For my project, I chose to analyze data from my perspective as a food blogger who develops and posts recipes throughout the year. My goal is to create recipes that are both popular and well rated.

My primary questions to be answered are as follows:

- Can the popularity of a recipe be predicted by its ingredients?
- What factors (other than quality of the recipe) might influence the sentiment of a review?
- What type of recipes/ingredients are popular throughout the different seasons of the year?

To accomplish this task, I used the Kaggle dataset from Food.com which contains data on 10,000 recipes with 234,115 corresponding reviews, including scoring and review text. Additionally, I used the Guardian News API to extract recipe headlines throughout the 2020 year. I used Jupyter Notebook for my analysis with several libraries and packages (Appendix A).

I first performed an exploratory analysis. Please see appendix B for the most impactful charts. The primary takeaways were:

- Average review scores are high overall, but desserts in particular have a high number of 5 star reviews
- Most instructions are under 1500 characters and those above 2000 have a high rating (>4.5)
- Season doesn't seem to have a large influence on review scores

Next, I constructed a Naïve Bayes model to classify recipes as either highly reviewed (>10 reviews) or not based on the IDF of recipe ingredients. The overall accuracy was 61%; however, the model correctly classified 87% of all "not highly reviewed" recipes. A few of the most impactful ingredients were sugar, butter and flour. See Appendix C. Next, I constructed a Logistic Regression model using the recipe ingredients, feature engineered length of instructions and category of recipe. Despite the additional features, the predictive power of the model was not significantly different than the Naïve Bayes model (Appendix D).

After modeling, I performed a sentiment analysis. As expected, the reviews are overwhelmingly positive. The season the review was written does not seem to affect polarity; however, chicken has a wider distribution of polarity as compared to the other top recipe categories. This is interesting since we don't see this reflected in the numeric average score. See Appendix E.

To get a sense of the popularity of different recipes throughout the year, I used the Guardian API to obtain the headlines from articles concerning recipes for the year 2020. I created separate word clouds for each season (Appendix F). We see healthier foods and veganism popular during the spring and summer and comfort foods and sweets popular during the winter and fall.

Lastly, I performed a network analysis (Appendix G) to see if there were any recipes several high frequency reviewers chose to review. I found Rosemary Chicken, No-Bake Chocolate Oatmeal Cookies, and Leftover Potato Pancakes were all reviewed by more than one high frequency reviewer.

I faced several implementation challenges with the Guardian API. I was able to overcome the default page limit of 10 records by pulling separately for each season and setting the page limit at the max of

200. Another big challenge was figuring out how to combine the output from my tfidf transformer with other features. I successfully implemented a DataFrameMapper to accomplish this task for my logistic regression. I also learned about pipeline and feature union. Minor issues that took time to figure out included: converting an ISO formatted date, using mapping to create a season variable from a date and extracting a dictionary within a dictionary.

In conclusion, there are many insights to be gained from this analysis. While the models aren't great at predicting a popular recipe, it does well at identifying recipes that will not be popular. As a food blogger, I could use this to stay away from certain ingredients when I need to prioritize number of reviews. The sentiment of a review does seem to be influenced by the recipe type. I should expect baked goods such as desserts and breads to have a narrower range of sentiments than full meals such as chicken and one dish meals. As a dessert focused, vegetarian blogger, this is good to hear. Lastly, while seasonality didn't seem to have a big effect in my analysis there are obvious seasonal food trends in the Guardian data. I used the date a review was written as an estimator for the date someone made the recipe. It is possible this is a flawed methodology.

