“when shit happens, Data Analysts save the day”

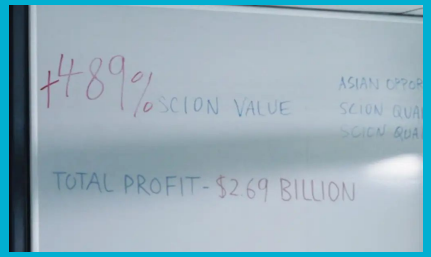
reddit.com Web APIs & Classification

Problem Statement

Money

reddit

r/ValueInvesting
r/technicalanalysis



The overall success of the prediction model shall be based upon its Accuracy and a balance of other metrics like Sensitivity, Specificity and Precision.

Analysts Team
- binary classification problem
- JSON API and NLP

Marketing Director
“find a way to process the salvaged data and re-classify them as accurately as possible into the respective subreddits.”

Data collection and cleaning

1 Collection

- Packages - Requests, Pandas, etc
- reddit.com JSON API
- 25 posts/run x 50 runs
- 2488 rows x 115 columns

2 Data Dictionary

- Only 5 columns required
- subreddit, selftext, title, author, created_utc
- Data Dictionary (required fields)
- Feature, Description, Datatype, Range, Example

Feature	Description	Datatype	Range	Example
subreddit	name of the subreddit	string	undefined	"ValueInvesting" "technicalanalysis"
selftext	sentences made by the author	string	undefined	"Benjamin Graham often spoke about the importance of bonds in an intelligent portfolio"
title	title of the post	string	undefined	"Research on price movements due to large investors"
author	name of the author	string	undefined	"tropicalcoconut29"
created_utc	date and time of post creation in UTC format	string	varies	"1606597322"

3 Cleaning

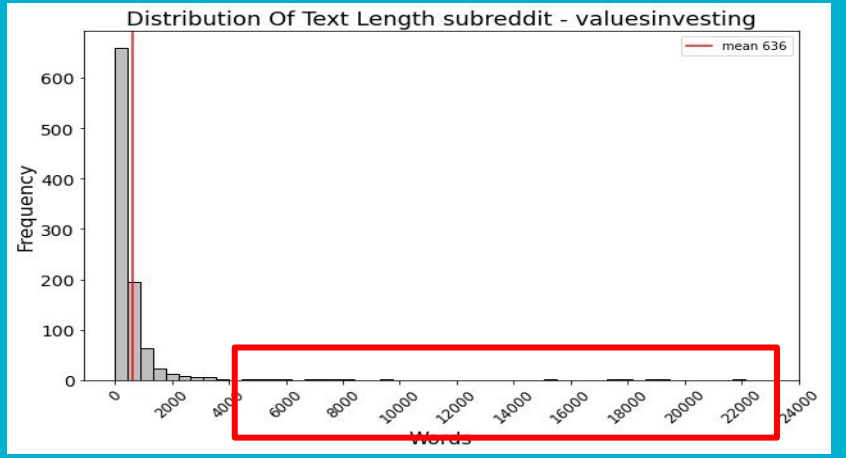
- Remove duplicated posts (1844 rows x 5 columns)
- Impute NaN with "---"
- pd.to_datetime on 'created_utc'
- Combined 'selftext' and 'title' (feat eng)
- Mapped Class 1 and 0 (encoding)

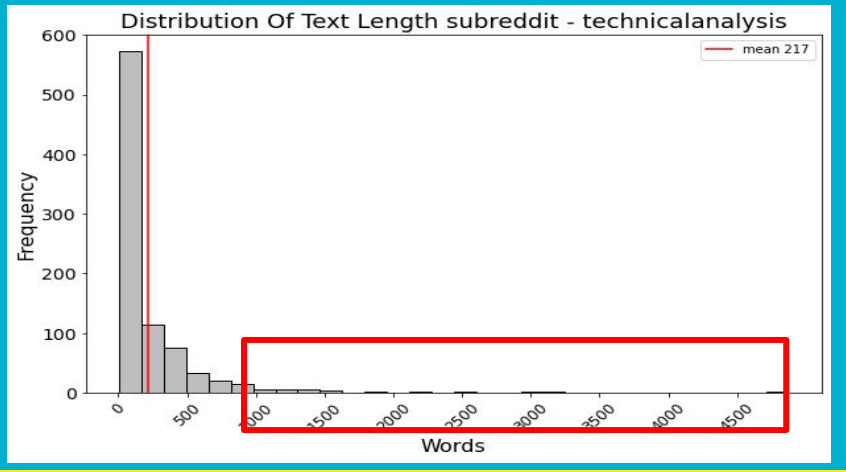
Exploratory Data Analysis

ANALYSIS ON POSTS

	r/valueinvesting	r/technicalanalysis
Range (min, max)	15 to 22,132 words	7 to 4871 words
Majority Posts Length	< the mean length of 636 words	< mean length of 217 words
Long Posts	Several number of posts > 4,000 words	~ 500 to 4871 words
Period	2020-02-20 to 2020-11-28	2018-05-23 to 2020-11-28

