# Using Recipe Data to Predict Number of Ingredients

Cool CSE158 Students
*Computer Science and Engineering*
*University of California, San Diego*
email@ucsd.edu

## ABSTRACT

In our study, we utilized a natural language processing bag-of-words model and a linear regression model to predict the number of ingredients required for any given recipe. The model was trained based on features of the recipe that we thought would most likely affect the number of ingredients, such as the length of text in a recipe's steps and the specific words used in the recipe.

## 1. INTRODUCTION

Often, as college students wanting to learn how to cook, we look for recipes that fit our tight and packed schedules. Juggling cooking on top of classes and extracurricular activities can be exhausting. Thus, finding a timely and cost-effective recipe is crucial for college students. Part of this requires us to know how many ingredients we may need to prepare ahead of time and choose recipes that might require fewer ingredients. Our study aims to provide a solution by predicting the number of ingredients in any given recipe.

## 2. DATASET

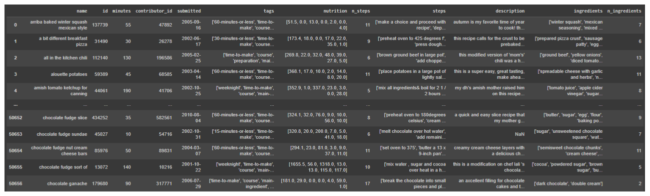
### 2.1 Identifying a Dataset

The dataset we decided to study includes 180K+ recipes and 700K+ recipe reviews taken from Food.com covering 18 years of user interactions and uploads. This dataset was most interesting to us because of our shared interest in food and cooking. The data also felt similar to those that we have worked with in the past and so we felt more confident in our abilities to successfully create a prediction model from it.

### 2.2 Exploratory Analysis

Before performing an exploratory analysis, we cleaned our dataset by removing any recipes that took more than a day to prepare and any recipes that took no time (i.e. 0 minutes) to prepare. We also reduced the dataset from 231,637 recipes to 50,000 recipes.

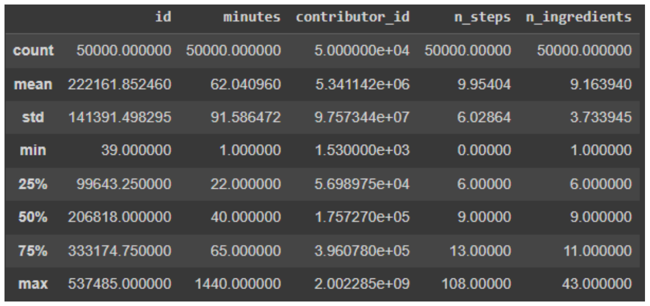
Not all of the features in our dataset were useful to our predictive task and so we simply ignored them when performing the analysis. All of the features in our dataset can be seen in Fig. 2.1.



