Covid Twitter Analysis

Group 3 - Jon, Giffin, Canxiu, Charles, Stephen

Business and Analytical Objective

Background: Negative sentiment raised rapidly during the beginning of pandemic which built up fears across our society. Given the public sentiments are an important indicator of crisis response, use social media platforms to monitoring sentiment can help with the end to balance exigency without adding to panic.

Goal and Task: To predict the current sentiment (Extremely Positive, Positive, Neutral, Negative, Extremely Negative) of our society in response of Covid by using the original Tweets during March and April 2020.

Analytical solution and Machine learning method:

1. Apply unsupervised machine learning to build a Natural Language Process (NLP) model that profiles and groups original tweets into different topics
2. Use the outputs from the unsupervised model as an input to build a supervised model to predict the sentiment by given any Tweets.

Data Exploration

The tweets (41157 instances) have been pulled from Twitter and manual tagging has been done then.

Target feature: Sentiment

3 Categorical features: Location, TweetAt, OriginalTweet

2 Numerical features: ScreenName, UserName (both features contain numerical assignments)

Data quality assessment:

- OriginalTweet: high cardinality (very high number of distinct values: 100% distinct values)
- Location: high cardinality with 12220 distinct values
- Location: 2801 (20.9%) missing values
- TweetAt and UserName are highly correlated at 92.9%
- ScreenName and UserName are highly correlated at 100% (both columns each have 100% distinct)
- UserName, ScreenName, and OriginalTweet are all uniformly distributed variables

NLP Unsupervised Modeling

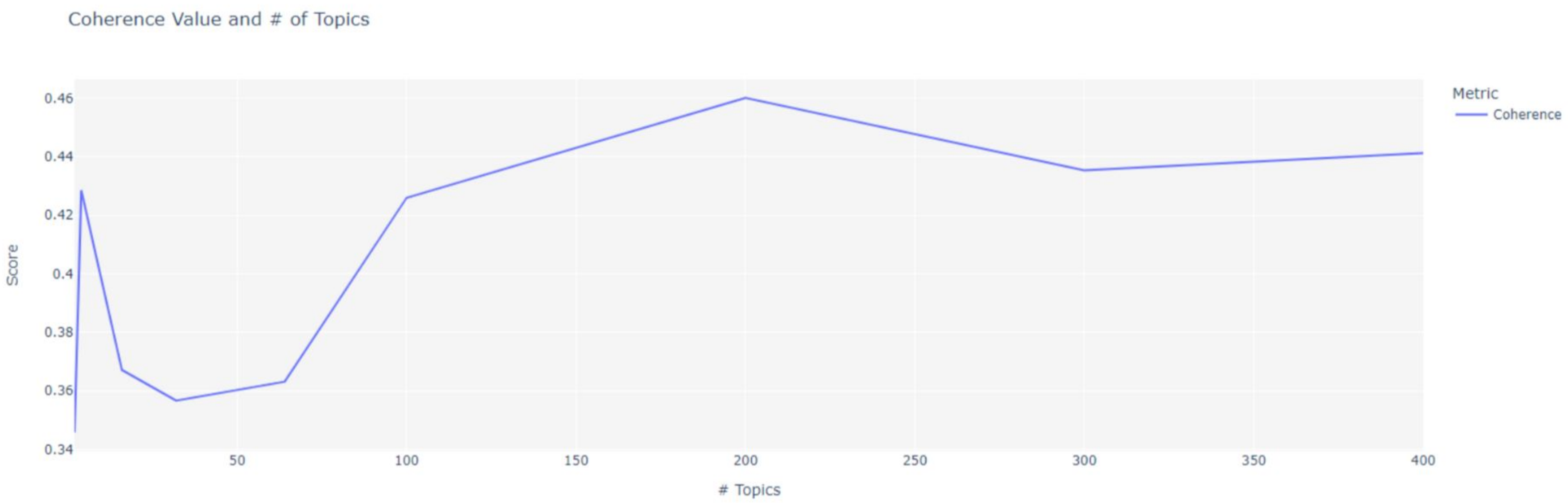
Unsupervised learning methods: Latent Dirichlet Allocation (LDA)

LDA Setup:

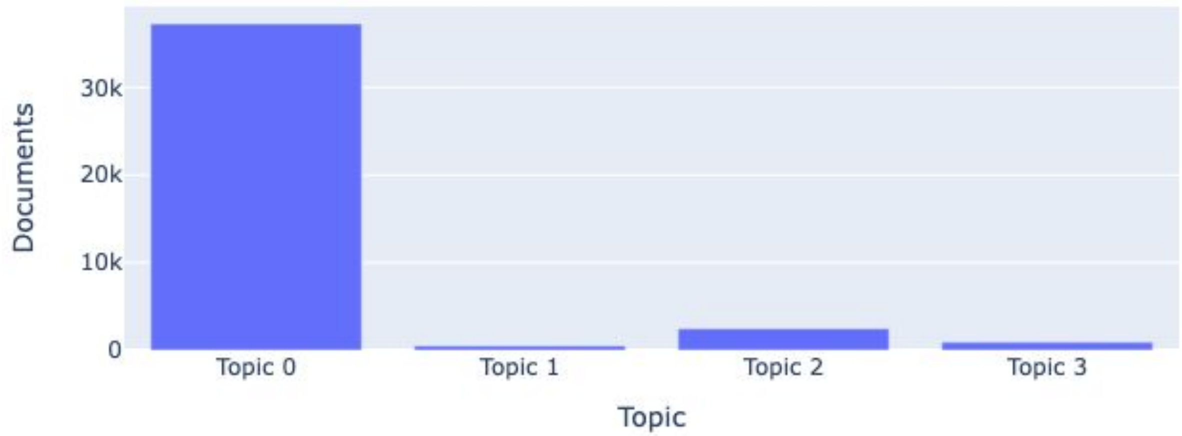
Hyperparameters

a. Custom_stopwords = ['covid', 'coronavirus', 'virus', 'pandemic', 'https', 'co']
b. Target = “OriginalTweet"

LDA Model Evaluation



Distribution by Topics



Keywords:

Topic 0: store, food, grocery, supermarket, go

Topic 1: new, may, use, high, company

Topic 2: consumer, demand, crisis, change, business

Topic 3: price, oil, market, low, lockdown

NLP Topic Modeling To Classification

1. Generate NLP model topics: lda_df = assign_model(lda)
2. Save NLP result for classification: lda_df.to_csv('lda_result_for_clasification.csv')
3. Example:

Original Tweet: "airline offer stock shelf…"

Topic_0	Topic_1	Topic_2	Topic_3	Dominant_Topic	Perc_Dominant_Topic
0.465304	0.178543	0.225528	0.130626	Topic 0	0.47

Setup/Data preparation For Multiclass Classification:

1. Dataset: Output from NLP model
2. Encoding sentiment (Extremely Negative: -2, Negative: -1, Neutral: 0, Positive: 1, Extremely Positive: 2)
3. Hyperparameters
    a. target= ‘encoded_sentiment’
    b. train_size= 0.8
    c. fold_strategy=’kfold’
    d. ignore_features=[‘UserName’, ‘ScreenName’, ‘Location’, ‘TweetAt’, ‘OriginalTweet’, ‘Sentiment’, Topic_0', 'Topic_1', 'Topic_2','Topic_3']
    e. normalize=True
    f. normalize_method= robust
4. Features: dominant topic (categorical), percentage dominant topic (numerical)

