Inferring Restaurant Ratings: San Francisco Restaurant Reviews

Darren Lyles

Image courtesy of: http://clipart-library.com

Agenda

- Introduction
- Overview of Dataset
- Data Wrangling
- Exploratory Data Analysis
- Machine Learning
- Results and Conclusion

Image courtesy of: http://clipart-library.com

Introduction

- Who is the primary audience?
  - Restaurant owners
  - Restaurant-goers

- What questions to consider?
  - Have you ever looked up the ratings and reviews of nearby restaurants to see what's good?
    Have you ever looked at your own restaurant ratings and reviews to see what others think?

Why bother?

- Restaurant-goers want a pleasant experience
  - Restaurant-goers tend to go to restaurants with high ratings and reviews while steering away from poorly rated restaurants and poor reviews

- Restaurant owners want to attract more customers
  - Restaurant ratings and reviews dictate how well the business does
  - Restaurant owners can use this feedback to improve customer satisfaction

Overview of Dataset

- Original dataset acquired from Kaggle.com contains restaurant reviews and ratings in the city of San Francisco, California
- Originally JSON file with 147 rows and 41 columns
- Majority of columns dropped due to irrelevance in the problem and reduced down to 15 columns

Address	**Fax**	**Longitude**	**Price**	**Reviews**	**Website**
Cuisine	**Hours**	**Locality**	**Rating**	**Tel**	
Email	**Latitude**	**Name**	**Region**	**Trip_advisor_url**	

Data Wrangling

- Need to convert JSON file to CSV
- The **reviews** feature contains features itself: name, review_url, review_title, review_text, review_rating, review_date
- **reviews** is unpacked and then merged with dataset
- The resultant dataset consists of 16500 rows and 20 columns

This means we have 16500 restaurant reviews at our disposal!

Feature Engineering Using Natural Language Processing

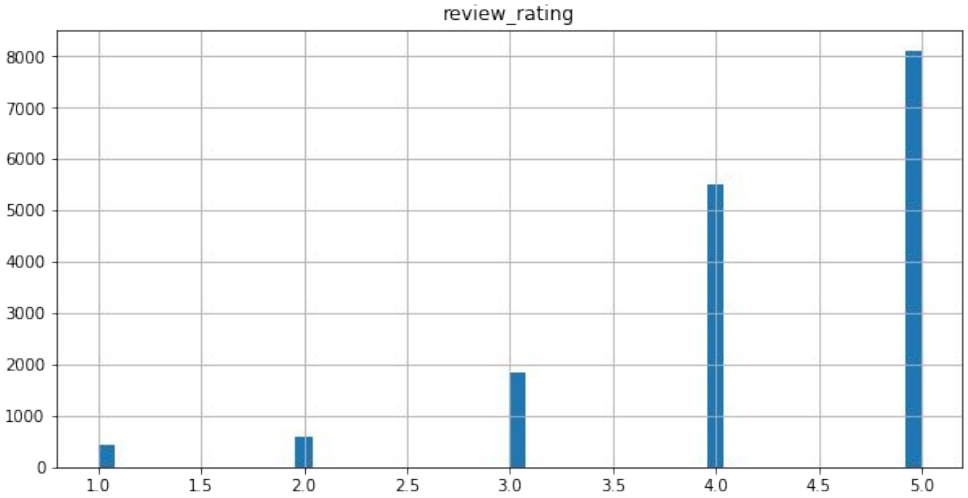
After the dataset was reformatted and cleaned up, I added an extra column which converts the raw restaurant review text into a tokenized version. The steps I used were:

1. Convert all text to lowercase
2. Tokenize all restaurant reviews
3. Filter out stopwords
4. Filter out punctuation
5. Lemmatization

The final dataset now has 16500 rows and 21 columns.

Exploratory Data Analysis

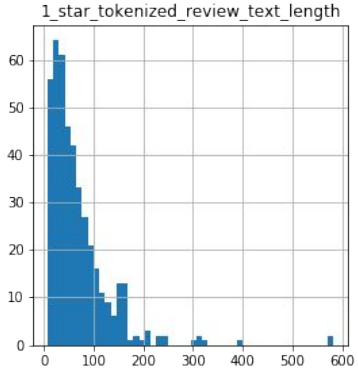
Exploratory Data Analysis

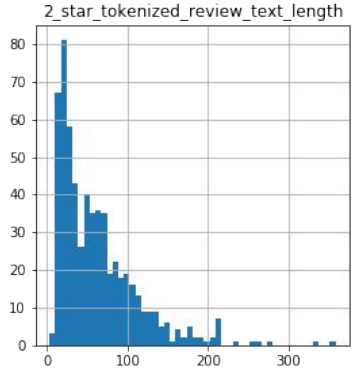


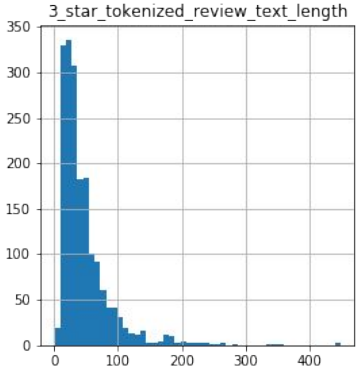
Review rating	Count
1	436
2	601
3	1833
4	5521
5	8109

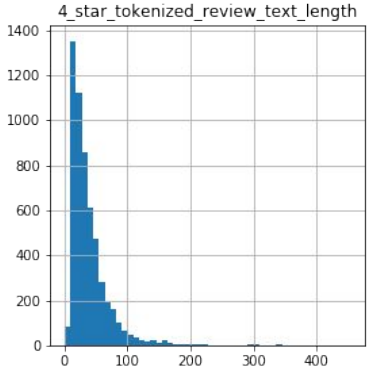
As we can see from above, we have a disproportionate amount of 5-star, 4-star, and 3-star reviews compared to 2-star and 1-star reviews. This class imbalance may cause our machine learning models to classify 5-star, 4-star and 3-star restaurant reviews with better precision. Since the 1-star and 2-star review ratings are much less in quantity, the model may have a lower precision metric, consequently lowering the overall accuracy score of the model.

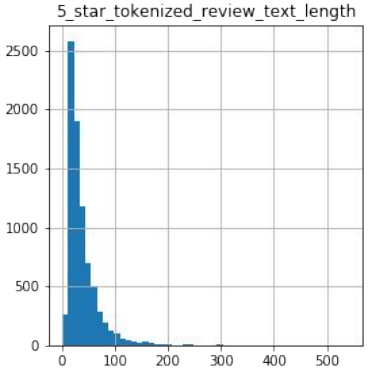
Tokenized Review Text distribution by Rating







