data together

```
df = pd.concat([teaching_2018, teaching_2019, mental])
del teaching_2018, teaching_2019, mental
```

```
df
```

	subreddit	author	date	post	automated_readability
0	teaching	wdead	2018/01/01	Looking for BIG digital timers for my 7th grad...	6
1	teaching	tiger-bulldog1318	2018/01/02	School turnaround from the teacher’s perspecti...	7
2	teaching	PMcommenter	2018/01/02	Where on high school & college campuses ca...	6
3	teaching	amylizzie12	2018/01/02	What have you used to keep your students busy ...	7
4	teaching	ravensandcrowsohmy	2018/01/02	Flexible, Free Vocabulary App/Website with a F...	7
...	...	...	...	...	
3355	mentalhealth	riverisaberry	2018/04/20	Does Anyone Have Chronic Hyperventilation Synd...	-0
3356	mentalhealth	eshmaalfatin	2018/04/20	Psychiatrist in Dubai & Abu Dhabi - Genera...	18
3357	mentalhealth	themonstrumologist	2018/04/20	My best friend just brought this to my attenti...	0
3358	mentalhealth	scndplace	2018/04/20	Why do I get depressed instead of sad? i wanna...	1
3359	mentalhealth	anniehall330	2018/04/20	What are the events that made you traumatised ...	4

4736 rows × 350 columns

```
j = 0
for i in df.columns:
  if i[:5] == 'tfidf':
    print(j)
  j +=1
```

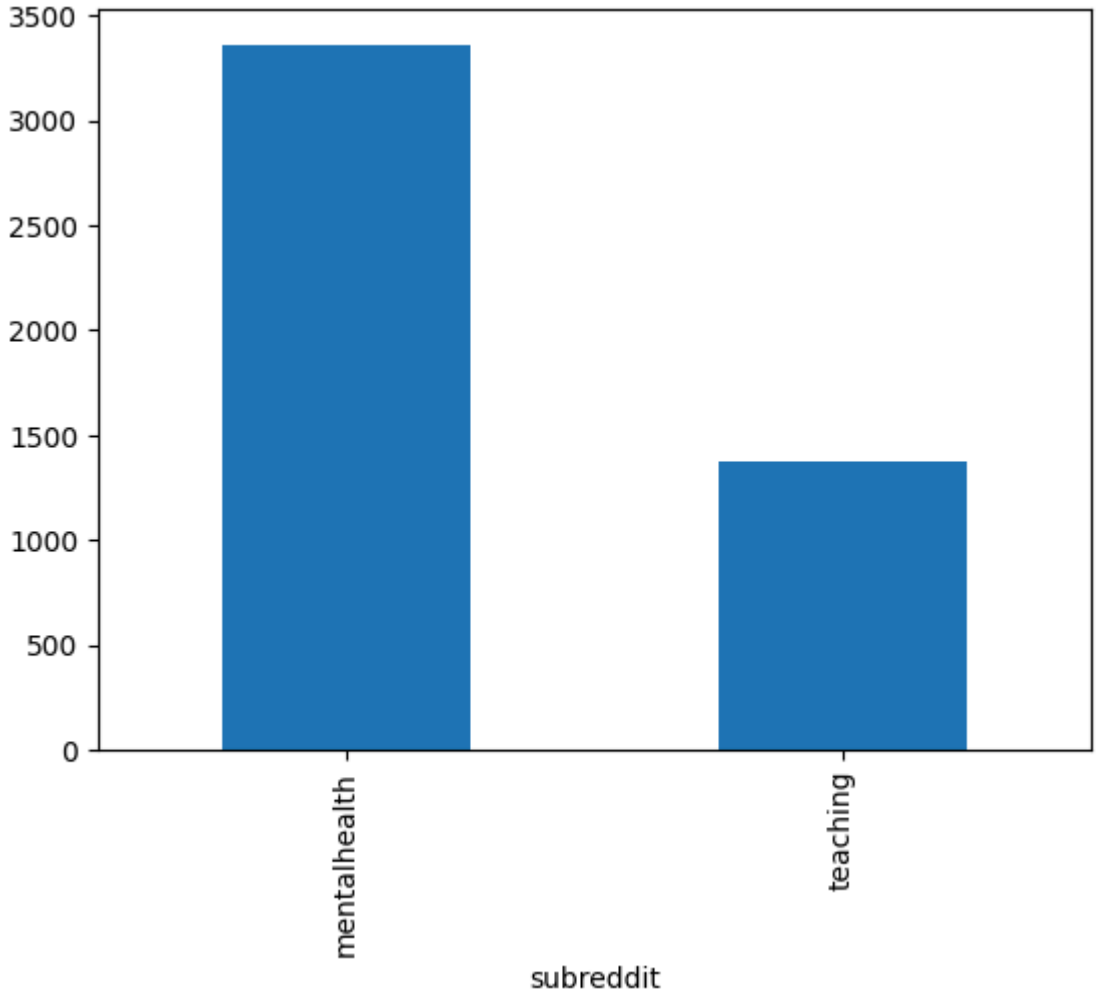
```
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
```

```
147
148
149
150
```

