Sentiment Analysis Machine Learning Presentation

By Timothy Chan

Agenda

1. Introduction to Sentiment Analysis
2. Pre-processing
3. Other steps - Tokenization, Lemmatization
4. Models used and selection
5. Optimization
6. Test results + Live Demo
7. Next Steps

Sentiment Analysis for Business

What is Sentiment Analysis

Sentiment analysis is the process of analyzing the emotion expressed in a piece of text. It uses natural language processing and machine learning to categorize the sentiment as positive, negative, or neutral.

Business problems -

It is used for social media monitoring, brand reputation management, and customer feedback analysis. It is used for social media monitoring, brand reputation management, and customer feedback analysis.

- Identify and address negative sentiment
- improve customer satisfaction, based on customer feedback and market trends.

Objective of this project

Demonstrate the usage of machine learning to analyse tweets for sentiments. Explore possible prototypes/ use cases for further analysis.

Pre-processing

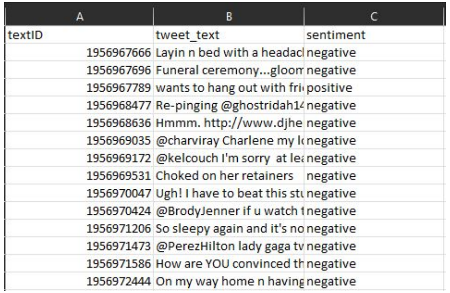
The data

18.7K Tweets from Twitter, sourced from Udemy

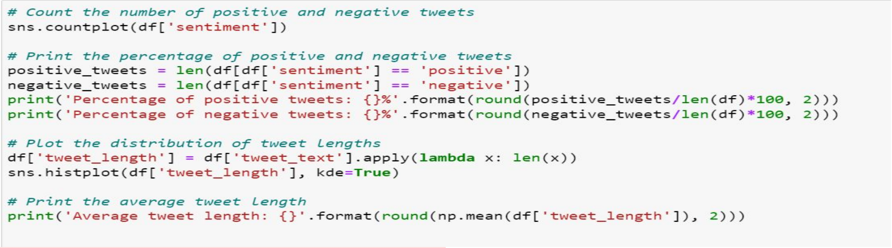
Positive tweets - 52.85%
Negative tweets - 47.15%

The Cleaning

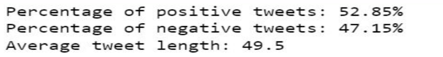
- Lower casing
- Replace all characters in the tweet_text column that are not alphabets (lowercase or uppercase) or hashtags (#) with a single whitespace character
- Stop words are common words such as "the", "and", "in", "of", etc. that are frequently used in a language but do not carry significant meaning on their own



