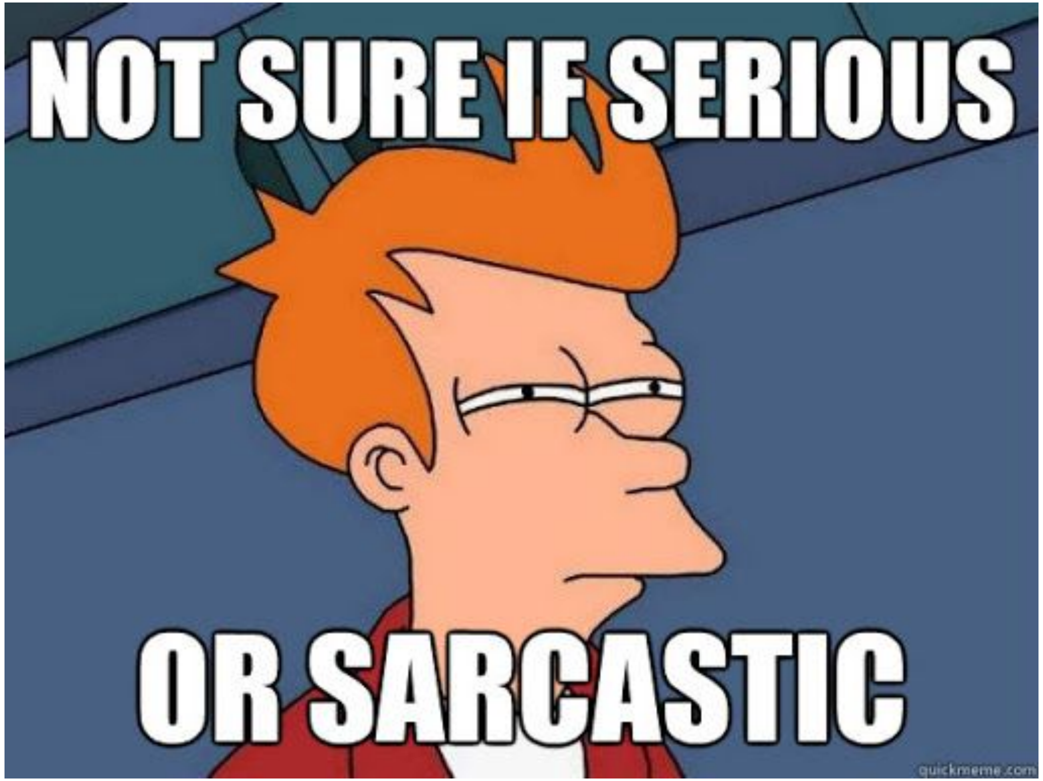
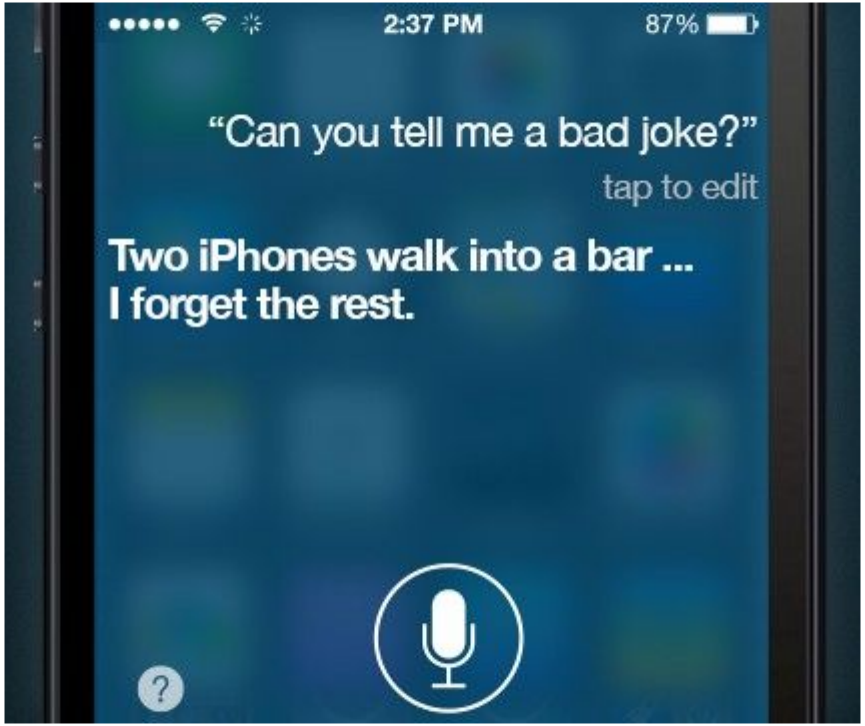
What's So Funny? - Humor Text Classifier



Erin Cox
https://github.com/coxem14/Capstone-2

Background



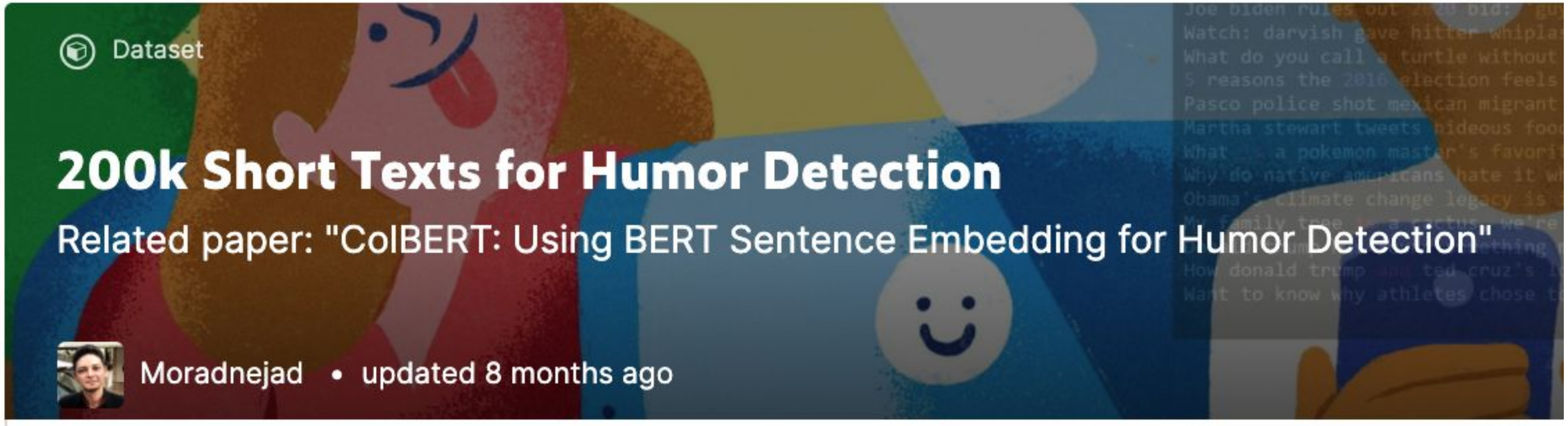
"Can you tell me a bad joke?"

Two iPhones walk into a bar ... I forget the rest.

Chatbots are helping many organizations achieve a higher level of customer service, and now many consumers have virtual assistants like Alexa in their homes, and Google Assistant and Siri on their phones.

Intent detection, like humor detection, is critical to improving and expanding the applications of natural language processing.

Data


Dataset

200k Short Texts for Humor Detection

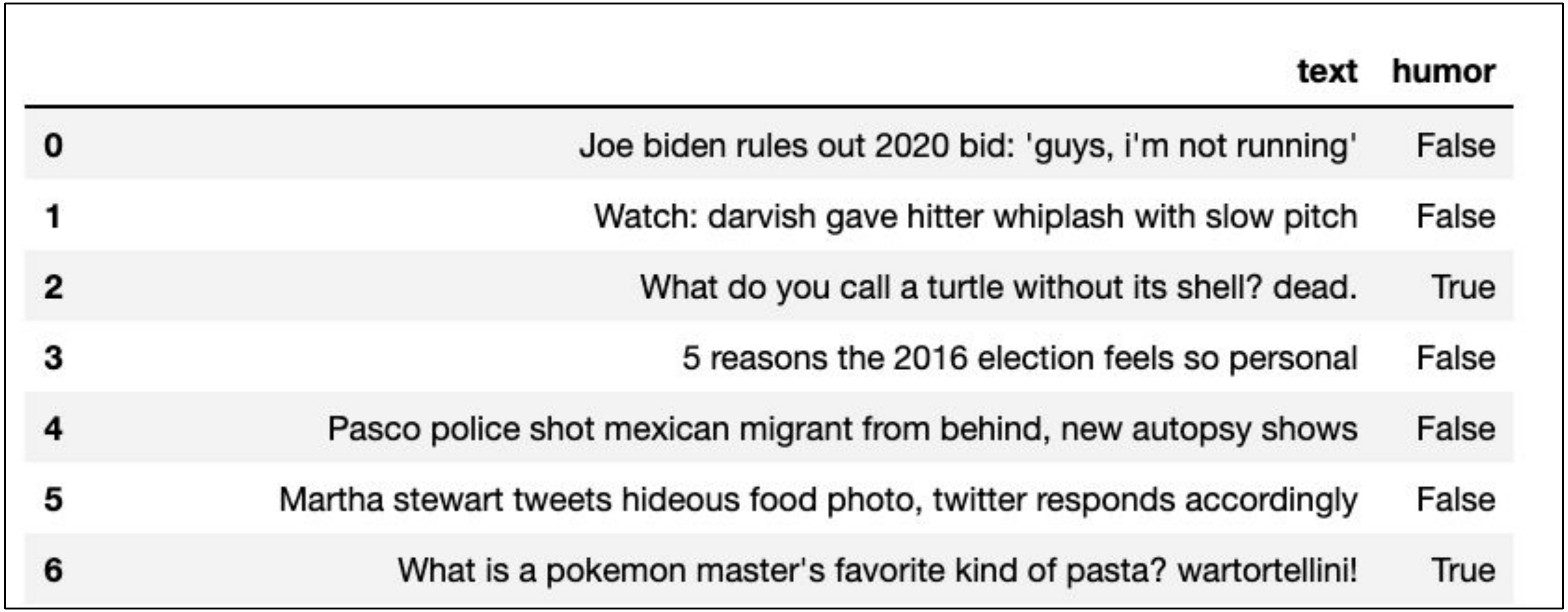
Related paper: "ColBERT: Using BERT Sentence Embedding for Humor Detection"

Moradnejad • updated 8 months ago

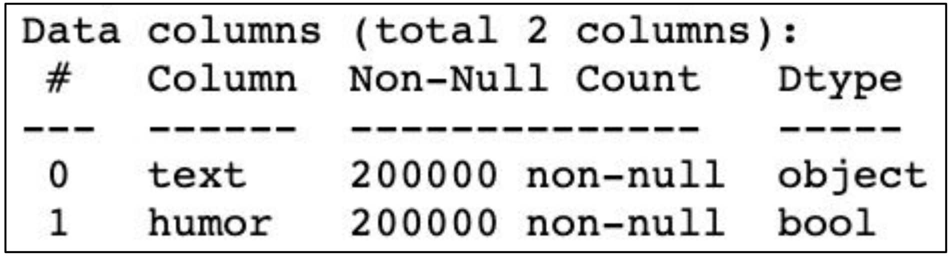
The dataset was created by Issa Annamoradnejad and team at Cornell University and was used in the paper *ColBERT: Using BERT Sentence Embedding for Humor Detection*. I downloaded the dataset from Kaggle.