Data exploration

```
df.subreddit.value_counts().plot(kind='bar')
```

```
<Axes: xlabel='subreddit'>
```




```
df[df.subreddit == 'mentalhealth']
```

	subreddit	author	date	post	automated_readabilit
0	mentalhealth	LilUziVertsAutotune	2018/01/01	Any idea what this is? So I came here for awns...	
1	mentalhealth	Kyzzen	2018/01/01	Advice, please. I've been getting so easily en...	
2	mentalhealth	somerandomperson93	2018/01/01	Can aggressive behavior in humans be modified ...	
3	mentalhealth	121802	2018/01/01	I heard my parents having sex and i need serio...	
4	mentalhealth	Mewdot	2018/01/01	From the inside out This will probably be a lo...	
...	...	...	...	...	
3355	mentalhealth	riverisaberry	2018/04/20	Does Anyone Have Chronic Hyperventilation Synd...	-(
3356	mentalhealth	eshmaalfatin	2018/04/20	Psychiatrist in Dubai & Abu Dhabi - Genera...	18
3357	mentalhealth	themonstrumologist	2018/04/20	My best friend just brought this to my attenti...	(
3358	mentalhealth	scndplace	2018/04/20	Why do I get depressed instead of sad? i wanna...	
3359	mentalhealth	anniehall330	2018/04/20	What are the events that made you traumatised ...	