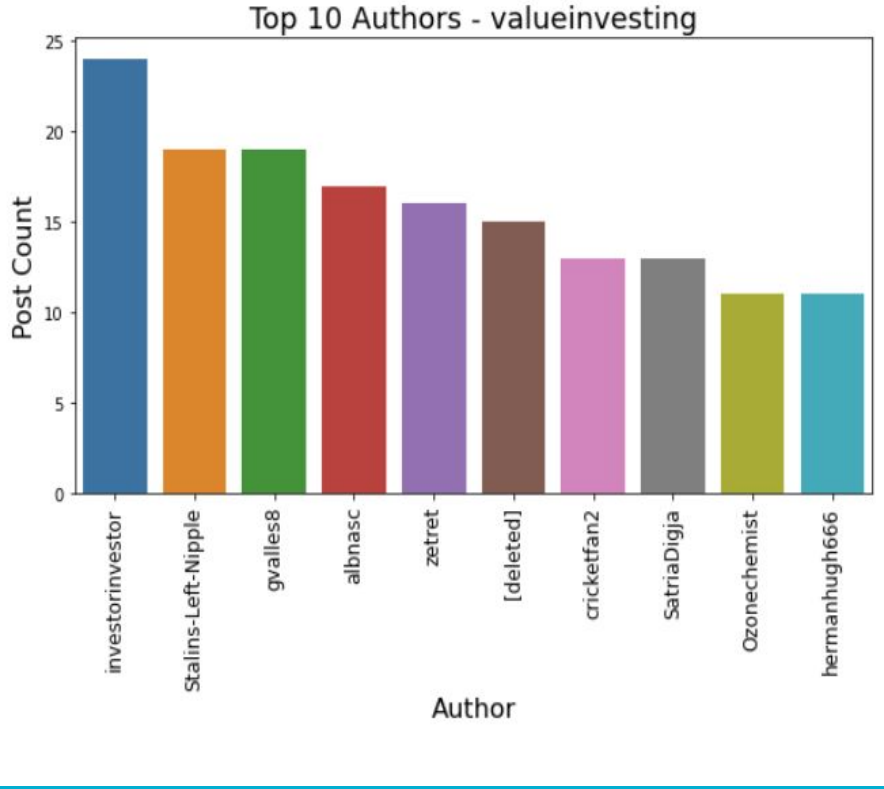
```
subreddit
technicalanalysis    2018-05-23 13:52:56
valueinvesting       2020-02-20 15:27:53
Name: created_utc, dtype: datetime64[ns]

subreddit
technicalanalysis    2020-11-28 21:15:45
valueinvesting       2020-11-28 23:11:35
Name: created_utc, dtype: datetime64[ns]
```