Data source:

https://www.kaggle.com/irkaal/foodcom-recipes-and-reviews

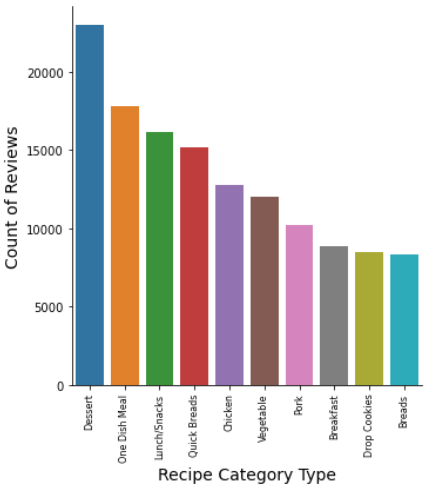
https://open-platform.theguardian.com/explore/

Appendix A: Libraries and Packages

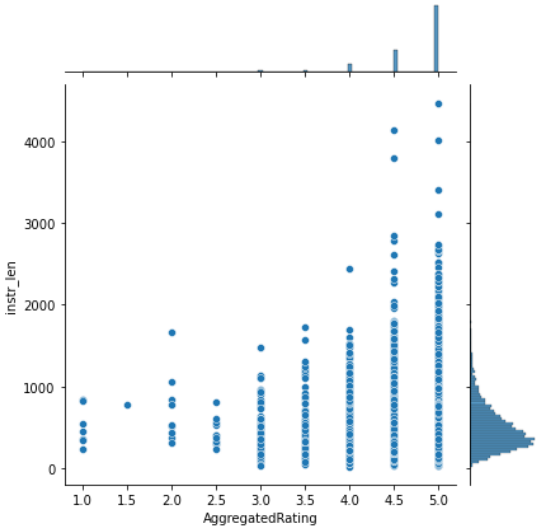
- *Data analysis:* NumPy, pandas
- *Plotting:* Matplotlib pyplot, Seaborn
- *Modeling:* sklearn (CountVectorizer, TfidfTransformer, train_test_split, accuracy_score, confusion_matrix, MultinomialNB, classification_report, DataFrameMapper, LabelBinarizer, LogisticRegression)
- *Text Analysis:* TextBlob, string, NLTK, wordcloud
- *API usage:* requests, json
- *Network Analysis:* NetworkX, pydot