The dataset includes 200k short text samples labeled as either True or False for humor.

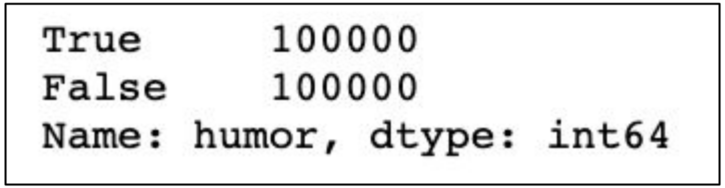
EDA - Initial Inspection



	text	humor
0	Joe biden rules out 2020 bid: 'guys, i'm not running'	False
1	Watch: darvish gave hitter whiplash with slow pitch	False
2	What do you call a turtle without its shell? dead.	True
3	5 reasons the 2016 election feels so personal	False
4	Pasco police shot mexican migrant from behind, new autopsy shows	False
5	Martha stewart tweets hideous food photo, twitter responds accordingly	False
6	What is a pokemon master's favorite kind of pasta? wartortellini!	True

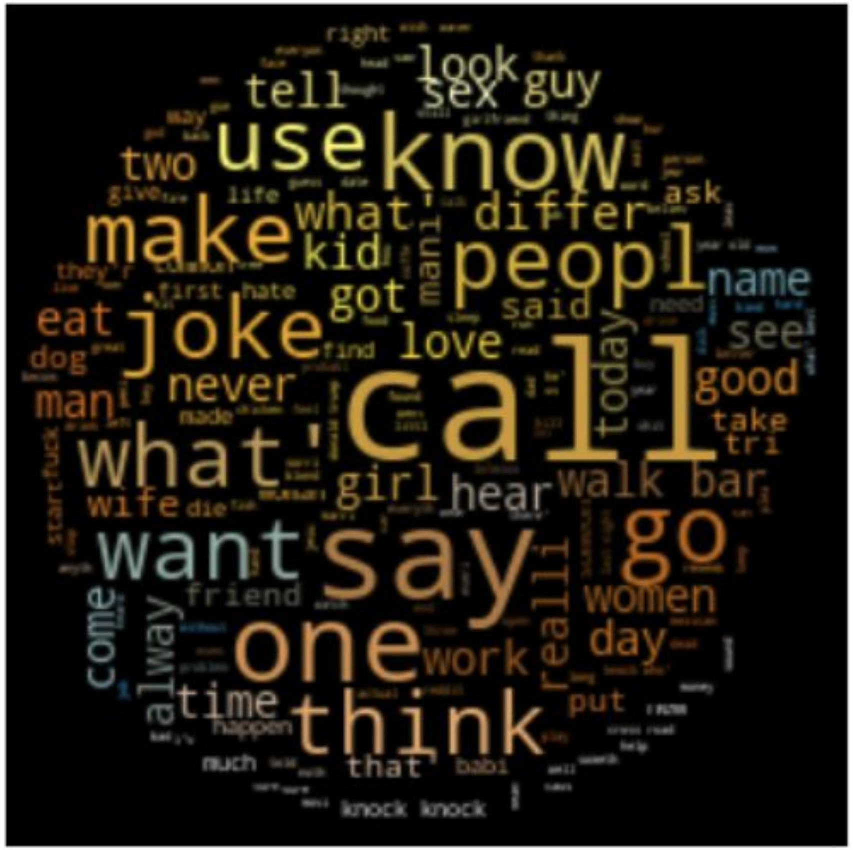


```
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   text    200000 non-null  object
 1   humor   200000 non-null  bool
```

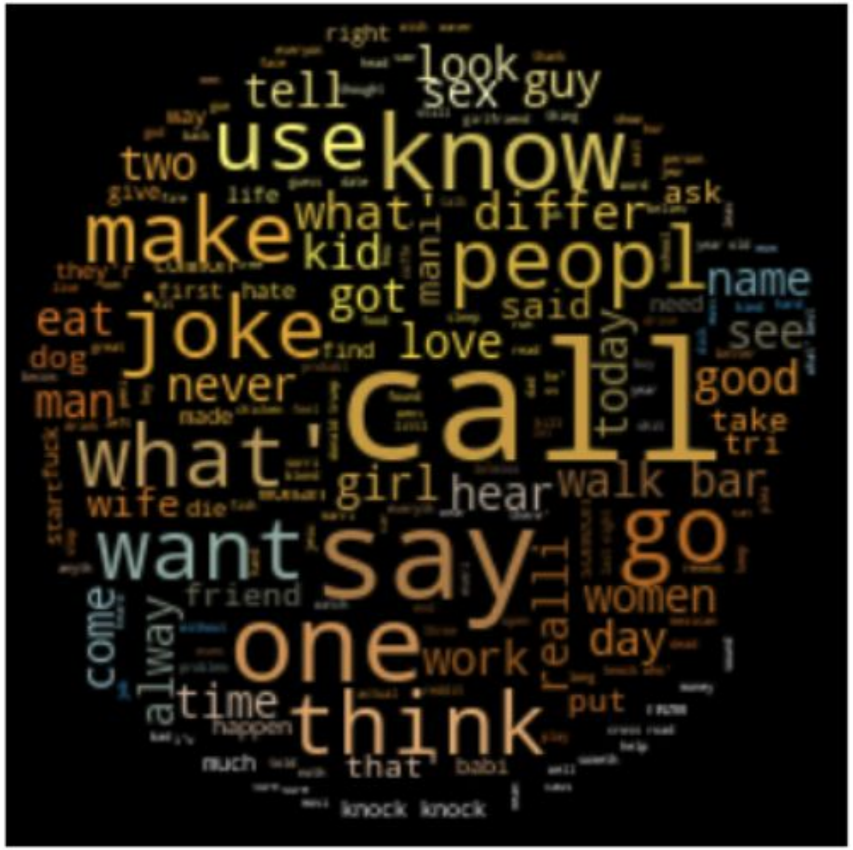


```
True      100000
False     100000
Name: humor, dtype: int64
```

EDA - Word Clouds



EDA - Word Clouds



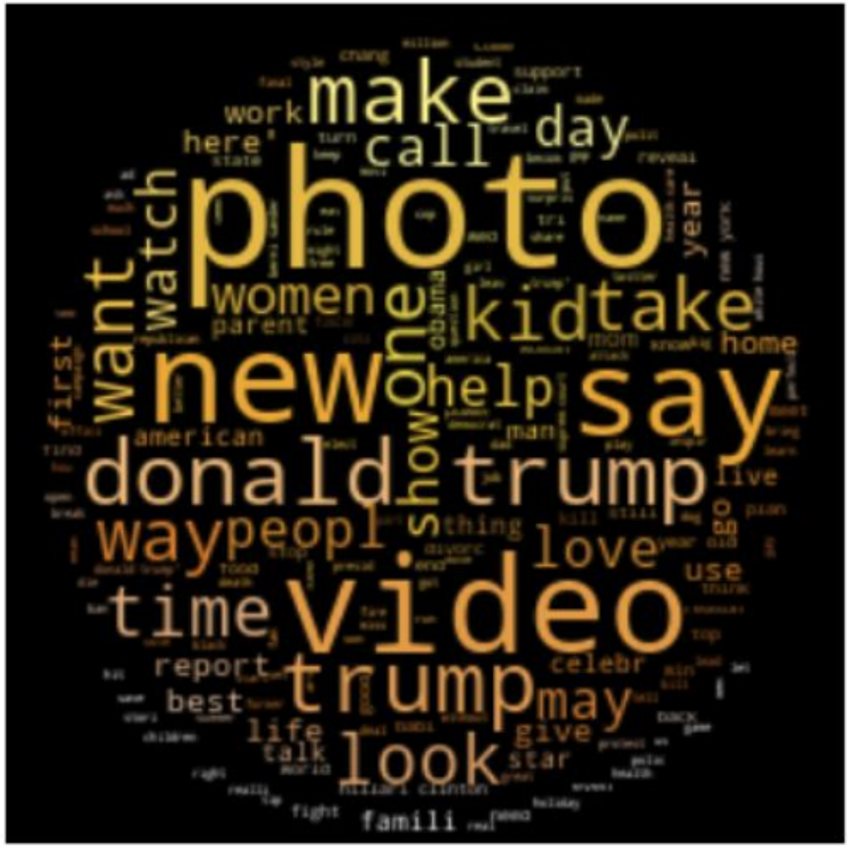
Humorous texts patterns seem to be:

- What did one _ say to the other _ ?
- What do you call a _ that _?
- What's the difference between _ and _?
- A _ walks into a bar
- 'Knock Knock' jokes

