1. Dataset and Exploratory Analysis:

In this assignment, we studied a dataset containing recipes and review data from Food.com [1]. From the dataset, we get information on the user id, recipes, ingredients, steps, techniques, ratings, and other descriptions. The dataset is quite large and has sufficient data for us to analyze and create models on as the size of the dataset is 1,132,367.

In the dataset, there are 231,637 unique recipes. There are 25,076 different users. The technique metadata in the dataset is stored as a binary array for each point of data, so we used the indexing of the array to analyze the techniques. There are 174 different techniques.

When performing an exploratory analysis on the raw data, we used a random sample size of 150,000. We initially took a look at the data for all ratings on our sample data. In **Figure 1**, we see that the distribution of ratings is primarily ratings of 5. To check if this was a mistake we checked the ratings of the entire dataset. A very similar distribution of ratings to the sample size was seen over the entire dataset which showed that our sample size of 150,000 should have been sufficient to identify any interesting phenomena from the dataset.

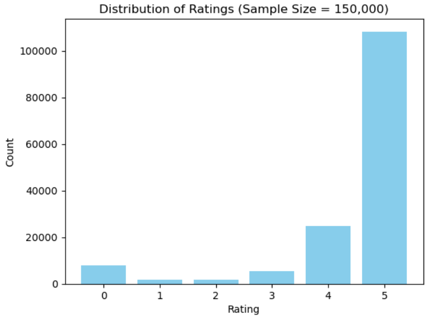


Figure 1: Bar graph of the count of ratings collected from the sample size

We constructed a correlation map to see if there was any correlation between elements of our data. In **Figure 2a**, we discovered that there is very little correlation between ratings and other elements. Minutes also has a weak correlation to other elements seen in the correlation matrix. The number of steps (n_steps) and number of ingredients (n_ingredients) has a medium correlation with each other. However, the correlation between calories and nutrients data is very high. As shown in **Figure 2b**, calories have a very high correlation. In particular, there is an especially significant connection between fats, sugar, and carbohydrates to calories.

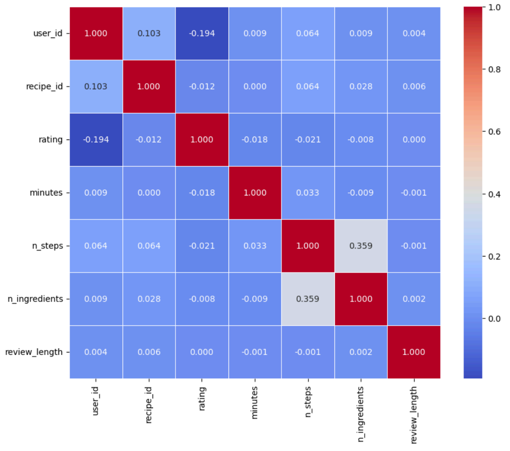


Figure 2a: Correlation matrix for raw data

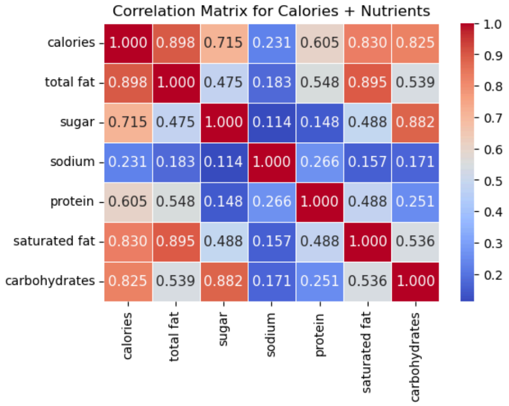


Figure 2b: Correlation matrix for calories and nutrients

Looking at data related to techniques and recipes, we saw that there were many techniques shared across many different recipes. From the data, we could determine which techniques were most popular based on the number of recipes it was used in. In

Figure 3, we constructed a plot that shows the top 40 most popular techniques based on the number of recipes that use these techniques. The top 3 techniques are at index 0, 9, and 33. This graph shows the popularity of a technique and shows how many recipes have similarities to each other through techniques.

