# Reddit NLP -
# Who Are the Dungeon Masters?

General Assembly DSI092021 - Project Three
by Mary Schindler (McAteer)

# Problem Statement:

We have been contracted by **Wizards of the Coast,** the owners of **Dungeons & Dragons,** to analyze text to see if we can determine whether the poster is simply a D&D fan or a D&D **Dungeon Master** (**DM**), a driver for game sessions. Reviewing comments from the subreddits for the 5th addition of the game ‘dndnext’ and a general subreddit for DMs ‘dmacademy’, we will attempt to build a model to fit their needs.

Why?

Wizards wants to start developing some new campaigns and modules. If they can determine what topics are trending, both among players and DMs, the business can better develop and market materials. As DMs are responsible for running games, the business can try catering to them specifically (they already do).

Who?

Wizards of the Coast’s Data Analytics Team

# The Data –

# What are we working with?

Using the **Pushshift API** we attempted to gather submissions from the subreddits **‘dndnext’** and **‘dmacademy’**.

Sadly,

- Only 3513 submissions in ‘dndnext’ going back to (GMT) Wednesday, June 25, 2014 12:22:55 AM
- Only 3090 submissions in ‘dmacademy’ going back to (GMT) Saturday, July 2, 2016 8:50:42 PM

So we looked at comments instead:

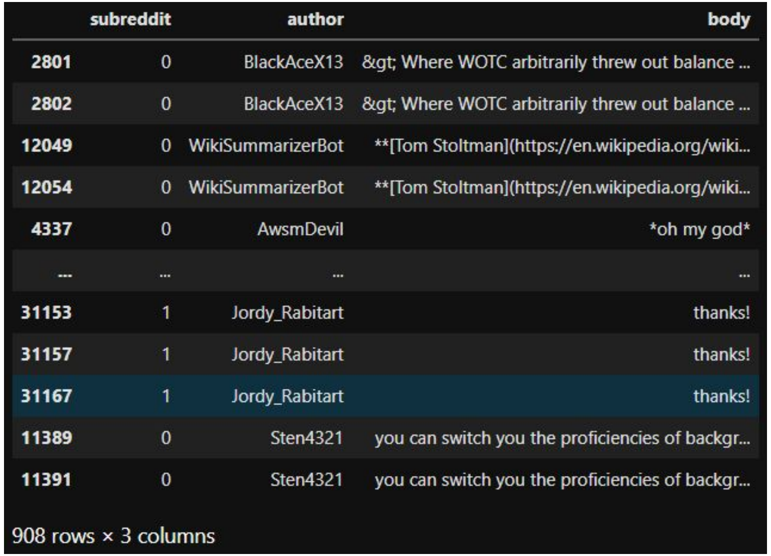
- 20,000 comments from ‘dndnext’ going back to (GMT) Friday, October 22, 2021 5:04:32 PM
- 19,999 comments from ‘dmacademy’ going back to (GMT) Tuesday, October 19, 2021 7:19:17 AM

These are *very* discussion-heavy, active subreddits.

# The Data –

# Cleaning It Up!

- Surprisingly there we no null values anywhere in our data (subreddit, author, and body)!
- There were 797 duplicate items, a total count of 908, so we took a closer look at those:

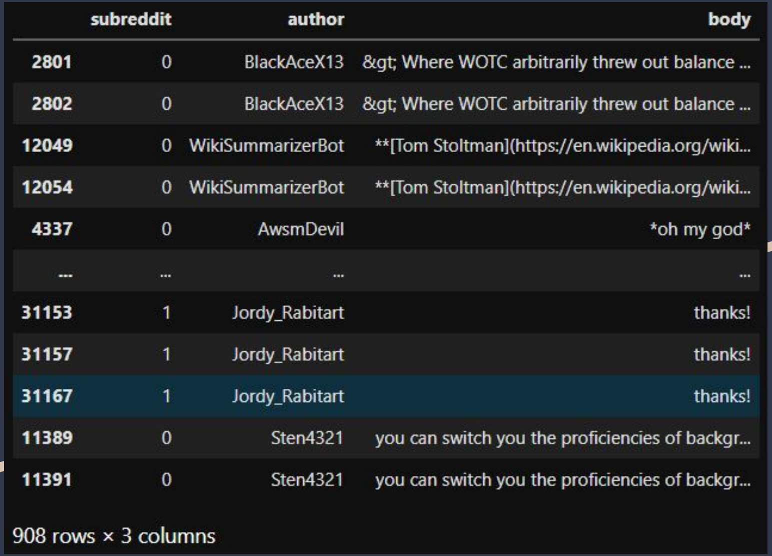


| | subreddit | author | body |
|---|---|---|---|
| 2801 | 0 | BlackAceX13 | &gt; Where WOTC arbitrarily threw out balance ... |
| 2802 | 0 | BlackAceX13 | &gt; Where WOTC arbitrarily threw out balance ... |
| 12049 | 0 | WikiSummarizerBot | **[Tom Stoltman](https://en.wikipedia.org/wiki... |
| 12054 | 0 | WikiSummarizerBot | **[Tom Stoltman](https://en.wikipedia.org/wiki... |
| 4337 | 0 | AwsmDevil | *oh my god* |
| ... | ... | ... | ... |
| 31153 | 1 | Jordy_Rabitart | thanks! |
| 31157 | 1 | Jordy_Rabitart | thanks! |
| 31167 | 1 | Jordy_Rabitart | thanks! |
| 11389 | 0 | Sten4321 | you can switch you the proficiencies of backgr... |
| 11391 | 0 | Sten4321 | you can switch you the proficiencies of backgr... |

908 rows × 3 columns

# The Data –

# Cleaning It Up!



| | subreddit | author | body |
|---|---|---|---|
| 2801 | 0 | BlackAceX13 | &gt; Where WOTC arbitrarily threw out balance ... |
| 2802 | 0 | BlackAceX13 | &gt; Where WOTC arbitrarily threw out balance ... |
| 12049 | 0 | WikiSummarizerBot | **[Tom Stoltman](https://en.wikipedia.org/wiki... |
| 12054 | 0 | WikiSummarizerBot | **[Tom Stoltman](https://en.wikipedia.org/wiki... |
| 4337 | 0 | AwsmDevil | *oh my god* |
| ... | ... | ... | ... |
| 31153 | 1 | Jordy_Rabitart | thanks! |
| 31157 | 1 | Jordy_Rabitart | thanks! |
| 31167 | 1 | Jordy_Rabitart | thanks! |
| 11389 | 0 | Sten4321 | you can switch you the proficiencies of backgr... |
| 11391 | 0 | Sten4321 | you can switch you the proficiencies of backgr... |

908 rows × 3 columns

The most frequent duplicate posts were as follows:

| | | |
|---|---|---|
| - | [deleted] | 497 |
| - | [removed] | 82 |
| - | Thank you! | 21 |
| - | Thank you. | 15 |
| - | Thanks! | 12 |
| - | No | 11 |
| - | Yes. | 10 |
| - | Yes | 10 |
| - | Thanks | 7 |
| - | I participate! | 5 |

I removed the duplicate posts with frequencies above 12: [deleted], [removed], Thank you!, Thank you., and Thanks!

In total 627 of the 908 duplicates were removed.

# The Data –

# Exploring Sentiments

Using nltk’s SentimentIntensityAnalyzer we took a look at the overall ‘sentiment’ of comments across both subreddits:

Highest Compound Score:

**‘dndnext’** - 'Seriously. Any character should be able to do their "thing" by level 5ish, in my opinion, even if they can\'t do it *well*. Everything after that should give you more options, stronger versions, or simply make your character "better".\n\nNobody wants to play DnD for months (if not years) just so they can *eventually* have a dragon around.'