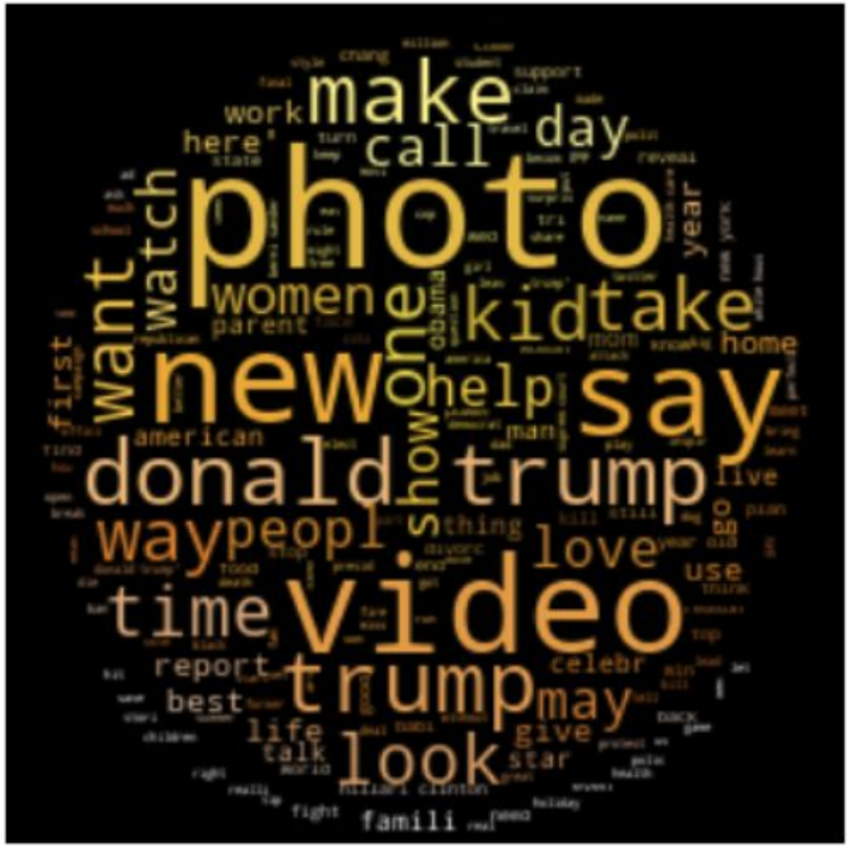
Top 10 words in humorous texts:

- call, say, one, know, go, make, what', joke, peopl, want

EDA - Word Clouds



EDA - Word Clouds



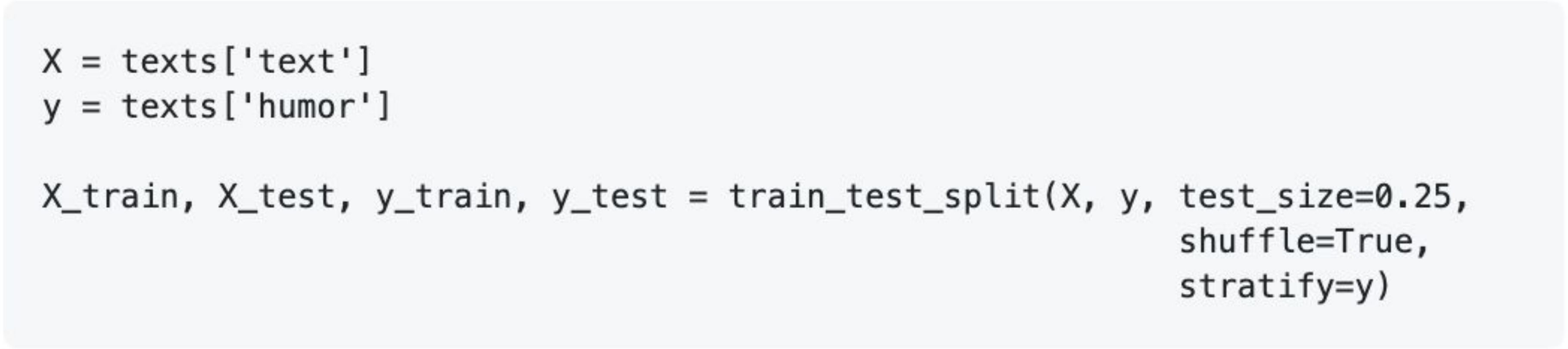
Serious texts patterns seem to be:

- News headlines (new, video, watch, photo, report, first, say, show)
- Article/Blog post titles
- Donald Trump comes up a lot

Top 10 words in serious texts:

- photo, video, new, say, donald trump, trump, make, one, kid, take

Train, Test, Split



```
X = texts['text']
y = texts['humor']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
                                                    shuffle=True,
                                                    stratify=y)
```

The resulting training datasets had 150,000 texts, while the testing datasets had 50,000 texts.

Model Selection

The primary models I built are as follows:

- Multinomial Naive Bayes
- Bernoulli Naive Bayes

I also explored:

- Random Forest Classifier
- Multilayer Perceptron
- KMeans Clustering - Appendix

Pipelines

```
mnb_pipeline = Pipeline([('tfidf', TfidfVectorizer()),
                         ('model', MultinomialNB())])
```

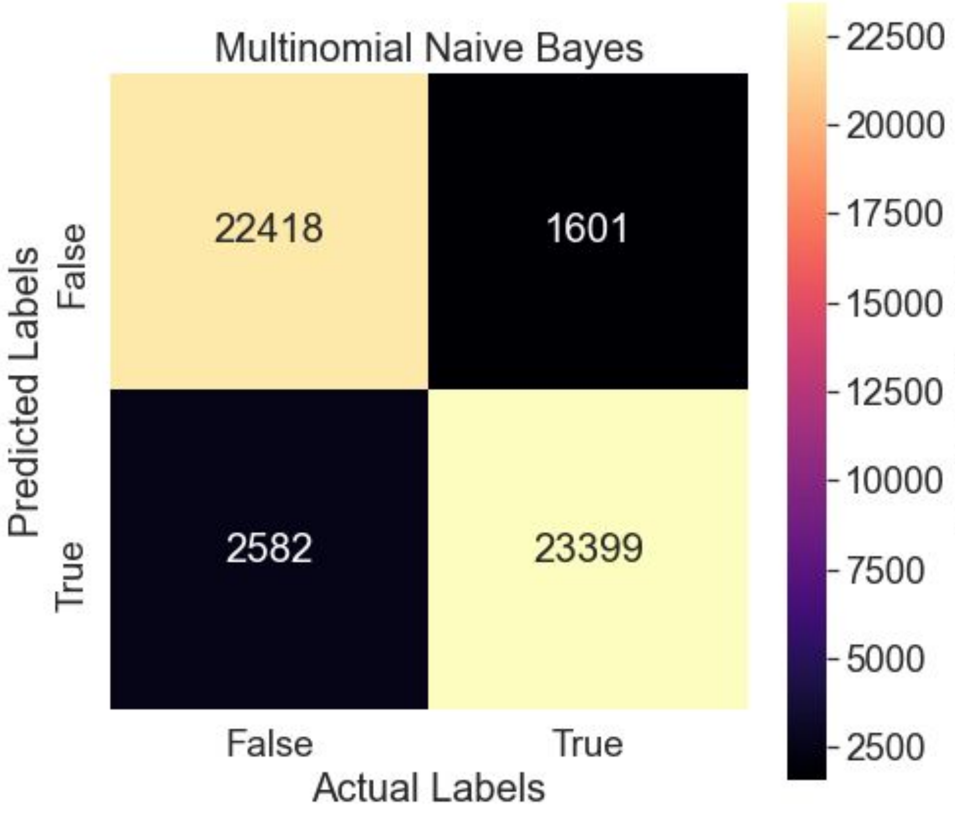
Fitting the Naive Bayes Models

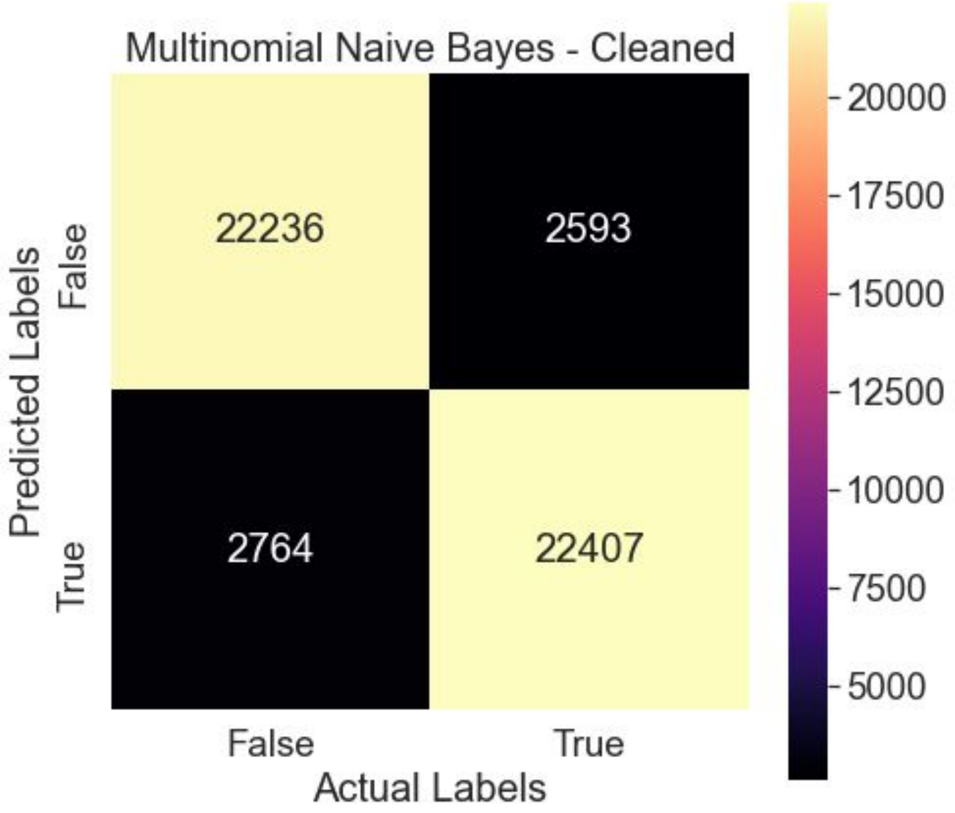
I wanted to see how the models performed with cleaned and uncleaned data inputs, so I ran X_train and X_test through the corpus cleaner function (same as I used for word clouds) prior to fitting and predicting, respectively.