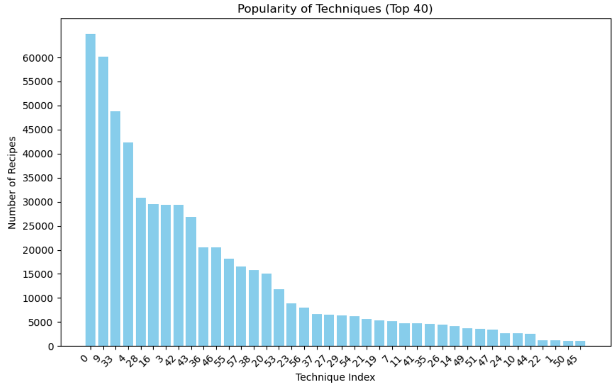


Figure 3: Bar graph of the popularity of a technique. The y axis shows the total count of recipes related to the technique index shown on the x axis.

Similarly, we looked at the data between recipes and users. We saw that similarities between recipes or users can be discovered between these two elements of data through the high count of users that made the recipes. In **Figure 4**, we constructed a graph that shows the popularity of recipes based on the number of users. The graph shows the top 100 recipes.

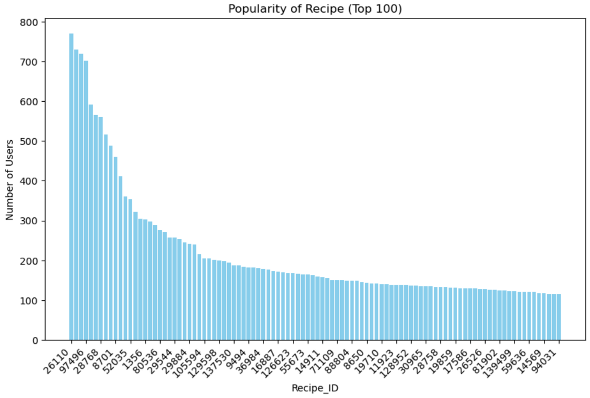


Figure 4: Bar graph of the top 100 most popular recipes. It is the number of users verses the recipe_id

Lastly, we discovered an interesting phenomenon where the month may have a correlation to when a recipe is made. We took a look at the popularity per month of recipes. When analyzing the data, we saw that some recipes had a higher count during certain months which suggest that the recipe is a seasonal recipe and that there is a connection between month and recipe. One example is the recipe “the most wonderful gingerbread cookies” whose recipe id is 80156. In figure 5, we see that this recipe has the highest popularity in december. Other recipes have a similar connection with popularities for certain months while others remain popular for all month. One such recipe is “best banana bread” whose recipe id is 2886. For this recipe, its popularity remains consistent for all months as seen in figure 6.

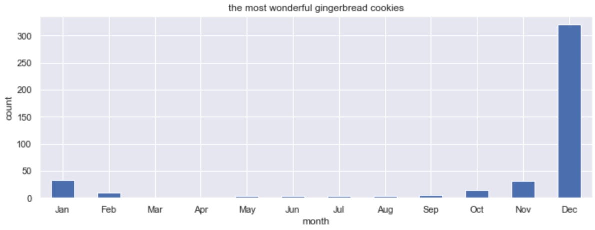


Figure 5: Popularity of “The most wonderful gingerbread cookies”

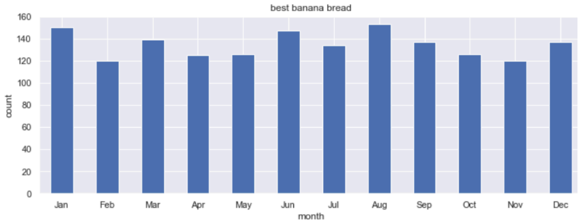


Figure 6: Popularity of “best banana bread” recipe

2. Identify Predictive Tasks:

In the course of our predictive analysis, we endeavored to investigate four distinct challenges. Upon thorough exploration of the data, our focus narrowed to the following objectives:

1. Predicting the user-assigned rating for a given recipe.
2. Determining the anticipated preparation time for a recipe.
3. Estimating the caloric content of a specified recipe.

