Calorie Level Classification Task on Food.com Dataset

Hao Feng[1] , **Pai Tong**[2] , **Kaiqi Jin**[3] and **Mingqing Xu**[4]
[1]A59004117
[2]A59004757
[3]A59002544
[4]A59012910
{hafeng, patong, kajin, mix014}@ucsd.edu

Abstract

Analyzing calories based on recipes is a critical and meaningful task for many platforms. In this paper, we aim to predict the calories of recipes from Food.com dataset into three categories based on their ingredients, making techniques and analysing recipe's preparation steps. We made a detailed study on datasets, using Logistic Regression, Support Vector Machine, Random Forest Classifier, Multi-layer Perceptron Classifier, XGBoost, LSTM and LSTM with GloVe word embedding models, and optimized them by tuning and adjusting various hyperparameters and model structures. We also compared the performance between different models and different features. In conclusion, the Logistic Regression model has the best performance on techniques and ingredients features with 0.5145 F1 score, and the LSTM with GloVe word embedding model performs best on steps features with 0.50118 F1 score. Our result is very helpful for the platforms to use recipes information provided by users to create value.

1 Introduction

With the enrichment of the material life, healthful eating has gradually come to be one of the hot topics in our daily life. As food varieties and ingredient combinations grow, more and more recipes are created. For people who pay attention to healthful eating, like dieters, calories of recipes have gradually become a significant indicator for them to select a recipe. But in the fast-paced world, sifting through ingredients and making techniques or analysing recipe's preparation steps to figure out how many calories are in a recipe will be time-consuming.

To alleviate this problem, our report aims to divide the recipes of the Food.com dataset into three categories based on their calorie levels. After selecting the dataset, we initially conducted the literature research on the dataset to identify prediction tasks that can be studied and decided to predict the calorie level of those recipes in the dataset. After that, we explore the dataset and complete the visualization analysis based on the dataset. Subsequently, we predicted the calorie level of the recipes in the dataset in two cases according to the difference of the selected features, so as to explore the influence of selected features on model performance. We selected F1 score as the indicator to evaluate the model performance. First of all, based on our dataset, we comprehensively used ingredients ids and making techniques as features and implemented various basic models for training. For the basic models training task, we selected Logistic Regression algorithm, Random Forest algorithm, Multi-Layer Perceptron algorithm, XGBoost algorithm and Gradient Boosting algorithm according to the characteristics of our dataset. For feature selection work of the basic models training task, we selected techniques, 500 ingredients ids, 1000 ingredients ids, techniques+1000 ingredients ids or ingredients GloVe word vectors as features to train basic models respectively. For the basic models training task, the result demonstrates that when Logistic Regression algorithm is selected for training and techniques +1,000 ingredients IDs were used as the features, we can obtain the highest F1 score, 0.5145. For the advanced models training task, according to the characteristics of our dataset, steps are selected as the features and LSTM model was selected for training. For model selection, three advanced models including LSTM without Initial Word Embedding, LSTM with GloVe and LSTM with trainable GloVe were used for training respectively. The result demonstrates that the LSTM with GloVe word embedding model performs best on steps features with 0.50118 F1 score. The results of these two tasks indicate that when Logistic Regression algorithm is selected for training and techniques +1,000 ingredients IDs are used as the features, our model has the best performance with 0.5145 F1 score.

1.1 Description of Dataset

Our dataset comes from the paper *Personalized Recipes from Historical User Preferences* Majumder *et al.* [2019]. The dataset contains 180K+ recipes and 700K+ user comments. This paper first collected 230K+ recipes and 1M+ user comments from Food.com from 2000 to 2018 and resampled the dataset by selecting the recipes which have at least 3 steps and 4-20 ingredients. In addition, users with less than four comments were also discarded. Finally, they proposed the dataset of 180K+ recipes and 700K+ user comments. In this paper, the dataset was used to generate reasonable and personalized recipes from incomplete input specifications by leveraging historical user preferences. They combined data-to-text gen-

eration task with personalized recommendation task. Their model took the names of particular dishes, some key ingredients and calorie levels as user input. They passed these loose input specifications to an encoder-decoder framework and attended on user profiles to recommend a recipe based on the user's tastes.

1.2 Related work

As a kind of basic datasets closely related to people's daily life, recipes related datasets have been applied in various machine learning related tasks in the past.

The Allrecipes.com dataset, which is similar to the Food.com dataset, has widely been used in the past to study the health of recommended recipes. In *Exploiting Food Choice Biases for Healthier Recipe Recommendation* Elsweiler *et al.* [2017], the authors used this dataset based on the FSA health scores to replace the recommended recipes with Healthier alternatives. In this paper, they obtained the highest accuracy (84.78%) when using random forest model. Additionally, the paper named *Investigating the Healthiness of Internet-sourced Recipes* Trattner and Elsweiler [2017] uses the same dataset to measure the health of recommended recipes. Based on WHO health Scores and FSA health scores, adopting the LibRec framework, they evaluated nine algorithms(random Item Ranking, Most Popular Item Ranking (MostPop) , User- and item-based collaborative filtering (UserKNN and ItemKNN) , Bayesian personalized Ranking (BPR) , Sparse Linear Method (SLIM) , Weighted matrix factorization (WRMF) , Association Rules (AR) and Potential Dirichlet Allocation (LDA)) , and finally concluded that the accuracy of LDA and the accuracy of WRMF were higher than the accuracy of other algorithms, and the health score of randomly recommended recipes was the highest one.

