SENTIMENT ANALYSIS

- ➢ Machine learning tool to extract emotions from text using Natural Language Processing (NLP)
- ➢ By training ML tools with example of emotions in text, machines automatically learn how to detect sentiment without human input

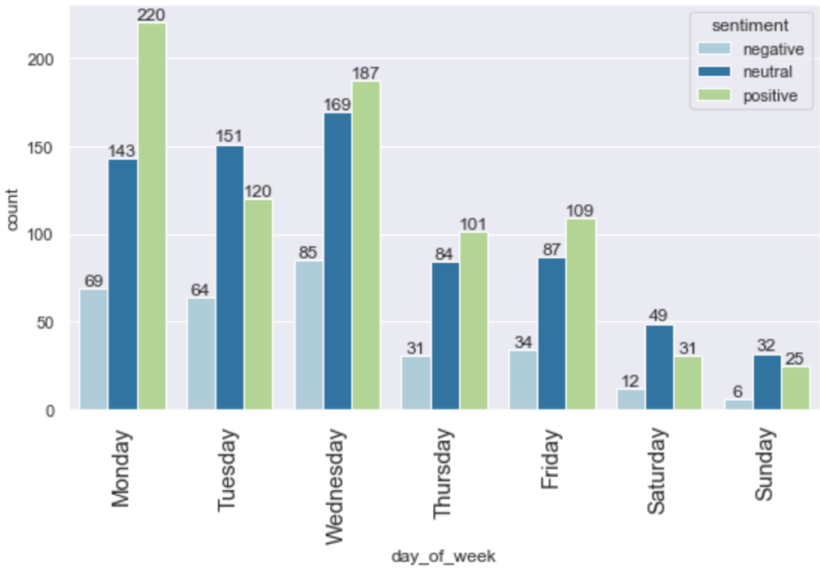
- ➢ The dataset contains approximately 2000 different (scrapped) tweets with the following attributes:

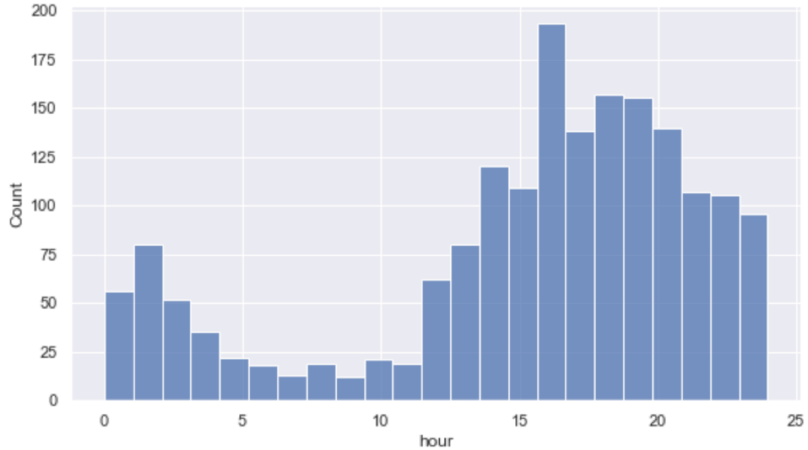
- ***'id'*** : unique 19 digit id for each tweet
- '***created_at'*** : date & time of each tweet (or retweet)
- ***'text'*** : tweet details/ description
- ***'location'*** : origin of tweet

- ➢ **'sentiment'** : target column is created for each tweet using a lexicon based functionality TextBlob (https://textblob.readthedocs.io/en/dev/) with values either 'neutral', 'positive' or 'negative'

Feature Engineering & Exploratory Data Analysis

➢ New features (day_of_week & hour) were created using 'created_at' datetime attribute





➢ **day_of_week**

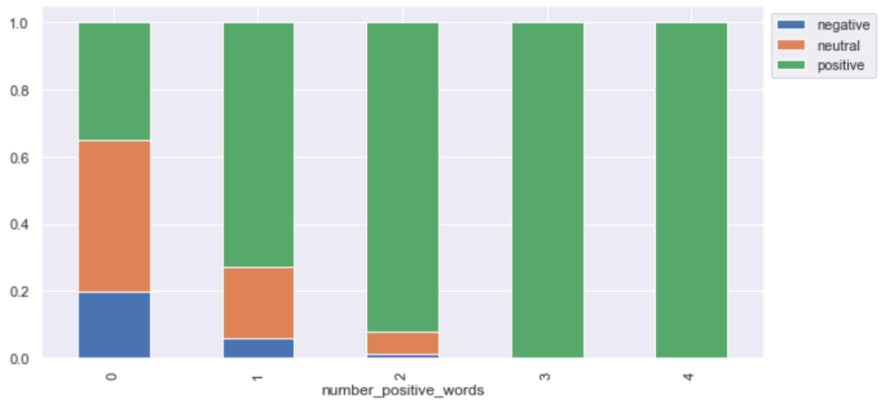
- All weekdays (except Tuesdays) have majority of tweets as positive tweets, while weekends & Tuesdays have majority of tweets as neutral tweets