Evaluating the Models

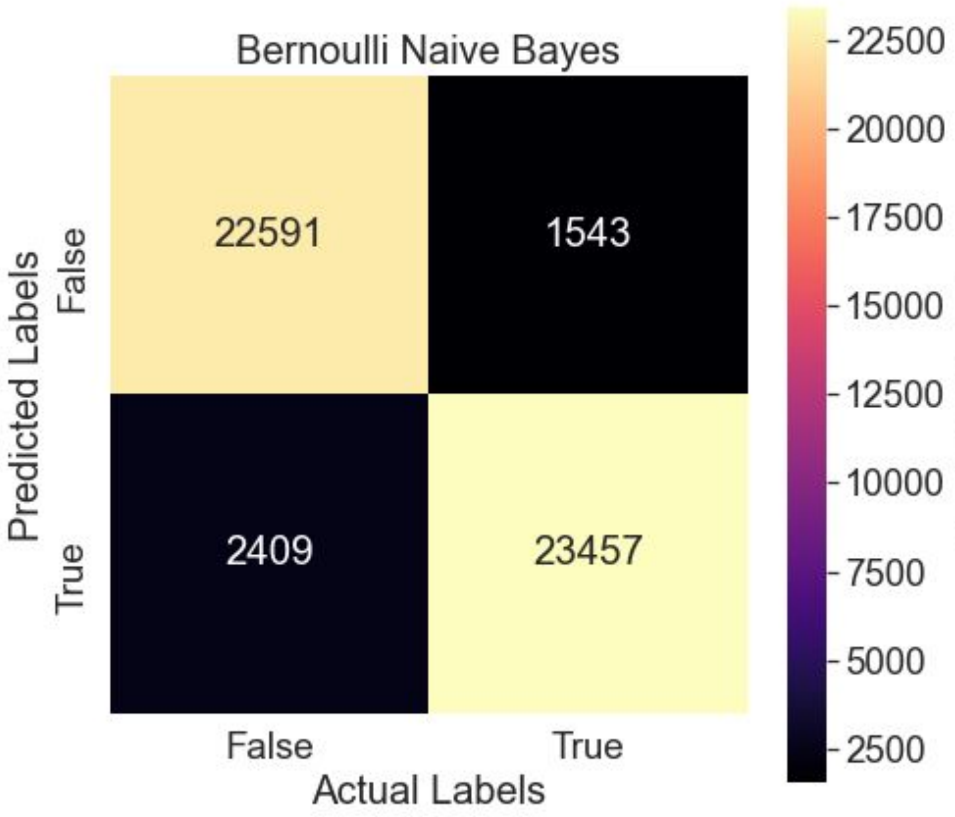
To determine which model performed the best, I fit each model with X_train and y_train, got the predictions, and compared accuracy, precision, recall scores, confusion matrices, and ROC plots.