In addition, several similar datasets have been used in the past to solve Food image recognition problems. These datasets include the pic2kcal dataset Ruede *et al.* [2021], the Food-101 dataset Bolanos and Radeva [2016]; Ege and Yanai [2017]; Meyers *et al.* [2015], the FCD dataset +RagusaDS dataset Aguilar *et al.* [2017], and the Food + Non-food images combination dataset Kagaya and Aizawa [2015]; Kagaya *et al.* [2014]. According to *Highly Accurate Food/ non-food Image Classification Based on a Deep Convolutional Neural Network* Kagaya and Aizawa [2015], the accuracy of food image recognition based on NIN model is the highest one (99.1%).

Similar datasets, such as QA-style datasets created based on large-scale food knowledge maps and health guidelines, have also been used to study personalized food recommendation systems that take into account dietary preferences and health factors in th past. For example, in the paper named *Personalized Food Recommendation as Constrained Question Answering over a large-scale Food Knowledge Graph* Chen *et al.* [2021], the pFoodReQ model proposed by the authors based on such a dataset is significantly superior to BAMnet model and P-MatchNN model for all evaluation indicators (MAP, MAR or F1 scores). In the paper named *Health-Aware Food Recommender System* Ge *et al.* [2015], the authors also proposed a personalized food recommendation system. They estimated the daily calories that users need based on users' personal information they collected before. Based on the formula util(u,rp) = $W_P \times$pref(u, RP) + $W_H \times$health(u, RP), a personalized recipe recommendation system taking account of users' dietary preferences and nutritional factors was proposed.

Most relevant to the Food.com dataset we used was the RecipeDB dataset. This dataset contains 118,071 recipes from AllRecipes.com and FOOD.com. In *Nutritional Profile Estimation in Cooking Recipes* Kalra *et al.* [2020], the authors used the USDA Standard Reference (USDA SR) database as a reference to calculate nutritional profiles. In the recipe database (36 calories per serving) , MAE was used as the model evaluation indicator to estimate the nutritional profiles of the recipes. The author used the Stanford Named Entity Recognition Model for training and obtained an F1 score of 0.95 on the test set validated by 5-fold cross-validation. According to the preliminary literature research, this is also the state-of-the-art method for research similar to ours.

Compared with the USDA Standard Reference (USDA SR) database used in Nutritional Profile Estimation in Cooking Recipes, due to the differences between the dataset used in this paper and that of our report, we ultimately selected calories to measure the extent to which a recipe is healthy or unhealthy on a discrete scale. We divided the recipes into three categories based on three calorie levels. Similarly, because of the differences between these two datasets, in this report, we comprehensively considered several features including ingredients ids, techniques and raw text sequence to generate predictions. Similar to the paper *Nutritional Profile Estimation in Cooking Recipes* Kalra *et al.* [2020], we also use F1 score to evaluate model performance. After trying various algorithms including Logistic Regression algorithm, Random Forest algorithm, Multi-Layer Perceptron algorithm, XGBoost algorithm and Gradient Boosting algorithm to predict the calorie level of the recipes, we found that when techniques and 1,000 ingredients ids were used as the features and the Logistic Regression algorithm was used for training, we can obtain the highest F1 score of the model, i.e. 0.5145. For advanced models training task, after trying three advanced models, including LSTM without initial word embedding, LSTM with GloVe and LSTM with trainable GloVe, to predict the calorie level of the recipes, we found that when raw text sequence was used as the feature and LSTM with GloVe model was used for training, we can achieve the highest F1 score of the model, i.e. 0.50118. Therefore, our experiments demonstrate that the highest F1 score, 0.5145, can be obtained when Logistic Regression model was used for training and techniques +1,000 ingredients IDs were selected as features. Although the performance of our model was not so good as that of the paper *Nutritional Profile Estimation in Cooking Recipes* Kalra *et al.* [2020], because of the differences of these two datasets, selected features, prediction tasks and selected models, the discrepancy between the results of the two experiments is reasonable.

2 Exploratory Analysis

Food.com Recipe and Review data from Majumder *et al.* [2019] is a dataset contains 231,637 recipe details and their

interaction with users. The interactions dataset contains 1,132,367 review data, each has user and recipe id, a rating, the date this review is written, and review content. The recipe dataset contains recipe calories and other nutrition facts, ingredients, user-given tags, detailed steps and other metadata.