| | name | id | minutes | contributor_id | submitted | tags | nutrition | n_steps | steps | description | ingredients | n_ingredients |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | arriba baked winter squash mexican style | 137739 | 55 | 47892 | 2005-09-16 | ['60-minutes-or-less', 'time-to-make', 'course... | [51.5, 0.0, 13.0, 0.0, 2.0, 0.0, 4.0] | 11 | ['make a choice and proceed with recipe', 'dep... | autumn is my favorite time of year to cook! th... | ['winter squash', 'mexican seasoning', 'mixed ... | 7 |
| 1 | a bit different breakfast pizza | 31490 | 30 | 26278 | 2002-06-17 | ['30-minutes-or-less', 'time-to-make', 'course... | [173.4, 18.0, 0.0, 17.0, 22.0, 35.0, 1.0] | 9 | ['preheat oven to 425 degrees f', 'press dough... | this recipe calls for the crust to be prebaked... | ['prepared pizza crust', 'sausage patty', 'egg... | 6 |
| 2 | all in the kitchen chili | 112140 | 130 | 196586 | 2005-02-25 | ['time-to-make', 'course', 'preparation', 'mai... | [269.8, 22.0, 32.0, 48.0, 39.0, 27.0, 5.0] | 6 | ['brown ground beef in large pot', 'add choppe... | this modified version of 'mom's' chili was a h... | ['ground beef', 'yellow onions', 'diced tomato... | 13 |
| 3 | alouette potatoes | 59389 | 45 | 68585 | 2003-04-14 | ['60-minutes-or-less', 'time-to-make', 'course... | [368.1, 17.0, 10.0, 2.0, 14.0, 8.0, 20.0] | 11 | ['place potatoes in a large pot of lightly sal... | this is a super easy, great tasting, make ahea... | ['spreadable cheese with garlic and herbs', 'n... | 11 |
| 4 | amish tomato ketchup for canning | 44061 | 190 | 41706 | 2002-10-25 | ['weeknight', 'time-to-make', 'course', 'main-... | [352.9, 1.0, 337.0, 23.0, 3.0, 0.0, 28.0] | 5 | ['mix all ingredients& boil for 2 1 / 2 hours ... | my dh's amish mother raised him on this recipe... | ['tomato juice', 'apple cider vinegar', 'sugar... | 8 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 50652 | chocolate fudge slice | 434252 | 35 | 582561 | 2010-08-04 | ['60-minutes-or-less', 'time-to-make', 'course... | [324.1, 32.0, 76.0, 9.0, 10.0, 56.0, 10.0] | 8 | ['preheat oven to 180degrees celsius', 'cream ... | a quick and easy slice recipe that my mother g... | ['butter', 'sugar', 'egg', 'flour', 'baking po... | 9 |
| 50653 | chocolate fudge sundae | 45027 | 10 | 54716 | 2002-10-31 | ['15-minutes-or-less', 'time-to-make', 'course... | [320.8, 20.0, 200.0, 7.0, 5.0, 41.0, 18.0] | 6 | ['melt chocolate over hot water', 'add remaini... | NaN | ['sugar', 'unsweetened chocolate square', 'wat... | 7 |
| 50654 | chocolate fudge nut cream cheese bars | 85976 | 50 | 89831 | 2004-03-07 | ['60-minutes-or-less', 'time-to-make', 'course... | [294.1, 23.0, 81.0, 3.0, 9.0, 37.0, 11.0] | 11 | ['set oven to 375', 'butter a 13 x 9-inch pan'... | creamy cream cheese layers with a delicious ch... | ['semisweet chocolate chunks', 'cream cheese',... | 11 |
| 50655 | chocolate fudge sort of | 13072 | 140 | 10216 | 2001-10-22 | ['weeknight', 'time-to-make', 'course', 'main-... | [1655.5, 56.0, 1318.0, 13.0, 13.0, 115.0, 117.0] | 10 | ['mix water , sugar and cocoa over heat in a h... | this is a modification on chef lali 's chocola... | ['cocoa', 'powdered sugar', 'brown sugar', 'bu... | 5 |
| 50656 | chocolate ganache | 179680 | 90 | 317771 | 2006-07-29 | ['time-to-make', 'course', 'main-ingredient', ... | [181.0, 29.0, 0.0, 0.0, 4.0, 59.0, 1.0] | 17 | ['break the chocolate into small pieces and pl... | an axcellent filling for chocolate cakes and t... | ['dark chocolate', 'double cream'] | 2 |

**Fig. 2.1** Features in our dataset

The dataset's basic statistics shown in Fig. 2.2 allowed us to further verify that the dataset we were using was realistic. It was interesting to see the range of recipes that existed.



| | id | minutes | contributor_id | n_steps | n_ingredients |
|---|---|---|---|---|---|
| count | 50000.000000 | 50000.000000 | 5.000000e+04 | 50000.00000 | 50000.000000 |
| mean | 222161.852460 | 62.040960 | 5.341142e+06 | 9.95404 | 9.163940 |
| std | 141391.498295 | 91.586472 | 9.757344e+07 | 6.02864 | 3.733945 |
| min | 39.000000 | 1.000000 | 1.530000e+03 | 0.00000 | 1.000000 |
| 25% | 99643.250000 | 22.000000 | 5.698975e+04 | 6.00000 | 6.000000 |
| 50% | 206818.000000 | 40.000000 | 1.757270e+05 | 9.00000 | 9.000000 |
| 75% | 333174.750000 | 65.000000 | 3.960780e+05 | 13.00000 | 11.000000 |
| max | 537485.000000 | 1440.000000 | 2.002285e+09 | 108.00000 | 43.000000 |

**Fig. 2.2** Basic statistics from our dataset

We tried various combinations of plots to see if there were any clear trends in our dataset between features. The most notable trends we observed were that the number of ingredients tended to increase as recipes got longer (either in steps or in time) and this can be seen in Fig. 2.4-2.6. According to Fig. 2.3, there seemed to also be more recipes that required 5-10 ingredients, which could be an indication that recipes with that number of ingredients might be more popular. Fewer ingredients would probably also be optimal for college students.