Score: neg - 0.037, neu - 0.749, pos - 0.214, compound - 0.9991

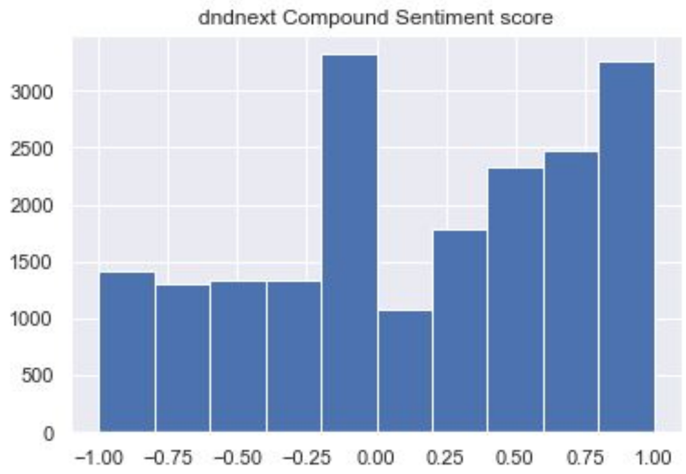
**‘dmacademy’** - 'Your goblin is going to be slightly less of a “good  wizard” than a gnome I guess, but they certainly aren’t just worse. You have better dexterity, which Wizards honestly really need, and you can get out of combat more easily and escape, something Wizards honestly really need.'

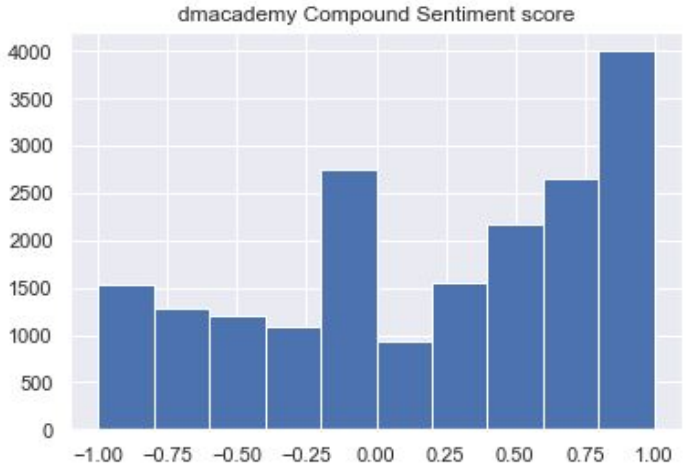
Score:  neg - 0.045, neu - 0.711, pos - 0.244, compound - 0.9996