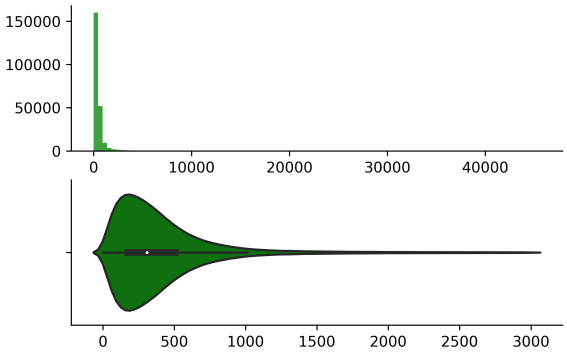


Figure 1: (a) Calories distribution of recipes; (b) Calories distribution of recipes with calories under 3,000

Calories are our major concern and predicting target in determining which meals are relatively healthy. Figure 1 (a) shows the distributions of calories of all recipes with calories less than 50,000. Among 231,637 data points, there are only 2 recipes with calories larger than 50,000. This is caused by inconsistency of units in the dataset. Most data points use kilo calories as units while some of them use calories. Figure 1 (a) also shows that calories values are centralized from 0 to 3,000, in fact, 228,486 data points (98.64%) are in this range. The detailed distribution of calories of these data points is shown in Figure 1 (b).

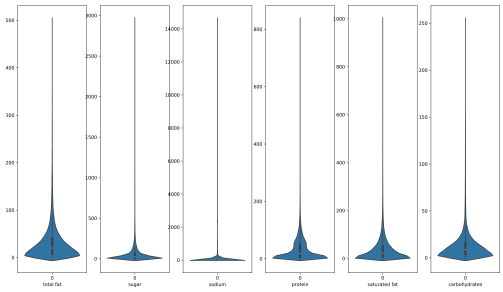


Figure 2: Distributions of six categories of nutrition of recipes with calories under 3,000

Figure 2 shows the distributions of six categories of nutrition of all recipes with calories less than 3,000. Though most data points are centralized in a small range, all categories of nutrition contain extreme values which makes all their distributions contain long tails. Also, all distributions of nutrition show a unimodal-like pattern, except for protein.

The relationships between user ratings and whether a

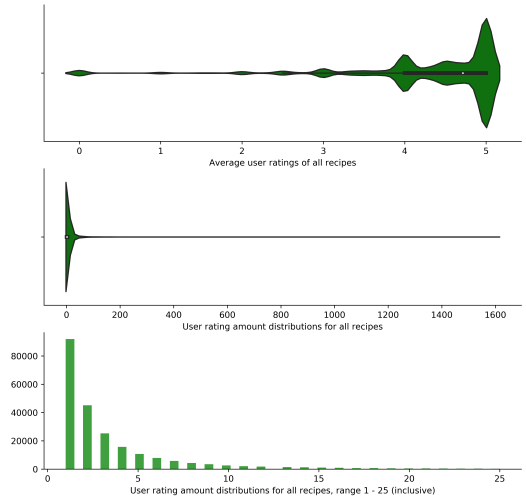


Figure 3: Distributions of average ratings for recipes

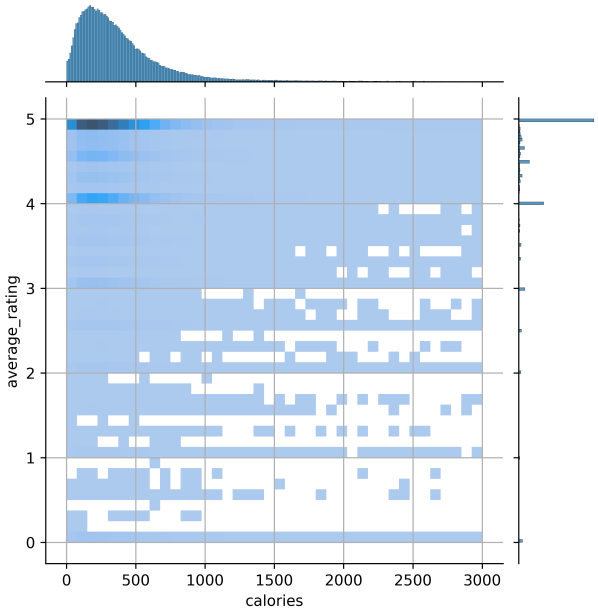
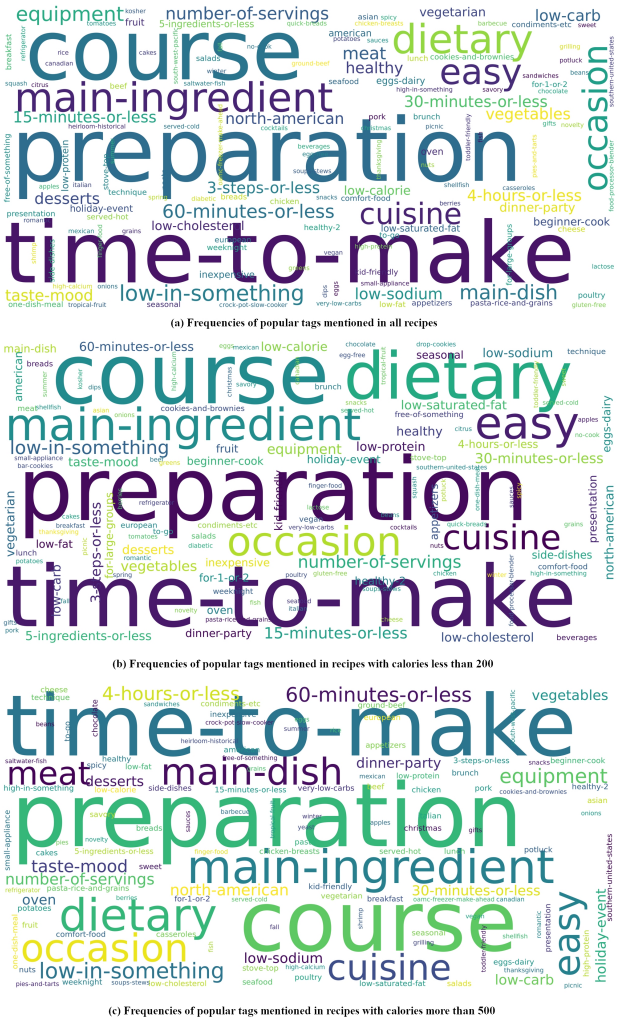