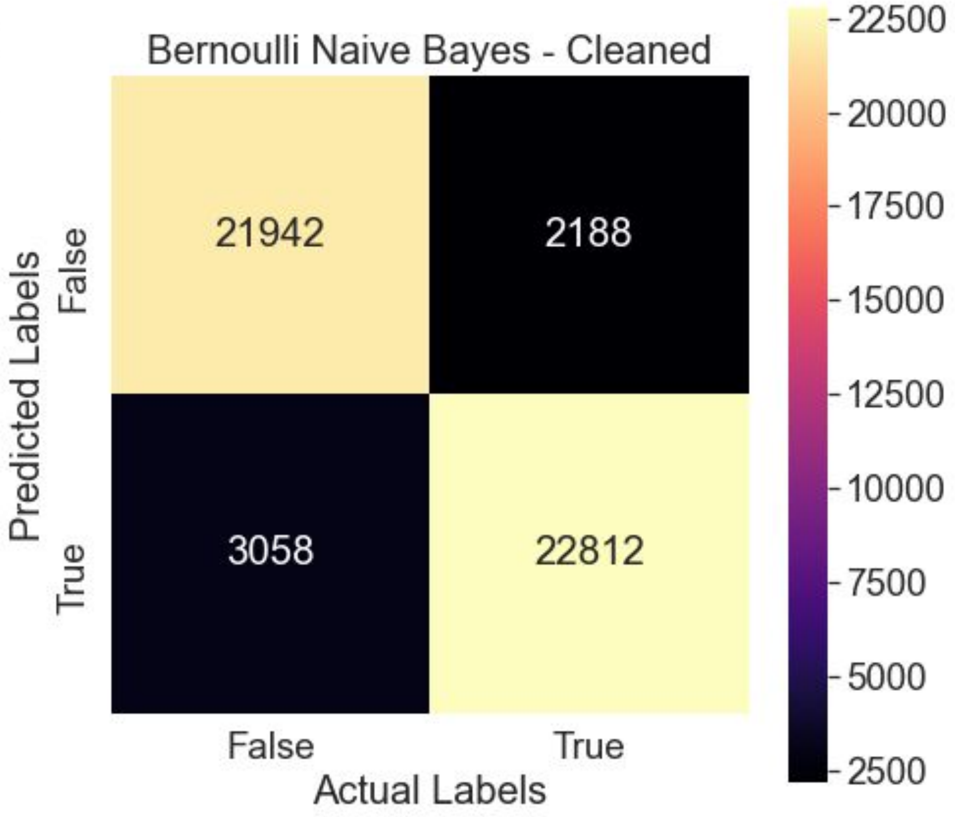
Multinomial Naive Bayes



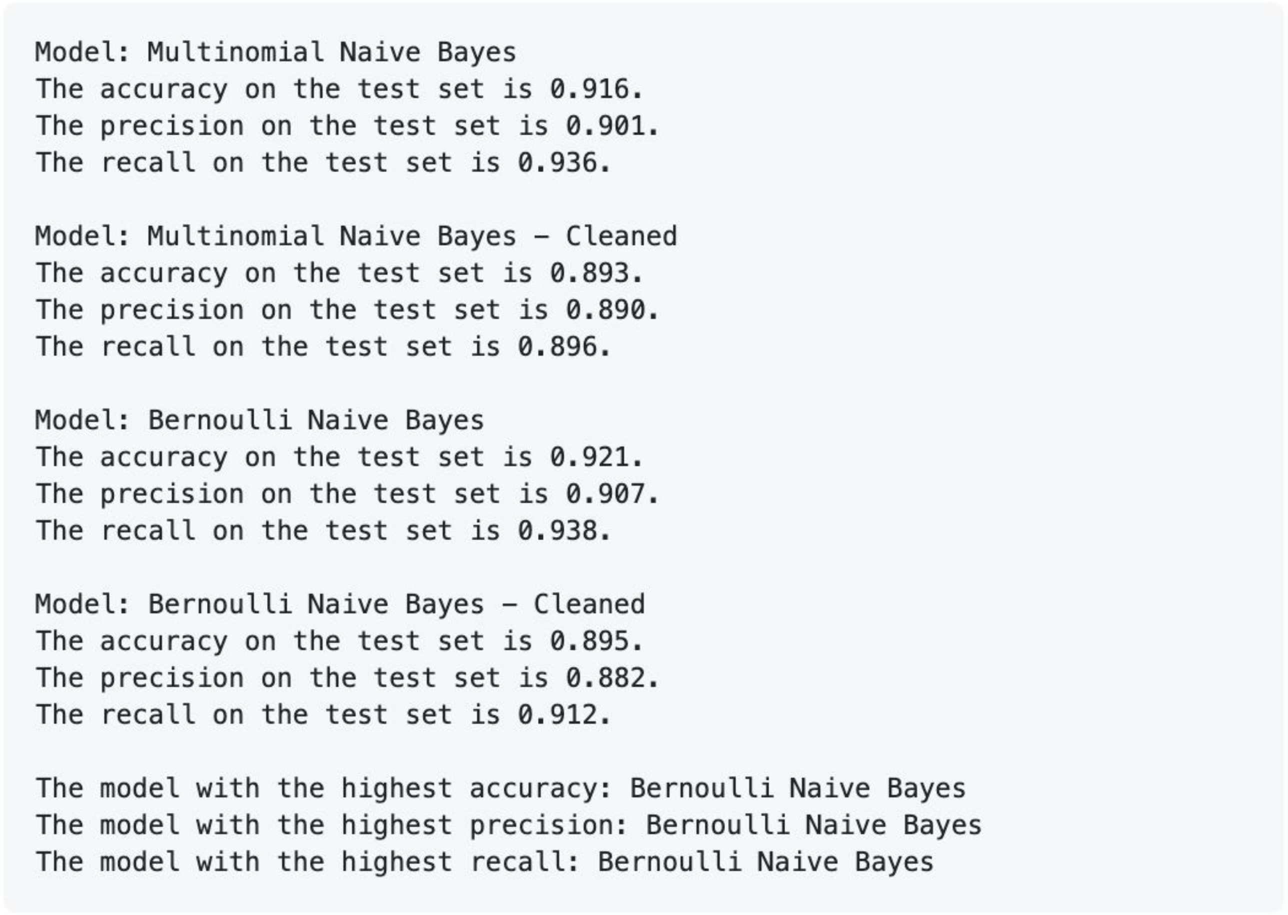


Bernoulli Naive Bayes



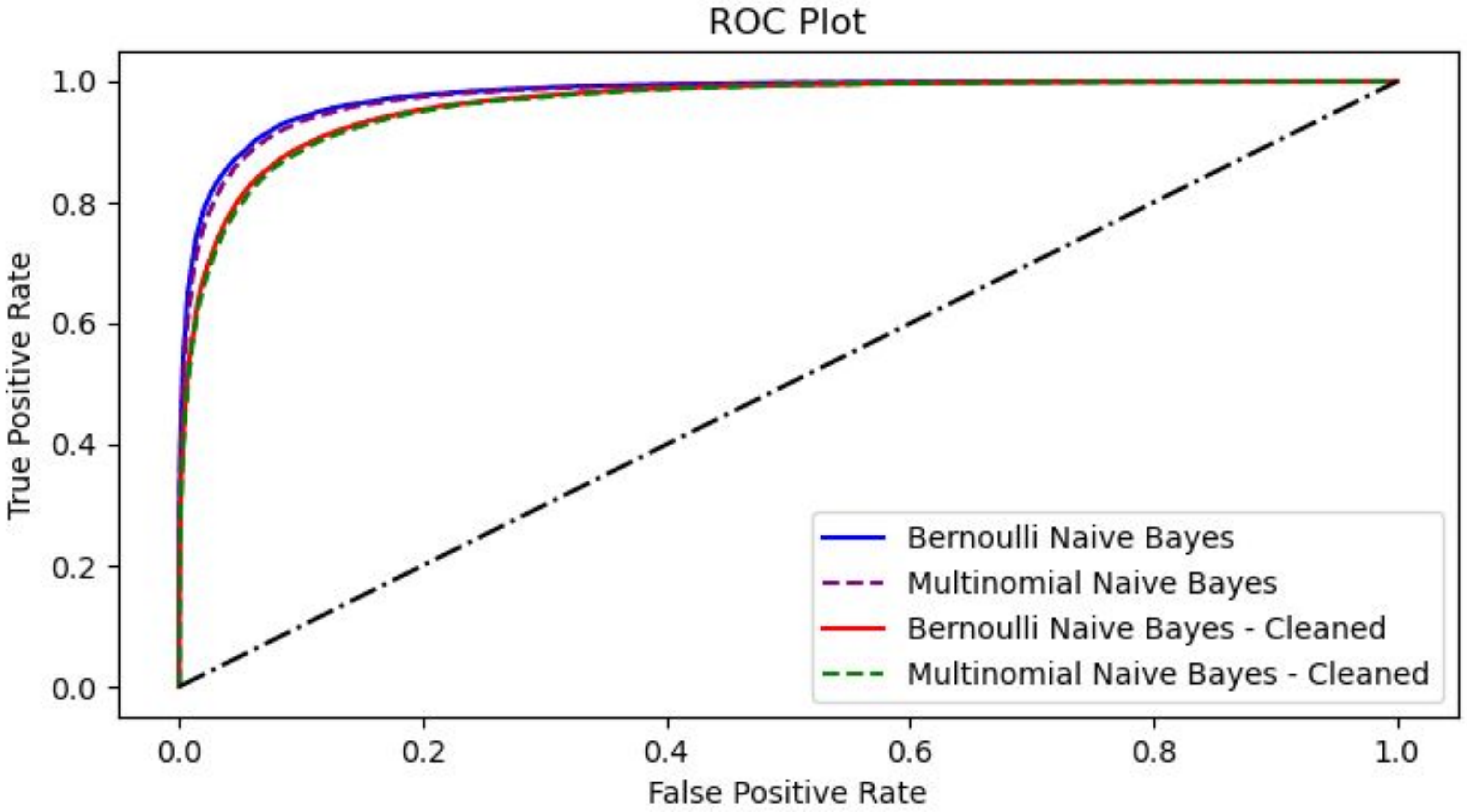


Accuracy, Precision, and Recall Scores

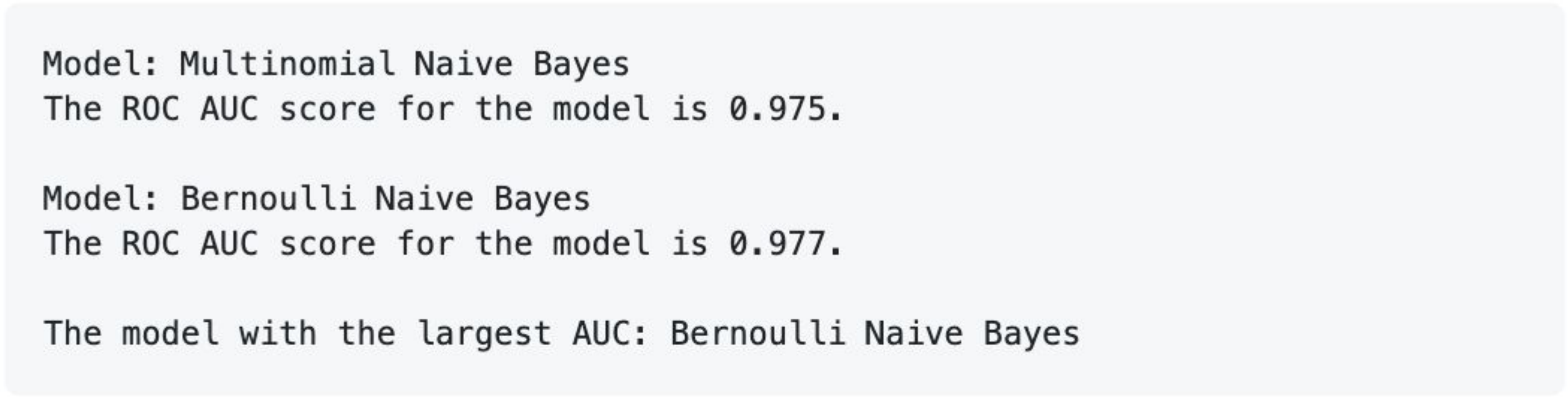


```
Model: Multinomial Naive Bayes
The accuracy on the test set is 0.916.
The precision on the test set is 0.901.
The recall on the test set is 0.936.

Model: Multinomial Naive Bayes - Cleaned
The accuracy on the test set is 0.893.
The precision on the test set is 0.890.
The recall on the test set is 0.896.

Model: Bernoulli Naive Bayes
The accuracy on the test set is 0.921.
The precision on the test set is 0.907.
The recall on the test set is 0.938.

Model: Bernoulli Naive Bayes - Cleaned
The accuracy on the test set is 0.895.
The precision on the test set is 0.882.
The recall on the test set is 0.912.

The model with the highest accuracy: Bernoulli Naive Bayes
The model with the highest precision: Bernoulli Naive Bayes
The model with the highest recall: Bernoulli Naive Bayes
```

Naive Bayes ROC Plot

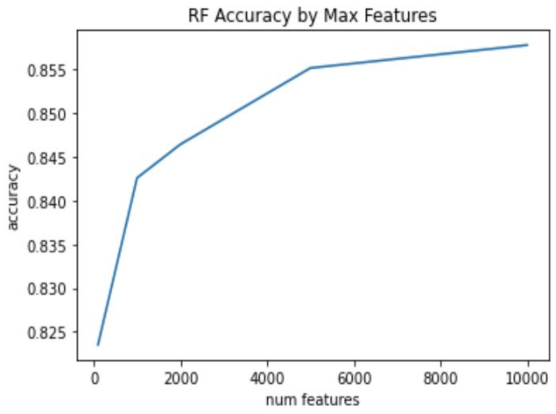


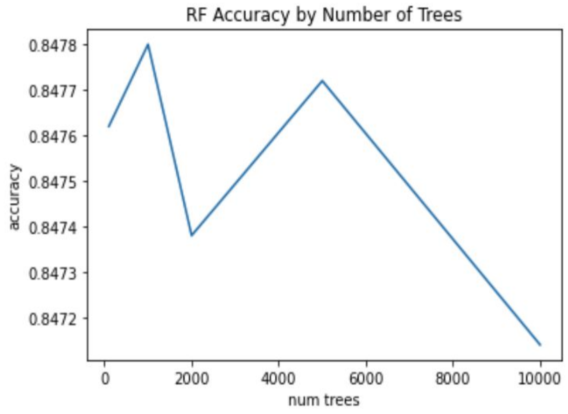
Naive Bayes ROC AUC Scores

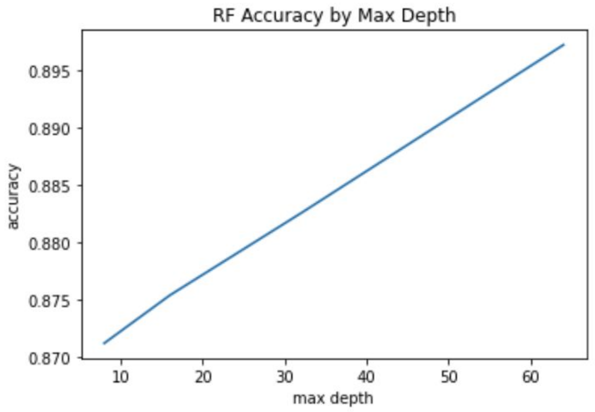


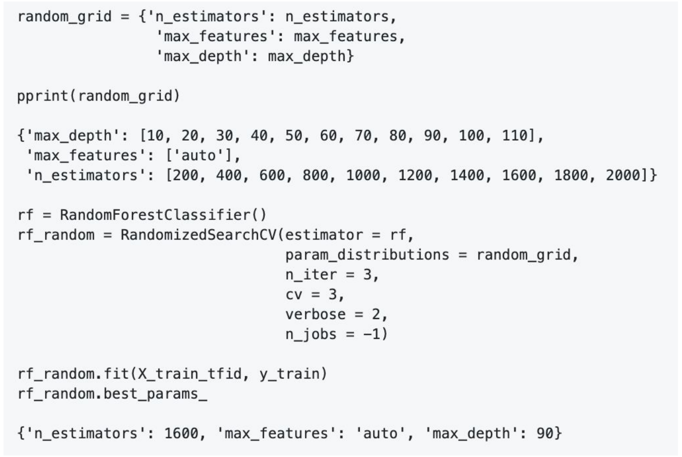
```
Model: Multinomial Naive Bayes
The ROC AUC score for the model is 0.975.

Model: Bernoulli Naive Bayes
The ROC AUC score for the model is 0.977.

The model with the largest AUC: Bernoulli Naive Bayes
```