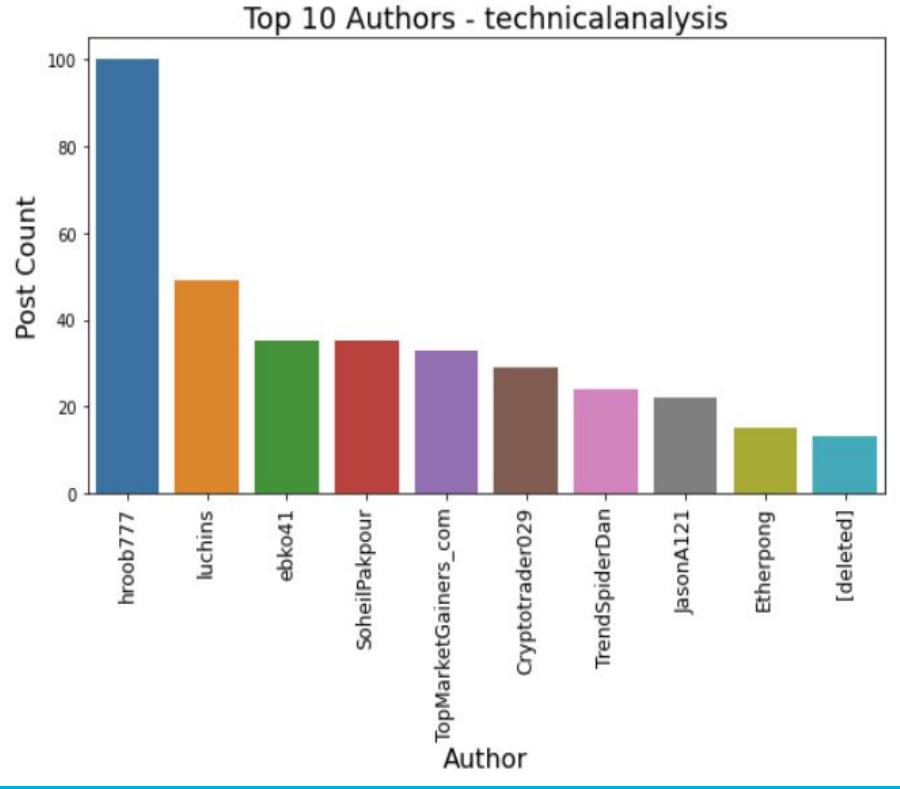




Exploratory Data Analysis

TOP AUTHORS ANALYSIS

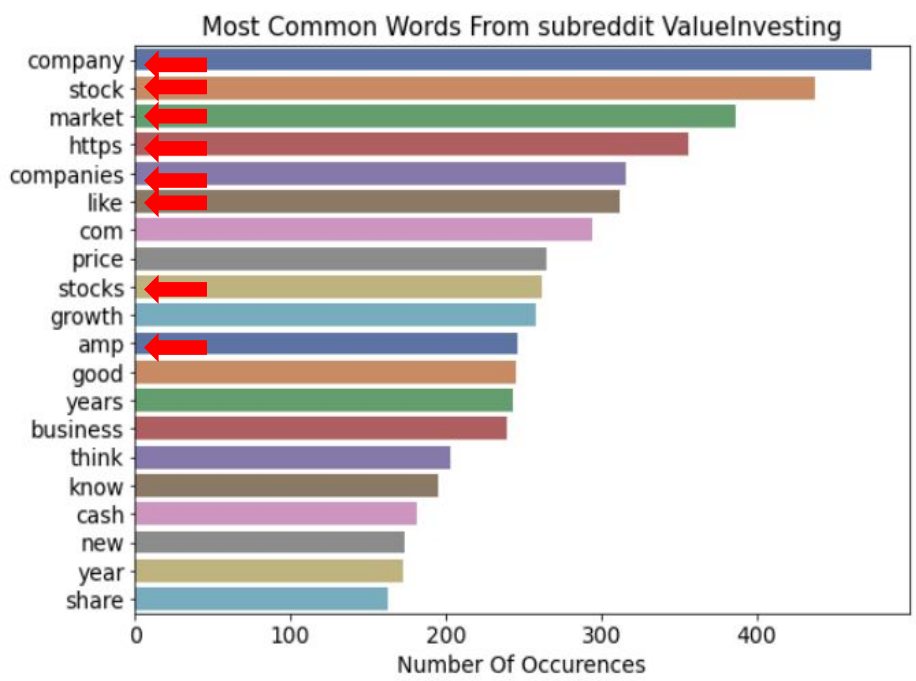


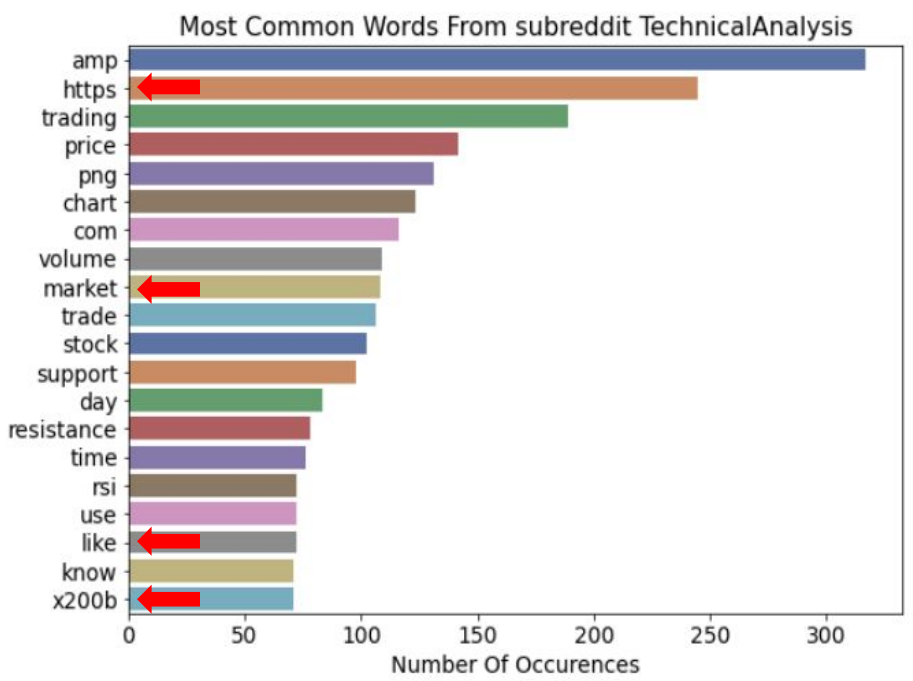


- appeared to be separate group of authors with none frequently posting in both subreddits

Exploratory Data Analysis

TOP WORDS ANALYSIS

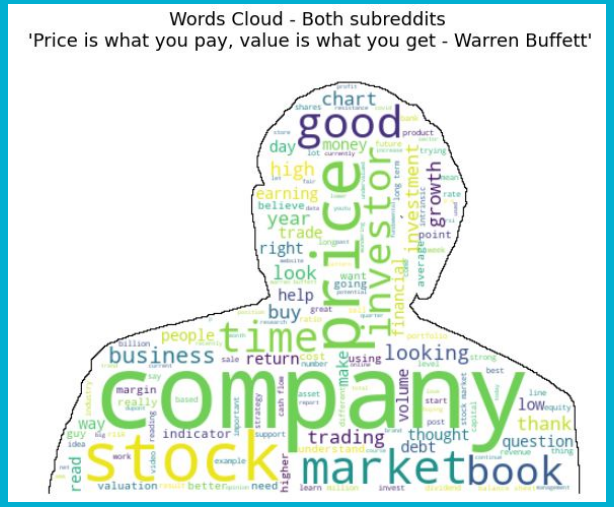


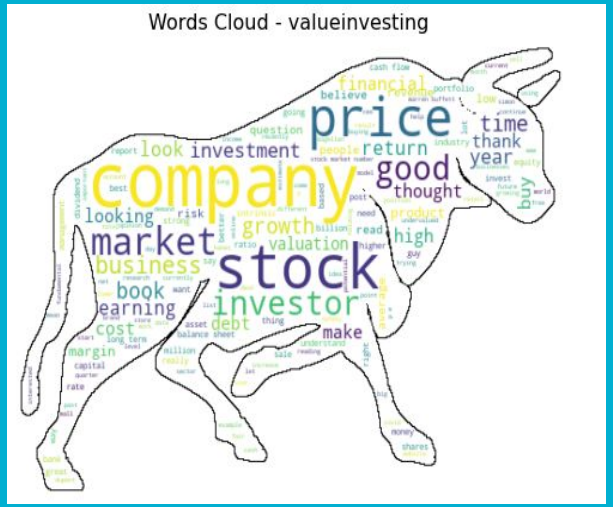


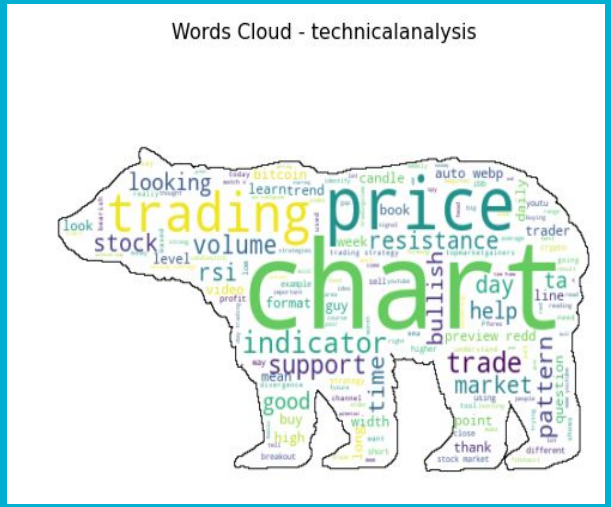
- many words occurring in both subreddits ('https', 'market')
- lemmatize or stem was necessary, multiple forms of words ('stock'-'stocks', 'company'-'companies')
- words that did not make sense ('amp', 'x200b')
- update the custom stopwords list with some of these words

Exploratory Data Analysis

WORDS CLOUD ANALYSIS







- many common words between the subreddits that could make it hard to classify them correctly
- customized stopwords list was definitely necessary but not to the point where it becomes too hard to differentiate the subreddits
- there were other parameters that could be used to tune this heuristic

