

aarongalbraith / flatiron-phase5-project

🔍

📁

👤

<> Code

🔗 Issues

🔗 Pull requests

🎬 Actions

📁 Projects

📖 Wiki

🛡 Security

📈 Insights

⚙

👁

🔗

☆

☆ 0 stars

🔗 0 forks

👁 1 watching

📈 Activity

🌐 Public repository

🔗 main ▾

⋮

🔗 Branches

🏷 Tags

👤 aarongalbraith

last changes

...


now ⌚ 43

View code

☰ README.md

✎

Aaron Galbraith Flatiron Data Science Capstone Project

alt text

Overview

Pfizer seeks better information about the sentiment of its potential customer base toward various prescription birth control methods. We analyzed user-generated reviews of birth control drugs from drugs.com and made recommendations to Pfizer based on our findings. Pfizer can track developing trends by using our modeling tool to analyze conversations happening elsewhere online.

Business and Data Understanding

Business Understanding

Following the US Supreme Court ruling in [Dobbs \(2022\)](#), many states began changing laws regarding reproductive health care rights. In the new reproductive environment created by this ruling, Americans who are concerned with family planning are showing greater interest in birth control options and are more likely to consume and practice the birth control methods (that remain legal) in greater numbers than before. Pfizer can capitalize on this trend by understanding public perceptions of the various methods and responding to these perceptions in their marketing.

In 2018, researchers Surya Kallumadi and Felix Gräßer at UC Irvine created the [UCI ML Drug Review Dataset](#) after collecting reviews from [Drugs.com](#) that users had written about various drugs between 2008 and 2017. A substantial portion of these reviews addressed birth control and emergency contraception drugs.

People often share similar sentiments with each other in online spaces such as [Reddit](#) and [Quora](#). Our project analyzes the Drug Review Dataset in order to 1) learn what the Dataset can tell Pfizer about sentiments toward the various methods of birth control and 2) train a model that can be applied in other online spaces to determine what birth control methods users are discussing and how they feel about them. With this tool, Pfizer can more effectively market their products to the increased demand created by the Dobbs ruling.

Data Understanding

After a substantial amount of cleaning, the data set included 21,779 records. Each record (initially) had these features:

Drug Name

These labels varied greatly. Many were specific brand names, while others were generic or chemical names, or even combinations of chemical names.

Condition

This feature had many missing labels. We eventually trimmed this feature to just two labels: "Birth Control" and "Emergency Contraception". In fact there was some cross-mixing of these two conditions, i.e. records labeled "Birth Control" that actually reviewed drugs for emergency contraception purposes and vice versa.

Review

This was the text of the review that a user posted on drugs.com. There were a great deal of duplicate reviews, as explained further below.

Rating

Users submitted a rating between 1 and 10 accompanying each review.

Date

The records spanned from February 2008 to November 2017. The number of reviews surged in 2014.

Useful Count

This feature counted the number of "upvotes" recorded by other users. This did not factor into our analysis. In further inquiry, it would be wise to note that the increase in the number of records from 2014 onward likely correlates with an increase in upvotes that does not necessarily reflect *better* reviews but simply *more* of them. Any analysis of this feature should perhaps calculate upvotes as a percentage of the total upvotes during a certain timespan, such as a day or a month.

Data Preparation

Duplicates, drug names, and missing condition labels

The majority of the records were entered twice: once with a brand name in the `drugName` feature and once with a generic or chemical name. *Some* of these duplicates had *one* missing condition label. By recognizing the nature of these special pairs, we were able to restore many of the missing condition labels (by matching them with their pair-mate).

For the remaining missing condition labels, we assigned the label that most commonly corresponded with the drug name listed. For example, if a record specified a drug name of "Viagra" but had no condition label, we would assign it the condition of "Erectile Dysfunction", as that was the most common condition associated with Viagra.

Once we had successfully restored as many missing condition labels as possible, we dropped the remaining records with missing condition labels and further dropped all records with condition labels other than "Birth Control" or "Emergency Contraceptive".

There were still more duplication instances beyond the special brand/generic pairs described earlier. This involved instances of the same review (unmistakably verbatim) appearing in multiple records, sometimes on different dates, usually with differing numbers of upvotes. We assumed in these cases that the same user had posted a review multiple times. We collapsed these reviews into a single record and modified the `usefulCount` to reflect the *total* number of upvotes from all instances. In at least one case, a single representative `date` label had to be chosen arbitrarily from two options that were only one day apart.