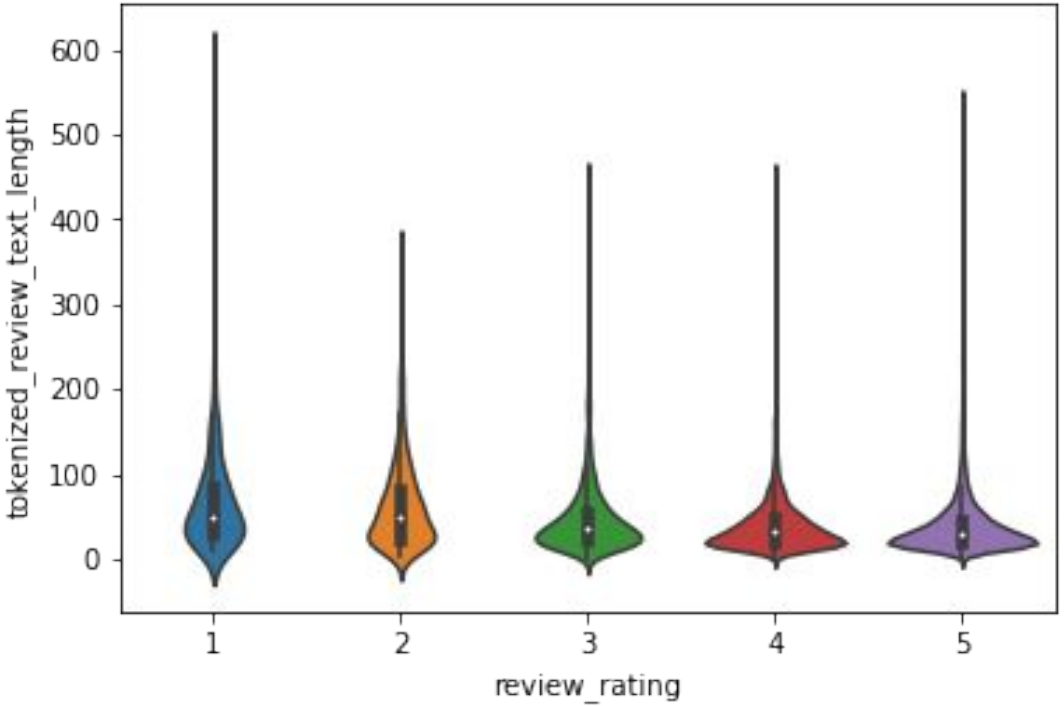




We can see that irrespective of the restaurant review ratings, the distribution of word counts are skewed to the right and the peak frequency is around 20-30 words for each review rating category.

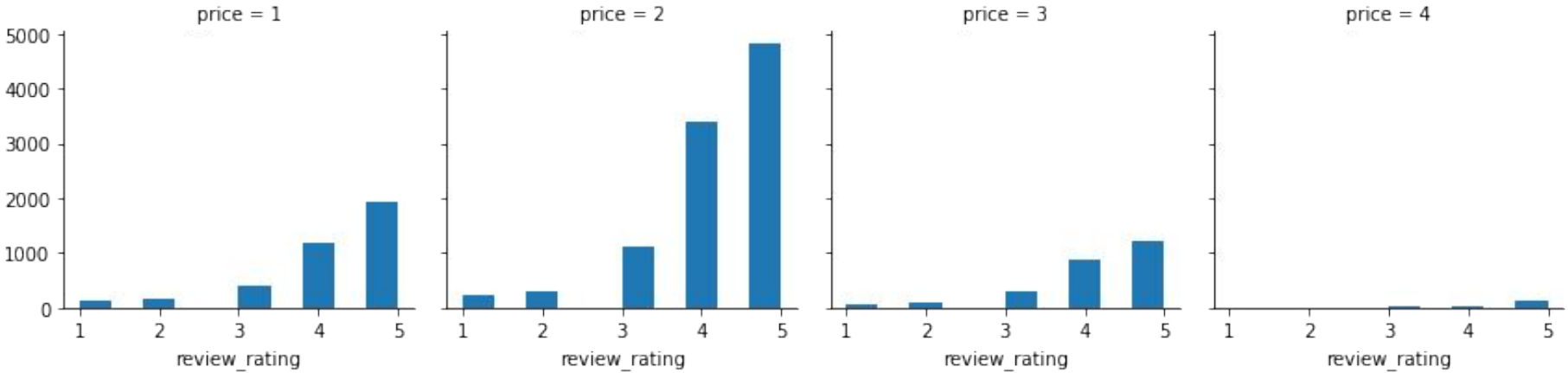
The 1-star and 2-star ratings have wider distributions, which means that some reviewers had a bit more to say about their unsatisfactory experience. Restaurant owners need not take this lightly.

Violin Plot Representation



Here, we have a violin plot showing the same information in the previous slide. As mentioned before, the 1-star and 2-star reviews have a wider distribution which is most likely due to lengthy complaints.

Rating Distribution by Restaurant Pricing



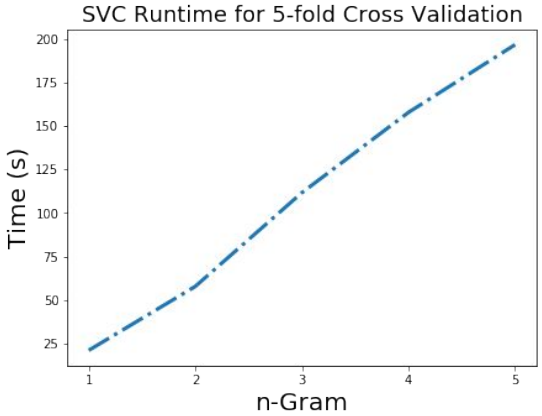
The following plots shows the number of ratings with respect to the price. It seems that restaurant reviewers in San Francisco tend to go to restaurants with two dollar signs. Not only that, but two dollar sign restaurants have the highest number of 5 star ratings than any other restaurants. This should not however, imply that the more 5 star restaurant ratings, the better. This is an example of voluntary response bias since not every person rates and reviews a restaurant they visited, and the frequency of restaurant reviewers for each restaurant is not uniform. This means the number of reviewers for each restaurant is not the same in general.

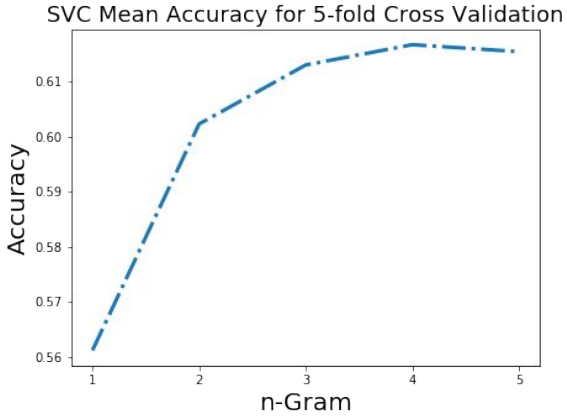
Machine Learning Analysis

Overview

- We are trying to infer the rating of a restaurant based on the content in any given review.
- This is a supervised learning problem which will involves raw text and tokenized text as input, and the output would be a rating from 1 to 5.
- The features used in this problem are **review_text** and **tokenized_review_text**
- The target is **rating**
- The models used in performance evaluation are:
    - Support Vector Classifier
    - Naive Bayes Classifier
    - Random Forest Classifier
    - Extreme Gradient Boosting Classifier
- Each model uses the TF-IDF-Vectorizer for **review_text** and the Count-Vectorizer for **tokenized_review_text**