3360 rows × 350 columns

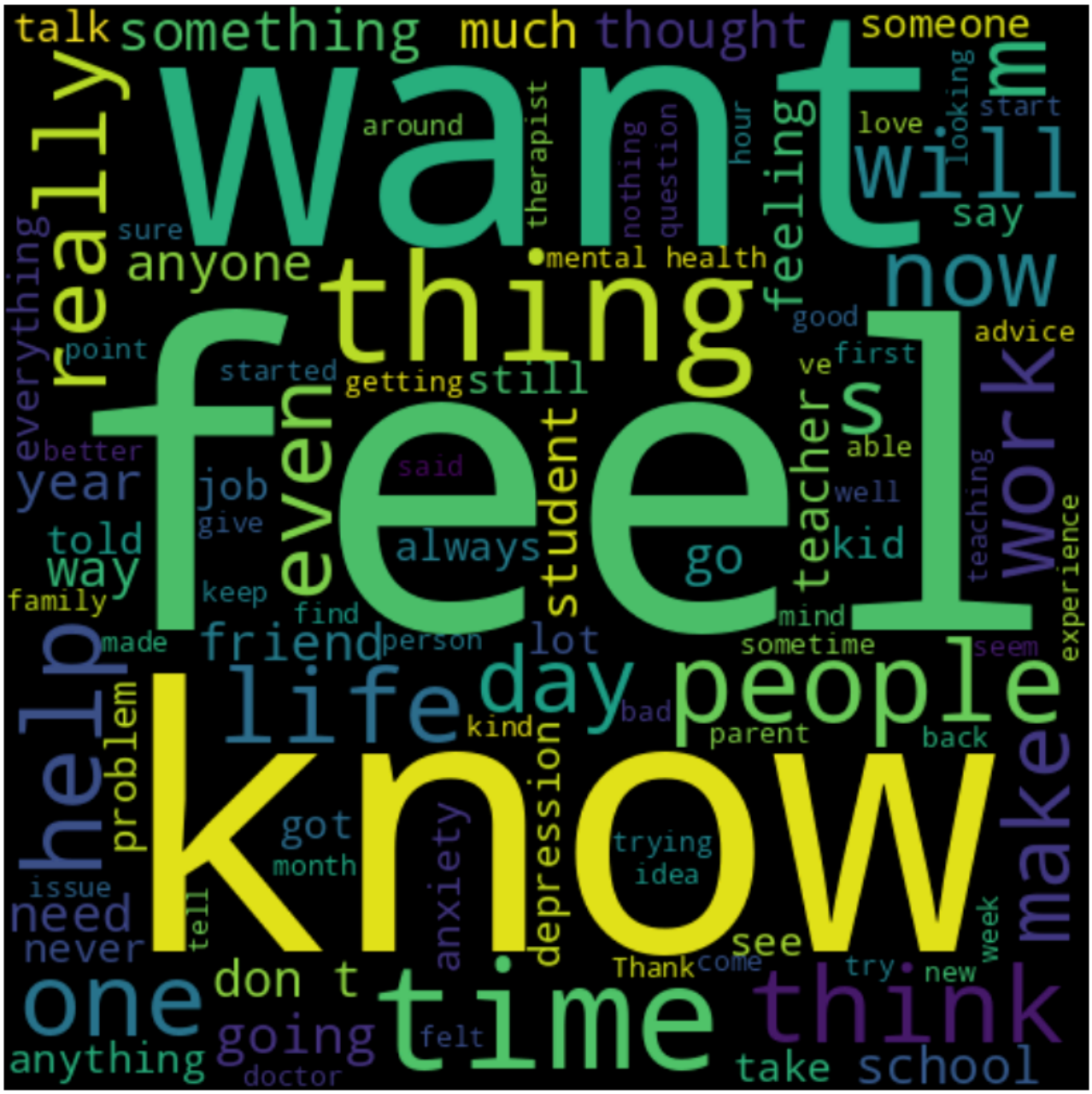
```
txt = ' '.join(df[df.subreddit == 'teaching'].post.values)
word_list = txt.split()
Counter(word_list).most_common()
```

```
[('I', 7943),
 ('to', 6939),
 ('the', 6099),
 ('and', 5863),
 ('a', 5715),
 ('in', 3447),
 ('of', 3422),
 ('my', 2788),
 ('for', 2679),
 ('that', 2127),
 ('is', 2076),
 ('have', 2004),
 ('with', 1628),
 ('this', 1399),
 ('be', 1352),
 ('but', 1347),
 ('on', 1323),
 ('it', 1298),
 ('you', 1246),
 ("I'm", 1083),
 ('was', 1070),
 ('are', 1038),
 ('at', 1038),
 ('me', 1036),
 ('as', 1031),
 ('or', 1024),
 ('not', 971),
 ('am', 969),
 ('do', 937),
 ('school', 892),
 ('teaching', 888),
 ('so', 875),
 ('about', 867),
 ('they', 823),
 ('an', 812),
 ('if', 751),
 ('just', 737),
 ('would', 733),
 ('like', 725),
 ('what', 695),
 ('teacher', 688),
 ('can', 666),
 ('students', 662),
 ('get', 660),
 ('from', 635),
 ('know', 632),
 ('any', 628),
 ('out', 582),
 ('all', 573),
 ('been', 570),
 ('some', 564),
 ('them', 545),
 ('I’m', 543),
 ('how', 536),
 ('their', 507),
 ('want', 503),
 ('will', 491),
 ('had', 470),
```

```
txt = ' '.join(df[df.subreddit == 'mentalhealth'].post.values)
word_list = txt.split()
Counter(word_list).most_common()[:20]
```