# The Data –

# Exploring Sentiments

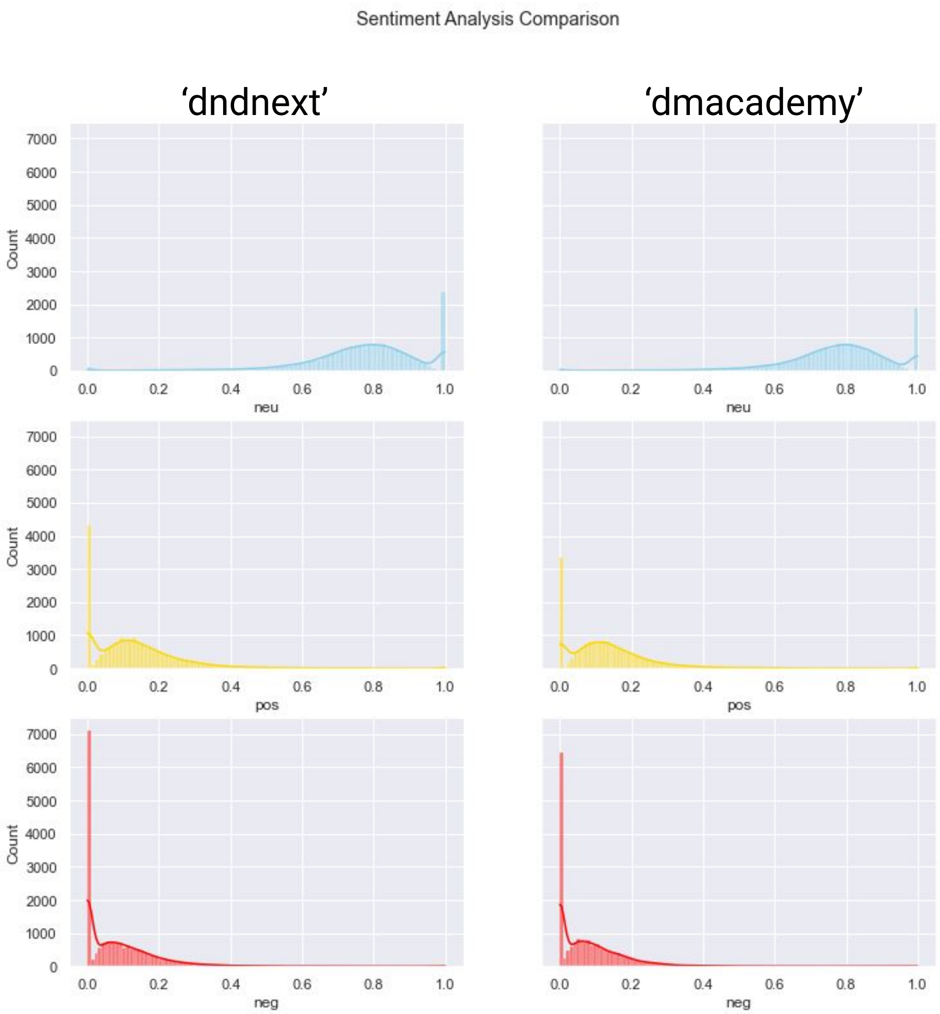
*please note these plots are not on the same scale





# The Data –

# Exploring Sentiments



# The Data –

# Exploring Sentiments

source: Kraken Dice

Overall, Sentiment Analysis is not a good way to observe this particular dataset. Too often commenters will use words that the text analyzer weights on opposite ends of the spectrum while the author clearly intends for the text to be read a certain way, eg.

“Our party viciously slaughtered the gang of evil goblins harming the village”

Score: neg: 0.49, neu: 0.381, pos: 0.129, compound: -0.8225

This statement is highly negative but depending on context, the party may have completed a positive task.

Who knew D&D players could be morally ambiguous?

# The Data –

# Extra Stop Words!

Considering how similar the posts in each subreddit may be, I added additional stop words, the union of the most popular words from each subreddit:

dndnext_words = cvec.get_feature_names()
&
dmacademy_words = cvec.get_feature_names()

The union of these lists resulted in 35 popular words in common:

'amp',  'campaign', 'character', 'combat', 'damage', 'did', 'does', 'game', 'going', 'good', 'just', 'know', 'level', 'like', 'lot', 'magic', 'make', 'need', 'party', 'people', 'play', 'player', 'players', 'point', 'really', 'roll', 'say', 'spell', 'thing', 'things', 'think', 'time', 'use', 'want', 'way'

These were added to the stop_words list.

# Modeling -

# Time to Work It!

Baseline Model - value_counts(normalize):

dndnext, probability of 0.502
dmacademy, probability of 0.498

Wizards of the Coast has contracted us so we need to produce a model that produces results more accurate than flipping a coin.

**Let's build and compare models!**

Set up two 'types' of models to be run:

- Pipelines using CountVectorizer, which counts frequencies of words
  &
- Pipelines using TF-IDFVectorizer, which gives weight to the importance of words.