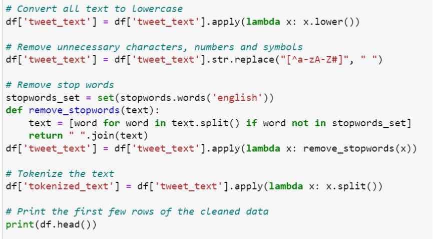
A	B	C
textID	tweet_text	sentiment
1956967666	Layin n bed with a headac	negative
1956967696	Funeral ceremony...gloom	negative
1956967789	wants to hang out with fri	positive
1956968477	Re-pinging @ghostridah1	negative
1956968636	Hmmm. http://www.djhe	negative
1956969035	@charviray Charlene my l	negative
1956969172	@kelcouch I'm sorry at le	negative
1956969531	Choked on her retainers	negative
1956970047	Ugh! I have to beat this st	negative
1956970424	@BrodyJenner if u watch t	negative
1956971206	So sleepy again and it's no	negative
1956971473	@PerezHilton lady gaga tv	negative
1956971586	How are YOU convinced th	negative
1956972444	On my way home n having	negative



```python
# Count the number of positive and negative tweets
sns.countplot(df['sentiment'])

# Print the percentage of positive and negative tweets
positive_tweets = len(df[df['sentiment'] == 'positive'])
negative_tweets = len(df[df['sentiment'] == 'negative'])
print('Percentage of positive tweets: {}%'.format(round(positive_tweets/len(df)*100, 2)))
print('Percentage of negative tweets: {}%'.format(round(negative_tweets/len(df)*100, 2)))

# Plot the distribution of tweet lengths
df['tweet_length'] = df['tweet_text'].apply(lambda x: len(x))
sns.histplot(df['tweet_length'], kde=True)

# Print the average tweet length
print('Average tweet length: {}'.format(round(np.mean(df['tweet_length']), 2)))
```



```
Percentage of positive tweets: 52.85%
Percentage of negative tweets: 47.15%
Average tweet length: 49.5
```



```python
# Convert all text to lowercase
df['tweet_text'] = df['tweet_text'].apply(lambda x: x.lower())

# Remove unnecessary characters, numbers and symbols
df['tweet_text'] = df['tweet_text'].str.replace("[^a-zA-Z#]", " ")

# Remove stop words
stopwords_set = set(stopwords.words('english'))
def remove_stopwords(text):
    text = [word for word in text.split() if word not in stopwords_set]
    return " ".join(text)
df['tweet_text'] = df['tweet_text'].apply(lambda x: remove_stopwords(x))

# Tokenize the text
df['tokenized_text'] = df['tweet_text'].apply(lambda x: x.split())

# Print the first few rows of the cleaned data
print(df.head())
```

Other steps

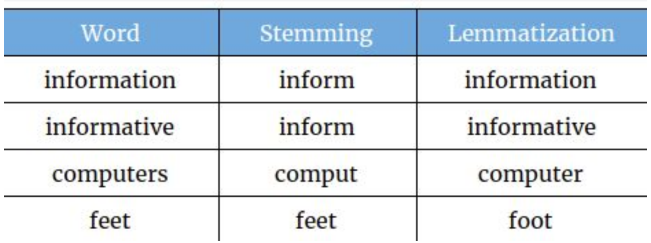
Tokenization

Tokenization helps to convert unstructured text data into structured data that can be processed and analyzed by algorithms

tweet_text	text_lower	tokenized_text	lemmatized_text
Choked on her retainers	choked on her retainers	['choked', 'retainers']	choke retainer

Lemmatization

Lemmatization is the process of transforming a word into its base or dictionary form, known as the lemma. The goal of lemmatization is to reduce inflectional or variant forms of a word to a common base form, which can help to improve the accuracy of natural language processing or machine learning algorithms.



Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

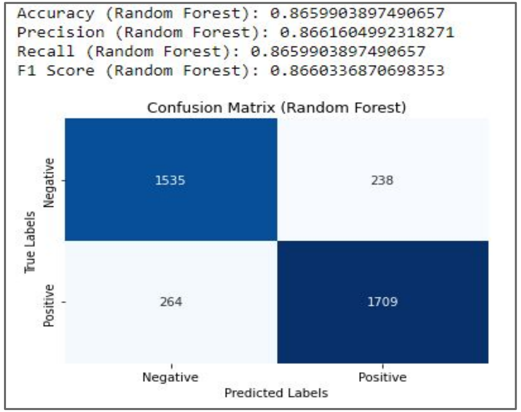
Bag of Words (BoW)

Its used in Natural Language Processing (NLP) to convert a piece of text into numerical features that can be used in machine learning algorithms. BoW representation represents the text as a bag of its words, disregarding grammar and word order, but keeping track of the frequency of each word.

CountVectorizer from Scikit-learn, which is a BoW technique that converts the text into a matrix of token counts.