Random Forest Classifier - Tuning Parameters







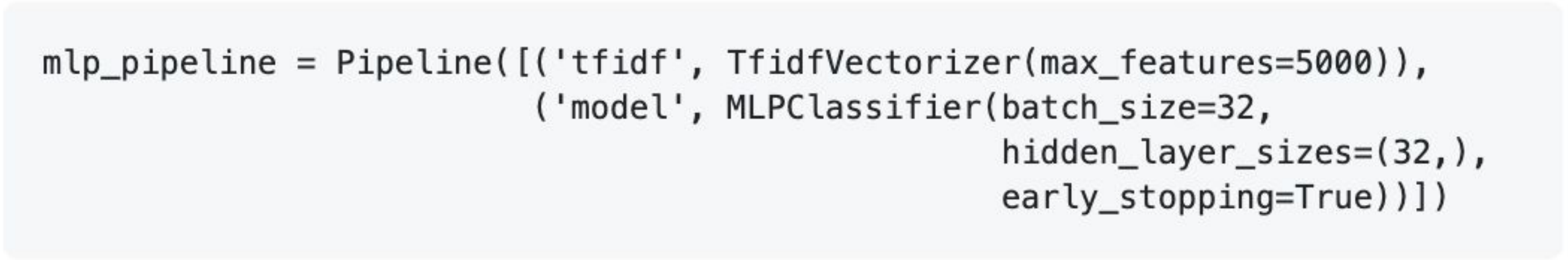


```
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth}

pprint(random_grid)

{'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110],
 'max_features': ['auto'],
 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}

rf = RandomForestClassifier()
rf_random = RandomizedSearchCV(estimator = rf,
                               param_distributions = random_grid,
                               n_iter = 3,
                               cv = 3,
                               verbose = 2,
                               n_jobs = -1)

rf_random.fit(X_train_tfid, y_train)
rf_random.best_params_

{'n_estimators': 1600, 'max_features': 'auto', 'max_depth': 90}
```

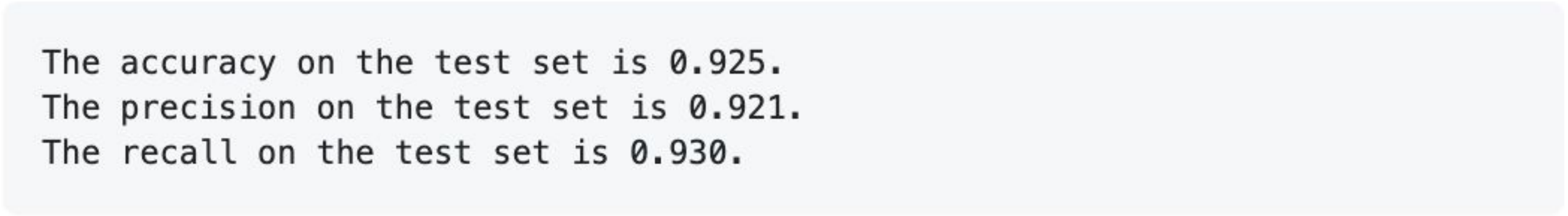
```
The accuracy on the test set is 0.851.
The precision on the test set is 0.888.
The recall on the test set is 0.804.
```

```
The accuracy on the test set is 0.907.
The precision on the test set is 0.905.
The recall on the test set is 0.909.

The ROC AUC score for the model is 0.968.
```

Multilayer Perceptron

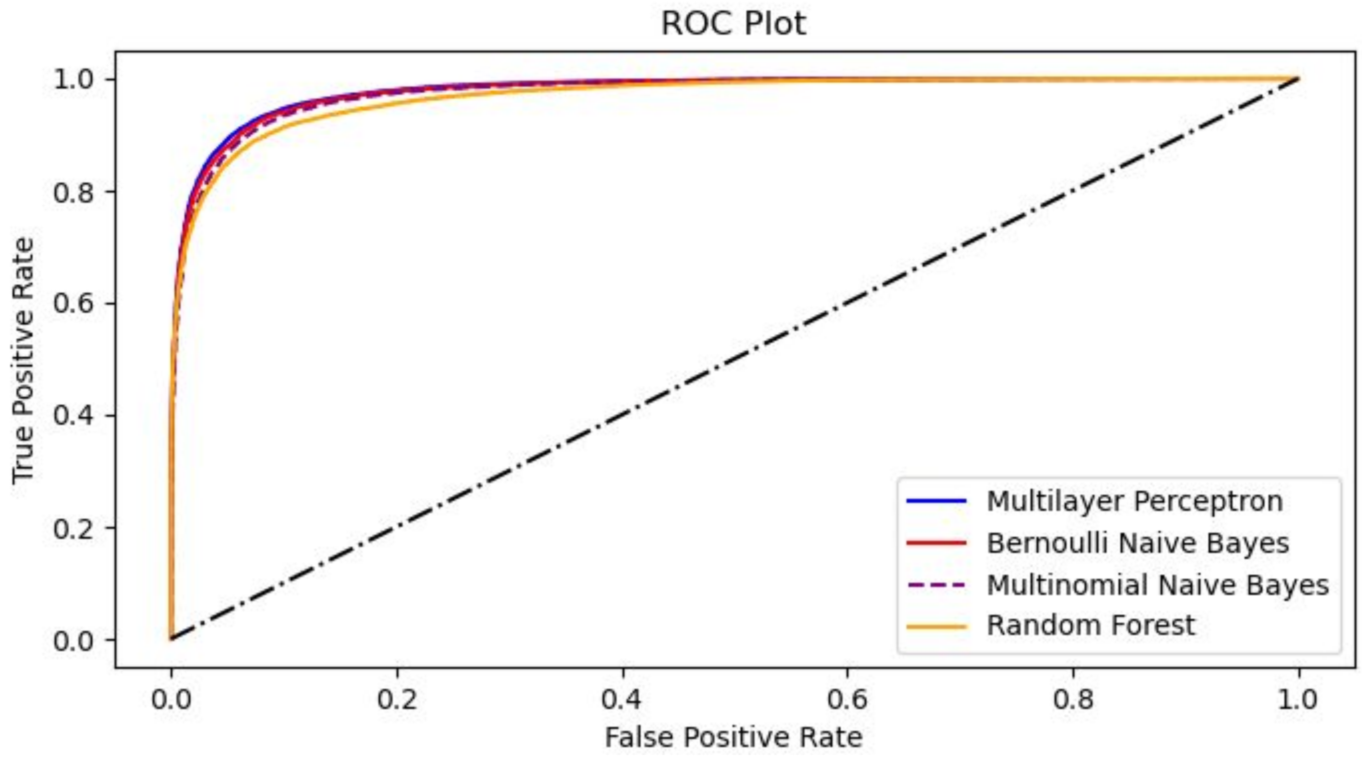


```
mlp_pipeline = Pipeline([('tfidf', TfidfVectorizer(max_features=5000)),
                         ('model', MLPClassifier(batch_size=32,
                                                 hidden_layer_sizes=(32,),
                                                 early_stopping=True))])
```

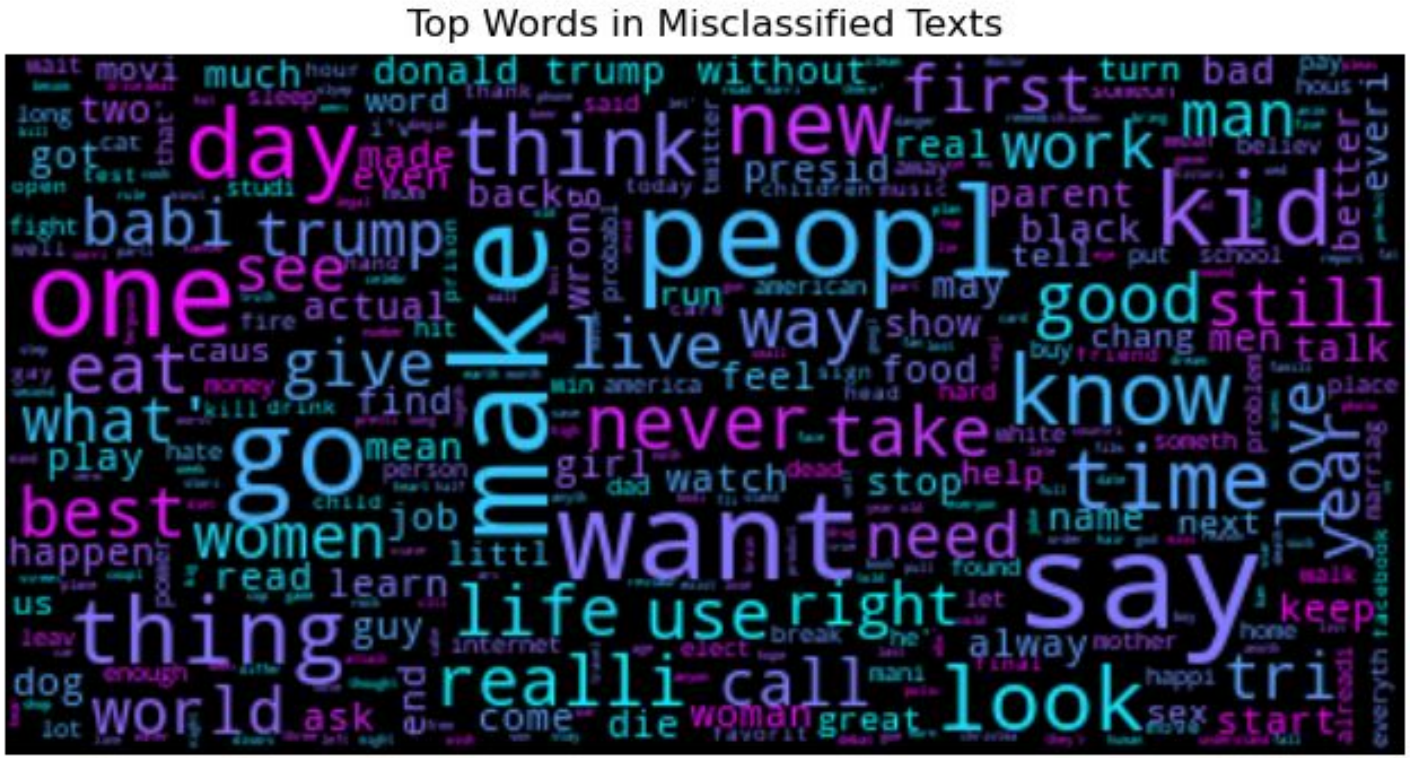


```
The accuracy on the test set is 0.925.
The precision on the test set is 0.921.
The recall on the test set is 0.930.
```

Combined ROC



Problem Words



Top 10 words in misclassified texts: say, make, want, people, go, one, kid, day, know, thing

Next Steps

Recap of Findings:

- MLP and Bernoulli Naive Bayes Models performed the best at classifying the texts and predicting on unseen data.
- Common misclassified word included many stopwords, and words which are equally likely to be present in humorous and serious texts.