➢ **hour**

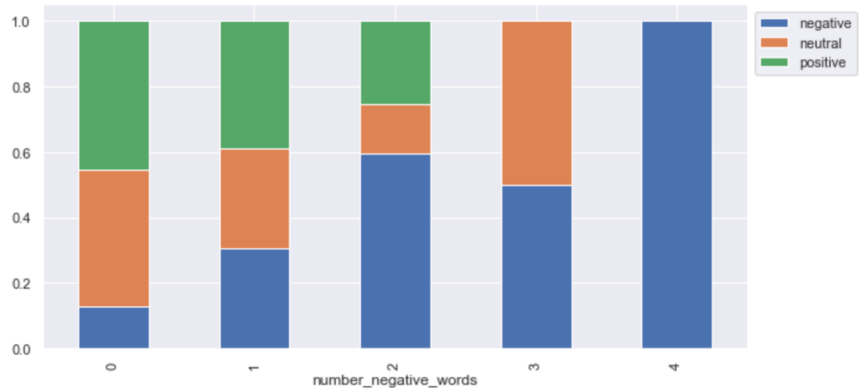
- Majority of the tweets are in the late afternoons & evenings (peaking after 3PM)

- New features were created by counting number of positive & negative words in each tweet. The list of all positive & negative words are borrowed from this study: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

sentiment number_positive_words	negative	neutral	positive	All
All	301	715	793	1809
0	285	654	505	1444
1	15	55	190	260
2	1	6	85	92
3	0	0	9	9
4	0	0	4	4



sentiment number_negative_words	negative	neutral	positive	All
All	301	715	793	1809
0	189	625	676	1490
1	82	82	105	269
2	28	7	12	47
3	1	1	0	2
4	1	0	0	1



- **number_positive_words**
  - Tweets with 1, 2, 3 or 4 number_positive_words are majorly a 'positive' sentiment as expected

- **number_negative_words**
  - Tweets with 2, 3 or 4 number_negative_words are majorly a negative' sentiment as expected

Text Pre-processing

text
@CIBC please explain to me why I want to remai...
RT @CIBCLiveLabs: We are pleased to announce, ...
CIBC World Markets Inc. Decreases Holdings in ...
Your home is a valuable asset. Use your equity...
Both #CIBC and #BMO ads at the Chicago Black H...

Conversion to lowercase &

Removing non-alphanumeric characters like ‘@’ or ‘#’

text
cibc please explain to me why i want to remai...
rt cibclivelabs we are pleased to announce ...
cibc world markets inc decreases holdings in ...
your home is a valuable asset use your equity...
both cibc and bmo ads at the chicago black h...

Tokenization

tokens
[cibc, please, explain, to, me, why, i, want, ...
[rt, cibclivelabs, we, are, pleased, to, annou...
[cibc, world, markets, inc, decreases, holding...
[your, home, is, a, valuable, asset, use, your...
[both, cibc, and, bmo, ads, at, the, chicago, ...

Lemmatization

tokens
[cibc, please, explain, to, me, why, i, want, ...
[rt, cibclivelabs, we, are, pleased, to, annou...
[cibc, world, market, inc, decrease, holding, ...
[your, home, is, a, valuable, asset, use, your...
[both, cibc, and, bmo, ad, at, the, chicago, b...

- Can see examples of words being broken down to the root like markets has become market & ads has become ad

Stemming

tokens
cibc pleas explain to me whi i want to remain ...
rt cibclivelab we are pleas to announc cibc in...
cibc world market inc decreas hold in ing groe...
your home is a valuabl asset use your equiti t...
both cibc and bmo ad at the chicago black hawk...