Figure 4: Joint distribution between recipe calories and the average ratings they received

recipe is healthy is also worth looking into. Figure 3 (a) shows the distributions of average ratings for each recipes. Most ratings are centralized around 4 and 5, the multiple peaks are probably because most recipes do not receive sizeable amount of ratings. Figure 3 (b) and (c) show the distributions of amount of ratings each recipe received. Though all recipes have at least one rating, most (81.49%) of them received less than or equal to 5 ratings from different users. Figure 4 shows the joint distribution between recipe calories and

the average ratings they received. Calories of recipes have different distributions under different average ratings, which makes average ratings a potential useful feature for predicting calories.



(a) Frequencies of popular tags mentioned in all recipes

(b) Frequencies of popular tags mentioned in recipes with calories less than 200

(c) Frequencies of popular tags mentioned in recipes with calories more than 500

Figure 5: Frequencies of popular tags mentioned in (a) all recipes, (b) recipes with calories less than 200, and (c) recipes with calories larger than 500

All recipes contain a user-chose tag and list of ingredients, which also give us insight about their calories and nutrition facts. Figure 5 shows the frequencies of popular tags mentioned in (a) all recipes, (b) recipes with calories less than 200, and (c) recipes with calories larger than 500. We can see while the most popular tags seem unchanged, there are still some small differences. In order to see these differences more clearly, Figure 6 shows the differences of frequencies of popular tags between recipes with calories less than 200 and those with calories more than 500. Recipes with tags "easy" and "15-minutes-or-less" tend to contain less calories, which is in line with our intuition. Also, there are some recipes have straightforward indication about their calories or nutrition facts. Similar approach is applied to see the relationship

Figure 6: The differences in frequencies of popular tags between recipes with calories less than 200 and those with calories more than 500

between ingredients and calories. Figure 7 shows the differences of frequencies of popular ingredients between recipes with calories less than 200 and those with calories more than 500. Recipes with ingredients like juice, egg and ice tend to contain less calories.

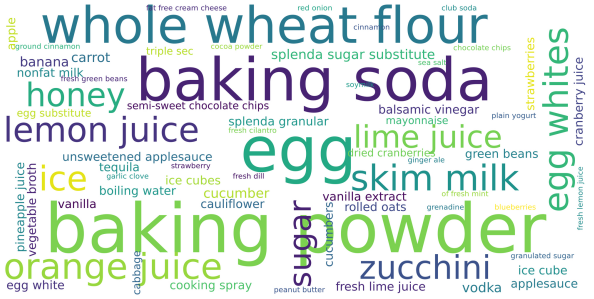


Figure 7: The differences in frequencies of popular ingredients between recipes with calories less than 200 and those with calories more than 500

3 Predictive Task

This part we would talk about the task can be studied on this dataset and give its definition. Besides we also explain methods to evaluate our model at this predictive task and assess the validity of model's predictions. Last but not least, we describe what features we will use and how to process the data.

3.1 Task Definition

In our assignment, our task is **calorie level classification task for recipes**, i.e, for every given recipe in this dataset, we need to predict its calorie level (high, medium, low). The dataset has a calories level label for each recipe with respect to its calories value, and the same standard is used in our task. So the prediction task is multi-class (three-label) classification.

3.2 Measurement Method

Since it is a multi-label classification task, accuracy measurement is not enough for this task. We adopt the F1 score to evaluate our model at this predictive task.

The formula of *precision* score is $P = \frac{TP}{TP+FP}$, and the formula of *recall* score is $R = \frac{TP}{TP+FN}$. Then the *F1 score* is

Dataset	Count	Low (calorie count)	Medium	High
Whole	178265	69699	63255	45311
Train	142612	55813	50646	36153
Test	35653	13886	12609	9158

Table 1: Experiment Dataset Split

Feature	Feature Length
Techniques	58
Ingredient:500	500
Ingredient:1000	1000
Ingredient:GloVe	300
Steps	300 * token_length (K)

Table 2: Feature Description

interpreted as a harmonic mean of the precision and recall,

$$F_1 = \frac{2PR}{P+R} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP}+\mathrm{FN})}$$

3.3 Experiment Design

We use the recipes in PP_recipes dataset and divided it into train and test dataset according to 80% - 20% ratio as shown in Tab.1.