Support Vector Classifier with CountVectorizer

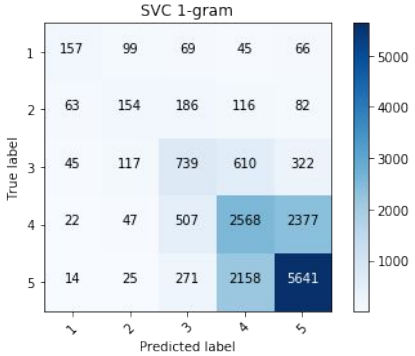


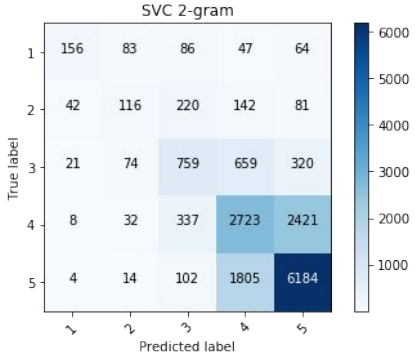


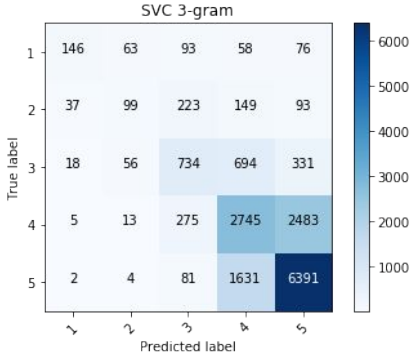
	Runtime (s)	Accuracy
1-gram	21.07	0.561149
2-gram	57.78	0.602302
3-gram	111.58	0.613030
4-gram	157.68	0.616727
5-gram	196.56	0.615456

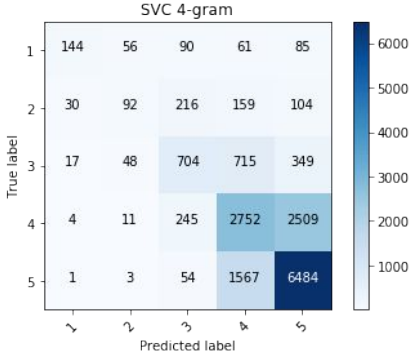
As we increase the number of grams, we see an initial significant rise in the accuracy as shown by the plot and table. However when comparing models that have 3-gram, 4-gram, and 5-gram features, the accuracy plateaus with 4-gram having the best accuracy score (61.67%). What we observe here is that the more features we add by setting the upper limit of n-grams, the slower the model takes to run. The runtime with respect to n-grams approximates linear behavior.

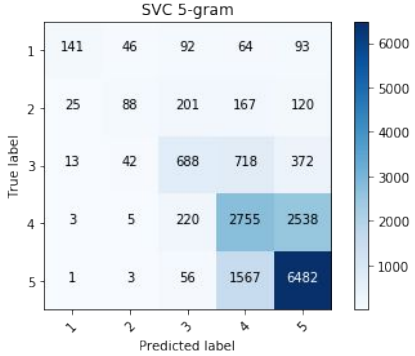
Support Vector Classifier with CountVectorizer (cont.)





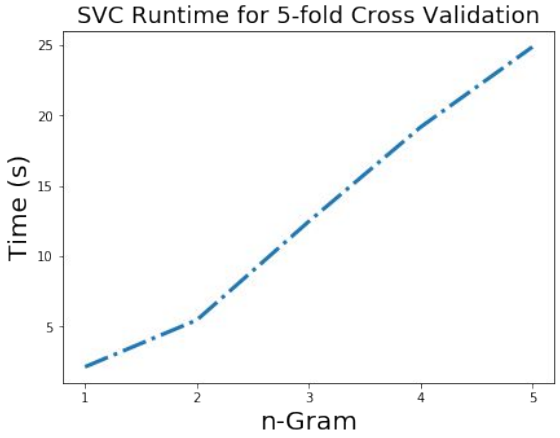


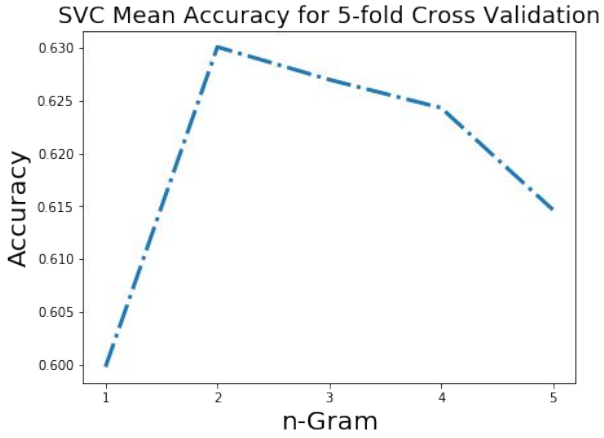




The small quantity of low ratings may well be the reason as to why our accuracy scores are relatively low. The SVC with 4-grams, which has the highest overall accuracy, also has the highest precision for 5-star ratings. This could well mean that the 5-star ratings are the dominant component in measuring the overall model accuracy.

Support Vector Classifier with TF-IDF Vectorizer

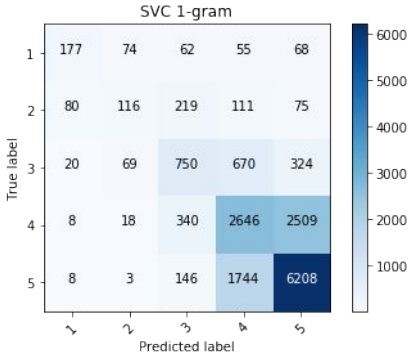


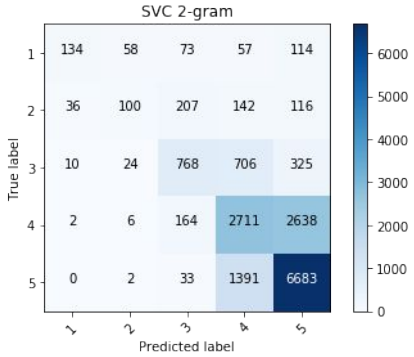


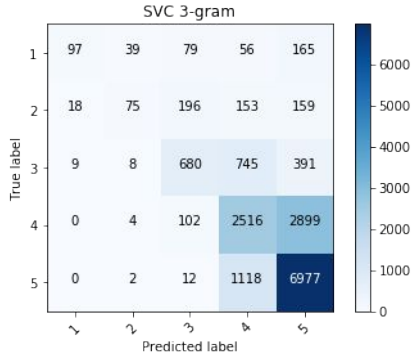
	Runtime (s)	Accuracy
1-gram	2.15	0.599814
2-gram	5.48	0.630060
3-gram	12.48	0.626970
4-gram	19.19	0.624304
5-gram	24.89	0.614668

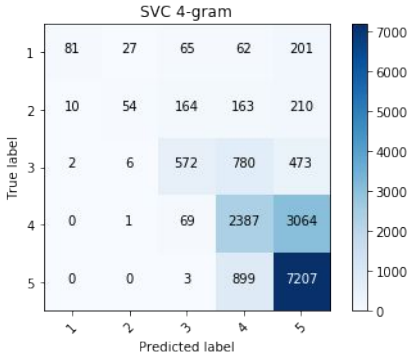
The 2-gram model in this set of tests has a average accuracy score of 63% and a runtime of about 5.5 seconds. As with the the previous set of models, it has a roughly linear runtime behavior with respect to n-grams.

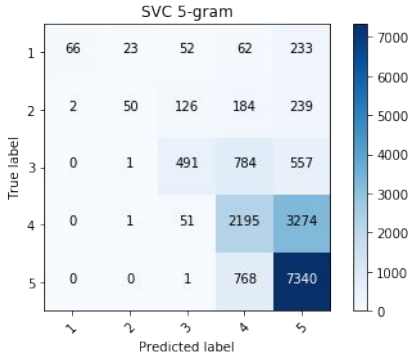
Support Vector Classifier with TF-IDF Vectorizer (cont.)