Multiclass Classification Supervised Modeling

```
best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.2795	0.5322	0.2031	0.1543	0.1443	0.0053	0.0164	2.4950
ridge	Ridge Classifier	0.2795	0.0000	0.2031	0.1543	0.1443	0.0053	0.0164	0.0310
lda	Linear Discriminant Analysis	0.2791	0.5322	0.2031	0.1683	0.1448	0.0053	0.0158	0.0500
ada	Ada Boost Classifier	0.2784	0.5431	0.2062	0.1933	0.1669	0.0095	0.0160	0.8610
dummy	Dummy Classifier	0.2775	0.5000	0.2000	0.0771	0.1207	0.0000	0.0000	0.0230
gbc	Gradient Boosting Classifier	0.2768	0.5430	0.2071	0.2204	0.1762	0.0105	0.0160	6.8190
rf	Random Forest Classifier	0.2763	0.5402	0.2092	0.2529	0.1866	0.0133	0.0182	1.0070
lightgbm	Light Gradient Boosting Machine	0.2754	0.5402	0.2087	0.2424	0.1862	0.0124	0.0169	1.7750
dt	Decision Tree Classifier	0.2742	0.5398	0.2085	0.2481	0.1878	0.0118	0.0158	0.0810
et	Extra Trees Classifier	0.2742	0.5398	0.2085	0.2477	0.1877	0.0118	0.0158	0.7630
qda	Quadratic Discriminant Analysis	0.2294	0.5339	0.2109	0.1507	0.1233	0.0115	0.0204	0.0970
knn	K Neighbors Classifier	0.2195	0.5101	0.2128	0.2254	0.2181	0.0149	0.0150	0.8840
nb	Naive Bayes	0.2109	0.5363	0.2193	0.1319	0.1443	0.0184	0.0234	0.0550
svm	SVM - Linear Kernel	0.1981	0.0000	0.2032	0.1684	0.1310	0.0041	0.0035	0.2460

Deployment

- Dual Model Deployment
  - Trying to import the model from the Linear Dirichlet Allocation(LDA) and the Logistic Regression(LR) caused issues
  - Recreated the LDA model in the Web App
  - Imported the LR Model to run the prediction
- Other Issues
  - Execution of predict function takes some time due to creation of LDA Model
  - Matched column headers to the amount of data passed
- Regular Expressions (RegEx) Validation
- Deployed on Heroku

https://tweet-sentiment-analysis.herokuapp.com

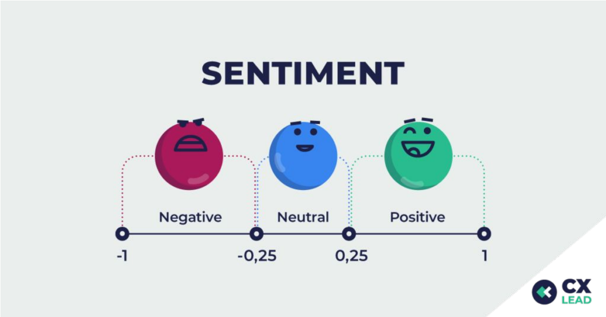
Next Steps

Rationale:

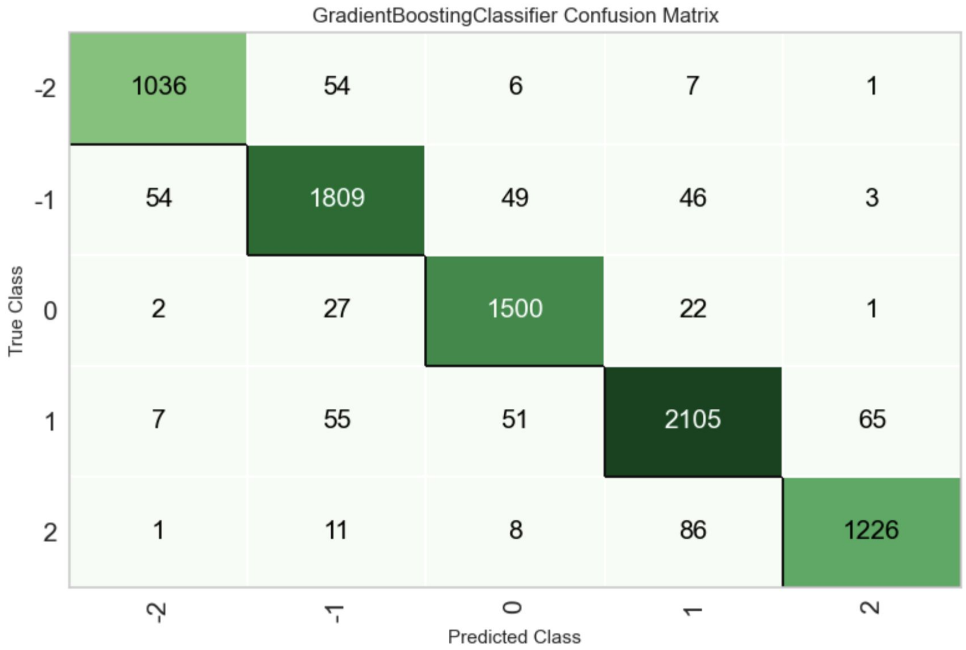
Given our desire to improve the accuracy score of our current model we looked at feature engineering to improve classification performance.

Instead of mapping our LDA generated topics to sentiment, we explored creating new features through pre-trained sentiment dictionaries that can instead be mapped to sentiment:

1. Polarity score
2. AFINN (Affective Norms for English Words) score



Next Steps: Evaluation




```
pred_unseen = predict_model(gbc, data = df_test)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Gradient Boosting Classifier	0.8994	0.9764	0.9017	0.8998	0.8994	0.8722	0.8723

Thank You!

Any questions?