# Predicting the Rating of a Recipe based on its Components

*Lakshyana KC, Nhat Pham*

## 1. Background

### 1.1 Motivation

Exchanging recipes over the internet has become more popular than ever before. There are numerous websites that allow us to upload our own recipes, search for others', as well as rate and review them. Such sites aggregate invaluable information, not only in terms of providing recipes, but also in providing information about cultural preferences with regard to food. For example, a comparison of popular recipes on an European versus an American website shows that recipes on the American website have more diverse ingredients and are more protein rich (Trattner et. al, 2018). Understanding what constitutes a high-rated recipe offers a lens through which we can study eating habits at a societal level. Food creators can also use this prediction to modify their recipe content in some ways to improve its reception and visibility in the community.

Personally, we are curious to learn if there are specific combinations of ingredients or certain instructions that have an influence on how much the users like the recipes. Assuming our preferences align with the public's, we can use this to expand our favorite recipe repertoire.

### 1.2 Dataset

We used the recipe information from Food.com, as taken from the Kaggle dataset "Food.com - Recipes and Reviews" (Alvin, 2020). The data includes a list of 500k+ recipes and a list of 1.4M reviews dated from 1999 to 2020. We chose this dataset because of its recency and large size. It contains rich text data including description, ingredients, cooking instructions, and users' reviews, as well as numeric data on nutritional values, cooking time, and other features.

### 1.3 Objectives

The objective of this project was to train and compare multi-class classification models for predicting the rating of an online recipe using the ingredients, recipe instructions, nutrition and healthiness, complexity, description and title attributes. We also trained a Doc2Vec embedding model using the concatenated recipes' ingredients and instructions text. We performed some visualizations of the recipe embedding models to better understand the dataset and characterize the embedding vectors for the recipes. We then trained multiple statistical models such as Naive Bayes (NB), Logistic Regression, Support Vector Machine (SVM), and Random Forest using different combinations of text and numerical features.

We also trained a feedforward classification model using the pre-trained embedding model weights to compare its performance with the other models. We will then evaluate the model performances using a common test dataset, and evaluation metrics such as the F1 score, Precision, Recall, and Accuracy metrics.

## 2. Data Preprocessing and Exploration

### 2. 1 Text Features

Before starting the model training, we first carried out some preprocessing steps to clean up the text data. First, we selected a subset of the dataset to only include recipes with the ratings labels, and non-null feature attributes. Then, from the subset of 270k recipes, we cleaned the text features by removing the symbols, punctuations, quotes and other symbols and characters; separating the alpha-numeric characters; lower-casing the text; sentence tags and other clean up steps. We then tokenized the texts, and applied stop-words removal and lemmatization, so we reduce the size of the vocabulary in the text and keep the most useful and distinctive parts of the text features. For the ingredients, we preserved each single ingredient as a single token. For instance, the ingredient "olive oil" was updated as "olive-oil" to treat it like a single token. Besides applying these text-preprocessing steps, we also created a combined text feature using both the ingredients and the instructions to train the embedding model with the concatenated feature.

### 2.2 Other Features

For the numerical features, we applied some rounding to the aggregated ratings data to limit the labels to be a discrete value such as 1, 2, 3, 4, or 5. We further combined the dataset for 1 and 2 ratings to one category, as they were significantly smaller in size compared to the recipes with higher ratings. Additionally, we assumed that qualitatively, predicting the recipes using 4 labels versus 5 labels may not have a significant difference for the purpose of our project.

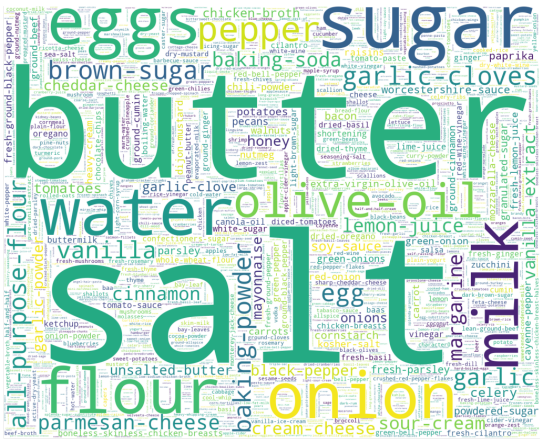
For the text description instead of using the text of the description, we added a new feature to instead capture the presence of absence of recipe description. We found that a lot of the recipes had the same default description added by default, whereas other recipes had a more distinct description related to the recipe. We also created a feature to capture the presence or absence of images in the recipe. Some of the features used for classifying the recipes are shown in Figure 1.