Appendix B: Exploratory Graphs

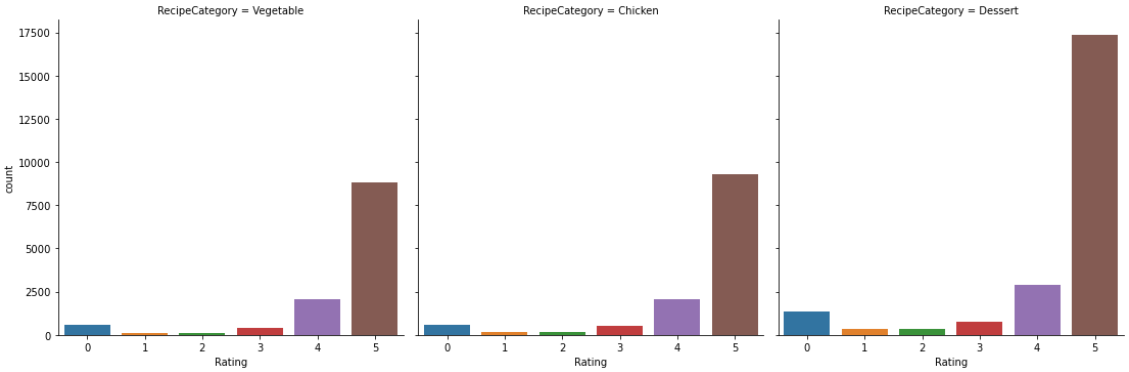
Review Counts by Top Category Types

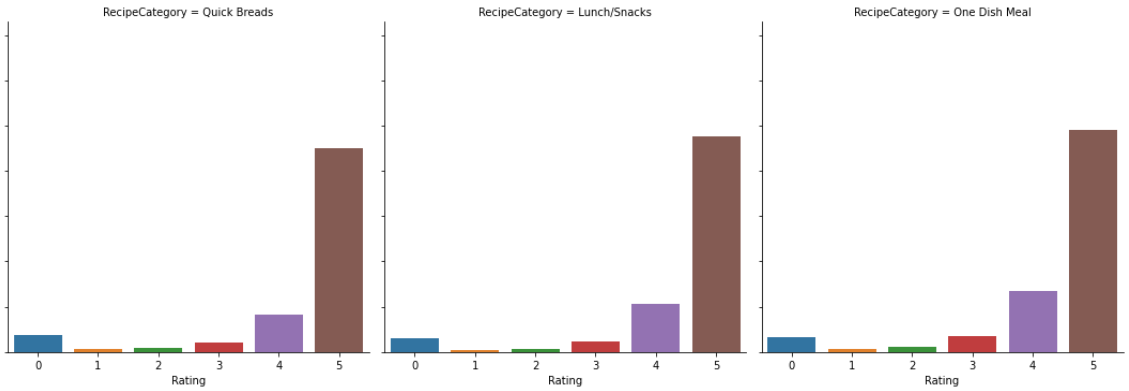


Average Rating by Instruction Length



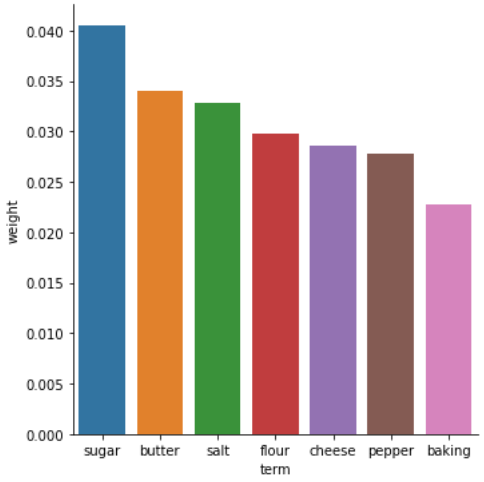
Rating distribution by Top Category Types





Appendix C: Naïve Bayes Output

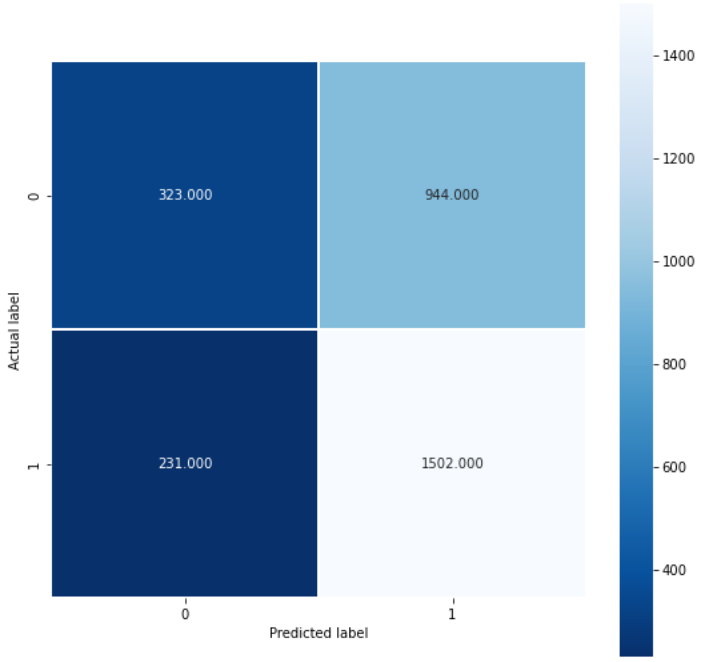
Highest IDF Weight by Ingredient



Model Output

	precision	recall	f1-score	support
High	0.58	0.25	0.35	1267
Not High	0.61	0.87	0.72	1733

accuracy			0.61	3000
macro avg	0.60	0.56	0.54	3000
weighted avg	0.60	0.61	0.57	3000

