WHAT ARE USERS EATING?

A study of the most popular food.com recipes

Capstone 2 Final Report

Jessica Williams

OBJECTIVE

Food.com is looking to have more diverse cuisine types within the recipes that they offer on their site. They have started to get an overwhelming majority of recipes for a few select cuisine types because these recipes have had the highest user interactions thus far. But what are the more specific factors that are giving these recipes high interaction? Is it only the cuisine type or could it be something else? The goal for this research is to find a significant correlation between the recipe attributes and user interaction and ratings. Finding the factors that contribute to user satisfaction with recipes outside of cuisine type can help us find recipes that have these specific attributes within different types of cuisine.

DATA

To examine this question, we will use a dataset that consists of food.com recipes.

This dataset contains 5 different dataframes with information as follows:

- ingredients map(pickle file)-The ingredients file contains all of the ingredients listed in the recipes. It categorizes the specific ingredients into broader categories (i.e. bib lettuce, romaine lettuce, butter lettuce all become lettuce).
- raw_recipes(csv file)- The raw_recipes file contains the raw recipe information for all of the recipes.
- PP_recipes(csv file)- The coded recipes file contains number tokens or id numbers for most of the columns in the raw recipes.
- raw_interactions(csv file)- The raw_interactions file contains different elements of the user interactions for the recipes.
- -PP_users(csv file)- The user_profile file contains a more detailed profile of the users that rated the recipes.

Food Kaggle Data: (Full Data Set)

METHOD

After examining the 5 different data sets (ingr_map, raw_recipes, PP_recipes, raw_interactions, PP_users), I determined that the raw recipes and raw interactions files contained the most useful data. My aim was to find recipe features that had a strong effect on the ratings of the recipes. With so much text and numeric data pertaining to the attributes of the recipes I first needed to pull more generalized features and components from the data through descriptive statistics and text feature analysis. From there I looked at how the features correlated with two different target variables for the recipes which were the mean rating of the recipe and the average polarity score of the reviews for the recipe. From here I tested a few models to see which had the best predictive power for the target variables.

DATA WRANGLING

After narrowing my focus on the raw_recipes and raw interactions data frame I took a few steps to improve the structure of the data.

- **General cleaning**

After examining the data types for the raw_recipes dataframe I notice a few variable data types that needed to be updated. I converted the minutes, number of steps and number of ingredients columns from integer to float data types. I also noticed that the submitted column, which lists the dates that the recipes were added to the site, was an object variable. Since this column contains dates, I converted this into a datetime variable. Next I searched for null values in this dataframe which turned up a tremendous amount of nulls in the description column and 1 null in the name column. The description column contains an overall description or summary about the recipe given by the recipe contributor. Because I anticipated most likely dropping this column in the long run I didn’t think these null values needed to be dealt with right away. For the name column I decided to leave this null value as is because the id# f will be used to identify the recipe throughout data analysis. Next I checked the id variable for duplicates to insure that there were no racipes listed more than once in the dataframe. All of the recipe ids were unique.

For the raw_interactions dataframe, the first thing I did was update a few column data types as well. I converted the rating column from an integer to a float and the date column from a string to a datetime column. I examined this dataframe for nulls as well and there were no null values in the columns.

- **Creating text features**

There was a lot of text data in the raw recipes dataframe in the ingredients, steps, and tag columns. I needed to pull some of the most commonly seen text in these categories to get a base for creating attribute features. I ran a count vectorizer on the steps description column, the tag description column, and the ingredients list column. In this stage I stuck to pulling only the single ngrams from the columns at first. From this I was able to create feature matrices for the top 50 word tokens in each of these categories that I applied to my final dataframe. Some of the top tokens for steps were ‘add, ‘heat’, ‘stir’, and ‘cook’. For the tag variable some of the most popular tokens for the tag variable were ‘low’, ingredient’ ‘minutes’, and dietary. For the ingredients variable, the popular tags were ‘salt’, ‘pepper’, ‘sugar’ and oil. As we can see most of the frequent word tokens were very common language used in all recipes.

- **Merging effectively**

The two main data sets I needed to combine were the raw recipes and raw interactions data frames. Because the raw interactions data frame contains multiple reviews and rating for all of the recipes, I wanted to create some columns that would summarize the ratings and reviews and make it easier to combine the two data frames. I created a few calculated columns:

- ‘rating_mean’ column that averaged all of the ratings for each recipe.

