Introduction

In this post, I'll summarize the exploratory data analyses I performed, explain the feature engineering and reduction steps I utilized, and present my final models to classify tweets feed

Problem Statement

The goal of this project was to classify twitter feeds into breakfast time, lunch time or dinner time, using only features derived from text from ten news organizations

Risks and Assumptions

I've chosen my source based on some renown news organizations and large corpus, but I recognize that this model may not generalize to news from other sources. My results will likely also be applicable to my determined timeframe, and a time interval since I'm utilizing tweets from October 2017 onwards.

Data Acquisition

The first step of this project is to acquire live twitter feeds via twitter API and store them as they are collected and use them to build my corpus. With my Api_key, I wrote a function to collect 2,000 tweets at once. These data are stored as a csv file before processing and analysis in Pandas.

Data Transformation

After acquiring the data, I wrote a script to clean up the dirty data, get rid of emoticons, hashtags, RT's ,?, http, etc. Then wrote another function to remove all the stop words. Of over 116,000 tweets, I selected over 23,000 to solve the problem at hand

EDA: Sentiment Analysis

Sentiment analysis is a natural language processing technique that seeks to identify the polarity of written text - whether the document expresses positive, negative, or neutral feelings. This technique is commonly applied to text that may express strong opinions, such as reviews or social media posts, I'm going to apply it to twitter feeds and see if there are any differences in the sentiments expressed by different news organizations.

The goal for this analysis is to derive scores for each tweet based on the positivity, neutrality, or negativity of the words they contain, and see if these scores might be

useful features in predicting what time of the day is from. Sentiment scores range from 1 (very positive) to -1 (very negative), and I'm expecting a relatively small range for feeds in this corpus, since news is presumably objective. Stop Words is a resource that maps tens of thousands of English words to a sentiment score. Since language is complex, and the meaning of a given word can vary significantly depending on the context in which it's used, these scores are imperfect, but they'll provide a general picture of the sentiment behind each twitter feed. From the 10 news organizations, I noticed that CBS NEWS has the highest level of positive sentiment or happiness.

EDA: Identify topics in the twitter feeds

LDA (Latent Dirichlet Allocation) is an unstructured machine learning technique that iteratively attempts to find clusters of words that are likely to happen together across multiple documents. We interpret the co-occurance of these words together to be analgous to different topics discussed in across a body of documents.

LDA works by iteratively guessing how likely a given word is to be part of a given topic until we tell it to stop. I pulled 10 main topics from my feeds and the topics are easy to understand for example topic_0 talks about sexual harassment, topic_2 is about Jerusalem and national security issues, while topic_3 talks about Flynn guilty plea, etc.

Determining Class for analysis

My aim for the project is to predict a twitter feed based on a time interval of the day. Thus, I decided to pick three time intervals that I deemed useful as predictors. Breakfast time, lunchtime, and dinnertime. In my data, I noticed that people tend to tweet more during the evening so to avoid unbalanced class, I widen a little bit breakfast and lunch time

Evaluating difference between groups

I used Bayesian strategy to evaluate difference between my groups based on the polarity. I used pymc3 (Monte Carlo simulation) to plot the posterior distribution of the group means and the standard deviations. Breakfast and lunch tweets have no difference in terms of tone. Breakfast and dinner tweets are different in terms of tone. Lunch time and dinner tweets have a difference in terms of tone.

Feature Selection

Since the TfidfVectorizer yielded approximately 119,000 unigrams and bigrams, feature reduction was a necessity. Every unique word in a document can represent a new feature, so it's important to identify which words are the most informative and train a model on those, discarding the less informative words in order to reduce dimensionality and save time transforming data and

fitting models. In addition, having a set vocabulary allows a classifier to work on new tweets that may contain words it's never seen before. I also use ngram range to check the relationship between words.

Model Building: Multiclass Classification

Naive Bayes are the go-to models for text classification tasks, as they are reasonably efficient and robust to errors, especially when the sample size is relatively small. However, other viable options include random forest, adaboost, gradient boosting, and Support Vector Machines.

I tested six models in total using Scikit-Learn: MultinomialNB, a random forest classifier, Adaboost, Gradient Boosting, and Support Vector Machines. I was able to use ngram with different range to find feature importance. Most of my models have an accuracy of about 43% except the Support Vector Machine that have no meaningful signal. This model is not appropriate to my dataset. My model scores are not that high because it was difficult to separate the classes