# Modeling –

# Time to Work It!

Bag of Holding - Let’s toss our words in there!

```
Using: CountVectorizer(analyzer = 'word',
                       stop_words = stop_words,
                       max_features = 10_000)
```

We ran the following:

MultinomialNB(),

- Mean cross_val_score of 0.7369
- Training score of 0.7733
- Testing score of 0.7391

LogisticRegression(max_iter = 10_000),

- Mean cross_val_score of 0.7195
- Training score of 0.8588
- Testing score of 0.7243

RandomForestClassifier(random_state = 42)

- Mean cross_val_score of 0.7195
- Training score of 0.9905
- Testing score of 0.7149

# Modeling –

# Time to Work It!

As the pipeline with MultinomialNB produced the highest testing score with the least amount of overfitting toward the training data, I choose to run GridSearchCV on that pipeline.

The parameters we searched:

pipe_params:

- 'cvec__max_features': [None, 1_000, 2_000, 4_000, 8_000],
- 'cvec__min_df': [1, 2, 3],
- 'cvec__max_df': [1.0, .75, .85, .95],
- 'cvec__ngram_range': [(1, 1), (1, 2)]

After fitting we found the best parameters to be:

'cvec__max_df': 1.0,
'cvec__max_features': None,
'cvec__min_df': 1,
'cvec__ngram_range': (1, 2)

# Modeling –

# Time to Work It!

After fitting we found the best parameters to be:

```
 'cvec__max_df': 1.0,
 'cvec__max_features': None,
 'cvec__min_df': 1,
 'cvec__ngram_range': (1, 2)
```

What do these parameters mean?

max_df = ignored terms appearing in the defined percent of documents. GridSearch selected the default of 1, no terms are ignored

max_features = when None is selected the vectorizer builds a vocabulary that only consider the top max_features, ordered by term frequency across the corpus

min_df = ignored terms appearing infrequently. GridSearch selected the default of 1, no terms are ignored

ngram_range = determines the value of the upper and lower bounds of different word n-grams. Here GridSeached selected unigrams and bigrams (individual words and pairs of words).

# Modeling –

# Time to Work It!

## Additional n-Grams

Working with ngrams specifically did not increase the accuracy of the model however it did lower the bias between training and testing scores:

```
CountVectorizer(analyzer = 'word',
                stop_words = stop_words,
                max_features = 10_000,
                ngram_range = (1,3)
```

MultinomialNB()

- Mean cross_val_score of 0.7359
- Training score of 0.7680
- Testing score of 0.7365

LogisticRegression(max_iter = 10_000)

- Mean cross_val_score of 0.7210
- Training score of 0.8620
- Testing score of 0.7235

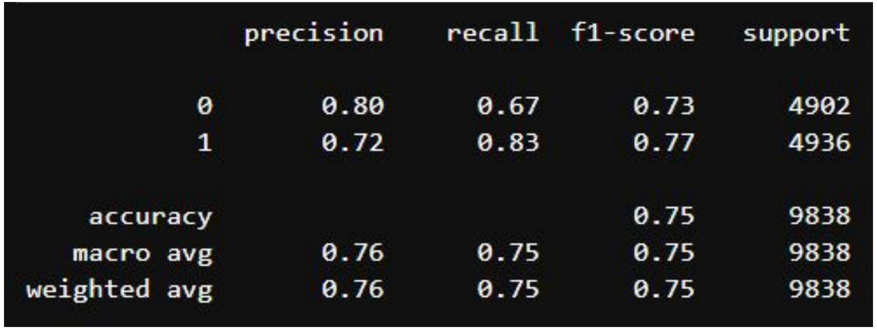
RandomForestClassifier(random_state = 42)

- Mean cross_val_score of 0.7123
- Training score of 0.9896
- Testing score of 0.7174