-‘review_count’ variable to sum the amount of review listed for each variable.

-'avg_review_len' column to aggregate the mean length of recipe reviews for each recipe.

- ‘polarity_avg’ column that averaged all of the polarity scores for each review of a given recipe.

I built these columns as a start to aggregating and summarizing user engagement and satisfaction with the recipes.

I also needed to make sure that the column for the id# is named the same across all dataframes. The two dataframes have different column names for the unique id# for each individual recipe. I think it makes the most sense to use the column name ‘recipe_id’, so I renamed the column in the raw recipes dataframe ‘recipe_id’.

From here I merged the two dataframes keeping all of the columns from the raw_recipes dataframe and adding the 'recipe_id', 'rating_mean', 'review_count','avg_review_len','polarity_avg' columns to the dataframe. I also merged all of our frequent word token features to the dataframe and came up with a final dataframe called ‘recipe_attributes’.

EXPLORATORY DATA ANALYSIS (EDA)

- **Correlations?**

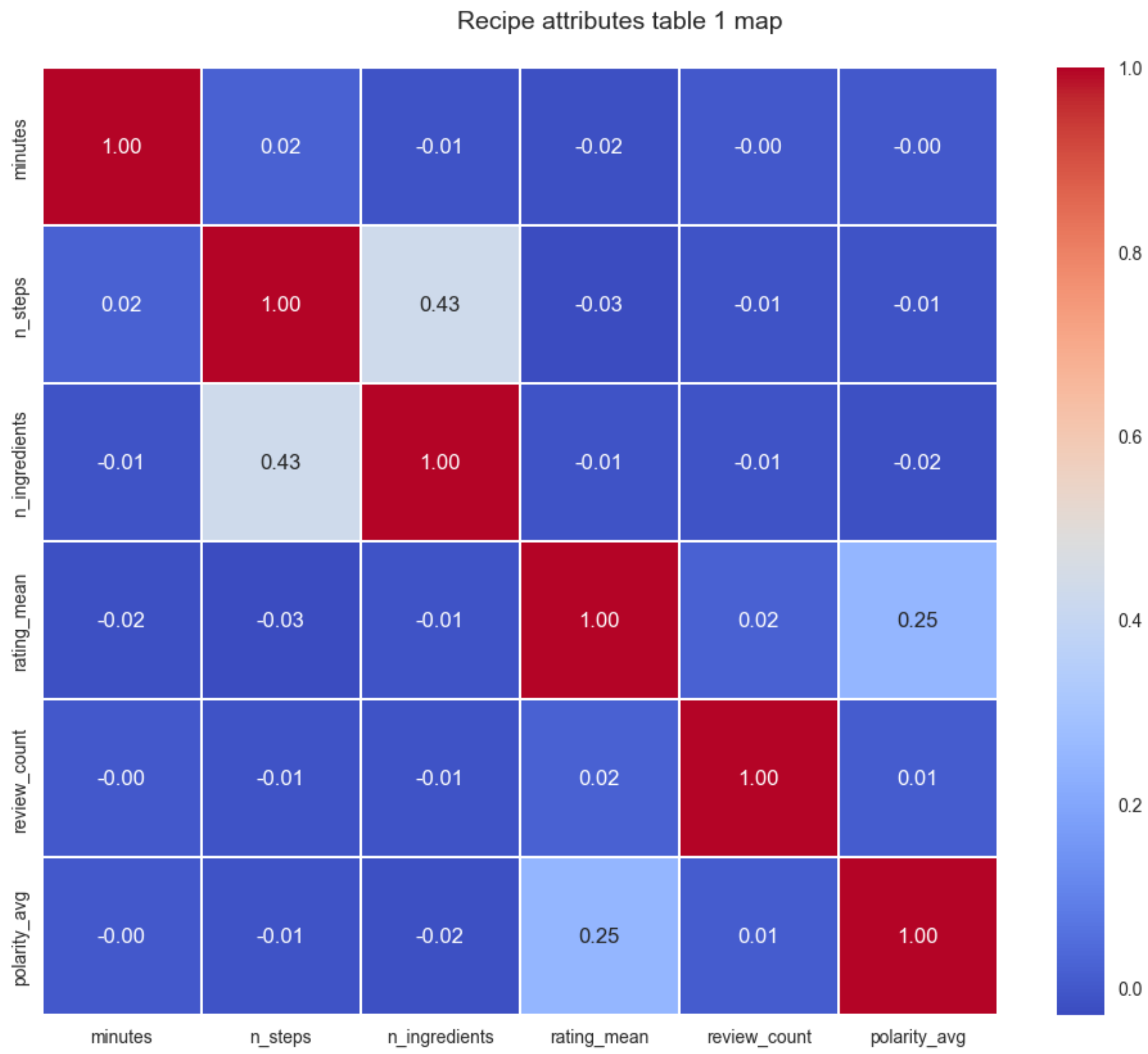
After running some general descriptive statistics on the data I noticed that most of the features have data with a wider spread. The mean sits far away from the max and min values and the standard deviations are fairly large. Next I searched for any significant correlations between the numeric attribute variables (number of ingredients, number of steps, and number of minutes) and the possible target variables(rating_mean and polarity_avg). From this I noticed a few things:

- Neither the rating mean or polarity average variable presents any clear relationships to these features. As a non-precise generalization we could say that a high concentration of the higher review polarity scores are for recipes with smaller numbers of steps.

- Looking at the plots for each feature we can see that there seems to be a slight inverse relationship between the rating mean and the both the steps and ingredients features. As far as the minutes feature, it doesn’t seem to have a much of a correlation with the rating mean at all.

Because we are dealing with a very large dataframe and I need to closely examine the relationships between individual variables, I decided to split the original dataframe and view smaller portions as I continued my analysis. I split the numeric features (number of steps, number of ingredients, minutes) into one dataframe with the target variables and all of the frequent text data into another dataframe along with the target variables. We can call them numbers dataframe and text dataframe from here on out.

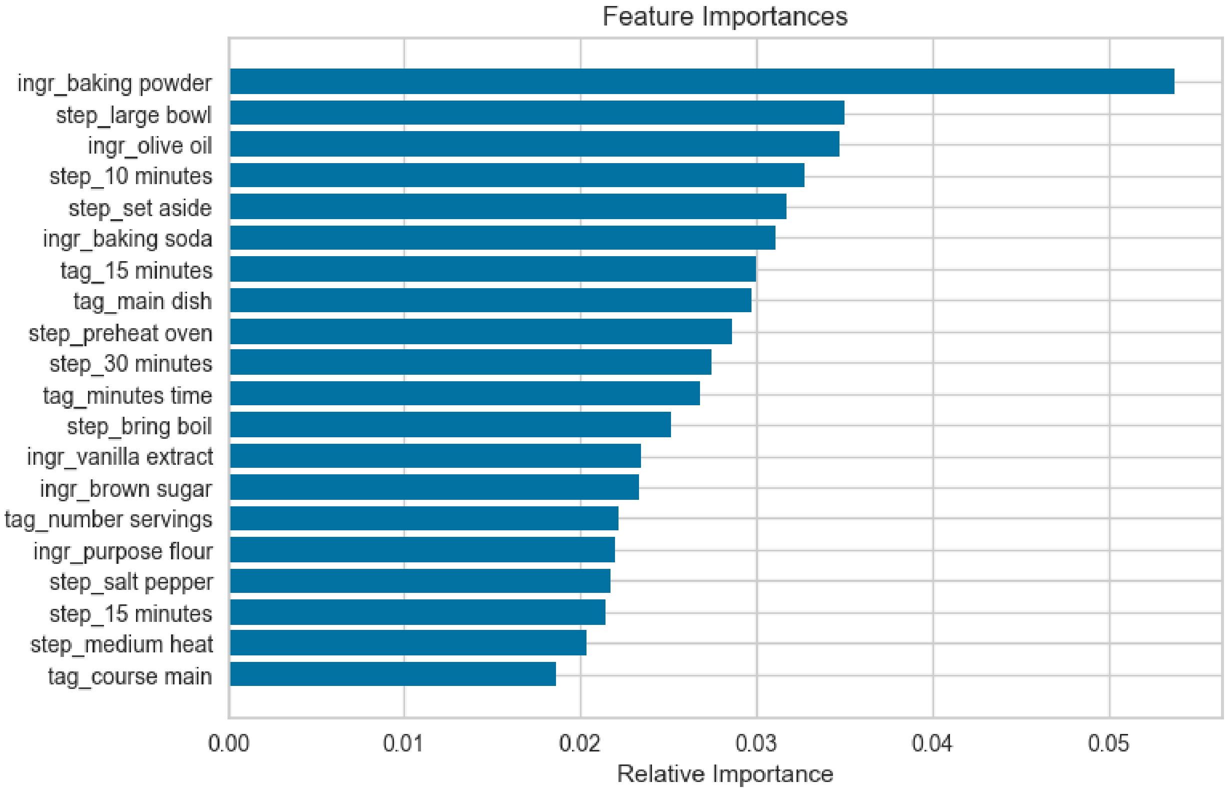
I created a correlation heat map for the numeric dataframe. The map showed no significant correlations not only between the features and the target variables but also the features with each other. I also examined the variability of these variables and found that the number of ingredients variable has the lowest variability.