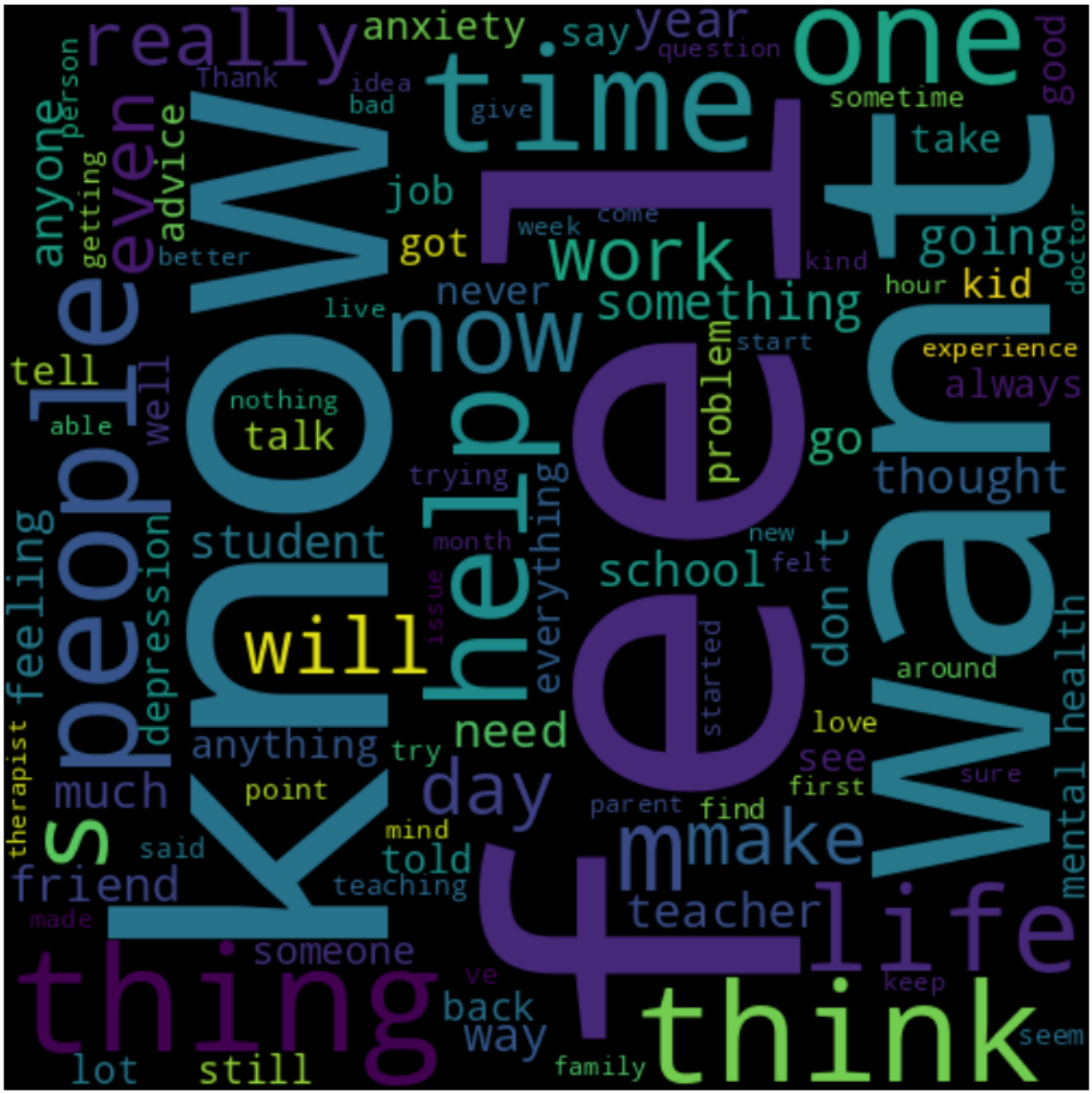
```
[('I', 40642),
 ('to', 25944),
 ('and', 24392),
 ('a', 17054),
 ('the', 16193),
 ('my', 13687),
 ('of', 12370),
 ('that', 8708),
 ('in', 8533),
 ('have', 7296),
 ('for', 7250),
 ('is', 7168),
 ('it', 6905),
 ('with', 6838),
 ('me', 6760),
 ('but', 6553),
 ('was', 5945),
 ('this', 5085),
 ('like', 5069),
 ("I'm", 4852)]
```

```
wordcloud = WordCloud(
        background_color = 'black',
        width = 500,
        height = 500,
        max_words=100,
        stopwords = set(STOPWORDS)).generate(str(' '.join(df.post.values)))

fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.imshow(wordcloud, interpolation='bilinear')
ax.axis("off")
fig.show()
```