4. Assessing the likelihood of a user preparing a given recipe.

When contextualizing these objectives within the framework of this course's materials, one can perceive the outcomes of these predictions as pivotal factors influencing the decision-making process for recipe recommendations. Acquiring an accurate estimation of the user's rating for a recipe proves essential in determining whether to endorse said recipe. For instance, consider a user pressed for time; it is evident that such a user would prefer a recipe with a quick preparation time. However, not all recipes provide explicit time indications. Therefore, the ability to predict the required cooking duration is of paramount importance. Analogously, for users conscientious about their caloric intake, obtaining an estimate of a recipe's calorie count is fundamental.

Ultimately, discerning whether a user intends to prepare a specific recipe represents the culmination of our predictive endeavors. It is plausible to utilize the insights gained from addressing the first three objectives to inform and guide the solution to the fourth problem. In the course of this project, our team is committed to applying the techniques acquired throughout this class to effectively address and resolve these four challenges.

3. Describe Your Models:

Would Make Model

<u>Most Popular (Baseline):</u>

For the baseline of the binary classification we decided to use a top percentile of the most interacted with recipes within the dataset(popularity based). This is done by iterating through the list of interactions and counting the number of times a certain *recipe_id* occurred; then sorting in descending order based on occurance. From here we would shorten the most popular list by a percentile. Then we would predict on the test set by seeing if the *recipe_id* occurred in the most popular list. If it exists in the most popular data then the model predicted 1 and if it did not exist it predicted 0. From tuning the model(iterating over the percentile cutoff) we were able to achieve an accuracy of ~70% on our baseline.

$Y = popularPredict$

<u>User Similarity:</u>

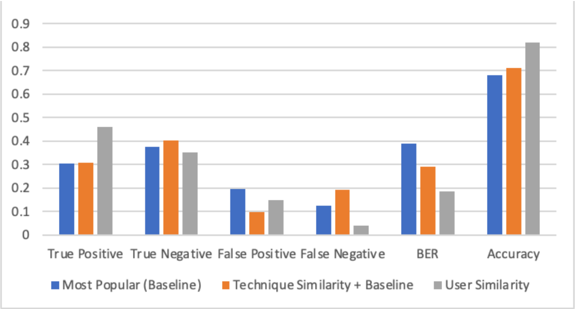
To build from our baseline predictor the first modification we tried was by finding the similarity between users and the recipes they have each made. To do this we build a dictionaries of *recipesPerUser* and *usersPerRecipe* which each hold sets of a user/recipe IDs corresponding to specific recipes/users IDs that are associated with it. This can then be used in a Jaccard similarity function where we look at a given user pair and see the overlap between the *recipe_id* sets. This allows us to judge the similarity between users based on the overlap of recipes. From the overall similarities we then can see if the given *recipe_id* and user pairs would result in a high enough similarity to assume if we can predict 1 or 0. In other words if the similarity between users is high enough and one user has made the recipe then we can predict if the other user would make it as well. This user similarity is then combined with the popularity similarity to determine if we can predict 1 or 0.

$Y = popularPredict + userSimilarity$

<u>Technique Similarity</u>

For this method we looked at the techniques required to make each recipe and compared them to the techniques each user is known to have used. To do this we followed a very similar model as using a Jaccard similarity and created both a *techniquesPerRecipe* and *techniquesPerUser* dataset. Using these two datasets for a given user recipe pair we computed the length of their intersection divided by the length of the recipe set. If the user has the techniques required this results in a number >=1 and if not a number <1. This resulted in a classifier that bisected the data into users that have the skill to make the recipe and those that don't. Using this result we grafted it onto the baseline model which allowed us to get an increase in accuracy.

$$Y = popularPredict + techniqueSimilarity$$



Calories in a Recipe

Global Average (Baseline):

Similar to Assignment 1, an intuitive baseline would be to select the average calories of all the recipes in the training set as the prediction. This is both very quick (in terms of implementation and to run) and a very good estimate in certain applications. For implementation you simply take the mean of the dataset and that is it. For elaboration on why this can be a good approach think of the following example. Suppose the data you are trying to predict is distributed in a Gaussian fashion. This means that the majority of the samples from your data are going to be near the mean. This is especially true if the variance of your distribution is small. Not only is the majority of the data near the mean but in fact they are very close to it.