Recipe attributes table 1 map

- **Dimensionality Reduction**

I moved on to our text dataframe. I decided to make some dimensionality reduction attempts with our word feature variables just to see what shows up in the top features. I applied the random forest model to that portion of the datframe. As a general hypothesis I would expect that the word category with the most feature importance would be ingredients.

After looking at this feature analysis of the word features I found that the category most highly represented in the top 30 features is the steps category. It is representative of the top 5 features and 22 of the total top 30 features. The ingredient features are only representative of 7 out of 30 features and the tags 1 out of 30. I followed up with a correlation matrix to check for any word features that correlate with the rating mean or polarity average. I found no correlations higher than 0.005.



Since I didn’t find any features that seemed to have any significant predictive power on my target variables, I tried extracting text features with longer ngram ranges from the data. At first glance this ngram range for word features in these columns gave clearer more descriptive attributes for what the words are referencing. We got tokens like ‘course main’ and ‘baking soda’. After getting the feature importances for these text features I noticed that once again step features have the most representation with 9 of 20. The steps category was followed by ingredients with 6 of 20 and tags with 5 of 20. But after examining a correlation matrix for the new text features there were no significant correlations between any of the text features and the target variables.

Finally I made some further dimensionality reduction attempts by looking at the correlation among the text features. I noticed that quite a few features in the tag category had very high correlations with each other and seemed to be representative of the same item or idea (i.e. ‘low sodium' and 'sodium low'). This is reflective in the fact that there as many tag features listed in the top important features. I decided to drop the features with correlations above 0.75.

- **EDA Findings**

After exploring this data further there aren't many clear correlations between the features and the target variables. A few observations to note:

-The ingredient variables don't seem to carry as much weight as I thought they would. This could be a good thing when trying to find factors that influence user interaction outside of cuisine type since ingredients closely reflect this attribute.

-Even though there are no significant correlations, the step features seem to stay at the head of the pack in terms of most important features and features connected to user interactions.

-It is possible that further feature reduction may help bring out more important features and higher correlations with the target variable of the rating mean.

-After exploring the review polarity as a target variable there doesn't seem to be much of a difference between its correlations with the features and the rating means correlations.

MODELING

- **Model choice**

Since this project aimed to find features that had predictive power for the target variable, I tested a few different regression models on the data. I tested 3 different regression models and the results were as follows:

First was a linear regression model:

Our R^2 for the train and test sets are 0.069 and 0.063.

Our mean absolute error for the train and test sets are 0.646 and 0.642.

Next was the Random Forest Regression model:

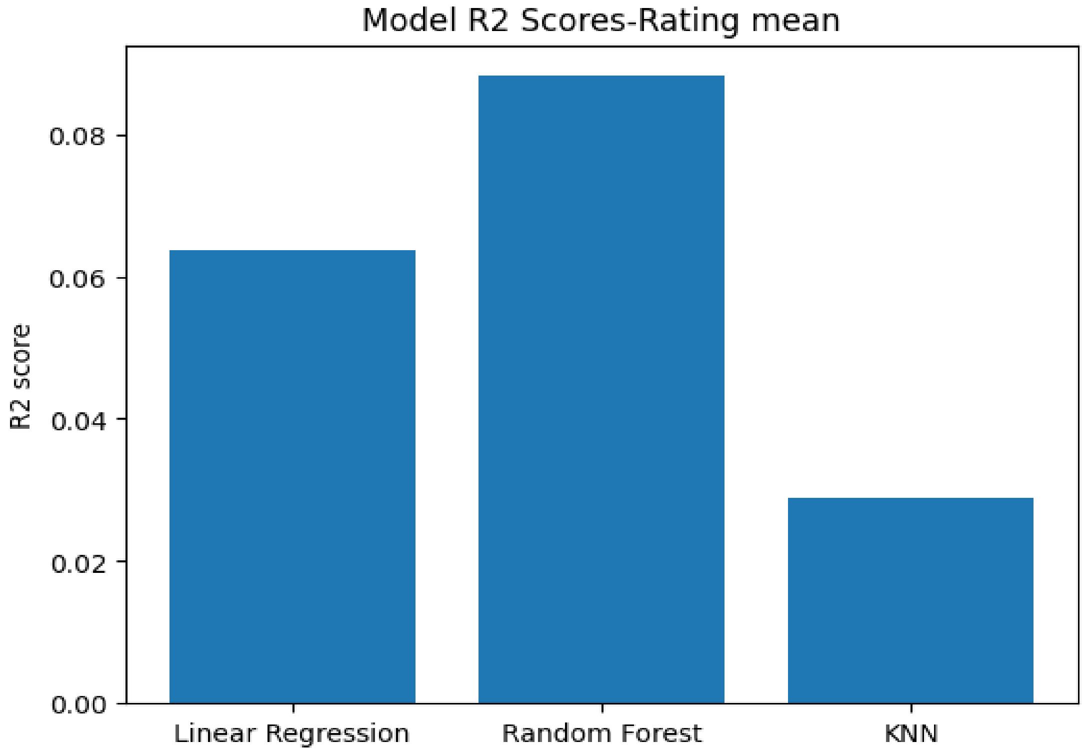
Our R^2 for the train and test sets are 0.093 and 0.088.

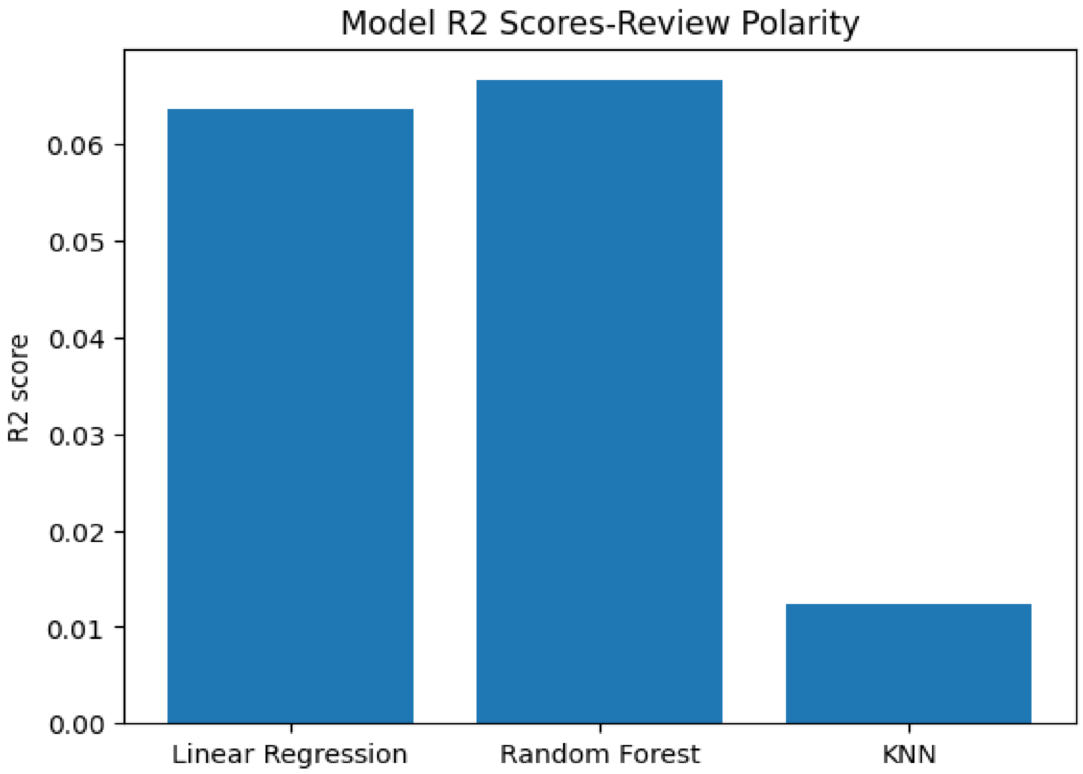
Our mean absolute error for the train and test sets are 0.647 and 0.642.