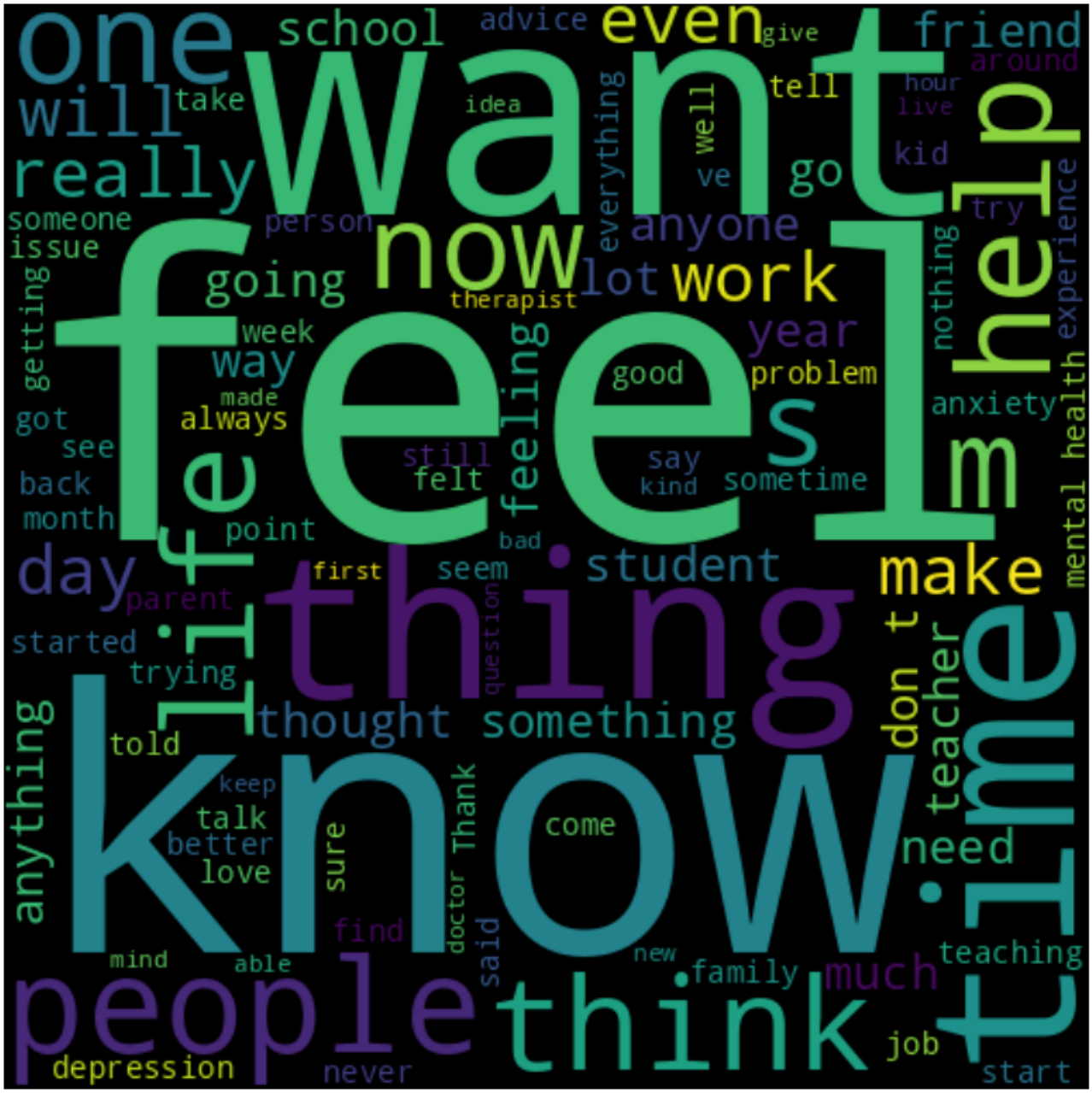

```
# สร้าง Wordcloud ประเภท Teaching ด้วยตนเอง
wordcloud = WordCloud(
        background_color = 'black',
        width = 500,
        height = 500,
        max_words=100,
        stopwords = set(STOPWORDS)).generate(str('Teaching'.join(df.post.values)))

fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.imshow(wordcloud, interpolation='bilinear')
```

```
ax.axis("off")
fig.show()
```




```
# สร้าง Wordcloud ประเภท Mentalhealth ด้วยตนเอง
wordcloud = WordCloud(
        background_color = 'black',
        width = 500,
        height = 500,
        max_words=100,
        stopwords = set(STOPWORDS)).generate(str('Mentalhealth'.join(df.post.values)))
```

```
fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.imshow(wordcloud, interpolation='bilinear')
ax.axis("off")
fig.show()
```



Feature extraction

```
vectorizer = CountVectorizer()
```

```
# ให้ทำการแปลงข้อความให้เป็น Feature โดยการระบุข้อมูลที่ใช้ในการแปลงให้ถูกต้อง

# X = vectorizer.fit_transform(_____)

X = vectorizer.fit_transform(df.post.values)


print(X.shape)
print(X.toarray())
print(vectorizer.get_feature_names_out())
```

```
(4736, 21793)
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
['00' '000' '00pm' ... 'مدرسة' 'محمد' 'ㄅㄆㄇㄈ']
```