Pre-processing, modeling and tuning

1 Tokenize, Stemming or Lemmatizing

- Tokenize first
- both Stemming and Lemmatizing return the root form of the words
- stemming might not return an actual word whereas lemmatizing does mostly
- Stemming is faster
- Stemming chosen

2 Vectorizer and Model Choice

- CountVectorizer
- TF-IDF Vectorizer
- Logistic Regression
- Multinomial Naive Bayes

- Pipeline
- GridSearch
- Parameters selection / choices

3 Train-Test-Split and Baseline

- If our model has an accuracy that is greater than 53.69%, we know that it is better than simply guessing the class of a post to be coming from subreddit r/valueinvesting (Class 1)

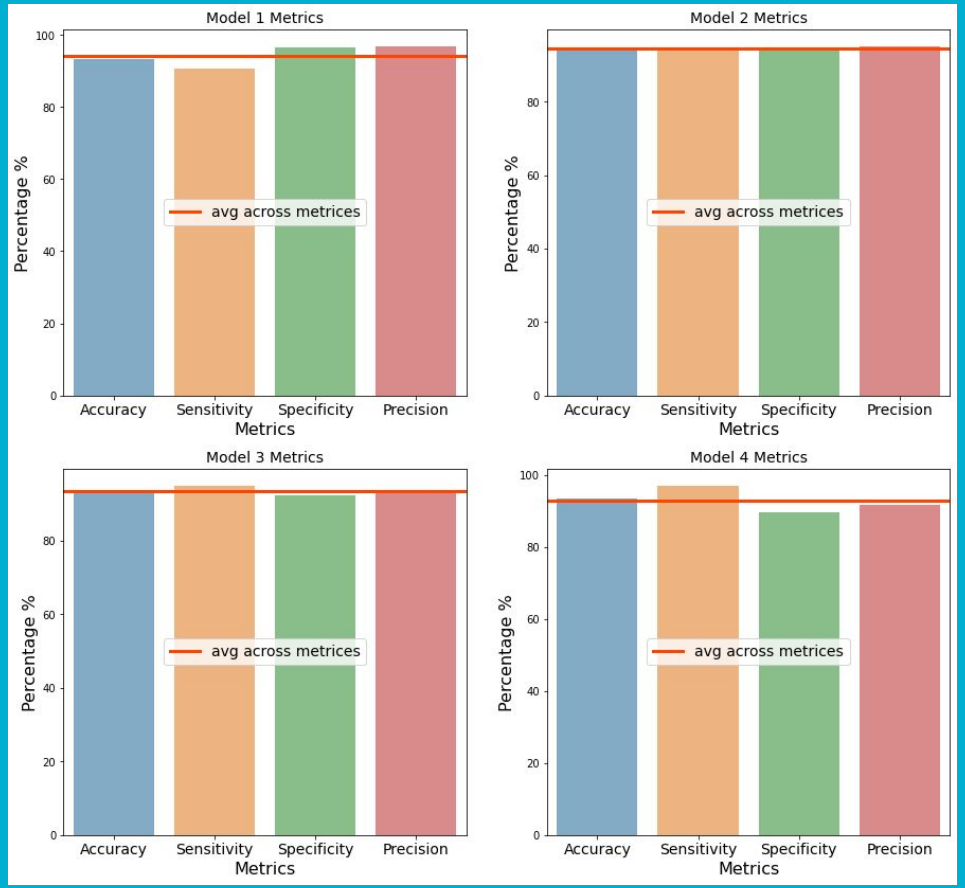
tok_text	stemmed_text	lemmatized_text
[google, fitbit, merger, potential, opportunit...	googl fitbit merger potenti opportun for satis...	google fitbit merger potential opportunity for...
[what, s, the, difference, between, a, triangl...	what s the differ between a triangl wedg amp p...	what s the difference between a triangle wedge...

```
pipe1 = Pipeline([('cvec1', CountVectorizer(lowercase=True,
                                            stop_words=custom_stopwords,
                                            analyzer='word')),
                  ('logreg1', LogisticRegression())])

# Search over the following values of hyperparameters
pipe1_params = {
    'cvec1__max_features': [3000,5000,7000,9000],  # since we have <9000
    'cvec1__min_df': [2,3],
    'cvec1__max_df': [0.8,0.9,1],
    'cvec1__ngram_range': [(1,1), (1,2)]}
```

```
# For a baseline estimator, we choose 1 which is
round(y_test.value_counts(normalize=True),4)*100
```

```
1    53.69
0    46.31
Name: subreddit_class, dtype: float64
```