Exploration

Modeling

There were so few tweets with negative sentiments that it caused a class imbalance issue. We decided to make a binary classifier between positive and non-positive tweets (by grouping negative and neutral tweets together as "non-positive"). Consolidating sentiments into a binary classification reduced the class imbalance problem. 67.4% of the records were labeled non-positive and 32.6% were positive.

The goal in this case was simply to predict these labels as accurately as possible overall. If we had chosen to classify all three sentiments, then it might have made more sense to choose precision or recall of positive or negative tweets, but the only metric we used was accuracy.

The only other consideration was overfitting; we discounted models whose training accuracy was significantly higher than their test accuracy, even if the test accuracy was better than that of other models.

All models involved removing a common list of stop words (as well as a list of stop words that we supplemented) and tokenizing, lemmatizing, and vectorizing the tweet text feature.

Naive Bayes (BASELINE)

The basic Naive Bayes model gave training/test accuracies of 79.4%/71.5%. This was our baseline model.

When we tuned the Naive Bayes model, we improved both scores (89.0%/72.2%). However, since it also widened the gap between training and test accuracy, we recognized this as an instance of overfitting the training data.

When we experimented with oversampling, in an attempt to address the moderate class imbalance, it improved the training accuracy but worsened the test accuracy (86.7%/68.0%). Following this result, we abandoned all attempts at oversampling.

Random Forest

The Random Forest models gave more examples of overfitting. The first result was 96.5%/73.2%, and the tuned model was 86.4%/72.5%.

Gradient Boost (FINAL)

The Gradient Boost model, in our view, gave the strongest result without obviously overfitting the training data (74.9%/72.3%).

Summary of Model Performance

We experimented with some other models as well, but none of the results were as relevant as the main three mentioned above.

Model	Training Accuracy	Test Accuracy
Logistic Regression (rough)	95.9%	62.6%
Logistic Regression (oversampled)	95.8%	61.7%
Naive Bayes (rough)	79.4%	71.5%
Naive Bayes (tuned)	89.0%	72.2%
Naive Bayes (oversampled)	86.7%	68.0%
Decision Trees (rough)	96.5%	70.5%
Decision Trees (tuned)	79.3%	71.6%
Bagged Trees	74.7%	71.7%
Random Forest (rough)	96.5%	73.2%
Random Forest (tuned)	86.4%	72.5%
Support Vector Machine	92.6%	74.3%
Adaboost	74.1%	70.6%
Gradient Boost	74.9%	72.3%
XG Boost	84.1%	73.3%

Confusion Matrix for Final Model

The normalized confusion matrix for the Gradient Boost model shows that its recall is much higher for non-positives than for positives, which is certainly a drawback of this model.

Evaluation

Most of the models betray evidence of overfitting the training data. The Bagged Trees, Adaboost, and Gradient Boost models were the only models we considered likely not to be overfit. Of these three, Gradient Boost had the best test accuracy.

Recommendations

1. Evidence suggests the pop-up store was very popular. This was an effective way to get people excited about the product at a time when they could share their excitement with others around them. This event should be repeated if possible.
2. Apple should consider addressing battery life and design issues with some of their products. These topics didn't fully dominate the discussion by any means, but they were the most significant of Apple's negative topics of any substance.
3. The party Google hosted was clearly very popular and appeared to drive a lot of what buzz they enjoyed at the festival. Apple should consider hosting parties at festivals in a similar manner.

Further Inquiry

More sophisticated modeling techniques might be able to better analyze either a direct positive v. negative comparison or even a multi-class analysis (positive, negative, and neutral). The class imbalances make this difficult.

More direct analysis could be done with the tweets that mentioned *both* Apple and Google brands. Perhaps these tweets feature direct comparisons that could be very illuminating.

With more time we would have liked to explore feature importances of the various models.

We would also like to have explored *why* the models were overfitting the training data so consistently and what aspects could have been changed to prevent this.

We would have liked to investigate other features, such as tweet length (counting both characters and words), to see if that added anything to the models.

It may also be worth rethinking our evaluation metric. It could be of more value to prioritize true positives (recall) than just focusing on accuracy.

Links to PDFs

Find the notebook [here](#)

Find the presentation [here](#)

Find the github repository [here](#)

Find reproducibility notes and instructions to run the notebook [here](#)



Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● Jupyter Notebook 100.0%