Split Data into training and test sets

```
df['label'] = pd.Categorical(df.subreddit).codes


df
```

	subreddit	author	date	post	automated_readability
0	teaching	wdead	2018/01/01	Looking for BIG digital timers for my 7th grad...	6
1	teaching	tiger-bulldog1318	2018/01/02	School turnaround from the teacher’s perspecti...	7
2	teaching	PMcommenter	2018/01/02	Where on high school & college campuses ca...	6
3	teaching	amylizzie12	2018/01/02	What have you used to keep your students busy ...	7
4	teaching	ravensandcrowsohmy	2018/01/02	Flexible, Free Vocabulary App/Website with a F...	7
...	...	...	...	...	
3355	mentalhealth	riverisaberry	2018/04/20	Does Anyone Have Chronic Hyperventilation Synd...	-0
3356	mentalhealth	eshmaalfatin	2018/04/20	Psychiatrist in Dubai & Abu Dhabi - Genera...	18
3357	mentalhealth	themonstrumologist	2018/04/20	My best friend just brought this to my attenti...	0
3358	mentalhealth	scndplace	2018/04/20	Why do I get depressed instead of sad? i wanna...	1
3359	mentalhealth	anniehall330	2018/04/20	What are the events that made you traumatised ...	4

4736 rows × 351 columns

```
# ให้ระบุคอลัมน์ ที่จะใช้เป็น label เพื่อที่จะใช้ในการสอนและทำนาย

y = df.label
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42)
```

Modelling

```
log = LogisticRegression(random_state=0).fit(X_train, y_train)

print(log.predict(X_test[:10]))
print(log.predict_proba(X_test[:10]))
print(log.score(X_test, y_test))
```

```
[0 0 0 0 0 1 1 1 0 0]
[[1.00000000e+00 3.09881006e-11]
 [9.86735561e-01 1.32644392e-02]
 [9.45501761e-01 5.44982389e-02]
 [9.86729070e-01 1.32709295e-02]
 [9.16042616e-01 8.39573837e-02]
 [1.10372408e-03 9.98896276e-01]
 [5.20349080e-04 9.99479651e-01]
 [1.20955290e-03 9.98790447e-01]
 [9.99999873e-01 1.26532163e-07]
 [9.99999663e-01 3.37263137e-07]]
0.971850809289233
```

ให้ทดลองสร้างโมเดล ด้วย Algorithms อื่นๆ ด้วยตัวเอง อีก 2 โมเดล

```
from sklearn.ensemble import RandomForestClassifier

# RandomForestClassifier model
clf1 = RandomForestClassifier(random_state=0).fit(X_train, y_train)

print(clf1.predict(X_test[:10]))
print(clf1.predict_proba(X_test[:10]))
print(clf1.score(X_test, y_test))
```

```
[0 0 0 0 0 0 1 1 0 0]
[[0.93 0.07]
 [0.78 0.22]
 [0.8  0.2 ]
 [0.84 0.16]
 [0.81 0.19]
 [0.53 0.47]
 [0.46 0.54]
 [0.35 0.65]
 [0.96 0.04]
 [0.97 0.03]]
0.9268121041520057
```

```
from sklearn.ensemble import GradientBoostingClassifier

# GradientBoostingClassifier model
clf2 = GradientBoostingClassifier(random_state=0).fit(X_train, y_train)

print(clf2.predict(X_test[:10]))
print(clf2.predict_proba(X_test[:10]))
print(clf2.score(X_test, y_test))
```

```
[0 0 0 0 0 1 1 1 0 0]
[[0.98553097 0.01446903]
 [0.83825515 0.16174485]
 [0.76252921 0.23747079]
 [0.93679944 0.06320056]
 [0.91158509 0.08841491]
 [0.04204465 0.95795535]
 [0.04645257 0.95354743]
 [0.00929147 0.99070853]
 [0.98229542 0.01770458]
 [0.99032615 0.00967385]]
0.9422941590429276
```

Evaluation

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล


y_true = y_test
y_pred = log.predict(X_test)


target_names = ['MentalHealth', 'Teaching']
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