- Can see examples of words being broken down to root (irrespective of the tense) like please & pleased has become pleas

- **Word Cloud before (left) & after (right) text pre processing**

- 'cibc', 'co', 'bank', 'thank' are some of the common word occurrences in tweets after text pre processing
- ‘https' was a common occurrence in the original tweets but is no longer a common occurrence after text pre processing

- ***‘location’*** attribute is too unclean and will be dropped from analysis

- **‘ sentiment’**

- There is imbalance in sentiment attribute; Maximum tweets have a positive sentiment followed by neutral sentiment

positive	793
neutral	715
negative	301

Machine Leaning

- Label encoding of Target (1:'positive', 0:'neutral' & -1:'negative') and one hot encoding of categorical columns like 'day_of_week'

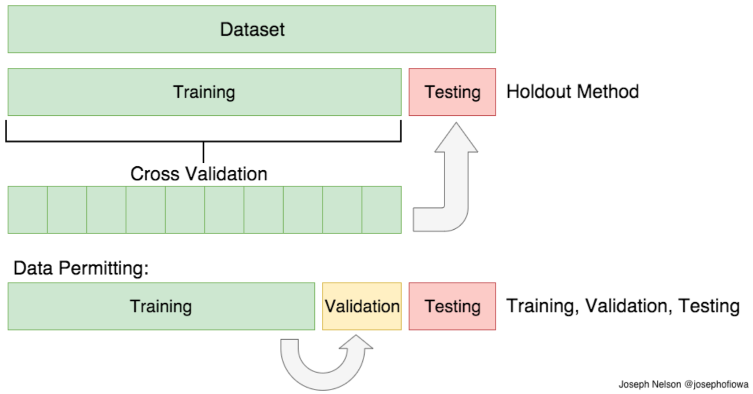
- TF-IDF vectorizer chosen for text feature engineering to convert text (after necessary pre processing) into a numerical matrix
  - This gives importance to 'rare' words in tweets which are not common across all other tweets
  - Use stop_words = 'English' to remove common English occurrences like 'if', 'but', 'or', 'an', 'the' etc.

- RandomOverSampler to account for target imbalance (specifically to account for number of negative tweets being less than positive & neutral ones)

- Choice of popular ML algorithms to work with text data: Multinomial Naive Baiyes, Linear Support Vector Classifier, Random Forest Classifier & XGBoost

- Choice of metric : 'F1' chosen to ensure the maximum possibility of correct predictions across each of the target classes

- ML Workflow
  - Split dataset into training & testing set
  - Perform cross validation on training set to choose best (base) vectorizer & ML algorithm
  - Perform Grid Search Cross Validation on training set using vectorizer & chosen ML algorithm to get best hyperparameters
  - Use the final classifier to make predictions on testing set



https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

5-fold Cross Validation Results

- Best scores were with Multinomial Naive Bayes & XGBoost. Multinomial Naive Bayes takes the least amount of time to fit & predict, while XGBoost takes the maximum amount of time
- Random Forest is easier to interpret with class_feature_importance built in the module with not much compromise in performance score & time

```
Cross Validation Model Performance on Training Set - TfidfVectorizer

SupportVectorMachine : 0.38172497048147835
Time : 0.4990891933441162


MultinomialNaiveBayes : 0.4179577924814408
Time : 0.15718913078308105


RandomForest : 0.37329228832046796
Time : 1.3154309272766114


XGBoost : 0.38960427193273184
Time : 33.96854658126831
```

Hyperparameter Tuning – GridSearchCV - Random Forest

- Build a pipeline to choose the best hyperparameters for both TF-IDF Vectorizer & chosen ML algorithm (Random Forest) using GridSearchCV object

```
pipe       = Pipeline(
                 [('TfidfVectorizer',TfidfVectorizer(stop_words='english')),
                  ('oversampler',    RandomOverSampler(random_state=1)),
                  ('RandomForest',   RandomForestClassifier(random_state=1))])

parameters = {
              'TfidfVectorizer__ngram_range' : [(1,1),(1,2)],
              'TfidfVectorizer__max_features': range(1000,10000,2500),
              'RandomForest__n_estimators'   : range(10,100,25),
              'RandomForest__criterion'      : ['gini', 'entropy'],
              'RandomForest__max_depth'      : range(3,15,3),
              'RandomForest__max_features'   : ['auto', 'sqrt'],
            }
```

```
Best Hyperparameters are:
 {'RandomForest__criterion': 'gini', 'RandomForest__max_depth': 12, 'RandomForest__max_features': 'auto', 'RandomForest__n_estimators': 85, 'TfidfVectorizer__max_featur
es': 1000, 'TfidfVectorizer__ngram_range': (1, 1)}
Best Score is:
 0.7199566069352127
```

Final Classifier

- **Performance on (unseen) Testing Dataset**
  - The final classifier is able to achieve a F1 score of ~65% for negative sentiments, ~>=75% for positive & neutral sentiments

Final Classifier Unbiased Testing Performance:

	precision	recall	f1-score	support
-1	0.84	0.52	0.64	60
0	0.70	0.91	0.79	143
1	0.88	0.76	0.81	159
accuracy			0.78	362
macro avg	0.80	0.73	0.75	362
weighted avg	0.80	0.78	0.78	362