Linear Regression:

One method we implemented in order to perform better than the baseline was the Linear Regression model. Linear regression is a statistical method used to model the relationship between a dependent variable (Y) and one or more independent variables (X) by fitting a linear equation to the observed data. The basic form of a simple linear regression equation is given by:

$Y = b_0 + b_1X + \varepsilon$

Here, Y represents the dependent variable, X is the independent variable, b_0 is the y-intercept, b_1 is the slope, and ε is the error term accounting for unobserved factors. The goal of linear regression is to estimate the coefficients b_0 and b_1 that minimize the sum of squared differences between the observed and predicted values. The model allows us to make predictions or understand the strength and direction of the relationship between variables. Multiple linear regression extends this concept to more than one independent variable. In our problem Y is the calorie count we are trying to predict. For X we selected the total fat, sugar, sodium, protein, saturated fat, and carbohydrates in any given recipe. The motivation for selecting these features was that during our exploratory phase we discovered that these were the factors that were most correlated with the calorie counts.

Support Vector Machines:

In order to account for the potential that our data might not behave in a linear fashion, we wanted to implement a non linear regression model. The model we chose was the regression SVM model. Support Vector Machines (SVMs) can also be extended to handle non-linear relationships through the use of kernel functions. In the case of SVMs for non-linear regression, the goal is to find a non-linear decision boundary that captures complex patterns in the data. The regression equation is expressed as:

$$Y = b + \Sigma(\alpha_i * K(X, X_i)) + \varepsilon$$

Here, Y is the predicted value, X is the input feature, b is the intercept, α_i are the support vector weights, X_i are the support vectors, K is the kernel function, and ε is the error term. The kernel function computes the similarity between input data points in a higher-dimensional space, allowing SVMs to model non-linear relationships. Common kernels include polynomial and radial basis function (RBF) kernels. This nonlinear extension enables SVMs to capture intricate patterns in the data, making them effective for a wide range of regression tasks. In this case our X and Y variables are identical to the linear regression model.

Time to Make

Global Average (Baseline):

For the baseline, we decided to go with the average time it took to prepare a recipe in our training dataset. The reasoning and justification for this choice is explained in the 'Calories in a Recipe' section preceding this section.

Preprocessing Notes:

While exploring the dataset for this task we discovered an interesting characteristic. Certain data samples were given unreasonable time values. For example 'unbaked granola bars' were labeled as taking 4000 years to make. This makes it very difficult to train a reasonable model because the model will try to minimize the error caused by these ridiculous outliers. For this reason we decided to constraint the dataset to only have recipes that are 12 hours or less. This seems like a much more reasonable question to ask because the majority of the users will not be interested in recipes that take more than 12 hours.

Linear Regression:

Again, in order to tackle this prediction problem we decided to implement a linear regression model first. In this case our Y is prediction of the number of minutes a user will have to spend in order to prepare a recipe. Our X consists of the number of steps, number of ingredients, and the techniques required to prepare the recipe.

Support Vector Machines:

In order to improve upon our results in the linear regression model we also implemented an SVM regression model. This was for the sake of exploiting any non-linearity characteristics the data might have. For this problem we used both the RBF and Poly kernel.

Recipe Rating Model

Logistic Regression:

For our recipe rating model, we logistic regression to create our model. This rating prediction model is very similar to the rating model done in chapter 3 exercise 3.1 from the workbook from class. We built a one-hot encoding where the features were based on a user's review_length and the popularity of the recipe. Our inputs into the model were a user's review and the recipe id

Always Predict Rating of 5 (Baseline):

For a comparison, our baseline is to always predict a rating of 5 as this was the most common rating as seen in our exploratory data, Figure 1. For both the baseline and our model, accuracy was measured by comparing the predicted rating to the actual rating from our test dataset.