Lastly, the K-Neighbors Regression model:

Our R^2 for the test set is 0.028.

Our mean absolute error for the test set is 0.656.





Looking at our models we can see that our features don't show much predictive power. I would also like to note that I also fit the model to the rating polarity target variable as well, but the results weren't much different. Out of the 3 models, I got the best results from our Random Forest Regression.

- **Hyperparameters**

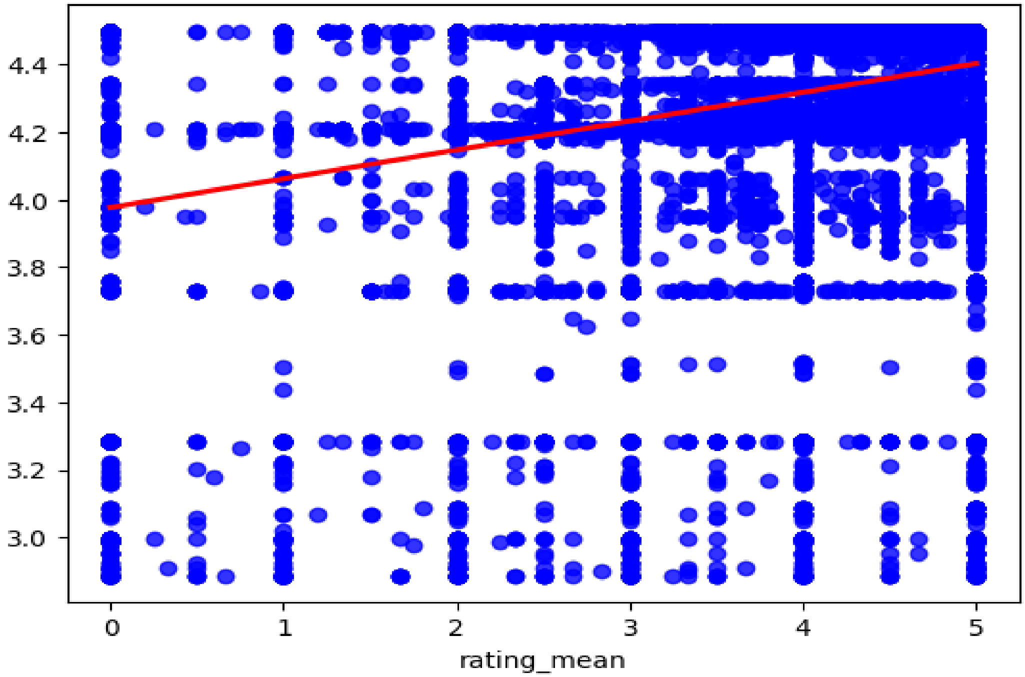
Because I am dealing with a very large data set, I ran a RandomizedSearchCV for the hyperparameters individually. This decreased the time for the CV search dramatically. The max_features parameter was the most important to me so I started there using values of 10, 20, 30, 40 and 50. The CV search determined the best max feature parameter was 50.

Next I wanted to find the best number of estimators. I tested the values 100,200,300 and 400. The best n_estimators parameter was 300.

Next I tested a few values for max_depth. Of the values 2, 4,6 and 8, the best parameter value was 8. Overall the hyperparameter values for this dataset seem to give the best performance when they are larger.