Model evaluation result



METRICS ANALYSIS

	metrics	model1_%	model2_%	model3_%	model4_%
0	Accuracy	93.27	94.25	93.60	93.60
1	Sensitivity	90.52	94.19	94.80	96.94
2	Specificity	96.45	94.33	92.20	89.72
3	Precision	96.73	95.06	93.37	91.62

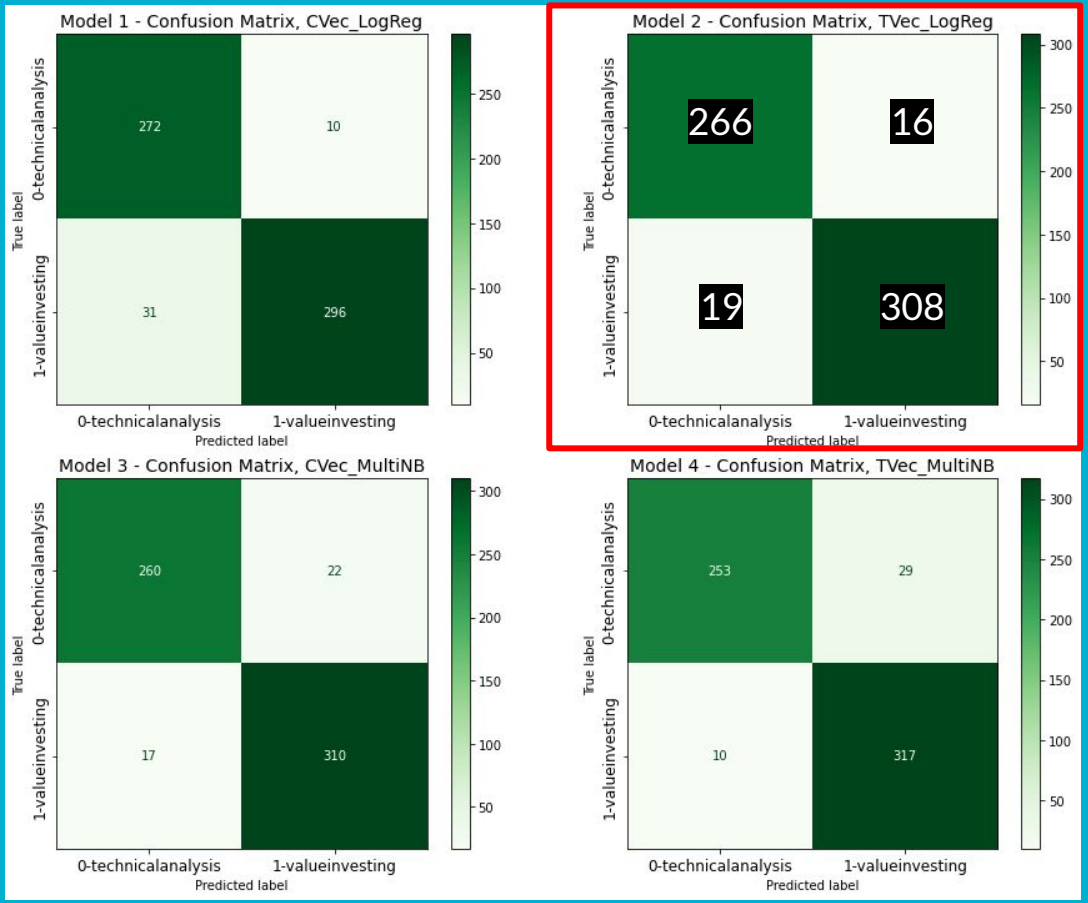
```
Model1 sum of matrices % =  376.97
Model2 sum of matrices % =  377.83
Model3 sum of matrices % =  373.97
Model4 sum of matrices % =  371.88

Model1 mean of matrices % =  94.24
Model2 mean of matrices % =  94.46
Model3 mean of matrices % =  93.49
Model4 mean of matrices % =  92.97
```