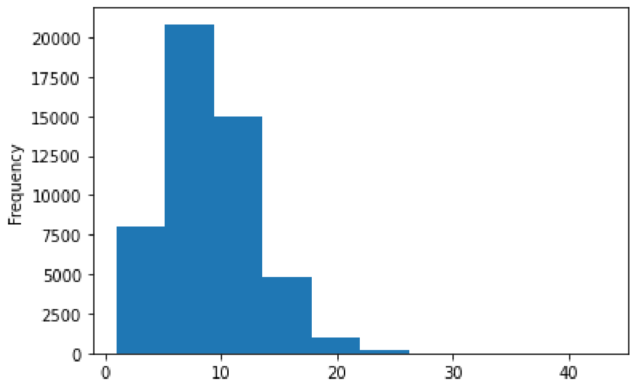


**Fig. 2.3** Histogram of the frequency of # of ingredients in recipes

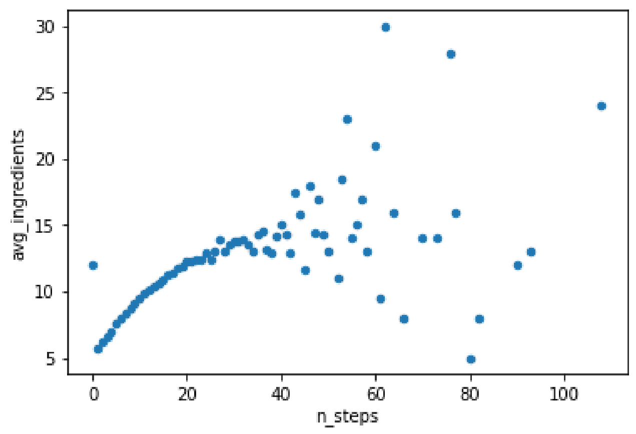


**Fig. 2.4** Plot of average # of ingredients vs. # of steps

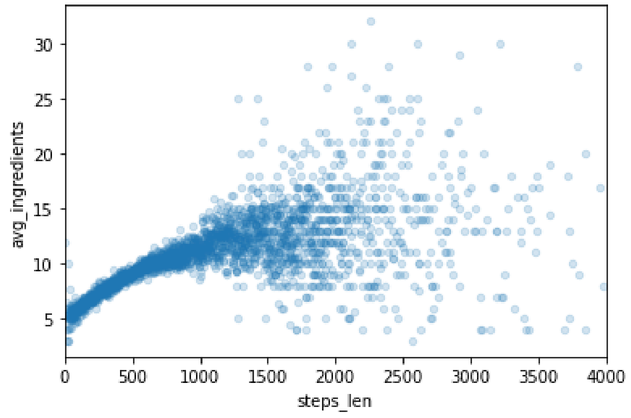


**Fig. 2.5** Plot of average # of ingredients vs. length of steps text

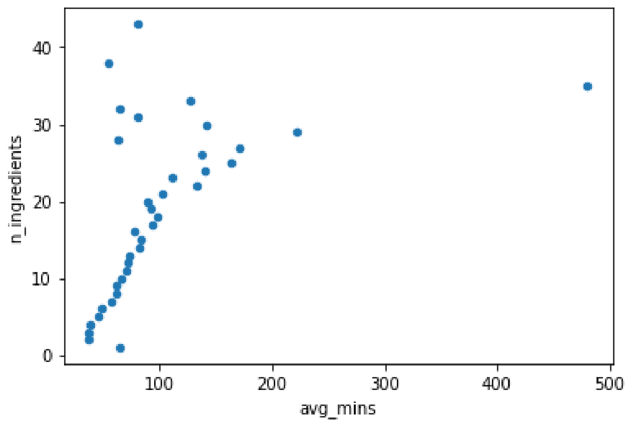


**Fig. 2.6** Plot of # of ingredients vs. average prep time

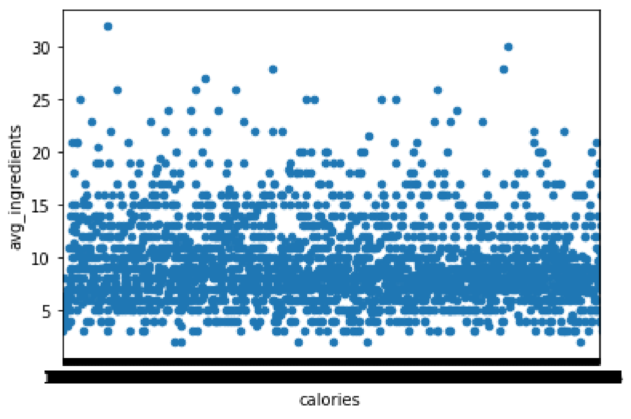


**Fig. 2.7** Plot of average # of ingredients vs. calories

## 3. PREDICTIVE TASK

### 3.1 Identifying a Predictive Task

Using cooking recipes and recipe review data from Food.com, we wanted to predict the number of ingredients in a recipe based on the recipe's metadata.