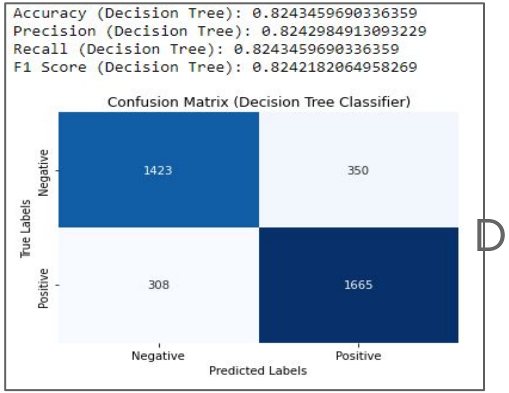
Results of different models

Random Forest



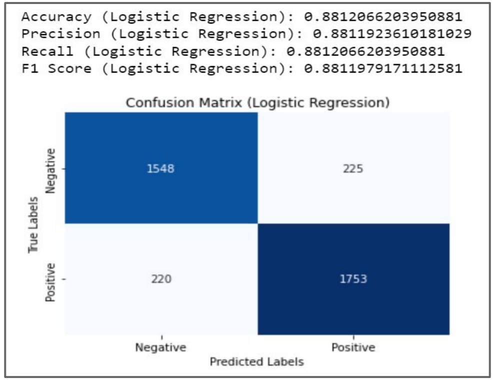
```
Accuracy (Random Forest): 0.8659903897490657
Precision (Random Forest): 0.8661604992318271
Recall (Random Forest): 0.8659903897490657
F1 Score (Random Forest): 0.8660336870698353
```

Decision Tree



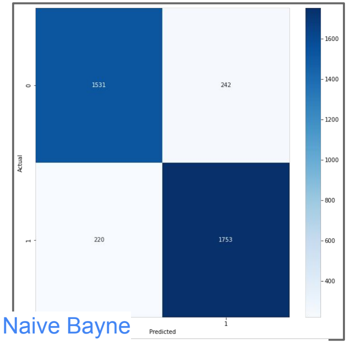
```
Accuracy (Decision Tree): 0.8243459690336359
Precision (Decision Tree): 0.8242984913093229
Recall (Decision Tree): 0.8243459690336359
F1 Score (Decision Tree): 0.8242182064958269
```

Logistics Regression



```
Accuracy (Logistic Regression): 0.8812066203950881
Precision (Logistic Regression): 0.8811923610181029
Recall (Logistic Regression): 0.8812066203950881
F1 Score (Logistic Regression): 0.8811979171112581
```

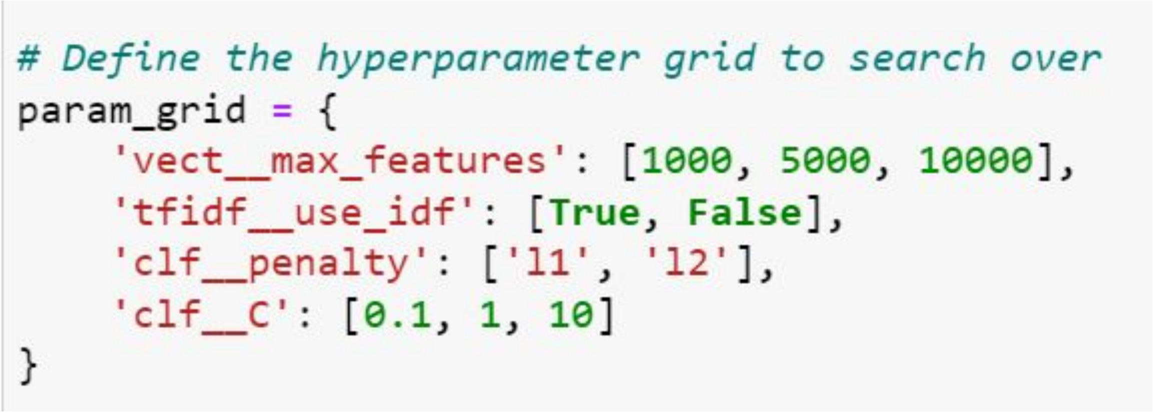
Naive Bayne



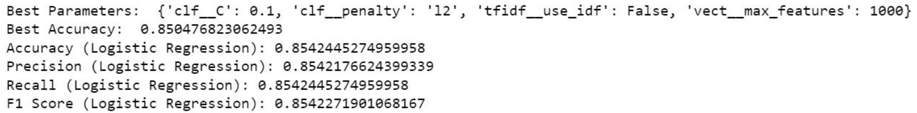
Naive Bayne
Accuracy: 0.8766684463427656
Precision: 0.8766429623329077
Recall: 0.8766684463427656
F1 Score: 0.8766253696557692