- **Results**

After applying the model to both data splits with our tuned hyperparameters the results I got were as follows:

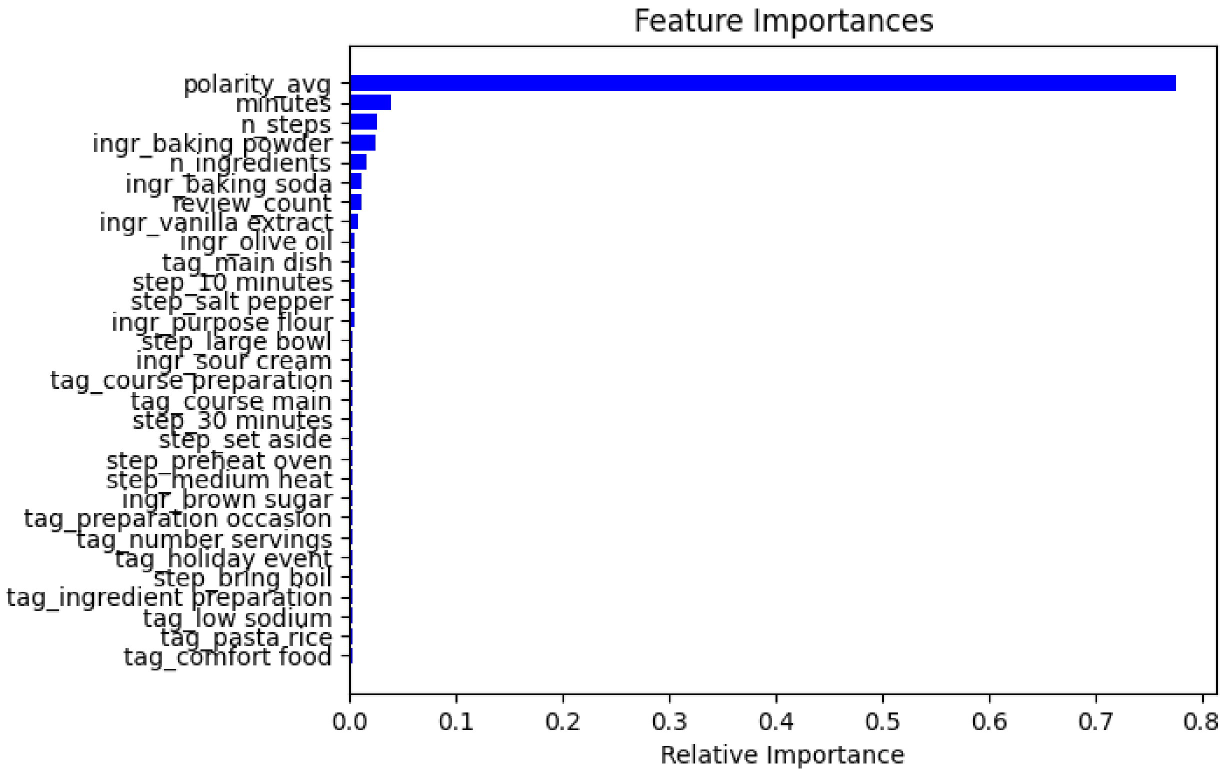


R-squared = 0.10

Our R^2 for the train and test sets are 0.134 and 0.105.

Our mean absolute error for the train and test sets are 0.632 and 0.634.

In comparison to the scores for the review polarity dataset (R^2 of test set= 0.089, MAE of test set=0.634), the metrics for the rating mean data set were minimally better so I decided to stick with the rating mean as the target variable. Although the results could be better, using these hyperparameters for the Random Forest Regression Model give us the best metrics we have seen thus far.



Looking at the feature importances for our final model we can see that the original features of number of minutes, number of steps, and number of ingredients within the top five features. As far as the text features go, our final model ranks more of our ingredient text features toward the top then we saw with other models. Baking powder baking soda and vanilla extract sit towards the top of the list. This could be an indicator that recipes in the baking category have more of an effect on user interaction.

CONCLUSIONS

After building and applying our model, it seems that the recipe attribute categories don’t have much predictive power over our rating mean. All of the metrics for model evaluation where fairly low. I think a good future approach would be to create some more solid feature categories with the test descriptions.

Maybe a longer ngram range for some of the text data would give us more specific features that could hold more significant and precise predictive power. Another step could be actually create a ‘cuisine type’ recipe attribute for all of the recipes. This could get us a step closer to more concisely categorizing our recipes as well as give us another attribute to use as a predictive feature for our user satisfaction variable. I believe that categorizing the recipes by cuisine type would be the most productive next step.

Overall it looks like there is more work that needs to be done to use if we want to make use of our recipe attributes for predicting user interactions. Although this first pass hasn’t shown a significant relationship between the features and user engagement, I think it is worth pursuing further.