# Modeling – Evaluation

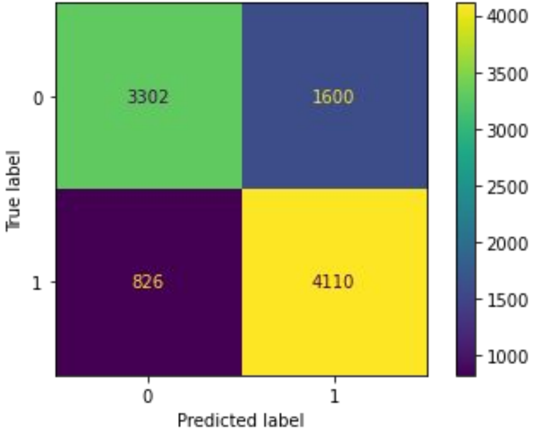
Evaluation of GridSearch’d CountVectorizer Model:

Classification Report:



```
              precision    recall  f1-score   support

           0       0.80      0.67      0.73      4902
           1       0.72      0.83      0.77      4936

    accuracy                           0.75      9838
   macro avg       0.76      0.75      0.75      9838
weighted avg       0.76      0.75      0.75      9838
```

Confusion Matrix:



# Modeling –

# Time to Work It!

```
Using: TfidfVectorizer(analyzer = "word",
                  stop_words = stop_words,
                  max_features = 10_000)
```

We ran the following:

MultinomialNB()

- Mean cross_val_score of 0.7388
- Training score of 0.7890
- Testing score of 0.7243

LogisticRegression(max_iter = 10_000)

- Mean cross_val_score of 0.7403
- Training score of 0.8158
- Testing score of 0.7243

RandomForestClassifier(random_state = 42)

- Mean cross_val_score of 0.7163
- Training score of 0.9902
- Testing score of 0.7227

# Modeling –

# Time to Work It!

Running a GridSearchCV on the TF-IDF pipeline with MultinomialNB:

The parameters we searched:

pipe_params =

- tf__max_features: [None, 1_000, 2_000, 4_000, 8_000],
- tf__min_df: [1, 2, 3],
- tf__max_df: [1.0, .75, .85, .95],
- tf__ngram_range: [(1, 1), (1, 2), (1,3)]

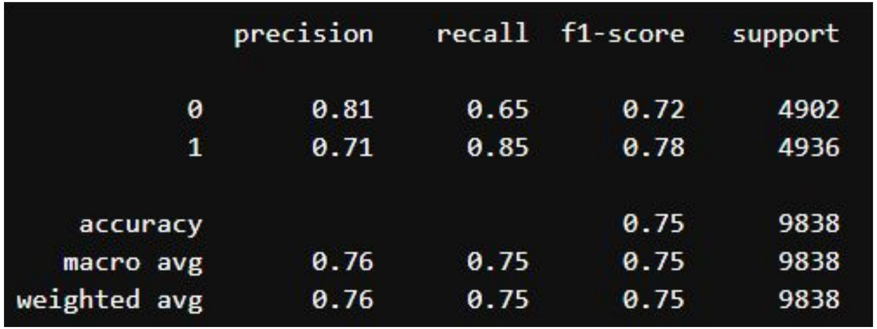
After fitting we found the best parameters to be:

tf__max_df: 1.0,
tf__max_features: None,
tf__min_df: 1,
tf__ngram_range: (1, 3)

# Modeling – Evaluation

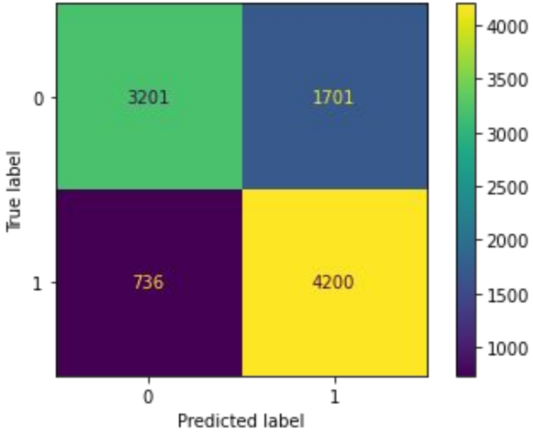
Evaluation of GridSearch'd TF-IDFVectorizer Model:

Classification Report:

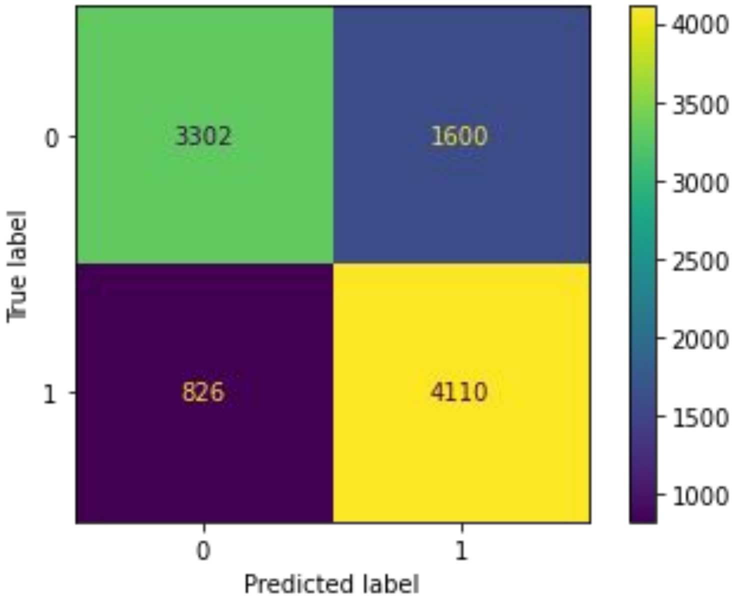


```
              precision    recall  f1-score   support

           0       0.81      0.65      0.72      4902
           1       0.71      0.85      0.78      4936

    accuracy                           0.75      9838
   macro avg       0.76      0.75      0.75      9838
weighted avg       0.76      0.75      0.75      9838
```

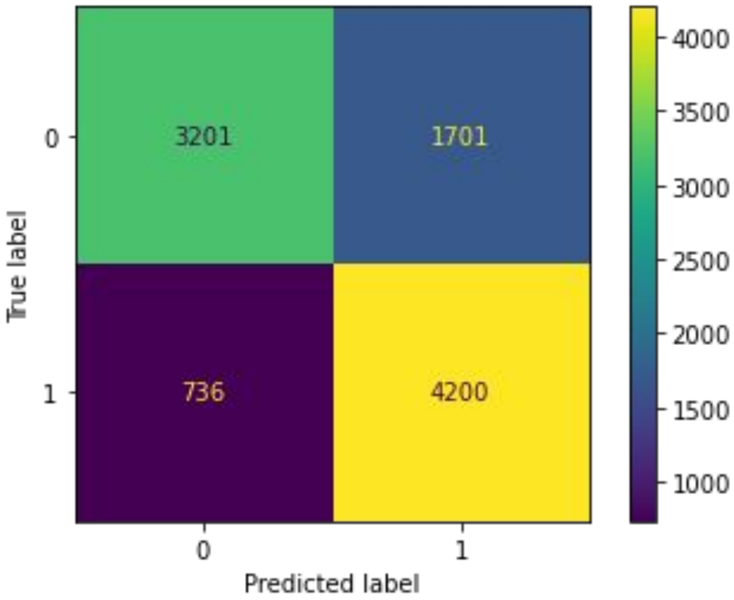
Confusion Matrix:



CountVectorizer with MultinomialNB



TF-IDFVectorizer with MultinomialNB



# Best Model –

# Trending Topics!

What words are good predictors of DMs?

