Predict the average recipe rating on BBC Good Food

To improve quality of BBC Good Food recipes

Group No. 6

黃珮茹 (Claire Huang)

周宛誼 (Wan Yi Chou)

施昱竹(Eva Shih)

Pornlada Ittipornpithak

Executive Summary

The primary stakeholder is BBC Good Food which is a recipe website related to BBC. The problem is that it is not the most popular website since it does not appear on the first page when user searching in Google website by keyword as "recipe website". Hence, the business goal of this project is to improve the quality of recipes on the website. By doing so, we aim to attract more people to visit the website and the analytic goal of the project is to predict the average rating value of new recipe before publishing.

The data were crawling by ourselves from the BBC Good Food website. The total numbers of data were about 8,400 records. After we had collected the data, we decided to handle the missing values by replacing them with the average value of each column. We also binned those predictors with too many categories, such as country and serving, into few categories. Besides, we derived a "total time" variable by plusing prepare time and cook time together. After the data were prepared, we processed some visualization to help us know the data better.

We picked four tools as the follows: prediction tree, KNN, multiple linear regression and Principal Component Regression to help us analyze. By comparing all the results, we chose multiple linear regression as our model which results were less error. Most importantly, we could get a specific result to show the client how the rating value does improve.

We suggest BBC Good Food that they could increase prepare time, cooking time of the dish, writing more article related to Side Dish and Starter, considering carefully when writing recipes about dinner or afternoon tea since it tends to get low rating value. We assume that readers might prefer something new and creative and choosing the dish level that is neither too hard nor too easy.

1. Problem Definition

- **Business goal**

Our main stakeholder is BBC Good Food, which is a recipe website related to BBC. Unfortunately, it is not the most popular recipe website. Our goal is to improve the quality of recipes on BBC Good Food website. We would like to make the website more attractive and to be well-known by word of mouth.

- **Analytics goal**

Our analytics goal is to predict the average rating value of each new recipe, supervised, predictive and forward-looking method and the outcome will be numerical. If the predicted rating value is close to the actual rating, we would consider it as success prediction. However, we still need some domain knowledge to define closeness.

2. Data Preparation

We used software "python" to crawl data from a recipe website, BBC Good Food. Additionally, we used software "R" to clean the data and converted to csv in order to run our models. Besides, all variables we collected were as the Table 1. in appendix A.

From cleaning processes (see Appendix A), we have 8,408 records, each record stands for a recipe, and there are 42 predictors (including all dummy variables we used) in the end (see Table.2).

Table.2 *Five Records of Dataset*

url	recipe	average_ratingvalue	kcalories	protein	carbs	fat	saturates	fibre	sugar	salt	binned_serving	step	ingredient	cooktime	preptime	total_time
http://ww	Panforte'	4.375	294	18	7	28	16	2	6	0.2	4	174	15	30	15	45
http://ww	'Butter pie	4.375	429	9	39	26	12	4	5	0.9	3	286	11	90	20	110
http://ww	'Doved' p	0	245	14	16	12	6	6	6	2.1	2	93	8	10	5	15
http://ww	10-minute	4.75	327	13	33	17	5	2	7	0.88	2	164	8	0	10	10
http://ww	10-minute	3.977275	494	37	69	10	2	4	9	2.91	2	95	9	5	5	10

cate_Dinner	cate_Main.course	cate_Side.dish	cate_Snack	cate_Brunch	cate_Treat	cate_Soup.course	cate_Afternoon.tea	cate_Dessert	cate_Vegetable.course	cate_Supper	cate_Lunch	cate_Breakfast
0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0	0	1	0

cate_Cocktails	cate_Drink	cate_Buffet	cate_Starter	cate_Condiment	cate_Canapes	cate_Pasta.course	cate_Fish.Course	country_america	country_asia	country_europe	country_other	level_Easy	level_For the keen cook	level_Moderately easy
0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

3. Data Exploration

- **Average rating value distribution**

We used histogram to see the distribution of rating number, we found out most of the values were distributed around 3~5 points. There were also couples of number at 0 point which need to be noticed. The rating value 0 may be caused by two following reasons: no one rates the recipe and the recipe is just posted on the website lately. Thus, we decided to keep the recipes which were post at least 1 month on the website (see Appendix B Figure B.1).

- **Total Using time (Cook time +Prepare time) vs. AVG Rating value**

In general, we would consider long cooking time as low rating value. After we plot the time versus rating value, graph we found that when the time got certain long enough, the value tend to be high (see Appendix B Figure B.2).

- **Ingredient vs. Average Rating Value**

We found that when the ingredient is fewer than 3.75, the rating value is centralized around 3 to 5 points. Comparing to the number larger than 3.75, people tend to give high rating value that needs few ingredients to cook (see Appendix B Figure B.3).

4. Data Mining Solution

- **Method 1- Regression Trees**

We tried running our data with prediction tree method to create model and we found that the most eight significant predictors were as follows: #kcalories, #category_drink, #category_condim, #protein, #cook time, #fibre, #sugar and #saturates. In addition, the performance was showing as Table.3 and also, the full tree was showing as (see Appendix C Figure C.1).

Table 3. Performance of Regression tree

	Total sum of squared errors	RMS error	Average Error
Training Data	5507.1109	1.4087386	7.62713E-16
Validation Data	6563.7478	1.537957903	-0.040601807
Test Data	6234.759326	1.476994012	-0.034513518

- **Method 2- KNN**

We also tried KNN method and imported all the variables to run the model, which was good at predicting continuous numerical output, and the performance was showing in the Table.4. However, in this case, we would not prefer KNN method, because it was a black box and could not provide variable selection to reduce dimensions. Also, it was not well of explaining data, and we were not able to give recommendation to our client.