The accuracy of the prediction task is measured by the mean squared error (MSE) of the training set and the test set we created from the dataset.

### 3.2 Relevant Features and Processing

There are several relevant features to be considered for the predictive task. The features described below were experimented with, but they do not all appear in the final model.

If `tags` are standardized and not user-inputted, they may be useful in identifying the number of ingredients. `tags` can be included in the feature vector by tokenizing the tags and setting the feature to '1' if the tag is included in the recipe and setting the feature to '0' if the tag is not included. This will most likely result in a sparse feature vector, so it may not be a good predictor. Alternatively, a bag-of-words model can be used to identify the common occurrences of words in `tags` across the dataset. `tags`, however, proved to be inconsistent and also did not appear to correlate with the number of ingredients in a recipe.

The `name` of a recipe may be relevant in predicting the number of ingredients of a recipe. The words in the name of the recipe can be tokenized in a similar fashion to the way `tags` are tokenized. Again though, this vector would be incredibly sparse if the set of individual words in `name` is large and the frequency of each word is small relative to the size of the set. The relationship between the name of a recipe and the number of ingredients may be useful. For instance, a chicken recipe may generally have a larger number of ingredients than a salad recipe. In our case, there were many recipes with many unique names. Since the names of recipes were not standardized in any way, it was difficult to consider them in our model as well.

The `ingredients` of a recipe can be tokenized as well but this feature is part of our prediction task, which aims to predict a recipe's number of ingredients.

There may exist a correlation between the integer value `n_steps` and the number of ingredients in a recipe. It is likely that the `n_ingredients` value of a recipe increases as the `n_steps` value increases.

Text mining can be used to leverage the `steps` text of a recipe. The character length of the recipe and a n-gram bag-of-words model may be useful predictors.

Similarly, `descriptions` written by the authors of the recipe could be used to predict the number of ingredients in the recipe, but this is unlikely as descriptions can be varying in content.

`nutrition` includes calories and the nutrient percentage of the daily value. These values can simply be appended to the model's feature vector or discarded since they are not necessary for our prediction.

Features from the user `review` data were also considered. A recipe's `rating` could be a relevant feature in the prediction task. A recipe's popularity could also be useful, where popularity is measured by the number of times a recipe has been reviewed. When building the model, we decided to exclude features that stemmed from the review data because we wanted to make the prediction based on only the recipe data. So in the end, we excluded a recipe's rating and a recipe's popularity from the model.

In order to process the data, we downloaded it from Kaggle in the form of a CSV file. Using Pandas and then eventually NumPy, we read the file into a DataFrame, which gave us columns with all the recipes' features. We further cleaned the data by attempting to remove any outliers (like recipes with outrageously large preparation times) and ignored any features deemed unnecessary (like nutrition).

### 3.3 Baselines

In order to compare the model to a naive baseline, we created a basic comparison metric by averaging the value of `n_steps` for each value of `n_ingredients`. More precisely, we grouped by each value of `n_ingredients` then we calculated the average `n_steps` for each of those groups. This baseline model made predictions on the number of ingredients of a given recipe by taking that recipe's `n_steps` and then finding the associated value of `n_ingredients` based on the previously calculated averages. The idea behind this baseline model was that the number of steps in a recipe is directly related to the number of ingredients in that recipe.

## 4. MODEL

The original model proposed consisted of a bag-of-words model based on the description of the recipe. The model was trained on the first 50,000 entries of the dataset and validated on the next 50,000. It first takes all of the descriptions from the training data and counts the frequency of each word. When counting the words, they were all stemmed to prevent similar variations of the same root word. Furthermore, punctuation and stop words were removed as they were deemed unimportant. Finally, the feature vector was built by counting the frequency of the top 1,000 words in each entry in the training set. The feature vector was then fit to a linear regression model since we were predicting a continuous variable.