| | RecipeIngredientParts | RecipeInstructions | Keywords | Calories | TotalTime | PrepTime | Description | Images | FatContent | S |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | [margarine, peanut-butter, brown-sugar, sugar, vanilla, rum, milk, chocolate-chips, flour, baking-powder] | <s> mix margarine butter peanut butter </s> <s> add sugar vanillarum milk mix thoroughly well blend </s> <s> stir chocolate chip </s> <s> mix flour bake powder use unsalted butter also add 1/2 tsp salt </s> <s> add flour mix wet ingredient stir flour incorporate </s> <s> optional throw freezer 20 30 minute </s> <s> dough easy work afterwards </s> <s> drop spoonful onto grease bake sheet </s> <s> press flatten </s> <s> bake 375 8 minute </s> <s> dont overbake </s> | [cookie-brownie, -30-mins, easy] | 142.7 | 23.0 | 15.0 | 1 | 0 | 8.9 | |
| 1 | [potatoes, boneless-skinless-chicken-breasts, diced-tomatoes, tabasco-sauce] | <s> layer potato veggie chicken slow cooker </s> <s> mix tabasco tomato pour top </s> <s> cook low 8 9 hour </s> | [chicken-breast, chicken, poultry, meat, low-cholesterol, healthy, easy] | 319.4 | 485.0 | 5.0 | 1 | 0 | 2.4 | |
| 2 | [low-fat-plain-yogurt, cardamom-powder, saffron, red-chili-powder, coriander-powder, garam-masala-powder, salt, chicken-breasts, onions, fresh-ginger, green-chili, tomatoes, cilantro] | <s> mix salt cardamom powder saffron chilli powder corriander powder garam masala powder yogurt </s> <s> marinate chicken 30 minute night </s> <s> heat oil skillet </s> <s> fry onion till golden colour </s> <s> add chop ginger green chilli cook stir 1 minute </s> <s> add chop tomato mix well </s> <s> add chicken cover cook till chicken tender </s> <s> garnish fresh cilantro serve </s> | [chicken, poultry, meat, asian, indian, -60-mins, stove-top] | 439.3 | 60.0 | 30.0 | 1 | 0 | 24.2 | |

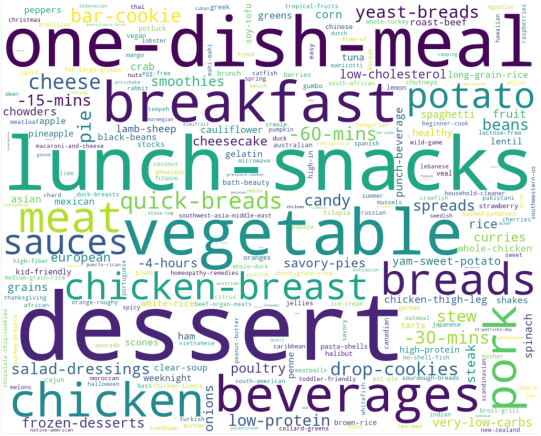
*Figure 1: Example Features Used For Classifying the Recipes.*

## 2.2 Data Exploration

To better understand the recipe dataset content, we carried out some exploration using word cloud visualization to get the most frequent ingredients in all the recipes, and the recipe categories and keywords. The visualizations from our exploration is shown in Figure 2.




a)




b)


c)

*Figure 2: Word Cloud Visualization: a) Frequency of ingredients, b) Frequency of categories, c) Frequency of keywords.*

## 3. Modeling

The modeling tasks for the recipe dataset was divided into two categories of experiments:

1) Statistical modeling using Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest classifiers by encoding the text features using the TF-IDF CountVectorizer method.
2) Neural network modeling for multi-class classification using a Doc2Vec Distributed Memory (DM) embedding model to encode the recipe ingredients and instructions text features.