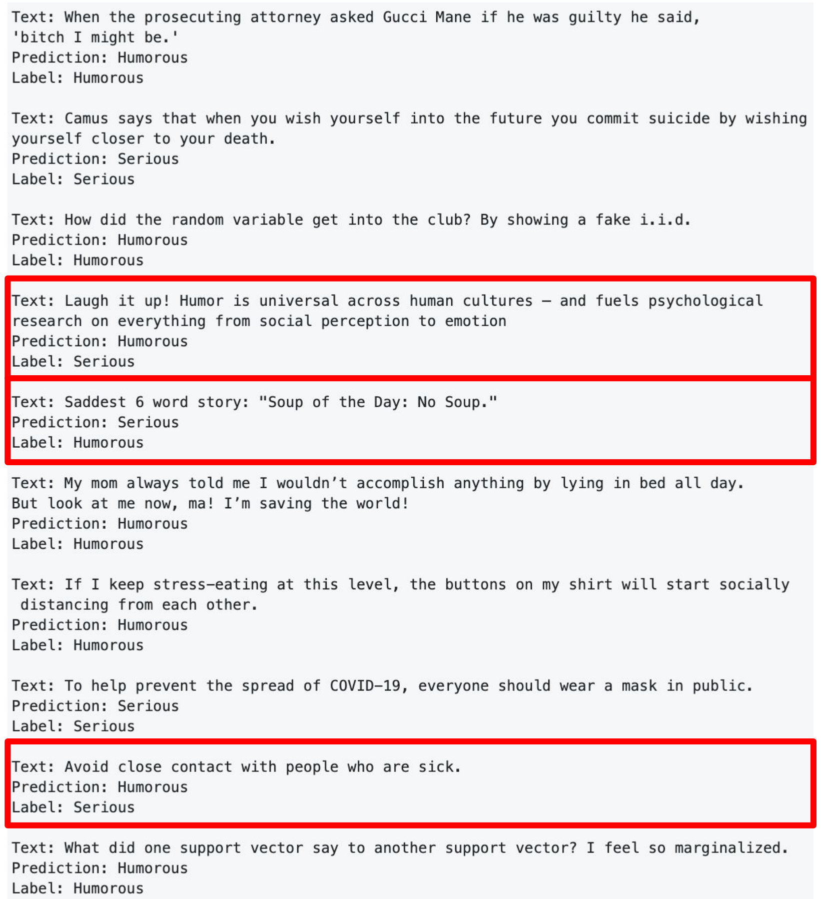
Future Steps:

- Further cleaning for word clouds
- Further tuning to improve models
- Train, test, split - try cross validation, changing test size
- Featurizing Text - explore n-grams (or sequence of words)
- Random Forest - more in depth hyperparameter tuning
- MLP - more in depth hyperparameter tuning
- Test multiple ks for KMeans
- Explore PCA
- Explore CNN/RNN

Appendix

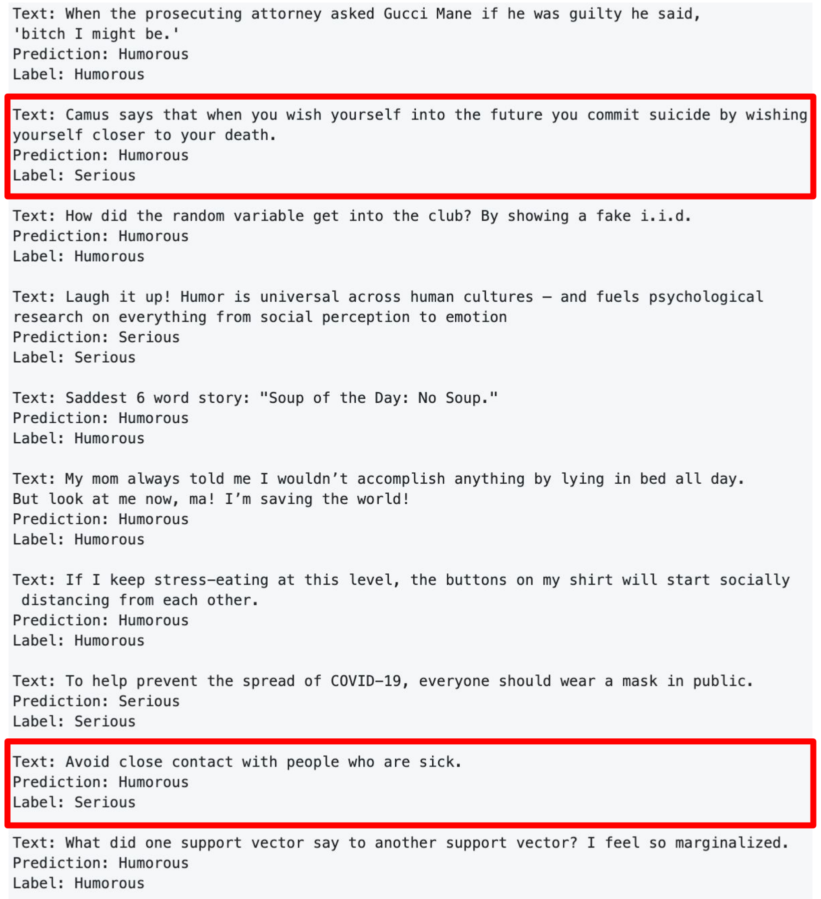
Results from Cohort Submissions

MLP



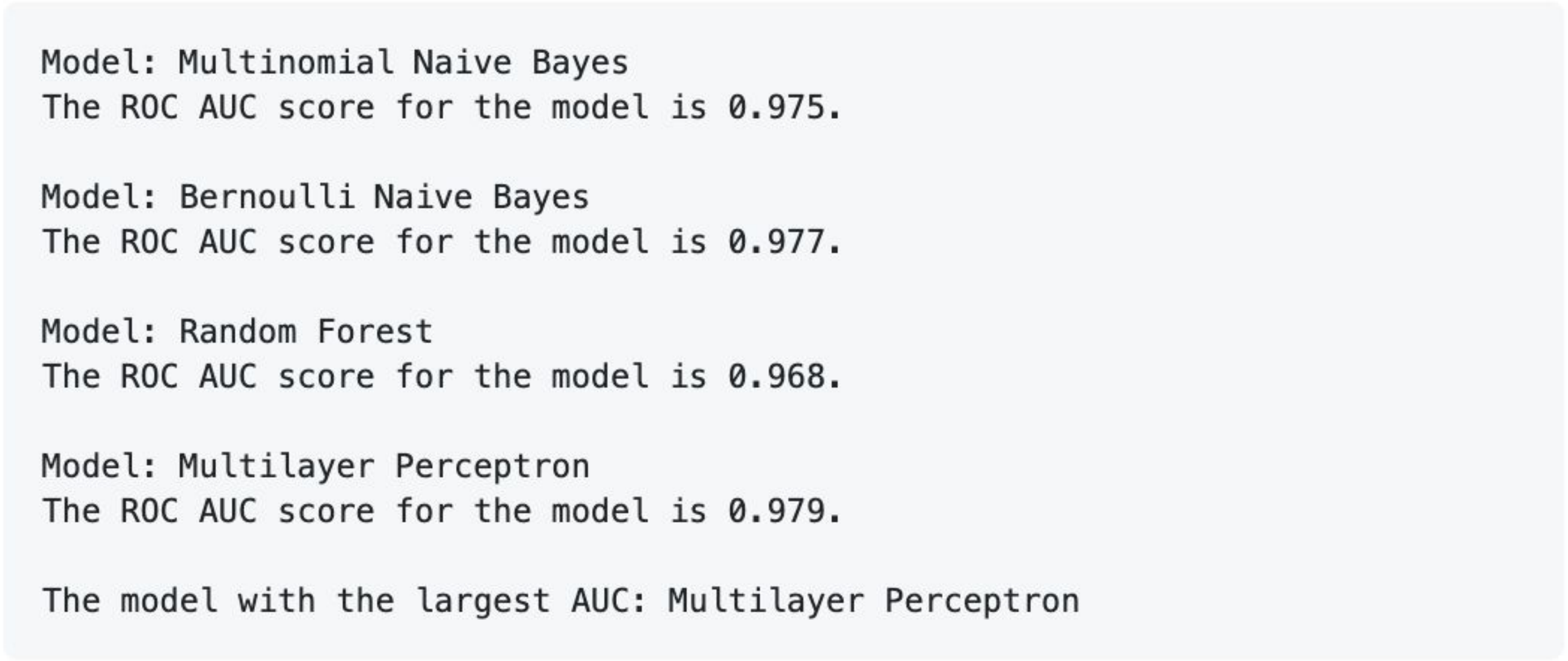
```
Text: When the prosecuting attorney asked Gucci Mane if he was guilty he said,
'bitch I might be.'
Prediction: Humorous
Label: Humorous

Text: Camus says that when you wish yourself into the future you commit suicide by wishing
yourself closer to your death.
Prediction: Serious
Label: Serious

Text: How did the random variable get into the club? By showing a fake i.i.d.
Prediction: Humorous
Label: Humorous

Text: Laugh it up! Humor is universal across human cultures — and fuels psychological
research on everything from social perception to emotion
Prediction: Humorous
Label: Serious

Text: Saddest 6 word story: "Soup of the Day: No Soup."
Prediction: Serious
Label: Humorous

Text: My mom always told me I wouldn’t accomplish anything by lying in bed all day.
But look at me now, ma! I’m saving the world!
Prediction: Humorous
Label: Humorous

Text: If I keep stress-eating at this level, the buttons on my shirt will start socially
 distancing from each other.
Prediction: Humorous
Label: Humorous

Text: To help prevent the spread of COVID-19, everyone should wear a mask in public.
Prediction: Serious
Label: Serious

Text: Avoid close contact with people who are sick.
Prediction: Humorous
Label: Serious

Text: What did one support vector say to another support vector? I feel so marginalized.
Prediction: Humorous
Label: Humorous
```