As stated in our results further in our report, our original model was not as successful as we hoped, therefore a second model was built featuring a different prediction task. In this second model, the number of steps, minutes value of the recipe, and bag-of-words model were combined into one feature vector in order to predict the number of ingredients.

This feature vector was still fit to a linear regression model as we were still predicting on a continuous scale. For the sake of simplicity and accuracy, the predictions were then rounded to the nearest whole number before being compared to the validation set.

## 4.1 Optimizing

A regularization pipeline was built to optimize the model's accuracy by tuning its parameters. The model's accuracy was also optimized by experimenting with different features that could feasibly improve the model. The experimentation with features was initially conducted by including features in the model that had seemingly obvious connections to the preparation time of a recipe. From there, experimentation with the features was conducted via trial and error.

## 4.2 Issues

We initially encountered issues with scalability due to the sheer size of the dataset. We solved this issue by truncating the dataset to 50,000 data points. We continued to prune the dataset by eliminating outliers.

We did not encounter any overfitting issues although our chosen model performed poorly on the prediction task. The issues we faced with the model and prediction task are described in more detail in Section 6.

## 4.3 Other Models

Originally there were a few models we built with different feature vectors and varying levels of success. For example, some of the models we originally experimented with were either fed features from the dataset with no manipulation or a bag-of-words model that did have any other features. Eventually, we combined a few of our models together into one streamlined model that provided us the best overall results. While we experimented with different feature vectors, the basis of the model was the same in all versions which was a Linear Regression model.

On its own, the bag-of-words model's weakness was that it was ineffective because the text in the recipe data did not contain enough varied sentiment. Most of the text contains positive or neutral sentiments, lacking negative sentiments that make this model more successful. On the other hand, this model's strength lies in the fact that text data covers a wide span. This allows the model to possibly account for hidden meaning in the data.

On its own, the model that was based purely on numerical relationships performed poorly because the data lacked diversity. However, there was a relationship between the number of steps in a recipe and the number of ingredients.

By combining the models, we were able to hide each model's individual weakness while exploiting each model's individual strength. The result was a slightly more successful model.

The biggest obstacle was that our original model proved to not perform very well. As discussed in Section 6, the Mean Square Error (MSE) was so high that our model was basically purposeless. As a result, we realized the best course of action was to pivot from our original prediction goal and to experiment with the data to see if there was a prediction task that could be implemented with greater accuracy.

## 5. LITERATURE

## 5.1 Dataset Origin

The food dataset we chose includes 180K+ recipes and 700K+ recipe reviews taken from Food.com covering 18 years of user interactions and uploads. For our testing purposes, we only required 50K recipes since we did not need to take into account recipe reviews since we are only predicting the amount of time a recipe takes to complete depending on the recipe's attributes.

In their study [1], Majumder et al. used this dataset to generate personalized recipes from historical user preferences (i.e. reviews and recipe data), building on top of existing recipe generation. They used users' historical preferences in conjunction with natural-text instructions using names and incomplete ingredient details and an attention fusion layer that combined technique- and recipe-level representations of users' previous recipes to control recipe text generation.

## 5.2 Similar Datasets

The closest study we found tried to predict the ingredients needed based on given recipe titles by Haryo Akbarianto Wibowo [5]. This study utilizes Seq2Seq Deep Learning architectures in Natural Language Processing, specifically Gated Recurrent Unit (GRU)+ Bahdanau Attention and Transformer. The dataset that Wibowo used also contained food recipes, specifically Indonesian recipes from Kaggle that were written in Indonesian. This data contained the name, ingredients, steps, loves, URL, and basic ingredients of each recipe.

Another similar study [3] attempted to predict the type of cuisine based on the recipe's ingredients. Srinivasasubramanian et al. used classification by grouping together ingredients that belonged to certain cuisines to predict other cuisines with similar ingredients. The dataset they used was from a Kaggle competition where lists of ingredients were tagged with the type of cuisine they most likely belonged to.

Other studies have mostly tried to predict the cooking time at restaurants for food delivery services. This is an important piece of information that contributes to the timing couriers are dispatched to pick up food and the estimated delivery time customers are given. Because of this, these food delivery service companies have tasked many engineering teams and competition participants to best predict food preparation times. Companies themselves have access to data regarding all 3 relevant agents: delivery partners, restaurants, and customers [2]. They have iterated from a single fixed guess value to using feature engineering for more comprehensive methods. Within competitions, companies have released datasets of restaurant menu items with descriptions and historical food preparation times of orders prepared by the restaurant with features including quantities of items ordered [4]. The winner of one of these competitions used one hot encoding and basic feature engineering.