Before training these models, we also analyzed the trained embedding model by projecting the recipe vectors using Tensorflow's embedding projector. These experiments are further elaborated in the following sections.

## 3.1 Embedding Model

To encode the recipe text features comprising ingredients and recipe instructions, we experimented with both word2vec and doc2vec models. Since the recipes had varying sizes, we decided to encode the entire recipe as one vector instead of combining individual word embedding vectors using averaging or summation. For the doc2vec model training, we further pre-processed the text dataset to add a document tag to identify each recipe. We then trained the model with the tokenized recipe for about 500 epochs with a vector size of 50.

To analyze the recipe embedding vectors, we projected the trained model outputs using Tensorflow's embedding projector, and used the Principal Component Analysis (PCA) for visualizing the vectors. The meta-data used for searching for sub-clusters within the full embedding cluster were the recipe names, category, instructions, and ingredients. We then used the search function within the projector to identify similar recipes to randomly picked recipes based on the cosine distance. We then isolated 100 similar recipes to the selected recipes for better visualization. We also used the search function to find any existing sub-clusters in the embedding representations of the recipes. Our findings from this exploration is shown in Figure 3 and Figure 4.

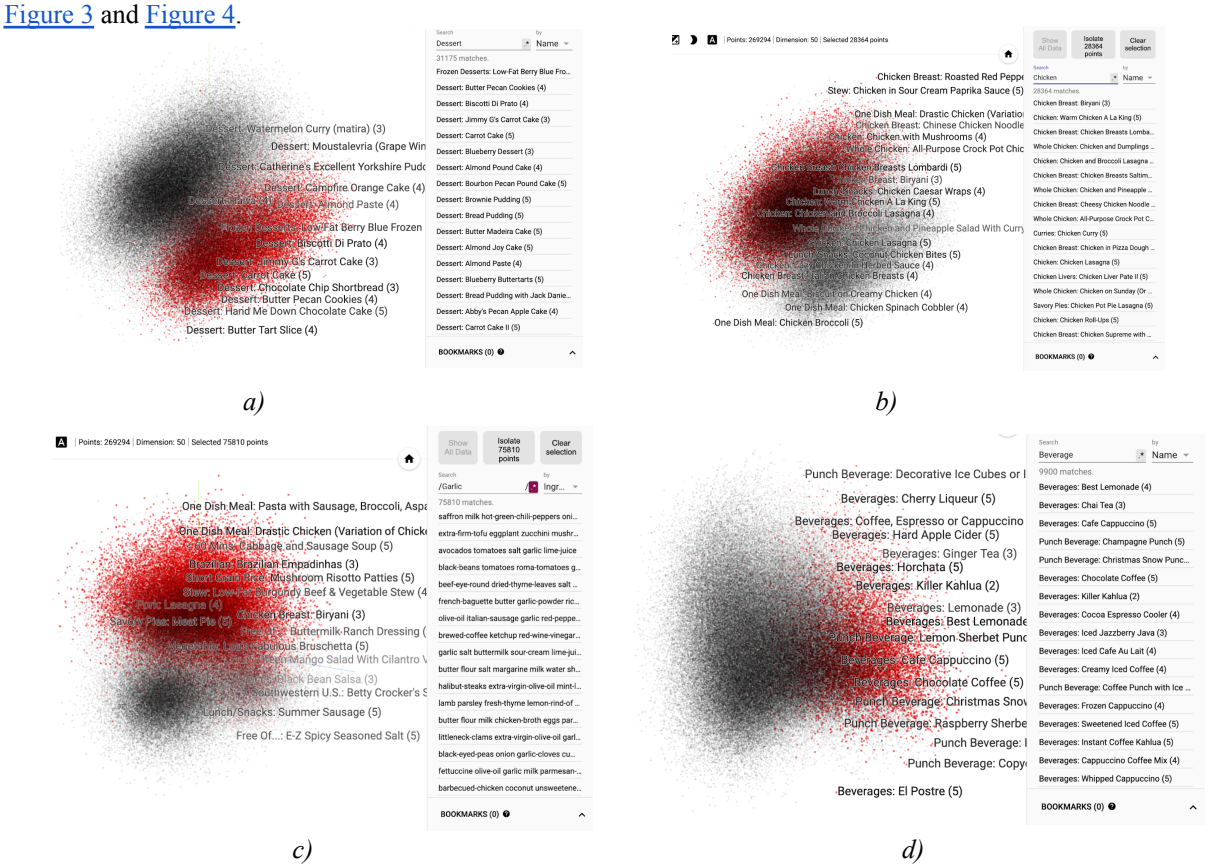



*Figure 3: Recipe Doc2vec Embedding Model Visualization: a) Dessert Recipe Cluster, b) Chicken Recipe Cluster, c) Garlic based recipe Cluster, d) Beverages Recipe Cluster*

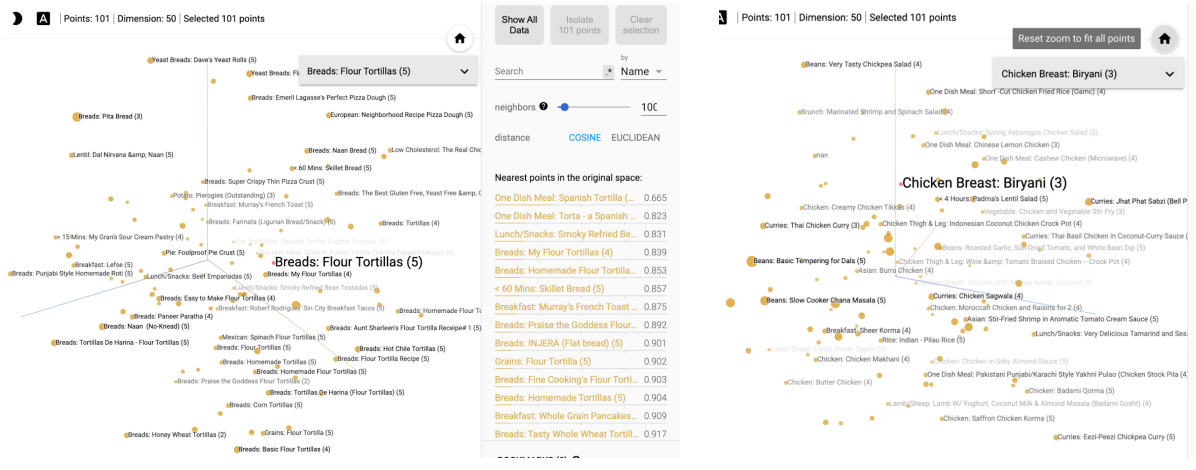



*Figure 4: Similar Recipe Embedding Vector Visualization by Cosine Distance: a) Similar recipes to Flour Tortillas, b) Similar recipes to Chicken Biryani*