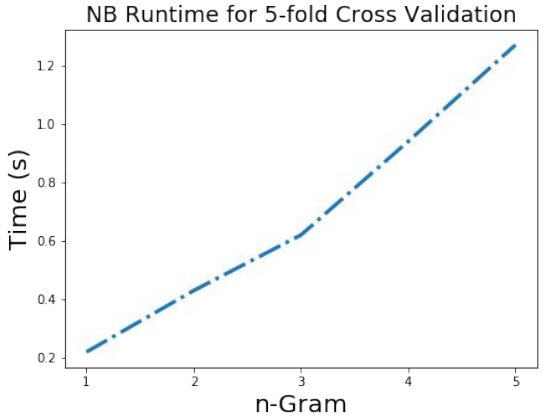


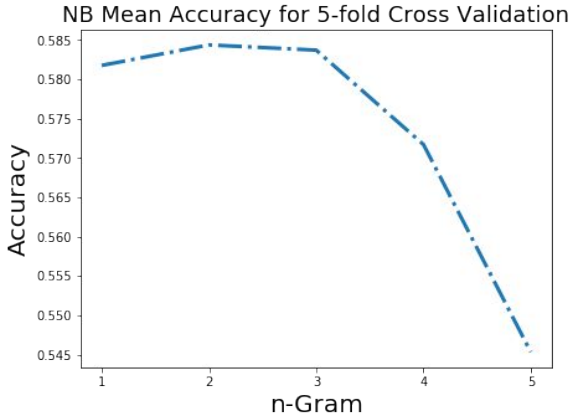




These confusion matrices provide details into the accuracy of each model using the TfIdfVectorizer, we can see that it is able to classify 5-star restaurants better than the models using the CountVectorizer, however in the case of 3-gram, 4-gram, and 5-gram features, the models are poorer classifiers for 1-star restaurants.

Naive Bayes Classifier with CountVectorizer

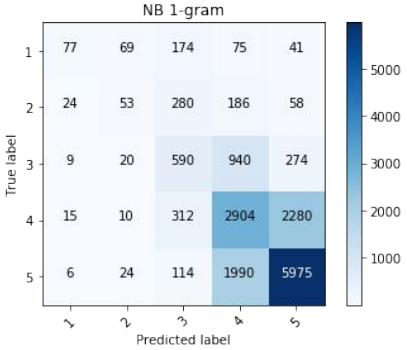


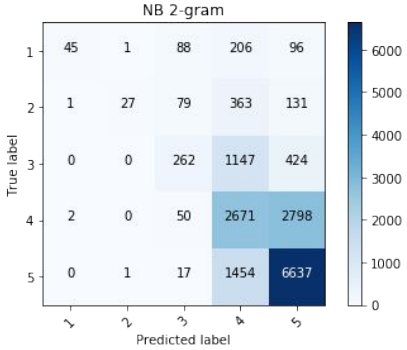


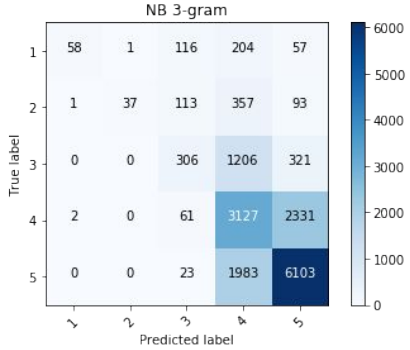
	Runtime (s)	Accuracy
1-gram	0.22	0.581759
2-gram	0.43	0.584364
3-gram	0.62	0.583698
4-gram	0.94	0.571700
5-gram	1.27	0.545338

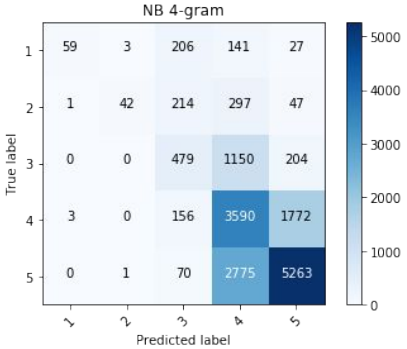
The model with the best accuracy is at 58% with a runtime of 0.43 seconds. As we increase the number of n for n-grams afterwards, the accuracy starts to lower, however since we have more input features with increasing n the runtime also increases and it shows roughly linear behavior.

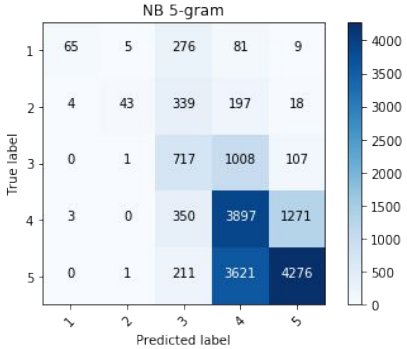
Naive Bayes Classifier with CountVectorizer (cont.)





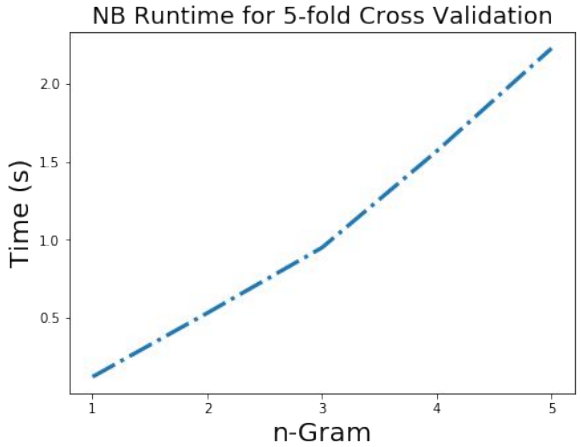


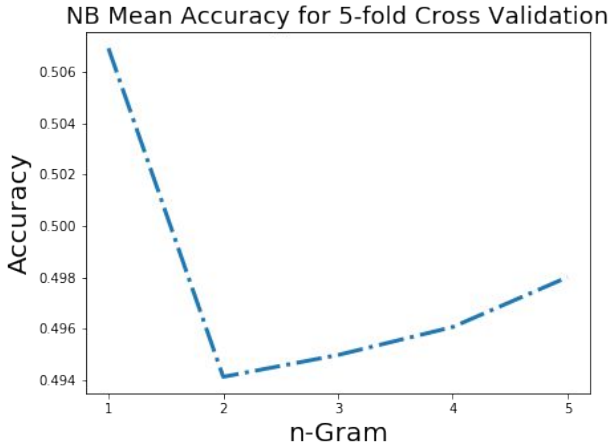




The best classifier among this group is determined by how many 5-star reviews it is able to correctly classify. Interestingly enough, it has the lowest precision amongst the other models for 1-star ratings. This classifier is great for predicting whether a restaurant review is a 5-star review, however it will have a high chance of missing a poor restaurant review by incorrectly classifying it.