4. Literature Review

We are using an existing dataset from *Generating Personalized Recipes from Historical User Preferences* [1] which is a dataset containing data on ratings and review data on recipes from Food.com. In the previously mentioned paper, the authors are experimenting with a way to generate personalized recipes from incomplete input details and past user preferences. For their research, they look at prior recipe usage and prior technique usage to generate a model. Their model has inputs of the recipe name as a sequence of tokens, a partial list of ingredients, and a caloric level. They built their own dataset based on recipes and user interactions from Food.com.

For their experiment, they made use of historical user preferences to produce results better than baseline as it improved generalization properties. The conclusion to their work was that their personalized generative models were successfully able to generate personalized and coherent recipes.

In relation to the models we explored in this assignment, we utilized the dataset in a different way so we will not have the same conclusions. Instead of using historical users and recipes data to make personalized generative models, we created models that use recipe and user information to make predictions based on the popularity of an element and its similarity to other users, recipes, nutrients, or techniques. All of our predictive tasks we made models for are covered in the second section of this paper.

In the paper *Spilling the beans: Food recipes popularity prediction using ingredient networks* [2], the paper set out to investigate which features are the most important in determining a recipe's popularity, finding the similarity of unrated recipes, and predicting popularity of unrated recipes. They obtained their data by compiling food recipe data and reviews from the website Allrecipes. The dataset contains information about the recipes, nutrition, ingredients, reviews, and more. The data contained is very similar to the dataset from paper [1], which is the one we are using. To analyze their data, they created a complement network which is a network of nodes and edges. The edges are based on the ingredients' pointwise mutual information which tell us if there is a similarity between two ingredients. They also create an algorithm to determine a recipe's similarity to others.

For their experimentation, they trained multiple models using different techniques. They used forest classifiers, regression algorithms, SVMs, and k-nearest neighbors. For their data they also encountered the same issue we had which was that more than 90 % of their review ratings were a 4 or 5. Their solution was to oversample the minority class and undersample the overrepresented. The results of their experimentation was that their models could accurately predict a users reference for inputted recipes if they reviewed both recipes and if there was similarity between the users.

From this reading, we saw that they also had a rating imbalance in their dataset. Because of this imbalance they had difficulty predicting lower ratings. However, our results differed as our model predicted lower results at times when it was supposed to be higher, such as rating a recipe a 4 when the correct rating was 5. Like the researchers from paper 2, we found that using the rating data for food review to be difficult as they were imbalanced.

5. Results

Recipe Rating Model:

We evaluated our model and the baseline model by checking the accuracy compared to the actual rating from the test data. By checking our prediction to the actual, we can determine the accuracy of the model in terms of percentages.

The result of our logistic regression recipe rating model had an accuracy of 72.09%. The baseline model (prediction of only rating 5) had an accuracy of 99.99%. After reviewing the exploratory data, this outcome makes sense as our predictive model would not be ideal for this dataset as the average rating is particularly high and skewed primarily toward a rating of 5 as seen in Figure 1. Additionally, the rating does not have a strong correlation to any particular element of data as seen in the correlation matrix from Figure 2a.

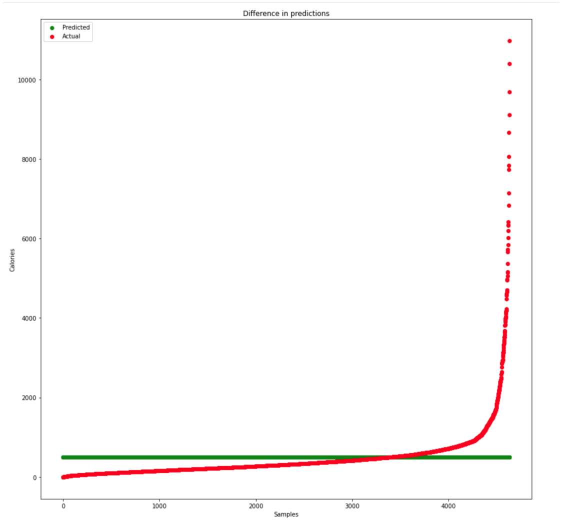
Calories in Recipe Model:

Since this is a regression model, the two metrics that were used to evaluate this problem were Mean Squared Error (MSE) and Root Means Squared Error (RMSE). The reason for also measuring the RMSE in addition to the MSE is to get better intuition about how off our model is.