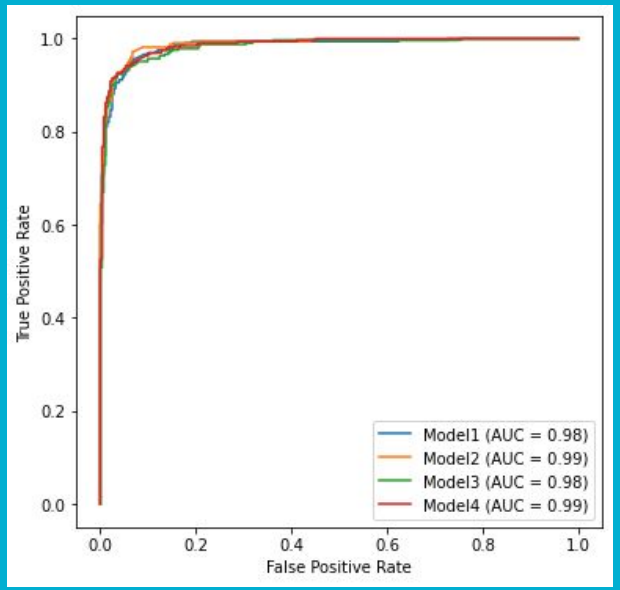
Generally higher was better in our case so a simple sum of mean of all the metrics percentage would suffice to give a holistic comparison.

- Accuracy: How many of all observations did the model predicted correctly?
- Sensitivity: Among all true positives, how many did the model predicted correctly? (also known as recall)
- Specificity: Among all true negatives, how many did the model predicted correctly?
- Precision: Among the true and false positives, how many did the model predicted correctly?