In general, we would train our models on the train dataset, then test its validity on test dataset for F1 score measurement. If a model get a higher F1 score, then we can conclude that this model is better than other models.

3.4 Feature Selection

We are curious about which features in recipes have influence on the the calorie level. According to previous analysis of dataset, we assume these features have connections with calorie level. The features we used are displayed in Tab.2

Techniques

Techniques adopted in a recipe would affect its final calorie level. Based on the common sense of life, the fry would cause much more calorie compared with steam since the prior technique would use more oil. So there would be connection between techniques used in a recipe and its calorie level.

Followed by Majumder *et al.* [2019], we adopted the processed most frequent technologies one-hot feature from processed recipe dataset.

The process is that we manually construct a list of 58 cooking techniques from 384 cooking actions collected by Bosselut *et al.* [2018] Then we approximate technique adherence via string match between the recipe text and technique list.

Ingredients

Ingredients is another important factor would greatly influence the calorie level. For example, sugar ingredient would bring lots of calories while water would bring nearly few calories. We tried two ways to extract the feature.

The first way is to use one-hot encoding to extract feature. We calculate the frequency of each ingredient and than use a 500-dimensional binary vector indicating the presence or absence of the 500 most popular ingredients across all recipes. Besides, we also used 1000-dimensional binary vector to contain more ingredient information.

The second way is use word vector to represent the ingredient. We use pre-trained GloVe model by Pennington *et al.* [2014] to convert the ingredient word string to 300-dimensional vector. Besides, since recipe often contains more than one ingredients, the simplest way is calculate the mean of word vectors from recipes.

Steps

This feature is text feature for recipe steps, in order. This feature contains most information the model may need, since the ingredient and technique would be included in this step instruction. Besides, it also contain the orders information as well as other necessary information such as time. Now the problem change to text-classification problem.

Since steps is text, this time firstly we concated the steps into one paragraph. Then we used tokenizer in tensorflow to tokenize the text. After that we still used pre-trained GloVe model to get vector sequence. Now we got a sequence of word vectors, it could be used by time sequence model.

4 Model

In this part, we would explain and justify the models we used for multi-label calorie-level classification task. We would talk about the details of models, comparison between models, and how to optimize it in section 5.

4.1 Basic Model

Since it is a classification task, there are several basic models we can adopt to predict the calorie level.

Logistic Regression

Logistics Regression classifier is the most common model which it comes into classification task. In multi-class classification task, we calculate a separate loss for each class label and sum the resultlos [2017]. It uses cross-entropy loss to minimize,

$$-\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right)$$

In this formula, M is the number of classes and y is the binary indicator (0 or 1) if class label c is the correct classification for observation o.

Support Vector Machine

Actually, support Vector machine do not support multi-class classification naively. We could use one-to-one approach to realize it, which break multi-class classification task down into several binary classification tasks.

As Fig.8 shows, SVM could find hyperplane between every two classes and then perform the prediction.

Random Forest Classifier

Random Forest can inherently deal with multiclass datasets. Random Forest ensembles with a series of decision tree which trained by a different subset of features. Then the output would be the combination of decision tree by using simple voting.

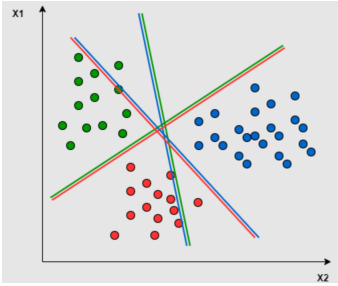


Figure 8: SVM One-To-One Strategy

Multi-layer Perceptron Classifier

Multi-layer Perceptron (MLP) as Fig.9 is a neural network supervised learning algorithm that learn the function from features to the label by using one or more hidden layers. MLP-classifier is more like a prediction model while given a set of label. So the output of MLPclassifier would be constrained to several given values. And MLP use Softmax as output function to support multi-class classification

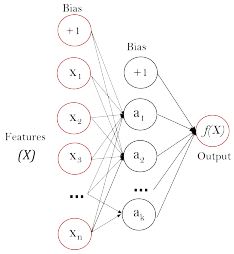


Figure 9: Multi-layer Perceptron

XGBClassifier

XGBoost is a popular and classic decision-tree-based ensemble model that uses a gradient boosting framework to predict. To perform multi-class classification task, we could use softmax output function.

Gradient Tree Boosting Classifier

In this section, we are using a Gradient Boosting Decision TreeMaklin [2019]. It is an ensemble model containing several decision trees. At each iteration, the estimator h_m for each tree is fitted to predict the negative gradients of the samples. The gradients are updated at each iteration. For multi-class classification task, K trees are built for K class at each iteration. The probability that X_i belongs to class k is modeled as a softmax value and choose the label with maximum probability as the return label.