- **Feature Importance**
  - The feature importance words & tweet sentiments make somewhat intuitive sense lending confidence in the explainability of the final model!

```
-1: [('growth', 0.020508279448409432),
 ('mortgag', 0.019366596158985994),
 ('flat', 0.01593803535685603),
 ('low', 0.013638174657405067),
 ('ceo', 0.00991227325635339),
 ('number_negative_words', 0.009865706883072735),
 ('singl', 0.008695714600743245),
 ('year', 0.007091239388495461),
 ('thi', 0.0036762564809373643),
 ('sorri', 0.0034910600221196805),
 ('digit', 0.0034875670207551697),
 ('forese', 0.0022939260500948098),
 ('nint', 0.001975114963768389),
 ('expect', 0.0019119478668486187),
 ('head', 0.0019059296524093635),
 ('close', 0.0015397788130216727),
 ('prioriti', 0.001397608266141923),
 ('poll', 0.0013932492910597326),
 ('fli', 0.0013557985039408462),
 ('wait', 0.0013532427413267963)]}
```

negative sentiments (-1)

```
[0: [('http', 0.005254903628625936),
  ('cibc', 0.0035686977763002403),
  ('outperform', 0.0012783284104765282),
  ('upgrad', 0.001105422848748587),
  ('canopi', 0.0008237733183353656),
  ('million', 0.0006391237213643123),
  ('affili', 0.0005750117591376953),
  ('pot', 0.0005617869930377333),
  ('pharmhous', 0.0005171659119547022),
  ('stock', 0.00048245308190496906),
  ('compani', 0.0003972520080258514),
  ('80', 0.0002840897183251984),
  ('loan', 0.0002778209808019329),
  ('recruit', 0.0002509295685181296),
  ('reaffirm', 0.00024354833156878063),
  ('provid', 0.00023361912402952383),
  ('specincanada', 0.00021524804917062436),
  ('gold', 0.00018683605095486946),
  ('price', 0.0001651093407695493),
  ('market', 0.00015872398701660342)],
```

neutral sentiments (0)

```
1: [('number_positive_words', 0.024674967297793084),
 ('thank', 0.0075800359501043595),
 ('new', 0.006824029545674138),
 ('latest', 0.004987000695066227),
 ('read', 0.004323775844940422),
 ('beat', 0.004250830164816125),
 ('daili', 0.0033137842100420677),
 ('click', 0.0030861135368116816),
 ('gundi', 0.002906447759835412),
 ('isg', 0.0021729156451157695),
 ('look', 0.0019328118519697814),
 ('love', 0.001404756888657215),
 ('market', 0.0013694959613604462),
 ('good', 0.0010149851813869692),
 ('great', 0.0010121532625143286),
 ('firstcaribbean', 0.0009225602339887848),
 ('wood', 0.0008998155189470076),
 ('home', 0.0007702234465535215),
 ('game', 0.0007399270199097473),
 ('cool', 0.0006142506427804738)],
```

positive sentiments (1)

Summary & Conclusion

- TextBlob was used to assign sentiment labels to tweets. There was inherently some imbalance in classes of sentiments for the dataset; negative : neutral : positive ratio was 0.15 : 0.43 : 0.40

- Data cleaning, visualization was followed by text analytics (removing non alphanumeric characters, tokenizing sentences, lemmatization, removing stop words & performing vectorization using TF-IDF). This was followed by random oversampling to handle target imbalance

- Four different base models were fitted to the training set. The best cross validated scores though were achieved with Multinomial NB & XGBoost; however Random Forest was chosen for hyperparameter tuning as it's easier to interpret with the class feature importance's, & with not much compromise with score & computational time

- A pipeline was further built for TF-IDF & Random Forest, thereby enabling tuning for the best hyper-parameters. Post tuning, performance was enhanced to achieve a F1 score of ~0.65 for negative sentiment & >0.75 for neutral & positive sentiments on unseen test data

- For the final model, feature importance was identified for all 3 classes (negative, neutral & positive tweets)
  - number_negative_words had high feature importance for predicting negative sentiments & number_positive_words had high feature importance for predicting positive sentiments
  - The words of importance's associated with negative tweets were found to be 'flat', 'mortgage', 'single', 'low', 'sorry', 'close' etc
  - The words of importance's associated with positive tweets were found to be 'new', 'thank', 'wood', 'latest', 'game', 'love', 'home', 'great', 'good'
  - The words of importance's associated with neutral tweets were found to be 'http', 'cibc', 'company', 'recruit', 'reaffirm', 'provide'

The feature importance words & tweet sentiments make somewhat intuitive sense.