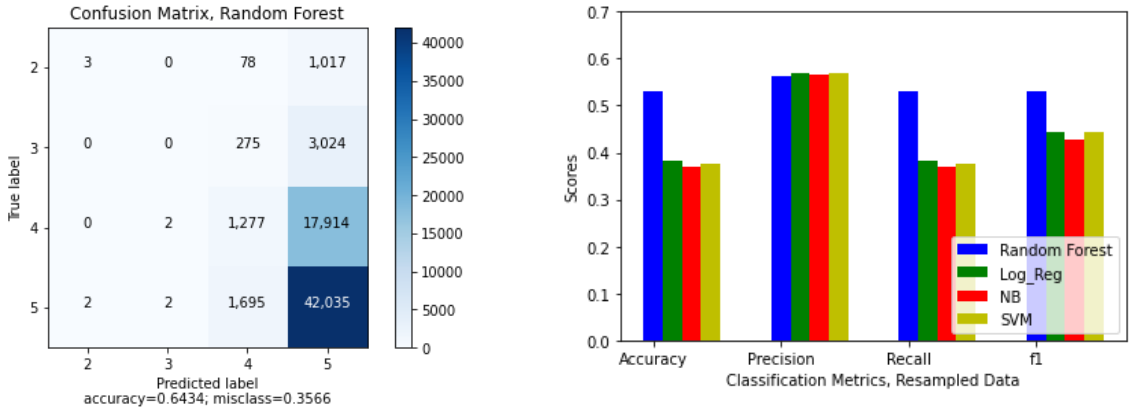
## 3.2 Statistical Models

*a. Model architecture*

We first converted each recipe's instructions, ingredients, and keywords to vector representations using the TF-IDF method. In this process, we removed words that are rare (appear in 5 documents or less) or too common (occur in more that 90% of the documents). We considered this a multiclass classification task, and ran the model through 4 common classifiers including Naive Bayes (NB), Logistic Regression, Support Vector Machine, and Random Forest. We first ran the model on text features only, then added other numerical features for comparison. Since the distribution of the target class is highly skewed, we also performed data resampling and rerun the models to compare the results.

*b. Results*

The results for models with just text features and models with both text and numerical features are very similar, hence we looked at the models with all features for completeness. Models' accuracy ranged from 0.37 (logistic regression) to 0.64 (Random Forest) (Figure 5 b). Yet, as can be seen from Figure 5 a, Random Forest classified almost all testing data as 5 (star). The model results with undersampled and oversampled data showed worse measures for all evaluation metrics. As we can see from the confusion matrix for SVM, we see more predictions for classes other than 5, than the model on original data, but also more misclassification for class 5. Overall, models are only good at predicting class 5, either because of the disproportionate presentation of this class in the training and testing data, or because our features are not distinctive among the five classes.




*a)* *b)*

*Figure 5: Training Results: a) Confusion Matrix (Random Forest), b) Performance Comparison on Resampled Data*

## 3.3 Feedforward Neural Network Model

The feedforward neural network model was trained using the pipeline shown in Figure 6. After training a Doc2Vec embedding model to get a vector encoding for each recipe, the pre-trained model weights were then used in the embedding layer added to the neural network model. The model comprised two dense layers, each followed by a dropout layer. In the final layer of the model, we adopted the CORAL method introduced by Cao et al. (2019), which stands for consistent rank logits, which can specifically handle classification tasks with ordinal labels, where there is a presence of order such as ranks and ratings as opposed to discrete labels. Using this method, the multi-class This method decomposes the multi-class classification task into binary classification tasks where the model learns to predict if the expected rating is greater than or equal to each of the individual ratings (Kennedy et al.). As the standard cross entropy loss function does not distinguish between a five-star rated recipe predicted as one star, versus a four-star, we used this classification method with the Ordinal cross-entropy loss and Ordinal softmax activation function to factor this in the loss computation.

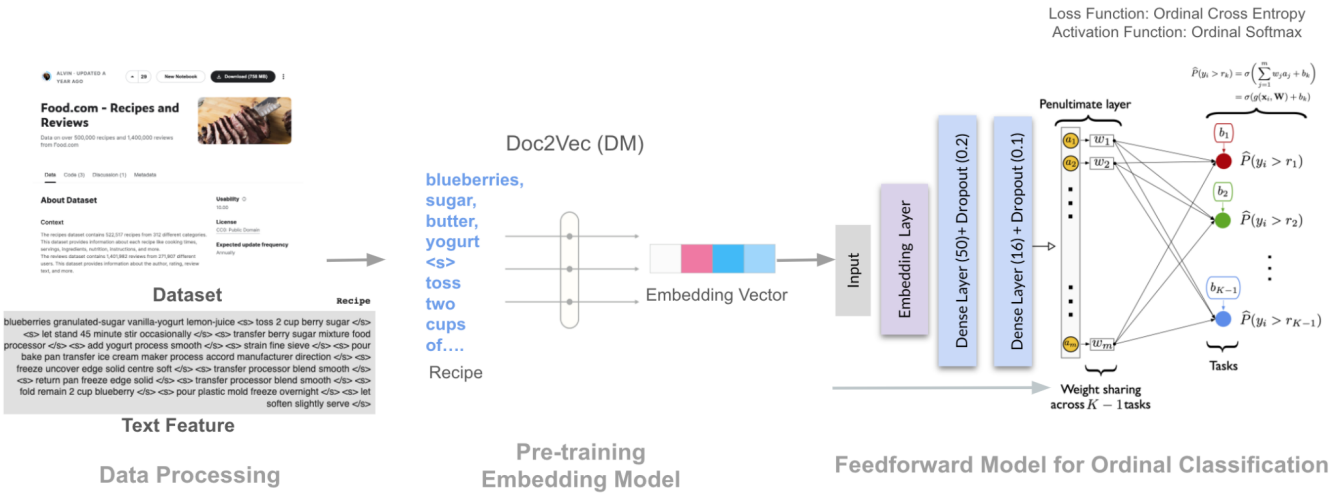
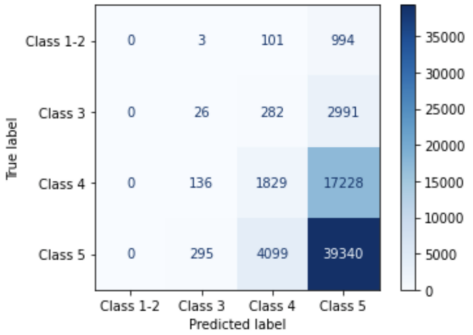