4.2 Advanced Model

Through observation, we speculate that the steps of recipe preparation contain a wealth of food calorie information. Modeling under this assumption, we reduce this problem to a sentence-level sentiment analysis problem. The input is a short essay, which describes how a recipe is prepared step by step, and the output is the calorie level.

In this case, our input is a sequence of text. One of the common ways of analyzing sequential text is using Recurrent Neural Networks. RNN has a memory that captures what has been calculated so far, which lets this model handle sequential information well.

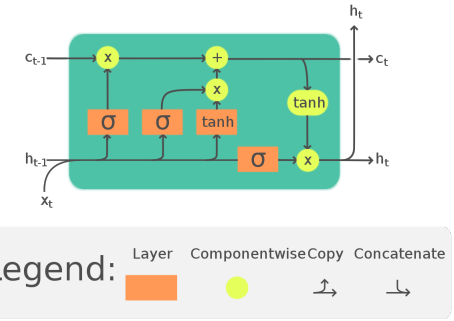


Figure 10: The Long Short-Term Memory cell can process data sequentially and keep its hidden state through time.

Therefore, we choose LSTM Hochreiter and Schmidhuber [1997]. Due to the unique design structure shown in figure 10, LSTM is suitable for processing and predicting important events with very long intervals and delays in the time series wik [2021], which is a perfect match for our task. In our LSTM model, a word embedding matrix is first initialized. In the process of processing the entire sequence, the word embedding matrix and other parameters will be continuously learned. After training, words with similar meanings often have similar vectors, and the output of the last layer is passed through a simple multi-classifier, through which we get the classification result.

5 Experiment and Result

We are unable to run experiments on all datasets each time, which runs unacceptable slow in some model (more than 24 hours for each), and we also found that when size of the dataset exceeds 50,000, it has no significant difference on the performance of the results according to table 6, therefore, we choose first 80,000 pieces of recipes as training and test data. For evaluation, we use F1 score to measure the classification effect of the model as described before.

We ran all the experiment using Python 3.6 on MAC OS 11.6. The implementations of basic models are based on scikit-learn library Pedregosa *et al.* [2011] and TensorFlow Abadi *et al.* [2016].

Besides, we use F1_score from sklearn.metrics and F1Score from tensorflow-addons.metrics to measure the validity of models,

5.1 Basic Models

Result Analysis

Tab.3 and Fig.11 demonstrates the F1 scores of various models trained before when selecting different features. As shown formerly, for our dataset, the performance of Random Forest model is exceedingly higher than that of other models except when selecting making techniques or ingredients GloVe word vectors as features. When the making techniques and 1,000

Features	LR	RF	MLP	XGB	GBDT
Techniques	0.3781	0.3636	**0.4081**	0.4079	0.3849
Ingredients_ids(500)	**0.4796**	0.3512	0.4305	0.4156	0.4312
Ingredients_ids(1000)	**0.5073**	0.3474	0.4477	0.4178	0.4336
Techniques + Ingredients_ids(1000)	**0.5145**	0.3583	0.4538	0.4308	0.4491
Ingredients GloVe word vector	0.4661	0.3842	0.4997	0.4301	**0.5023**

Table 3: F1 score result for basic models

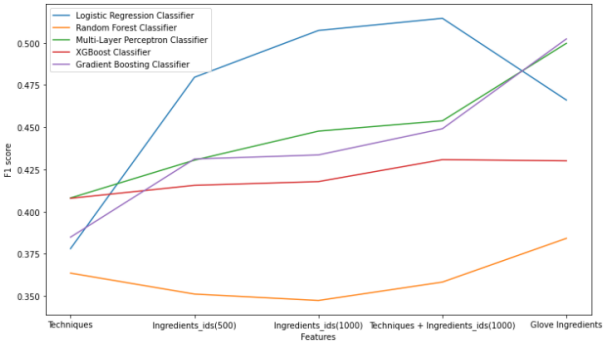


Figure 11: Model Performance Evaluation Curve

ingredients ids were used as features, the F1 score of Logistic Regression model reaches a peak of 0.5145. The performance of Random Forest model was consistently the worst one among those five models. With regard to feature selection, in terms of F1 score, it is apparent the model performance is consistently the worst one for those models trained before when the making techniques were used as features, except Random Forest model. For Logistic Regression model and XGBoost model, when the making techniques and 1,000 ingredients ids were used as features, these two models can both obtain their highest F1 scores (0.5145 and 0.4308). By contrast, when ingredients GloVe word vectors were selected as features, other three models, including Multi-Layer Perceptron model, Gradient Boosting model and Random Forest model, can all achieve their highest F1 scores (0.4997, 0.5023 and 0.3842).

Model Analysis

Logistic Regression Logistic Regression is simple, fast and powerful. Besides, it is easy to explain and implement. When the dimensional of feature is big enough, Logistic Regression Classifier could take only a little time to perform classification task.However, when the number of features is relatively small such as using technique only, its weakness that sensitive to the outlier data point would expose and get bad result.