<u>Baseline Results:</u>

After predicting the average calories for all the samples in the test set, these were our results:
MSE: 482851.837
RMSE: 694.875

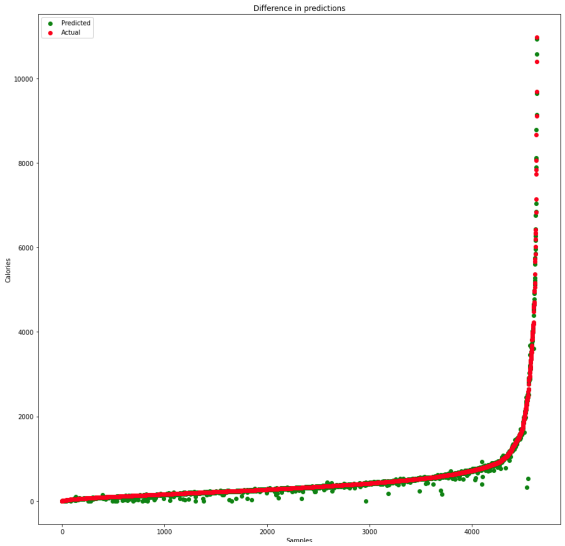
Plot:



The green line represents the predictions and the red line represents the actual values but sorted (better for visual purposes). We can see that this is a decent solution but fails greatly for significantly larger values in our data set.

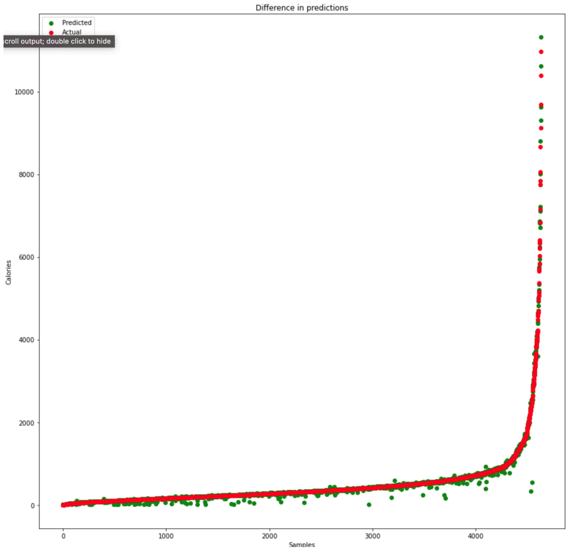
<u>Linear Regression Results:</u>

MSE: 2881.2286
RMSE: 53.677
Plot:



As you can see, the predictions are significantly better than the baseline. From our RMSE we can see that we are only off by 54 calories on average as compared to 695 calories on average in the baseline.

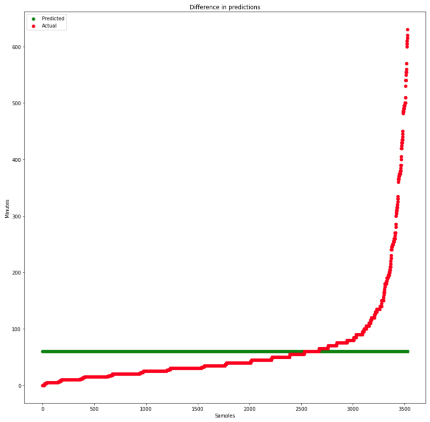
<u>Support Vector Machines Results:</u>
MSE: 3004.408
RMSE: 54.81248
Plot:



These results are near identical to the linear regression but take much longer to train. This can be an important factor to take into consideration when selecting a model. This also means that our data did not have any nonlinearity for us to take into account.