 MentalHealth       0.97      0.99      0.98       993
     Teaching       0.98      0.93      0.95       428

     accuracy                           0.97      1421
    macro avg       0.97      0.96      0.97      1421
 weighted avg       0.97      0.97      0.97      1421
```

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล clf1


y_true = y_test
y_pred = clf1.predict(X_test)


target_names = ['MentalHealth', 'Teaching']
```

```
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

 MentalHealth       0.91      1.00      0.95       993
     Teaching       0.99      0.77      0.86       428

     accuracy                           0.93      1421
    macro avg       0.95      0.88      0.91      1421
 weighted avg       0.93      0.93      0.92      1421
```

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล clf2
```

```
y_true = y_test
y_pred = clf2.predict(X_test)

target_names = ['MentalHealth', 'Teaching']
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

 MentalHealth       0.93      0.99      0.96       993
     Teaching       0.98      0.82      0.90       428

     accuracy                           0.94      1421
    macro avg       0.95      0.91      0.93      1421
 weighted avg       0.94      0.94      0.94      1421
```

ให้สร้างโมเดลด้วยตัวเองโดยใช้ LIWC features ที่มีให้ใน Dataset เพื่อใช้ในการ Train และ Test โมเดลด้วยตัวเอง

```
# คอลัมน์ LIWC features จะอยู่ในคอลัมน์ที่ 4 ถึง 94

df.columns[4:94]
```

```
Index(['automated_readability_index', 'coleman_liau_index',
       'flesch_kincaid_grade_level', 'flesch_reading_ease', 'gulpease_index',
       'gunning_fog_index', 'lix', 'smog_index', 'wiener_sachtextformel',
       'n_chars', 'n_long_words', 'n_monosyllable_words',
       'n_polysyllable_words', 'n_sents', 'n_syllables', 'n_unique_words',
       'n_words', 'sent_neg', 'sent_neu', 'sent_pos', 'sent_compound',
       'economic_stress_total', 'isolation_total', 'substance_use_total',
       'guns_total', 'domestic_stress_total', 'suicidality_total',
       'punctuation', 'liwc_1st_pers', 'liwc_2nd_pers', 'liwc_3rd_pers',
       'liwc_achievement', 'liwc_adverbs', 'liwc_affective_processes',
       'liwc_anger', 'liwc_anxiety', 'liwc_articles_article', 'liwc_assent',
       'liwc_auxiliary_verbs', 'liwc_biological', 'liwc_body',
       'liwc_causation', 'liwc_certainty', 'liwc_cognitive',
       'liwc_common_verbs', 'liwc_conjunctions', 'liwc_death',
       'liwc_discrepancy', 'liwc_exclusive', 'liwc_family', 'liwc_feel',
```

```
       'liwc_fillers', 'liwc_friends', 'liwc_future_tense', 'liwc_health',
       'liwc_hear', 'liwc_home', 'liwc_humans', 'liwc_impersonal_pronouns',
       'liwc_inclusive', 'liwc_ingestion', 'liwc_inhibition', 'liwc_insight',
       'liwc_leisure', 'liwc_money', 'liwc_motion', 'liwc_negations',
       'liwc_negative_emotion', 'liwc_nonfluencies', 'liwc_numbers',
       'liwc_past_tense', 'liwc_perceptual_processes',
       'liwc_personal_pronouns', 'liwc_positive_emotion', 'liwc_prepositions',
       'liwc_present_tense', 'liwc_quantifiers', 'liwc_relativity',
       'liwc_religion', 'liwc_sadness', 'liwc_see', 'liwc_sexual',
       'liwc_social_processes', 'liwc_space', 'liwc_swear_words',
       'liwc_tentative', 'liwc_time', 'liwc_total_functional',
       'liwc_total_pronouns', 'liwc_work'],
      dtype='object')
```

```
df.iloc[:, 4:94]
```

	automated_readability_index	coleman_liau_index	flesch_kincaid_grade_level	f
0	6.205652	6.216433	6.272283	
1	7.191104	10.905849	6.144719	
2	6.961154	8.008078	7.135026	
3	7.132836	9.284230	5.662866	
4	7.278080	8.537186	6.422609	
...	...	...	...	
3355	-0.313579	1.816079	2.206408	
3356	18.585441	21.801888	14.813529	
3357	0.995289	3.198568	2.559184	
3358	1.554975	3.657596	3.223969	
3359	4.084419	5.502131	5.342527	

4736 rows × 90 columns

```
X = df.iloc[:, 4:94]

df['label'] = pd.Categorical(df.subreddit).codes
y = df['label']


X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42)

log = LogisticRegression(random_state=0).fit(X_train, y_train)

print(log.predict(X_test[:10]))
print(log.predict_proba(X_test[:10]))
print(log.score(X_test, y_test))
```