Model evaluation result

ROC AUC CURVES AND CONFUSION MATRIX ANALYSIS

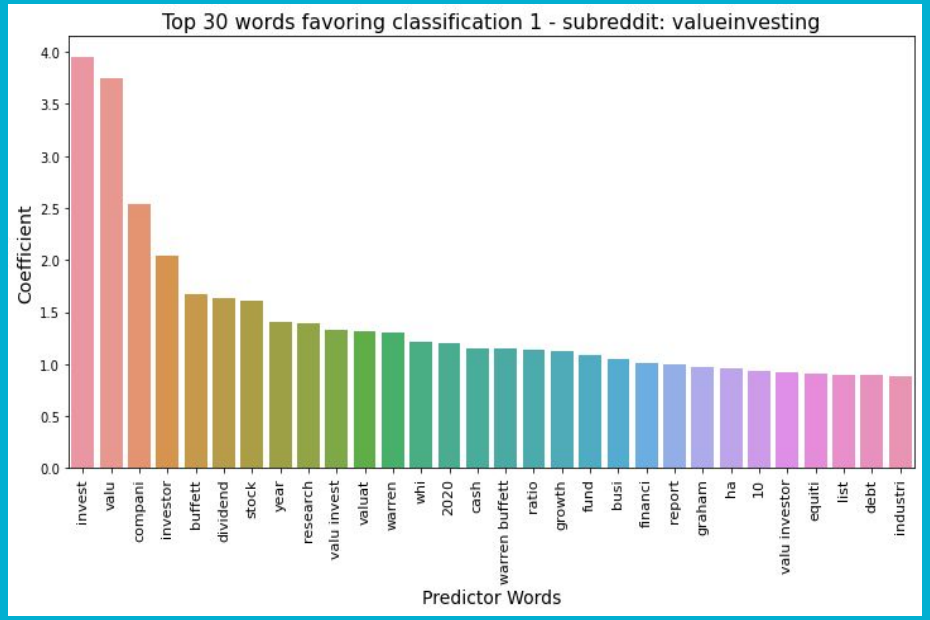


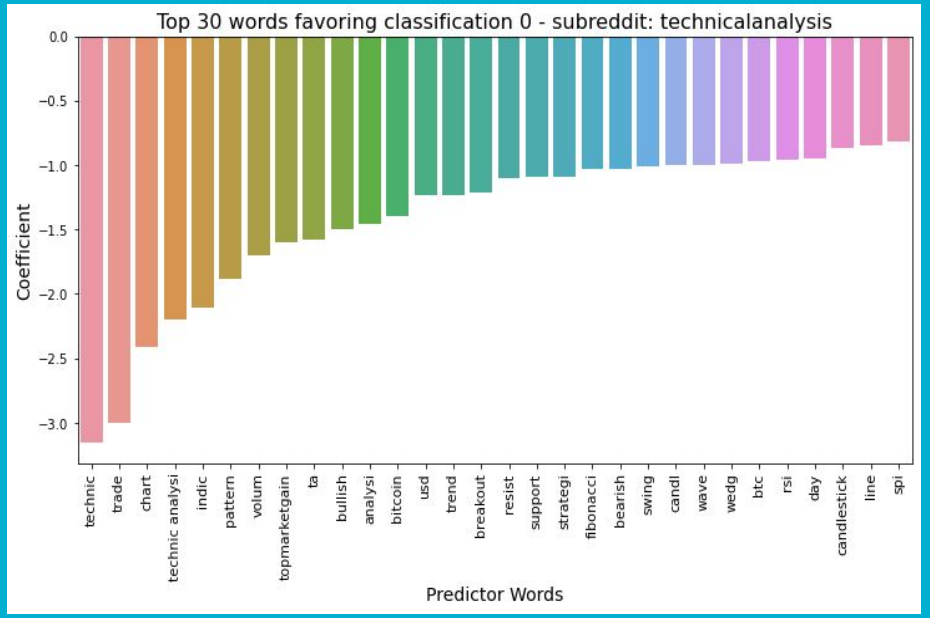


- all four models returned very high AUC scores
- Model 2: FP+FN = 16+19 = 35 (smaller is better meaning model is making lesser mistakes i.e. misclassification)
- Model 2: TP+TN = 308+266 = 574 (bigger is better meaning model is making more right predictions. i.e. accuracy)

Model evaluation result

MODEL 2 TOP PREDICTOR WORDS ANALYSIS





- Sanity check with domain knowledge, the top 30 predictor words were closely related to valueinvesting:
  - 'buffett' - value investor 'Warren Buffett'
  - 'valu' (stemmed from value) and 'valuat' (stemmed from valuation) - common in value investing
  - 'ratio', 'dividend', 'growth', 'fund' linked to value investing

- Sanity check with domain knowledge, the top 30 predictor words were closely related technicalanalysis:
  - 'indic' (stemmed of 'indicator'), 'pattern', 'ta', 'bitcoin' and 'trend' were words familiar with technical analysis
  - 'fibonacci', 'candl' (stemmed of candle), 'volum' (stemmed of volume), 'breakout' and 'resis' (stemmed of resistance) were technical analysis charting methods

Model evaluation result

MODEL 2 MIS-CLASSIFICATIONS ANALYSIS

	original_text	Class 0 - technicalanalysis_proba	Class 1 - valueinvesting_proba	stemmed_text	subreddit_class	predict_class
543	8 mind blow technic chart that mean absolut noth	0.859607	0.140393	8 mind blow technic chart that mean absolut noth	1	0
15	etf breakdown chart1 is there a quick way i ca...	0.728344	0.271656	etf breakdown chart1 is there a quick way i ca...	1	0
273	best stock analysi tool	0.613811	0.386189	best stock analysi tool	1	0

	coef	predictor_words
197	-1.457650	analysi
390	0.115504	best
2430	1.610023	stock
2685	-0.632651	tool

Simple method : visual comparison with original text and validate count of words against top 30 words in each subreddit. Domain knowledge would be applied here as well.

- Original post: Best Stock Analysis Tools---
- Stemmed text: best stock analysi tool
- >No of words found in valueinvesting top 30 = 1
- >They were ['stock']
- >No of words found in technicalanalysis top 30 = 1
- >They were ['analysi']

Enhanced method : tabulate coefficient scores of words found + not found in top 30 words to determine the favored classification by the model.

Conclusion

1. Model 2 - TF-IDFVectorizor and Logistic Regression was the best model in predicting whether a post came from subreddit r/technicalanalysis or r/valueinvesting.

2. Best parameters were:
   - max_df': 0.8, max_features': 3000, min_df': 3, ngram_range': (1, 2), stop_words: custom_stopwords

3. With the custom_stopwords, the Model 2was able classify the posts with an accuracy of 94.25% on test data.

4. Model 2 had the best balance of Accuracy, Sensitivity, Specificity and Precision suited for solving our problem.

5. The number of mis-classifications were low in our test data and we were able to trace and explain the reasons for the mis-classifications.

6. Henceforth, resolving the problem that the marketing team and analyst team were facing.

Recommendations

1. The models evaluated could be fine tune for better performance if 94.25% accuracy is not sufficient.

2. Introduce more hyperparameters to tune the models but at a balance of computational cost and accuracy/mis-classification trade-offs at some point.

3. Future improvements to consider:
   - collecting more data
   - using fewer or more features (depending on models) by setting max_features parameter when instantiating the Vectorizers
   - trying a non-default class priority on MultinomialNB if you have subject-matter expertise
   - adjusting L1, L2 penalties or changing solver (for Logistics Regression) or Alpha (for Multinomial NB) to improve regularization
   - rather than regularizing we can try a different model entirely like DecisionTrees, RandomForest, etc.

4. From a business application perspective, we also recommend that the model be expanded to be able to process more than 2 subreddits as there could be more investing related subreddits that the marketing team could target as well. In fact, even extending to other forums/websites

the end
Thank you

we are almost there