Support Vector Machine It is worth noting that the result did not include the SVM part actually. It may have good performance but it is too slow when it comes to large scale of data. It need to take several hours to fit. It is one of its biggest weakness and we have to give up this model.

Random Forest Unfortunately, the Random Forest's result is the worst. It is because the feature we given is sparse and high dimensional. The weakness of Random Forest is that it cannot process this kind of high-dimensional sparse feature, such as Ingredientsids(500). But its strengthen is that it do not require feature selection and fast.

Multi-layer Perceptron As the basic neural network model, MLP performed well on experiments especially on word vector feature. It has wide range of application scenarios but it is slow and not easy to converge. Its learning on feature maybe not enough during training, leading to the poor performance.

Gradient Tree Boosting Classifier GBDT had good performance especially on word vector. These kinds of feature has enough information and low dimensional which is very suitable to the GBDT. However, it is slow compared with XGBoost since it does not support parallel execution.

XGBoost Classifier Theoretically, XGBoost is the advanced model compared with GBDT, but the performance on experiment was worse than GBDT. We analyse that it may because it has so many parameters and we didn't dig the full potential performance. But since it support parallel execution, it would be faster than GBDT.

5.2 Advanced Models

Experiment

Considering that the steps texts and calories level labels are in two datasets respectively, as the processed dataset is a subset of the raw dataset, we delete the complement of the processed data in the raw data to obtain the steps list and the corresponding label as the input of our model.

LSTM requires the input sequence to have the same length, so we choose the first K (as a hyperparameter) tokens of each step's texts as input. Those insufficient ones are filled with zeros at the end of the sequence. The dimension of the word vector is fixed at 300 dimensions, which conforms to the common word embedding model. Inside the model, we use bidirectional LSTM as the core structure, which propagates the input forward and backward through the LSTM layer and then concatenates the outputs. This helps LSTM to learn long-term dependencies Greff *et al.* [2016]. For activation function in hidden layers, we try ReLU and tahn respectively and add a Dense layer with 3 units entered into softmax activation. When we have multiple outputs, softmax converts outputs layers into a probability distribution Li [2019]. In order to facilitate the calculation of F1 score, we convert the label to one-hot encoding, using categorical cross-entropy as loss function and adam as optimizer is a suitable choice.

Optimization

We tune the hyperparameters on the sub-dataset (10,000 recipes) following the single variable principle, and compare different models with the average performance of five epochs.

Activation function	F1 score
ReLU	**0.42472**
tahn	0.41848

Table 4: Experiment for choosing the activation function

Fixed sequence lengths	F1 score
9 (Min)	0.37996
50	0.43816
83 (Median)	0.4116
90 (Average)	**0.44446**
120	0.42024
150	0.41326
200	0.40628
233 (Max)	0.4153

Table 5: Experiment for choosing the sequence lengths

ReLU and tahn are respectively used as the activation function in our model for experiments, results on table 4 shows that ReLU will not cause the vanishing gradient problem in our task as mentioned in some articles but have better performance, thus ReLu is the idea activation function.

Since different recipe's preparation steps have different lengths, and the input sequence length of LSTM is fixed, the next step of optimization will focus on the selection of the length K of a input sequence. With statistics of the number of words in all input sequences, we select the shortest, longest, average, median and some other possible values, and compare performance of classification with them. The results are displayed in the table 5, through which we find that the more K deviates from the average length of the training sequence, the worse the performance, and the best K is 90, the average length of all recipe's preparation steps. This makes sense, because truncation and zero padding will introduce invalid information and introduce noises, and use average length can make the best use of input sequences.

On a small number of training sets, the loss of the validation set rises very fast, while on a large number of training sets, the loss will first drop and then rise. In order to avoid over-fitting, we increased the number of training sets as shown on the table 6. When the loss is the smallest, we will get the best results.

The choice of word embedding has a significant impact on classification. A basic idea is that we do not specify the initial word vector, but learn the word vector through continuous training. However, the randomly initialized word vector does not contain any semantic characteristics between words.

To make use of semantic information between words, we

Dataset size	F1 score	loss
10,000	**0.49652**	1.0151
80,000	0.47708	**1.01072**
50,000	0.47702	1.0537
10,000	0.44446	1.18442
1,000	0.34846	1.56206

Table 6: Experiment for avoiding over-fitting

Advanced Model	F1 score
LSTM	0.47708
LSTM + GloVe	**0.50118**
LSTM + trainable GloVe	0.50048

Table 7: Experiment on advanced model

further introduce the pre-training model GloVe Pennington *et al.* [2014] as the initial word vector, which has 840B tokens, 2.2M vocabularies, and 300 dimension vectors. GloVe is a word representation tool based on count-based & overall statistics. Its effect is usually better than word2vec Mikolov *et al.* [2013], whose biggest disadvantage is that it does not make full use of all the corpus. We counted the 20 most frequently occurring words in all recipes, and displayed their GloVe word vectors on a two-dimensional plane as figure 12 through SVD algorithm. It can be found that GloVe has a great representation on words' semantic characteristics, and the Euclidean distance between words with similar meanings is smaller.