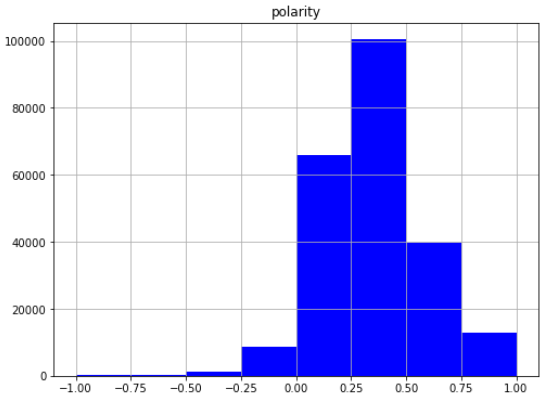


Appendix D: Logistic Regression Output

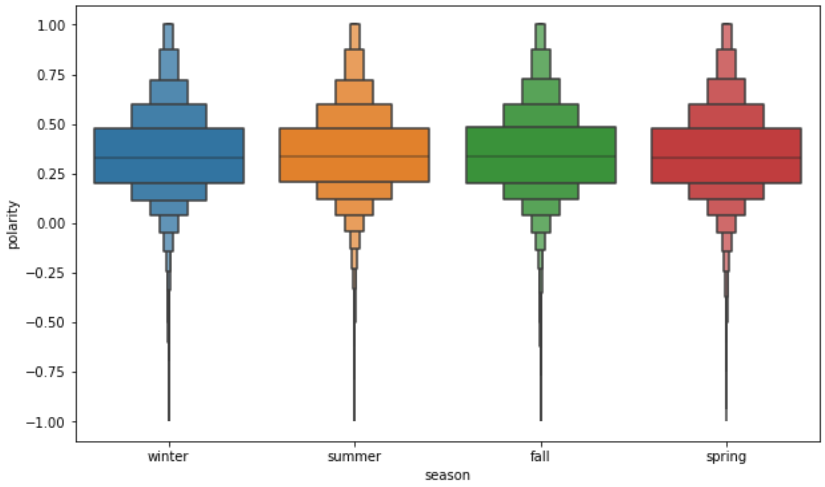
	precision	recall	f1-score	support
0	0.62	0.76	0.68	1733
1	0.53	0.37	0.44	1267
accuracy			0.59	3000
macro avg	0.57	0.56	0.56	3000
weighted avg	0.58	0.59	0.58	3000