Naive Bayes Classifier with TF-IDF Vectorizer

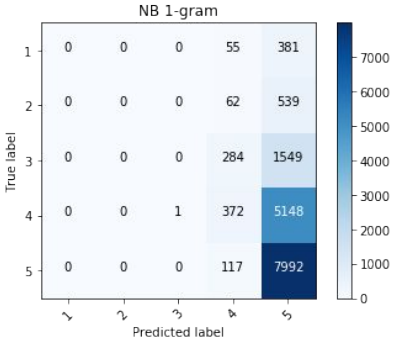


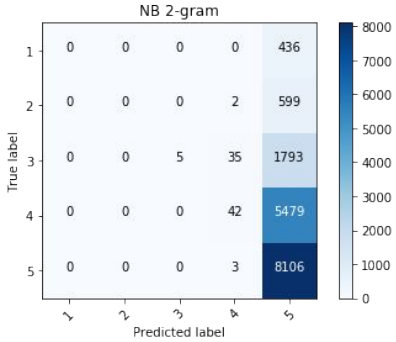


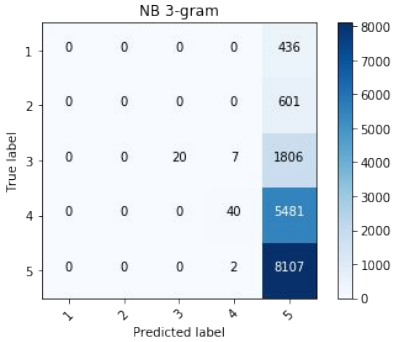
	Runtime (s)	Accuracy
1-gram	0.12	0.506910
2-gram	0.53	0.494122
3-gram	0.95	0.494970
4-gram	1.57	0.496061
5-gram	2.23	0.498000

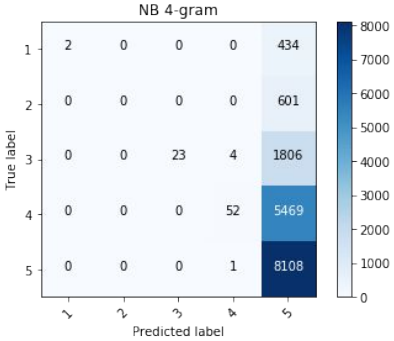
The model with the best accuracy in this set of tests is the naive bayes classifier using only unigrams. It has an accuracy of 51% and a runtime of 0.12 seconds. The runtime across all the models tested still exhibits linear behavior.

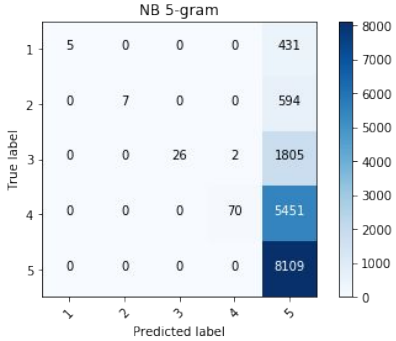
Naive Bayes Classifier with TF-IDF Vectorizer (cont.)





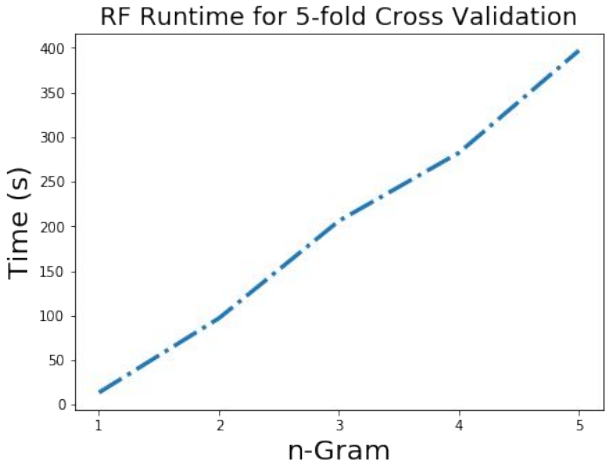


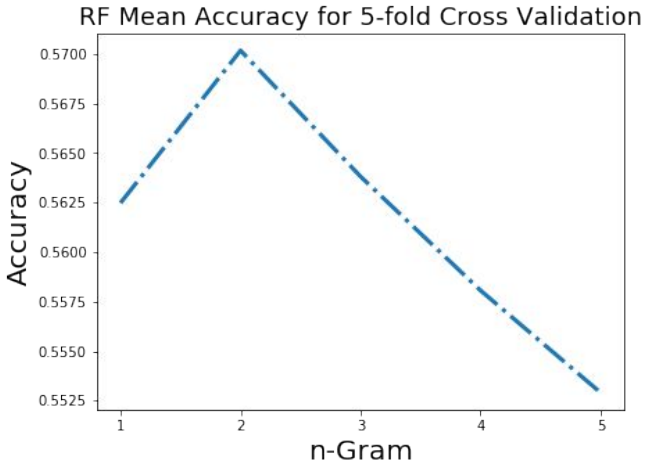




As we can see in the confusion matrices, this is an extremely poor classifier. A vast majority of the restaurant reviews are misclassified as being 5-star, whereas we can see that in the models taking 1-gram, 2-gram, and 3-gram features, they have no precision on classifying 1-star and 2-star reviews. The naive bayes classifier with 5-gram features is the only model in which has a diagonal with non-zero entries.

Random Forest Classifier with CountVectorizer

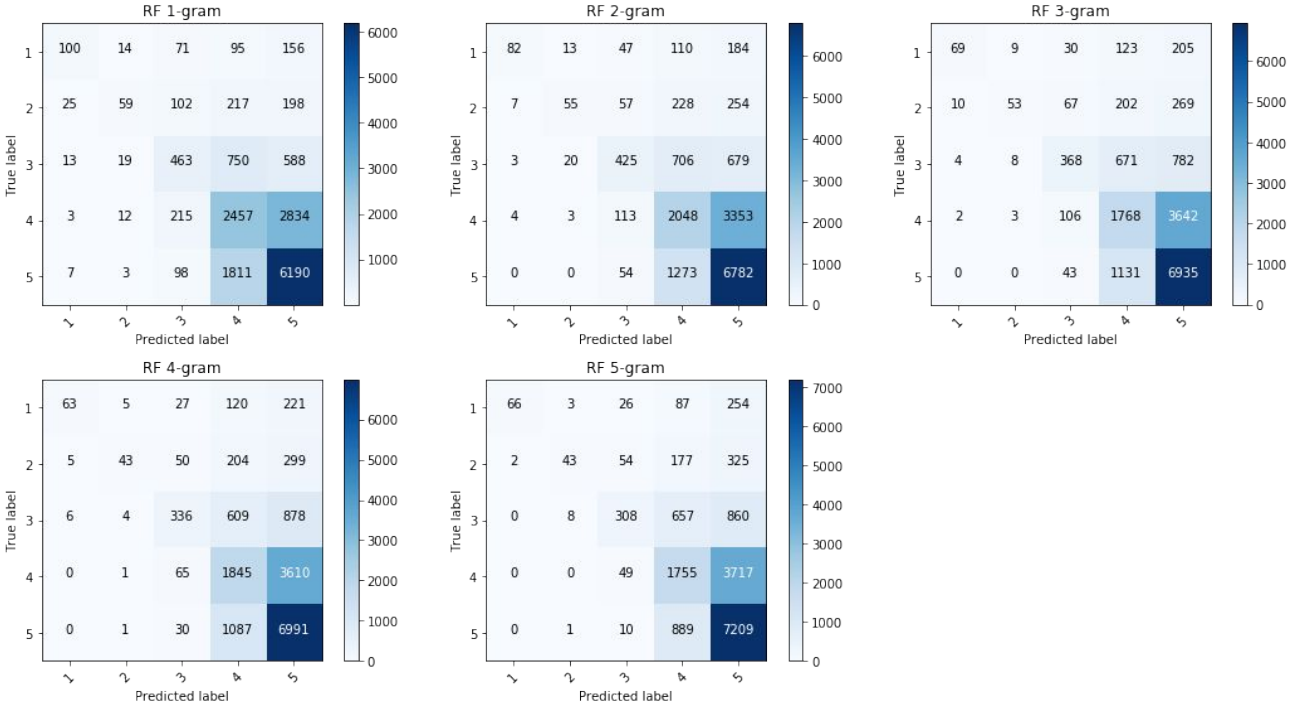




	Runtime (s)	Accuracy
1-gram	13.67	0.562485
2-gram	96.96	0.570182
3-gram	206.09	0.563819
4-gram	282.43	0.558064
5-gram	396.99	0.552908