```
[0 0 0 0 0 1 1 1 0 0]
[[9.99997891e-01 2.10869240e-06]
 [5.67882860e-01 4.32117140e-01]
 [6.43006119e-01 3.56993881e-01]
 [9.91066061e-01 8.93393940e-03]
 [8.26669573e-01 1.73330427e-01]
 [2.31554510e-02 9.76844549e-01]
 [3.82515020e-03 9.96174850e-01]
 [5.21654621e-03 9.94783454e-01]
 [9.99989076e-01 1.09237770e-05]
 [9.99696400e-01 3.03599767e-04]]
0.9176636171710063
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:469: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล


y_true = y_test
y_pred = log.predict(X_test)



target_names = ['MentalHealth', 'Teaching']
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

MentalHealth       0.93      0.95      0.94       993
    Teaching       0.88      0.84      0.86       428

    accuracy                           0.92      1421
   macro avg       0.91      0.89      0.90      1421
weighted avg       0.92      0.92      0.92      1421
```

ให้สร้างโมเดลด้วยตัวเองโดยใช้ TF-IDF features ที่มีให้ใน Dataset เพื่อใช้ในการ Train และ Test โมเดลด้วยตัวเอง

```
# คอลัมน์ LIWC features จะอยู่ในคอลัมน์ที่ 94 ถึง 350

df.columns[94:350]
```

```
Index(['tfidf_abl', 'tfidf_abus', 'tfidf_actual', 'tfidf_addict', 'tfidf_adhd',
       'tfidf_advic', 'tfidf_ago', 'tfidf_alcohol', 'tfidf_almost',
       'tfidf_alon',
```

```
            ...
            'tfidf_wish', 'tfidf_without', 'tfidf_wonder', 'tfidf_work',
            'tfidf_worri', 'tfidf_wors', 'tfidf_would', 'tfidf_wrong',
            'tfidf_x200b', 'tfidf_year'],
           dtype='object', length=256)
```

```
X = df.iloc[:, 94:350]
y = df['label']
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42)

log = LogisticRegression(random_state=0).fit(X_train, y_train)

print(log.predict(X_test[:10]))
print(log.predict_proba(X_test[:10]))
print(log.score(X_test, y_test))
```

```
[0 0 0 0 0 1 1 1 0 0]
[[0.99035168 0.00964832]
 [0.67261441 0.32738559]
 [0.67965393 0.32034607]
 [0.67705508 0.32294492]
 [0.85869157 0.14130843]
 [0.09668609 0.90331391]
 [0.3095809  0.6904191 ]
 [0.17407609 0.82592391]
 [0.9123111  0.0876889 ]
 [0.97088547 0.02911453]]
0.9289232934553132
```

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล
```

```
y_true = y_test
y_pred = log.predict(X_test)
```

```
target_names = ['MentalHealth', 'Teaching']
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

 MentalHealth       0.93      0.97      0.95       993
     Teaching       0.92      0.84      0.88       428

     accuracy                           0.93      1421
    macro avg       0.93      0.90      0.91      1421
 weighted avg       0.93      0.93      0.93      1421
```

ให้สร้างโมเดลด้วยตัวเองโดยใช้ TF-IDF และ LIWC features ที่มีให้ใน Dataset เพื่อใช้ในการ Train และ Test โมเดลด้วยตัวเอง

```
X = df.iloc[:, 94:350]
y = df['label']



X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42)

log = LogisticRegression(random_state=0).fit(X_train, y_train)

print(log.predict(X_test[:10]))
print(log.predict_proba(X_test[:10]))
print(log.score(X_test, y_test))
```

```
[0 0 0 0 0 1 1 1 0 0]
[[0.99035168 0.00964832]
 [0.67261441 0.32738559]
 [0.67965393 0.32034607]
 [0.67705508 0.32294492]
 [0.85869157 0.14130843]
 [0.09668609 0.90331391]
 [0.3095809  0.6904191 ]
 [0.17407609 0.82592391]
 [0.9123111  0.0876889 ]
 [0.97088547 0.02911453]]
0.9289232934553132
```

```
# ระบุ y_true และ y_pred ด้วยตัวเองให้ถูกต้อง เพื่อใช้แสดงประสิทธิภาพของโมเดล



y_true = y_test
y_pred = log.predict(X_test)



target_names = ['MentalHealth', 'Teaching']
print(classification_report(y_true, y_pred, target_names=target_names))
```