### 5.2.1 Inspiration

Since multiple similar studies used natural language processing with regard to the recipe's text and ingredients, we decided to build on top of those methods. In [3], Majumder et al. used the data-to-text generation abilities of natural language processing to generate recipe text based on data such as in automated journalism, question-answering, and abstractive summarization. However, in our case, we only needed to use natural language processing to help us predict the amount of time a recipe might take according to the descriptions used in the recipe text. In [4]. Wibowow also utilized natural language processing and so we wanted to use a similar method of tokenization.

We also took inspiration from the bag-of-words model and sentiment analysis we explored from the predictive tasks in our CSE 158 assignments involving predicting star ratings based on product reviews.

### 5.2.2 Related Conclusions

All of these studies' conclusions and results are rather dissimilar to our own because they were all predicting different tasks based on different features.

The most similar paper also concluded that their models have yet to learn certain general characteristics of recipes and the varieties of ways of cooking something [5].

## 6. RESULTS

With our approach, we received a relatively high MSE of ~9.784. When computing the accuracy of our model (whether the predicted number of steps was equal to the true number of steps), the accuracy was only marginally as good at an accuracy rate of around 0.13086. However, when we accounted for some tolerance of ± 1 step, the accuracy increased to around .2429.

### 6.1 Performance

To begin, our original plan was actually to predict the cooking time of the recipe based on a bag-of-words model based on the recipe description, similar to Assignment 1. Unfortunately, this model proved to be quite unsuccessful resulting in an MSE of >10000. With such an extremely high MSE, we decided to scrap this approach and shift to a different goal of predicting the number of ingredients in the recipe instead. The main conclusion drawn from this original model is that the bag of words model did not work as well in this instance because the recipe descriptions do not have words correlating to the overall length of the recipe. In comparison to Assignment 1 where a review text contained sentiments (either positive or negative) that were good indicators of an overall rating, recipe descriptions did not prove to be a good indicator for time.

Our revised approach that we shifted to perform better than the original, but still not as well as we originally hoped for. While our results showed that our models did not perform as well as expected, there were significant improvements to the model with each iteration. The primary conclusion that can be drawn from our results is that the number of ingredients has a low correlation with the description, time, and number of steps.

### 6.2 Interpretation

Initially, we wanted to predict a recipe’s cook time based on the recipe’s metadata. However, several different models failed to adequately make correct predictions. The mean squared errors for these models were beyond unreasonable. Variations of different models failed the prediction task because there was not enough information to predict relationships between cook time and other recipe data.

This lack of a relationship is most notably underlined by the description of a recipe. The model that used the recipes' description to make predictions was inspired by the bag-of-words model that used a product's review text to predict its star rating. The bag-of-words model was successful in predicting star ratings from product reviews because there exists some user sentiment within product reviews. This is not the case for recipe descriptions. In general, a recipe description will be positive or neutral, so there is no correlation between sentiment and cooking time. Additionally, a recipe description contains basic information about the dish itself, and this information is random. This randomness could also have contributed to the failure of this model.

When considering the results of our current approach, it makes sense that recipes can have varying amounts of ingredients that often have no correlation to other factors. We can thus still conclude that the correlation between recipe data and the number of ingredients is stronger than the correlation between recipe data and cook time.

## 7. REFERENCES

[1] Majumder, Bodhisattwa Prasad, Li, Shuyang, Ni, Jianmo, and McAuley, Julian. 2019. Generating Personalized Recipes from Historical User Preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5976–5982, Hong Kong, China. Association for Computational Linguistics. DOI=http://dx.doi.org/10.18653/v1/D19-1613.

[2] Pathirana, Amila. 2019. Food Preparation Time Prediction Competition- 1st Place Entry. *LinkedIn.* https://www.linkedin.com/pulse/food-preparation-time-prediction-competition-1st-place-pathirana/.

[3] Srinivasasubramanian, Sridhar et al. 2015. Identifying Cuisines From Ingredients. https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/039.pdf.

[4] Wang, Zi. 2019. Predicting Time to Cook, Arrive, and Deliver at Uber Eats. *InfoQ*. https://www.infoq.com/articles/uber-eats-time-predictions/.

[5] Wibowo, Haryo Akbarianto. 2020. 🍖🍲 Recibrew! Predicting Food Ingredients with Deep Learning!🍲🍖. *TowardsAI*. https://pub.towardsai.net/recibrew-find-out-the-foods-ingredients-dbc2a4e37383.