BNB



```
Text: When the prosecuting attorney asked Gucci Mane if he was guilty he said,
'bitch I might be.'
Prediction: Humorous
Label: Humorous

Text: Camus says that when you wish yourself into the future you commit suicide by wishing
yourself closer to your death.
Prediction: Humorous
Label: Serious

Text: How did the random variable get into the club? By showing a fake i.i.d.
Prediction: Humorous
Label: Humorous

Text: Laugh it up! Humor is universal across human cultures — and fuels psychological
research on everything from social perception to emotion
Prediction: Serious
Label: Serious

Text: Saddest 6 word story: "Soup of the Day: No Soup."
Prediction: Humorous
Label: Humorous

Text: My mom always told me I wouldn’t accomplish anything by lying in bed all day.
But look at me now, ma! I’m saving the world!
Prediction: Humorous
Label: Humorous

Text: If I keep stress-eating at this level, the buttons on my shirt will start socially
 distancing from each other.
Prediction: Humorous
Label: Humorous

Text: To help prevent the spread of COVID-19, everyone should wear a mask in public.
Prediction: Serious
Label: Serious

Text: Avoid close contact with people who are sick.
Prediction: Humorous
Label: Serious

Text: What did one support vector say to another support vector? I feel so marginalized.
Prediction: Humorous
Label: Humorous
```

Combined ROC AUC Scores



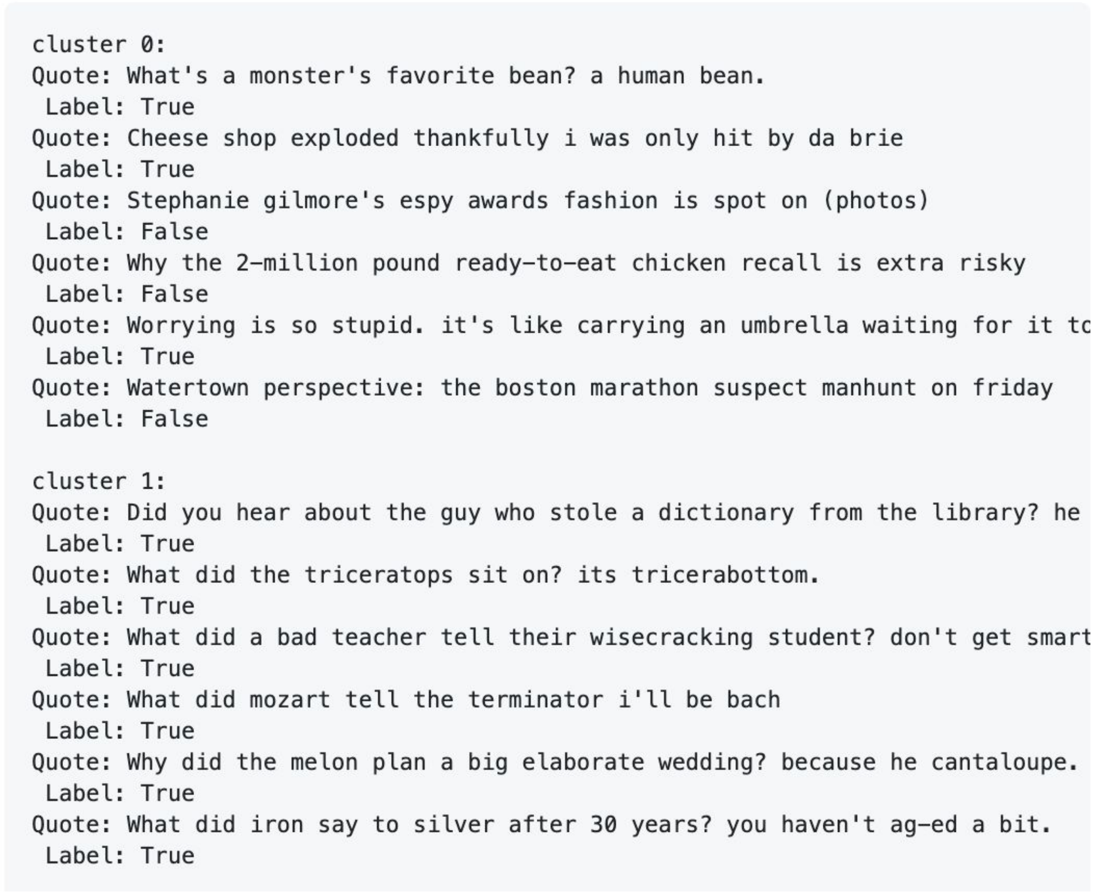
```
Model: Multinomial Naive Bayes
The ROC AUC score for the model is 0.975.

Model: Bernoulli Naive Bayes
The ROC AUC score for the model is 0.977.

Model: Random Forest
The ROC AUC score for the model is 0.968.

Model: Multilayer Perceptron
The ROC AUC score for the model is 0.979.

The model with the largest AUC: Multilayer Perceptron
```

KMeans Clustering

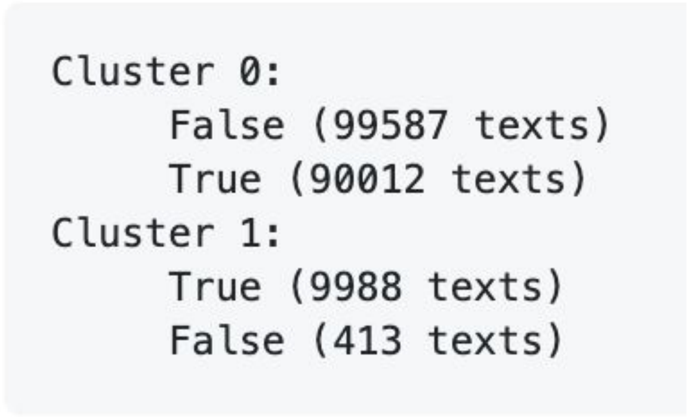
```python
k = 2
vectorizer = TfidfVectorizer(stop_words='english', max_features=1000)
X_vec = vectorizer.fit_transform(X)
features = vectorizer.get_feature_names()
kmeans = KMeans(n_clusters=k, verbose=2)
kmeans.fit(X_vec)

# Find the top 10 features for each cluster.
n_features = 10
top_centroids = kmeans.cluster_centers_.argsort()[:,-1:-(n_features+1):-1]
print("top features (words) for each cluster:")
for num, centroid in enumerate(top_centroids):
    print(f"{num}, {', '.join(features[i] for i in centroid)}")
```

```
top features (words) for each cluster:
0, like, trump, new, just, people, does, don, photos, make, know
1, did, say, hear, cross, road, got, man, chicken, know, guy
```

KMeans Clustering



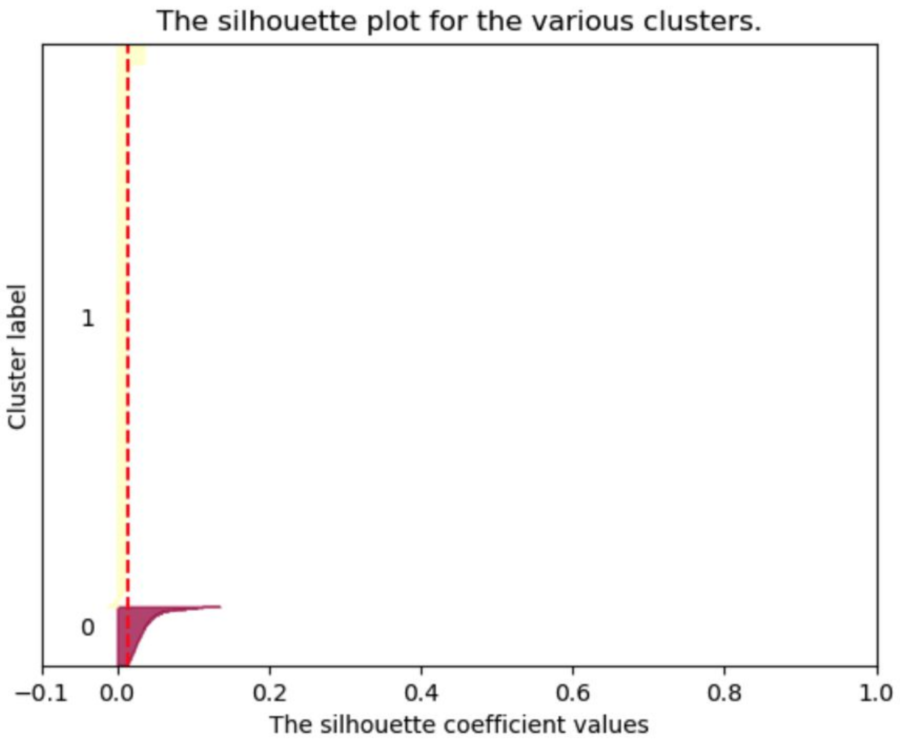
```
cluster 0:
Quote: What's a monster's favorite bean? a human bean.
 Label: True
Quote: Cheese shop exploded thankfully i was only hit by da brie
 Label: True
Quote: Stephanie gilmore's espy awards fashion is spot on (photos)
 Label: False
Quote: Why the 2-million pound ready-to-eat chicken recall is extra risky
 Label: False
Quote: Worrying is so stupid. it's like carrying an umbrella waiting for it to
 Label: True
Quote: Watertown perspective: the boston marathon suspect manhunt on friday
 Label: False

cluster 1:
Quote: Did you hear about the guy who stole a dictionary from the library? he
 Label: True
Quote: What did the triceratops sit on? its triceratbottom.
 Label: True
Quote: What did a bad teacher tell their wisecracking student? don't get smart
 Label: True
Quote: What did mozart tell the terminator i'll be bach
 Label: True
Quote: Why did the melon plan a big elaborate wedding? because he cantaloupe.
 Label: True
Quote: What did iron say to silver after 30 years? you haven't ag-ed a bit.
 Label: True
```



```
Cluster 0:
        False (99587 texts)
        True (90012 texts)
Cluster 1:
        True (9988 texts)
        False (413 texts)
```

KMeans Silhoutte Plot

```
For n_clusters = 2 The average silhouette_score is : 0.01396
```


The silhouette plot for the various clusters.
Cluster label
The silhouette coefficient values