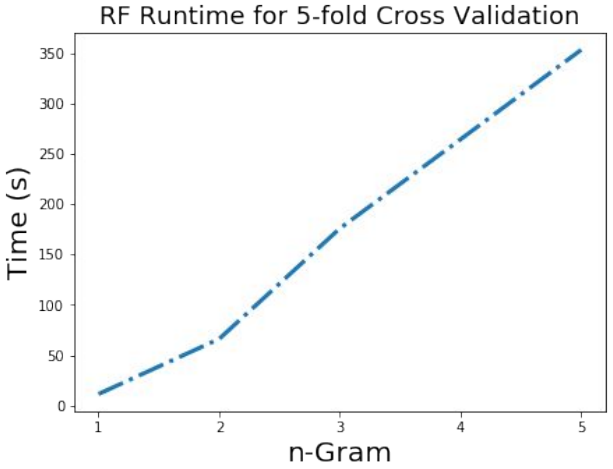
The best Random Forest Classifier using CountVectorizer has an accuracy of 57% and a runtime of 96.96 seconds. It contains bigram features. Just like in the previous models, the runtime behavior with respect to n-gram features is approximately linear. However, the more more features you include after 2-gram, the accuracy of the model decreases down to 2%.

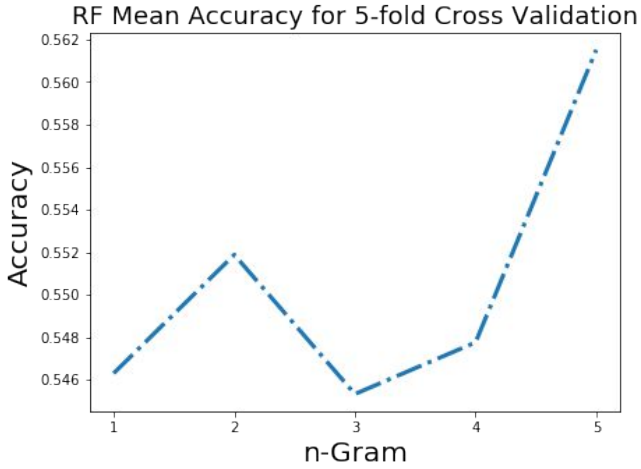
Random Forest Classifier with CountVectorizer (cont.)



The Random Forest Classifier using CountVectorizer also is a strong classifier for 5-star ratings, but very week for 1-star ratings just as the previous models. We can clearly see a pattern here where we lack strong precision in classifying low review ratings.

Random Forest Classifier with TF-IDF Vectorizer

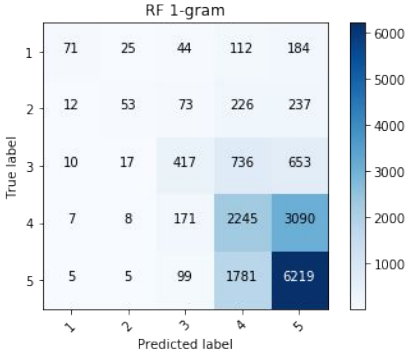


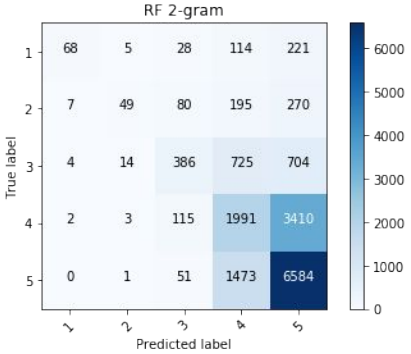


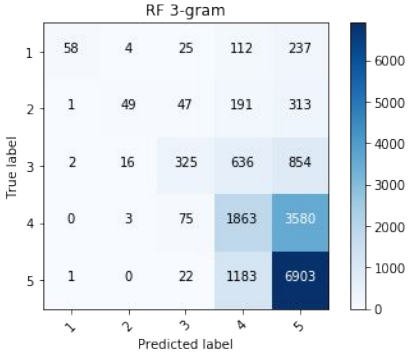
	Runtime (s)	Accuracy
1-gram	11.37	0.546306
2-gram	66.16	0.551882
3-gram	175.88	0.545334
4-gram	264.30	0.547763
5-gram	353.24	0.561514

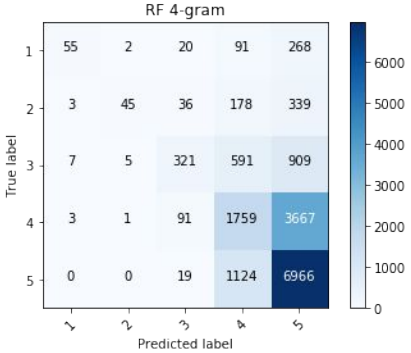
The best classifier in this group contains 5-gram features and an accuracy score of 56%. Its runtime is 353.24. Just as the other models, the runtime with respect to n-grams is roughly linear.

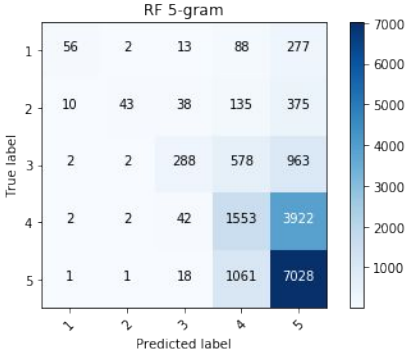
Random Forest Classifier with TF-IDF Vectorizer (cont.)





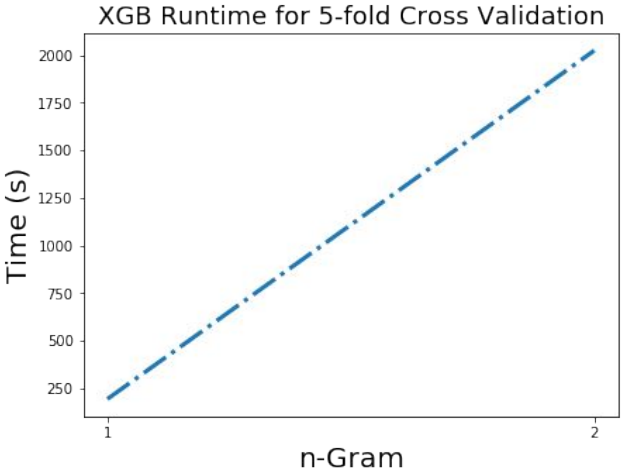


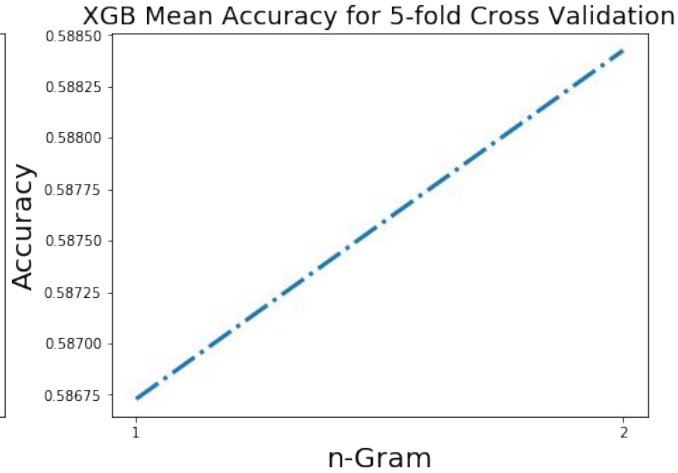




As with the Random Forest Classifier with CountVectorizer, we see roughly the same behavior. However the overall accuracy of the best model here is slightly lower by about 1%.