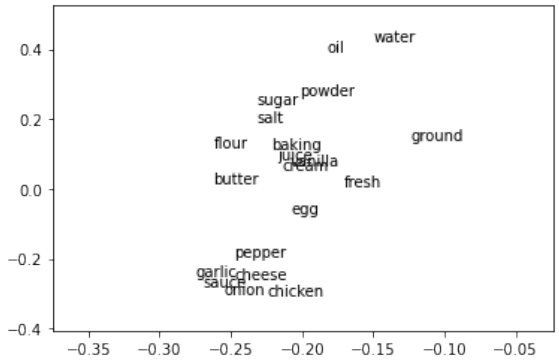


Figure 12: Top 20 word vectors on a two-dimensional plane

In order to further optimize the model, we consider that GloVe is not specifically trained on the cooking-related corpus. To make the word vectors more suitable for our dataset, we make GlovVe trainable in our model to fine-tune its word embeddings. The experiments should be carried out on these three models.

Model Analysis

After experimenting on the first 80,000 pieces of data, we have credible results on table 7 showing the adaptability of LSTM model to the text sequence feature of recipes' preparation steps.

LSTM without initial word embedding We found that only using LSTM to process the steps sequence in the data set could achieve quite a good performance outperforming Random Forest Classifier, Multi-layer Perceptron Classifier and XGBoost models, indicating that the recipes' preparation steps contain a wealth of information about recipe calories, and they could be caught by RNN.

LSTM with GloVe word embedding The LSTM model with GloVe embedding introduces additional semantic information to help prediction task become more precisely.

LSTM with trainable GloVe word embedding With the good performance brought by the introduction of GloVe, we further trained to fine-tuned GloVe. The result shows that LSTM with trainable GloVe model does not further improves the performance, though its better expression of the characteristics of food-related words.

6 Conclusion

Our task has emerged as an important tool for many Organizations and Companies as it gives useful insights into the customer base of a website which in turn helps their decision-making process Vaidya [2021].

When we have per-processed, tagged datasets, which already contain enough information as good features, we can get very good results with basic models. In particular, the Logistic Regression algorithm model that combines the two features of techniques ingredient-ids surpasses other models, which inspires us it is probably helpful to combine as many reasonable features as possible in the prediction tasks.

For advanced RNN model, the LSTM with GloVe word embedding, provides us with another way of using features. When the input of our model is a text sequence, i.e. steps, we will not need the specific ingredients of food or other per-processed, tagged, i.e. techniques, ingredient-ids to get higher precision predictions. This model has shown remarkable superiority and achieves 0.50118 F1 score, which easily outperforms many basic models run on per-processed and tagged features. Our conclusion is very useful in practices, because in common scenarios, the user's information is only a sequence of text describing a recipe.

A Core Code

The code of our work can be found in this repository.
https://github.com/Gilone/Healthy-Diet-is-All-You-Need

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

Eduardo Aguilar, Marc Bolaños, and Petia Radeva. Exploring food detection using cnns. In *International Conference on Computer Aided Systems Theory*, pages 339–347. Springer, 2017.

Marc Bolanos and Petia Radeva. Simultaneous food localization and recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3140–3145. IEEE, 2016.

Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. Simulating action dynamics with neural process networks. In *International Conference on Learning Representations*, 2018.

Yu Chen, Ananya Subburathinam, Ching-Hua Chen, and Mohammed J Zaki. Personalized food recommendation as constrained question answering over a large-scale food knowledge graph. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 544–552, 2021.

Takumi Ege and Keiji Yanai. Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 367–375, 2017.

David Elsweiler, Christoph Trattner, and Morgan Harvey. Exploiting food choice biases for healthier recipe recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 575–584, 2017.

Mouzhi Ge, Francesco Ricci, and David Massimo. Health-aware food recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 333–334, 2015.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hokuto Kagaya and Kiyoharu Aizawa. Highly accurate food/non-food image classification based on a deep convolutional neural network. In *International Conference on Image Analysis and Processing*, pages 350–357. Springer, 2015.

Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1085–1088, 2014.

Jushaan Kalra, Devansh Batra, Nirav Diwan, and Ganesh Bagler. Nutritional profile estimation in cooking recipes. In *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pages 82–87. IEEE, 2020.

Susan Li. Multi class text classification with lstm using tensorflow 2.0, Dec 2019.

2017. Accessed: 2021-12-1.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*, 2019.

C. Maklin. Gradient boosting decision tree algorithm explained, 2019. Accessed: 2021-12-1.

Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P Murphy. Im2calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1233–1241, 2015.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Robin Ruede, Verena Heusser, Lukas Frank, Alina Roitberg, Monica Haurilet, and Rainer Stiefelhagen. Multi-task learning for calorie prediction on a novel large-scale recipe dataset enriched with nutritional information. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4001–4008. IEEE, 2021.

Christoph Trattner and David Elsweiler. Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems. In *Proceedings of the 26th international conference on world wide web*, pages 489–498, 2017.

Ketan Vaidya. Sentiment analysis using lstm and glove embeddings, Jan 2021.

Long short-term memory, Nov 2021.