Optimizing the Model

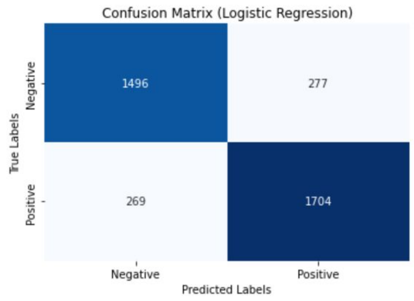
- Use techniques such as grid search or random search to optimize the hyperparameters of the best performing model.
- Evaluate the optimized model on the test set to ensure that it generalizes well to new data.



```
# Define the hyperparameter grid to search over
param_grid = {
    'vect__max_features': [1000, 5000, 10000],
    'tfidf__use_idf': [True, False],
    'clf__penalty': ['l1', 'l2'],
    'clf__C': [0.1, 1, 10]
}
```



```
Best Parameters:  {'clf__C': 0.1, 'clf__penalty': 'l2', 'tfidf__use_idf': False, 'vect__max_features': 1000}
Best Accuracy:  0.850476823062493
Accuracy (Logistic Regression): 0.8542445274959958
Precision (Logistic Regression): 0.8542176624399339
Recall (Logistic Regression): 0.8542445274959958
F1 Score (Logistic Regression): 0.8542271901068167
```

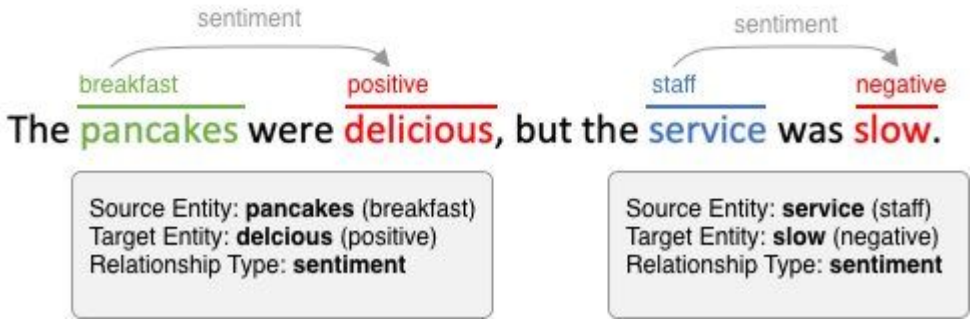


Test cases + Live Demo

Test sentence	Results
Today is sunday, I am going to have fun!	positive with probability 0.89.
I want to be outside having fun	positive with probability 0.86
I have wonderful plans for the weekend	positive with probability 0.87.
Today is monday, I have alot of work to do	negative with probability 0.66.
Today is a sad day as its the last day of the class	negative with probability 0.94.
I wish we had a garden, we don't have money to buy one	negative with probability 0.56.
I wish we had a garden, let’s go buy one now	positive with probability 0.53.

Next steps

- Develop script for Aspect / Featured based Sentiment Analysis
- Contextualise Sentiment Analysis for prototyping in different domains (eg: Mental health, Telco, Jewellery, Winery etc)



The pancakes were delicious, but the service was slow.

Source Entity: **pancakes** (breakfast)
Target Entity: **delcious** (positive)
Relationship Type: **sentiment**

Source Entity: **service** (staff)
Target Entity: **slow** (negative)
Relationship Type: **sentiment**