Extreme Gradient Boosting Classifier with CountVectorizer

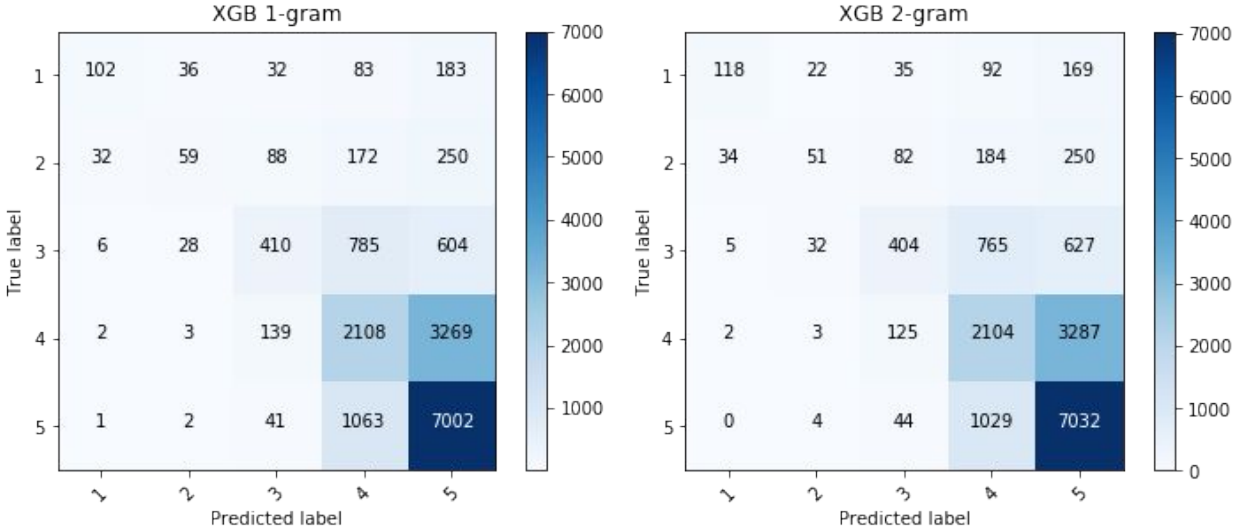




	Runtime (s)	Accuracy
1-gram	195.12	0.586729
2-gram	2023.43	0.588425

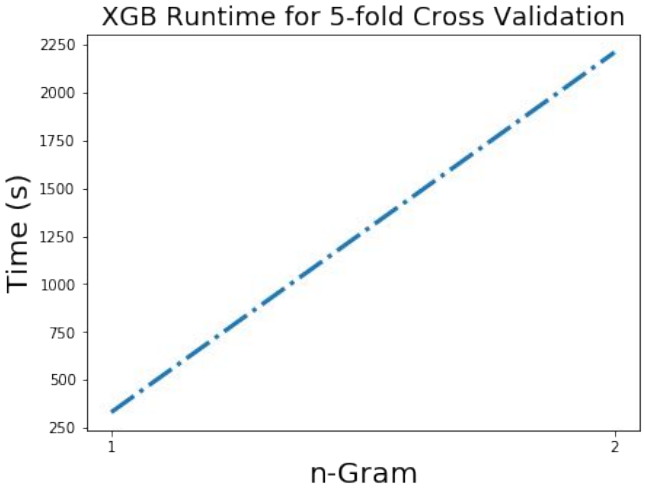
Unfortunately the XGBoost classifier has a lower accuracy than expected due to it being knows as a ground breaking machine learning algorithm. The best model, which includes bigrams, has an accuracy of 58.8% and a runtime of 2023.43 seconds.

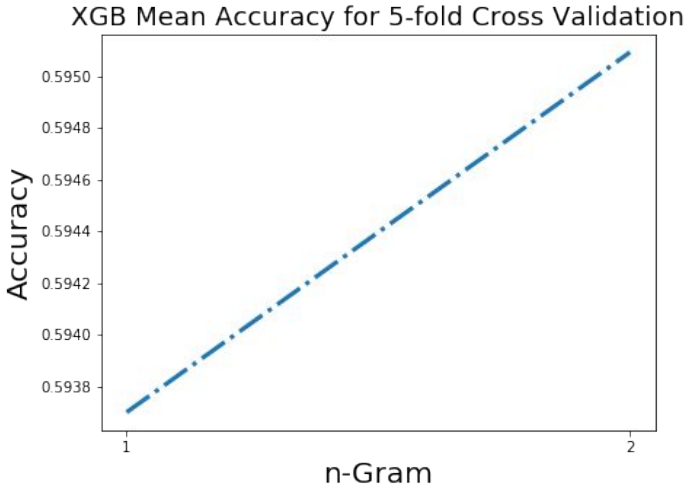
Extreme Gradient Boosting Classifier with CountVectorizer (cont.)



Like previous models, it classifies 1-star ratings poorly and 5-star ratings well.