*Figure 6: Pipeline for Recipe Classification for Neural Network Model*

The best performing model was trained with a batch size of 500, learning rate of 0.05, and 500 epochs with an early stopping callback based on the validation loss. The total training dataset comprised 201970 recipes and the testing data was split 80-20 for testing and validation. Similar to the non-neural network models, the model showed a similar performance and mostly predicted a five-star rating to the recipe. The other models trained using different hyperparameters achieved a slightly better performance in terms of the accuracy. However, the resulting accuracy was due to the model predicting a five-star rating for all the recipes. The neural network model was found to be relatively better at predicting lower-rated recipes. However, the model didn't predict any of the testing dataset as one-star or two-star rating. The results are shown in Figure 7.




Mean Absolute Error: 0.46794605192799

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 2 | 0.00 | 0.00 | 0.00 | 1098 |
| class 3 | 0.06 | 0.01 | 0.01 | 3299 |
| class 4 | 0.29 | 0.10 | 0.14 | 19193 |
| class 5 | 0.65 | 0.90 | 0.75 | 43734 |
| accuracy | | | 0.61 | 67324 |
| macro avg | 0.25 | 0.25 | 0.23 | 67324 |
| weighted avg | 0.51 | 0.61 | 0.53 | 67324 |

*Figure 7: Result for Feedforward Classification Model*

## 4. Limitations and Conclusion

In conclusion, our experiments showed that our models were much better at predicting a recipe with a five-star rating than lower ratings. This could be a result of multiple factors: 1) The highly imbalanced dataset, with significantly smaller dataset for lower rated recipes, 2) Highly varying size of the text features and that may result in further difficulties to classify the recipes. Due to the imbalanced nature of available data, with a much higher proportion of positive versus negative ratings, it was thereby challenging to develop a model that can reliably predict all of the four classes, despite experimenting with upsampling and downsampling techniques to balance out the dataset. In the future, we could further experiment with the hyperparameters and model options, and try using just the ingredient features to train the embedding model and the neural network model to compare with the performance of the non-neural models. We could also explore more complex neural network architectures to see if the model performance could be improved for the same dataset.

## References

Alvin (2020). Food.com - Recipes and Reviews [Dataset]. https://www.kaggle.com/irkaal/foodcom-recipes-and-reviews

Cao, Wenzhi, Vahid Mirjalili, and Sebastian Raschka. "Rank consistent ordinal regression for neural networks with application to age estimation." *Pattern Recognition Letters* 140 (2020): 325-331.

Kennedy, Chris J., et al. "Constructing interval variables via faceted Rasch measurement and multitask deep learning: a hate speech application." *arXiv preprint arXiv:2009.10277* (2020).

Trattner, Christoph, Dominik Moesslang, and David Elsweiler. "On the predictability of the popularity of online recipes." *EPJ Data Science* 7.1 (2018): 1-39