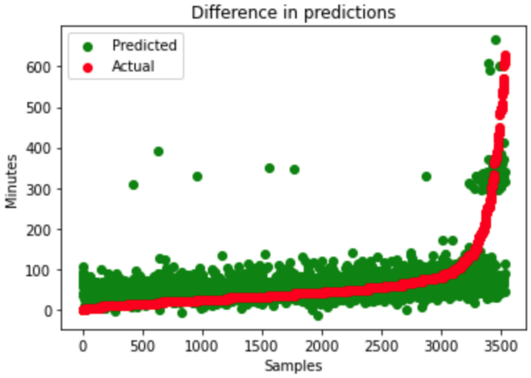
Cook Time Model:

<u>Baseline Results:</u>
MSE: 6739.561
RMSE: 82.0948
Plot:



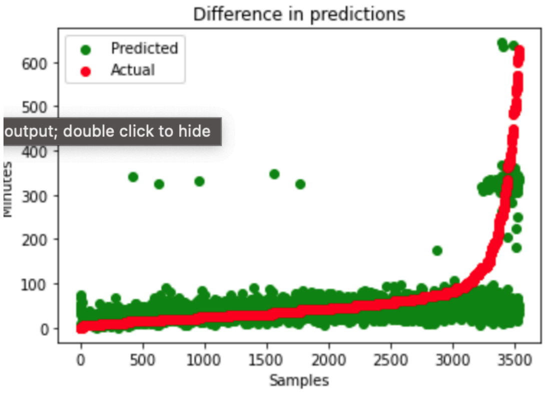
This method no longer is a good predictor because the variance in our data is too great. Despite using the mean, since our data is so far spread out our error is still too high.

<u>Linear Regression Results:</u>
MSE: 4362.764
RMSE: 66.0512
Plot:



This shows that the model is trying to learn something but it is very limited. From our exploratory phase we discovered that the features we are using are not very correlated with the predicted variable. As a matter of fact, it is mostly the contributions of the technique variable that is allowing us to make progress in our prediction task.

<u>Support Vector Machines Results:</u>
MSE: 4708.325
RMSE: 68.617
Plot:



Again this is really identical to our linear model, suggesting that there is no nonlinearity to take advantage of.

Would Make Model:
Looking at the model outcomes of the would make part in section 3 we can see that 2 of our models were able to improve upon the baseline of 68% accuracy. The first of the two models leveraged the baseline and a similarity between the techniques the user knows and the techniques required to make the recipe. This resulted in a 3% improvement over baseline. The second and best strategy implemented was the user similarity which leveraged a jaccard similarity in the model to relate other (user,recipe) pairs. This model resulted in an 82% accuracy which is a 14% increase over our baseline.

Looking at our two final models of the user and technique similarities these two models perform well compared to the baseline. Especially compared to other models we implemented on the dataset. Other models we tried were based on the ingredient IDs similarities which lead to minimal improvement over the popularity predictor as there is not a strong correlation between the two(around ~51% accuracy). Additionally, to improve this model we also tried including the number of steps similarity into the predictor however this had no impact on the accuracy of the model(~51% accuracy). From our attempts we found that data relating to counts or ingredients performed poorly whereas looking at other user specific characteristics provided more accurate metrics on which to build a model.

6. Citation

[1] Generating Personalized Recipes from Historical User Preferences
Bodhisattwa Prasad Majumder*, Shuyang Li*, Jianmo Ni, Julian McAuley
EMNLP, 2019
https://cseweb.ucsd.edu/~jmcauley/pdfs/emnlp19c.pdf

[2] Ruis, F. A. (2023). *Spilling the Beans: Food Recipe Popularity Prediction Using Ingredient Networks*. Retrieved from https://essay.utwente.nl/78726/1/Recipe_popularity_prediction_Paper.pdf