Extreme Gradient Boosting Classifier with TF-IDF Vectorizer

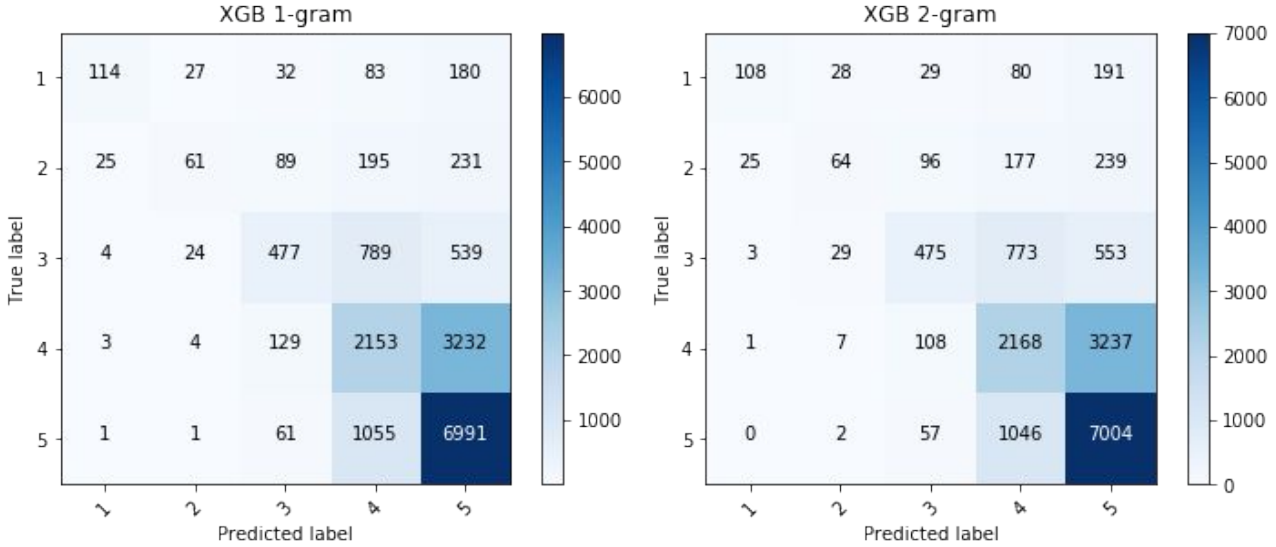




	Runtime (s)	Accuracy
1-gram	331.45	0.593699
2-gram	2208.28	0.595092

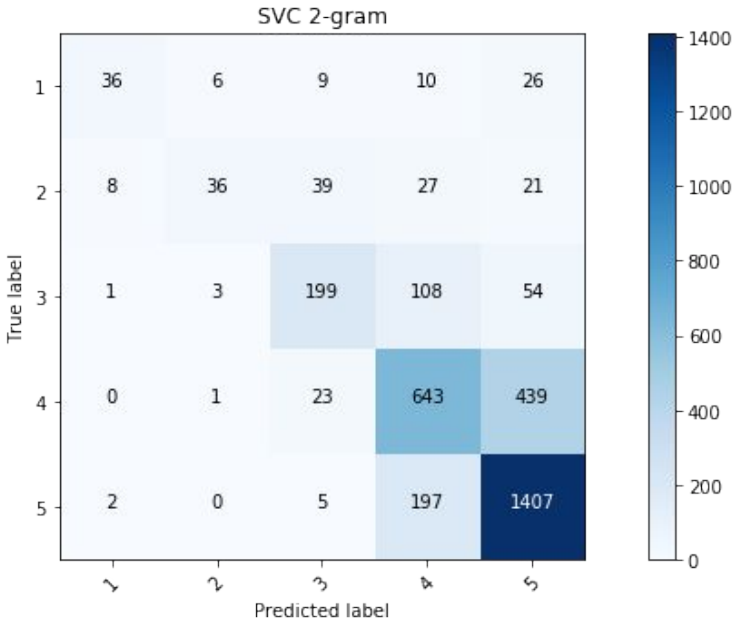
Using the TfIdfVectorizer did not make a significant difference in improving the accuracy of the model. The best model with TfIdfVectorizer has an accuracy of 59.5% and a runtime of 2208.28. Although slightly better, it has a significantly longer runtime by 200 seconds!

Extreme Gradient Boosting Classifier with TF-IDF Vectorizer (cont.)



Like previous models, it classifies 1-star ratings poorly and 5-star ratings well.

Results and Conclusion

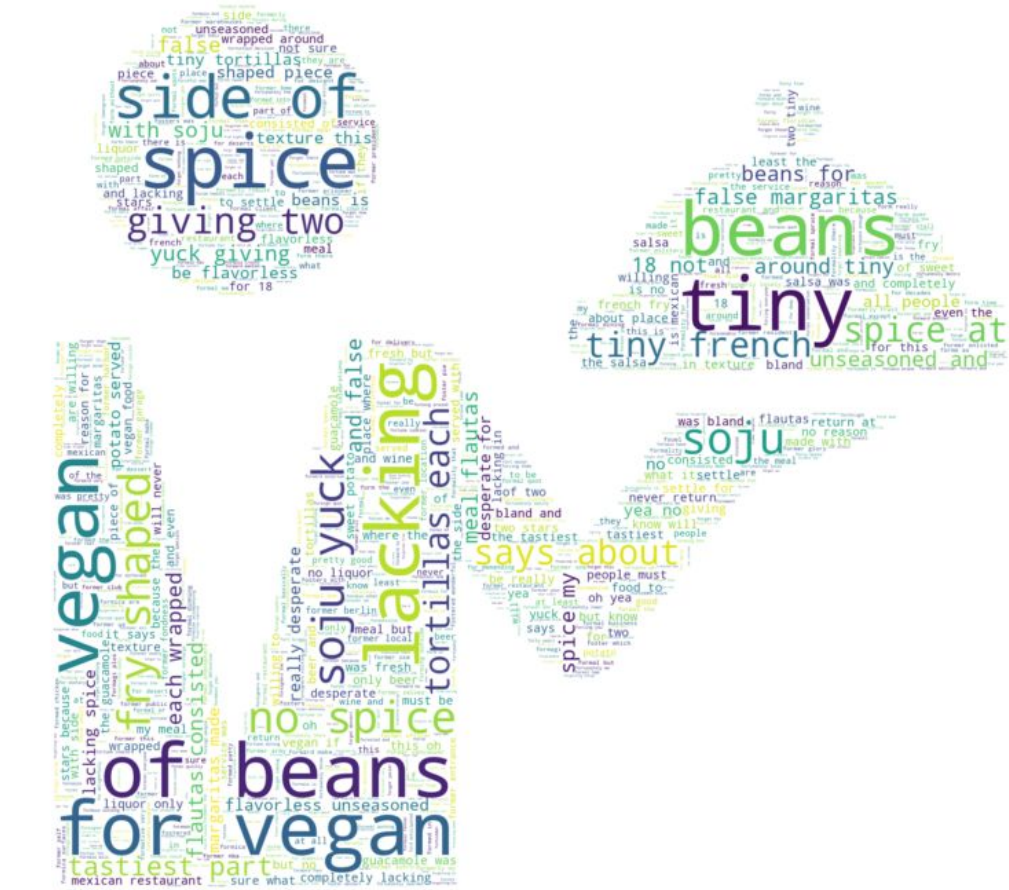


- ***Winner: Support Vector Classifier with bigram features using TF-IDF Vectorizer (accuracy: 63%, runtime: 5.48s)***
- Applying model to dataset yields accuracy of **70%** and a runtime of **1.27s**
- Due to the huge class balance between 5-star ratings vs the rest, this model will naturally classify 5-star reviews best.

Words commonly found in 5-Star Restaurant Reviews

- Some words which stand out in 5-star restaurants are: incredible, easy to, been mentioned, absolutely go, etc.
- As expected, these would be words for a positive review.
- These are typically what restaurant owners would like to see in their reviews as it helps them get more customers.
- Additionally positive reviews with such content written in them will lure in more customers, resulting in better business.
- ***The customer is happy and so is the owner!***

Words commonly found in 1-Star Restaurant Reviews



- Possibly due to class imbalance, the results of the 1-star word cloud may seem strange.
- Examples include: ***tiny, beans, vegan, soju yuck, etc.***
- Negative experience: ***lacking, no spice, never return, unseasoned, false margheritas.***
- It's obvious that we need to look into the context of the reviews to understand why certain words without a negative connotation itself are in this word cloud.
- This is important information for the restaurant owner so he/she can see where the customer is dissatisfied and take appropriate action. Unfortunately for the restaurant, such reviews can deter potential customers from coming, so taking action is a must!

Thank You

Image courtesy of: https://unixtitan.net