- From the MultinomialNB models:
    - 'characters', 'feel', 'fun', 'group', 'having', 'idea', 'let', 'maybe', 'new', 'rules', 'run', 'session', 'story', 'sure', 'work', 'world'
    -
- From the LogReg models:
    - ‘blind', 'conclave', 'paralyzed', 'torture', 'vtt'

# Best Model –

# Trending Topics!

In general DMs are talking about status conditions, the following words appeared as influential on our models:

- paralyzed
- blind
- blindness

They’re also tossing around some fun ideas:

- hydra
- goliath
- tarrasque
- trap
- traps

But they also need some tools:

- enforce
- generator
- tweak
- advice
- timer
- roll20

# Best Model –

# Trending Topics!

As an individual who plays D&D (but does NOT DM) I was surprised that Roll20 was not a more influential or popular term in our corpus. It did appear, though.

I'm also not surprised that instances of 'torture' are helpful in predicting posts from 'dmacademy'. Sometimes the DM just doesn't want to see you win.

It's interesting to see that 'vtt' is useful for determining and popular among DMs. Virtual TableTop is a way of gaming either digitally or in person (with the vtt), using virtually built sets to host the game. Rather than a classic hex grid for gaming, DMs are able to pre-build virtual settings in which to further immerse their players, see https://foundryvtt.com/. If Wizards of the Coast has not already thought about their own platform for vtt, they should consider it as DMs are talking about.

# In Conclusion:

Out of all of our models, CountVectorizer run with MultinomialNB produced the most accurate, least overfit predictions. While we did not reach the benchmark rate of 80%, our model was close, account for approximately 77% of all training and 74% of all testing data. While the RandomForestClassifier performs extraordinarily well on training data, it is very overfit.

If anything, we should re-collect our data to look at a greater date range of posts. Additionally, we should re-run our model to drop the additional stop words, the popular words in common with both subreddits.

In this analysis we looked at comments going back to 10/22/21 and 10/19/21 for the dndnext and dmacademy subreddits, respectively. Had we had the computing power available to a company like Wizards of the Coast, the model would have been built on additional data, likely going back to the last module release (The Wild Beyond the Witchlight, September 21, 2021).

Regardless, we were still able to gleam valuable insights from the data on behalf of Wizards of the Coast.

# Secret Sauce –

External Sources:
https://github.com/pushshift/api
https://www.epochconverter.com/
https://www.reddit.com/
https://regex101.com/
https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
https://stackoverflow.com/questions/64258622/gridsearchcv-with-tfidf-and-count-vectorizer
https://foundryvtt.com/

Images:
https://www.redbubble.com/i/sticker/U-Jelly-Dice-Bois-D6-Gelatinous-Cube-by-anniespjs/40840125.EJUG5
https://www.dndmini.com/products/d-d-icons-of-the-realms-gargantuan-tarrasque
https://www.krakendice.com/

Coding:
https://stackoverflow.com/questions/49188960/how-to-show-all-columns-names-on-a-large-pandas-dataframe
https://stackoverflow.com/questions/14657241/how-do-i-get-a-list-of-all-the-duplicate-items-using-pandas-in-python
https://data-science-blog.com/en/blog/2018/11/04/sentiment-analysis-using-python/
https://www.python-graph-gallery.com/25-histogram-with-several-variables-seaborn
https://stackoverflow.com/questions/50444346/fast-punctuation-removal-with-pandas
https://stackoverflow.com/questions/4328500/how-can-i-strip-all-punctuation-from-a-string-in-javascript-using-regex
https://stackoverflow.com/questions/19790188/expanding-english-language-contractions-in-python
https://regex101.com/
https://stackoverflow.com/questions/336210/regular-expression-for-alphanumeric-and-underscores
https://stackoverflow.com/questions/47557563/lemmatization-of-all-pandas-cells
https://www.geeksforgeeks.org/python-intersection-two-lists/
https://stackoverflow.com/questions/5511708/adding-words-to-nltk-stoplist
https://awhan.wordpress.com/2016/06/05/scikit-learn-nlp-list-english-stopwords/
https://stackoverflow.com/questions/57924484/finding-coefficients-for-logistic-regression-in-python