Appendix E: Sentiment Analysis

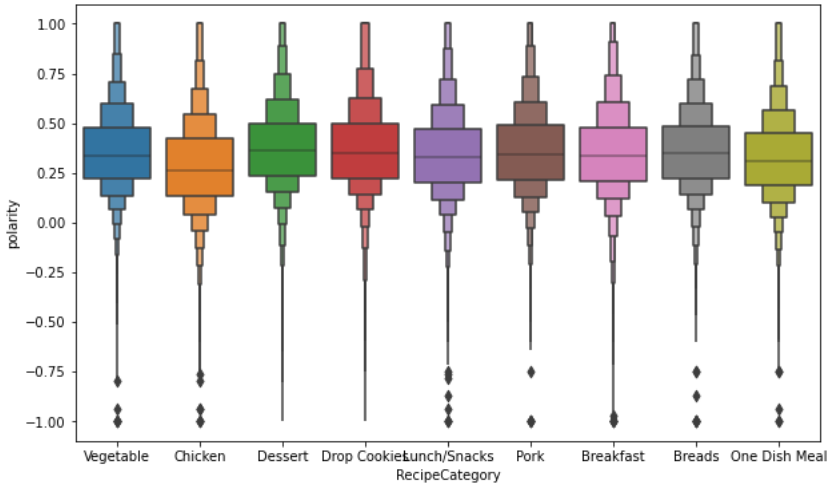
Overall Polarity



Polarity by Season

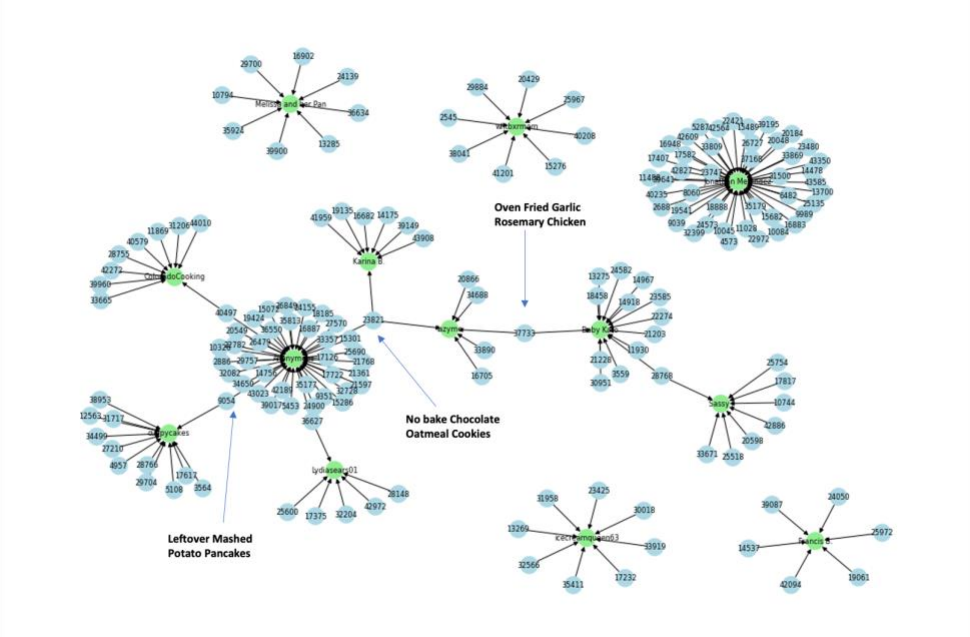


Polarity by Recipe Category



Appendix F: Word Clouds

Appendix G: Network Analysis



Citations:

Kristin's Kitchen

Network Analysis:

F., K. (2020, September 11). *Masking with WordCloud in python: 500 most frequently used words in German*. Medium. Retrieved March 10, 2022, from https://medium.com/swlh/masking-with-wordcloud-in-python-500-most-frequently-used-words-in-german-c0e865e911bb

moe_95, moe_95moe_95 36722 silver badges1313 bronze badges, Victoria StuartVictoria Stuart 3, & ilamaaailamaaa 41122 silver badges88 bronze badges. (2019, February 1). *Assign color to NetworkX node based on column name*. Stack Overflow. Retrieved March 10, 2022, from https://stackoverflow.com/questions/55342586/assign-color-to-networkx-node-based-on-column-name

samulKarlsamulKarl 5333 bronze badges, & yatuyatu 79.4k1111 gold badges5858 silver badges106106 bronze badges. (2021, February 1). *How to link two columns based on a third columns to build network*. Stack Overflow. Retrieved March 10, 2022, from https://stackoverflow.com/questions/67005486/how-to-link-two-columns-based-on-a-third-columns-to-build-network

Guardian API:

Causby, A. (2022, March 2). *Discovering powerful data: An NLP goldmine*. Medium. Retrieved March 10, 2022, from https://towardsdatascience.com/discovering-powerful-data-the-guardian-news-api-into-python-for-nlp-1829b568fb0f

ChamithChamith 7199 bronze badges, & invariantinvariant 7766 bronze badges. (2020, February 1). *Get all articles guardian API*. Stack Overflow. Retrieved March 10, 2022, from https://stackoverflow.com/questions/61031878/get-all-articles-guardian-api

Heinz, M. (2020, August 20). *Scraping news and articles from public apis with python*. Medium. Retrieved March 10, 2022, from https://towardsdatascience.com/scraping-news-and-articles-from-public-apis-with-python-be84521d85b9

Splitting dictionary into separate columns in pandas dataframe. (n.d.). Retrieved March 10, 2022, from https://www.skytowner.com/explore/splitting_dictionary_into_separate_columns_in_pandas_dataframe

The guardian API. | notebook.community. (n.d.). Retrieved March 10, 2022, from https://notebook.community/njtwomey/ADS/01_data_ingress/06_web_scraping_api

Data Mapping:

Sklearn-Pandas. PyPI. (n.d.). Retrieved March 10, 2022, from https://pypi.org/project/sklearn-pandas/1.5.0/

Sondosatwi. (2017, August 3). *Text classification and feature union with DataFrameMapper in python*. Sondos Atwi. Retrieved March 10, 2022, from https://sondosatwi.wordpress.com/2017/08/01/using-text-data-and-dataframemapper-in-python/