Table 4. Performance of KNN

	Total sum of squared errors	RMS error	Average Error
Training Data	1.36581E-27	7.01559E-16	-1.12023E-17
Validation Data	6095.5057377	1.482086109	-0.05842848
Test Data	6031.278301	1.45262091	-0.004456721

- **Method 3- Multiple Linear Regression**

At the beginning, we imported all variables and selected "stepwise" variable selection to let XLMiner provided us suggestion (see Appendix C Table C.1) that what kind of predictors combination performed well. We choose subset#14, which CP value was mostly close to coefficient, to running the regression model again, and the performance is showing in the Table.5. Finally, we got our best model in regression:

Y=0.00289***Saturates**+0.02035***Salt**+0.00086***Total_time**+(0.10653)***Dinner**+
0.12083***Side_dish**+(0.0977)***Afternoon_tea**+(0.08404)***Supper**+0.40145***Starter**+
0.06988***America**+(0.05537)***Asia**+0.03465***Europe**+0.0933***Level_Moderately easy**

Table 5. Performance of Multiple Linear Regression

	Total sum of squared errors	RMS error	Average Error
Training Data	1008.9	0.63911	4.09245E-15
Validation Data	1176.87	0.69026	-0.00173737
Test Data	1076.17	0.65015	0.21656965

- **Method 4- Principal Component Regression (PCR)**

We constructed regression models using 2 principal components we chose by screeplot (see Appendix C Figure C.5) as independent variables, trying to get less error. Additionally, in order to

run this model, we installed a package "pls" in R (R codes in Appendix C). Finally, we got SSE = 5720.424, RMS Error=1.51, and Aver. Error=0.0314.

5. Model Evaluation

- **Compare Four Methods**

Speaking of comparing the error, the number of data is important. Here we had the same number of data and set the same seed when running all the methods. We compared these results (see Table.5), we found that regression all got the smallest numbers among these errors item. Thus, we chose regression to be our final model.

Table 6. Model Evaluation between four Methods

	Total SSE	RMS Error	Aver. Error
Trees	6563.7478	1.54	-0.041
KNN	6095.51	1.48	-0.058
Regression	1176.87	0.69	-0.0017
PCR	5720.424	1.51	0.0314

- **Residual Evaluation**

We plotted 3 histograms of different residuals: from original model (see Appendix D Figure D.1), 2) from adjusted-log Y model (see Appendix D Figure D.2), and from adjusted- log Y model but transformed to original scale (see Appendix D Figure D.3). All are central with 0 but it shows that it has no big improvement after we adjusted the model by comparing figure D.1 and D.3. Therefore, we still choose the original regression model as our final model.

6. Conclusion

- **Advantages**

BBC Good food is able to acknowledge the average rating of recipe that they will easily get before publishing. They could make adjustments to improve the quality and increasing rating. According to p-value and coefficient sign of each predictor in regression model (see Appendix D Table D.1), we could present 4 recommendations to BBC Good Food, how to adjust article to get high rating as follows: 1) increasing prepare time, cooking time of the dish; 2) writing more article related to Side Dish and Starter; 3) considering carefully when you are writing recipes about dinner or afternoon tea since it tends to get low rating value. We assume that readers might prefer something new and creative; 4) choosing the dish level that is neither too hard nor too easy.

- **Limitation**

Since BBC Good food is a recipe website from England, most recipes in our dataset are European cuisine (around 70 percent), and our model was built based on this dataset. As a result, our model is likely to work well only with European recipes. Moreover, scale of rating value was quite small, from 0 to 5, errors of prediction might not be reflected what it is in reality.

- **Operational Recommendations**

From data exploration process, we found there were different units of measures in serving predictor such as people, gram or milliliter. Thus, user should be aware of this issue before running model and manually bin it into proper range. Besides, We could not judge the error by the number, since our scales are too small (from 0 to 5). We may need some domain knowledge to help us. To know how to define the benchmark, which would be valuable for our client. Besides, for other data mining projects regarding article data in the future, we would suggest them to know their data by playing and exploring them.

Appendix A

Data Preparation

Table.1 *Variables We Collected (white rows = variables we used finally)*

variable	Type	Explanation
average_ratingvalue	Numerical	Dependent variable
url	_	Link for recipes
recipe	_	Recipe name
country	Categorical	4 regions. Europe/America/Asia/Other
comment	Numerical	Number of people comment on website
ratingnumber	Numerical	Number of people give ranking on website
kcalories	Numerical	Nutrition
protein	Numerical	Nutrition
carbs	Numerical	Nutrition
fat	Numerical	Nutrition
saturates	Numerical	Nutrition
fibre	Numerical	Nutrition
sugar	Numerical	Nutrition
salt	Numerical	Nutrition
serving	Numerical	Number of people
level	Categorical	3 levels. Easy/for the keen cook/ moderately easy
step	Numerical	How many words for recipe step.
ingredient	Numerical	Number of ingredients
cooktime	Numerical	Time for cooking
preptime	Numerical	Time for preparing
totaltime	Numerical	Derived variable(totaltime= cooktime + preptime)
datePublished	Date	The date of publishing the recipe
cookmethod	Categorical	
Category of recipe	Categorical	21 categories (each recipe has at least 1 category). Dinner/Main.course/Side.dish/Snack/Brunch/Treat/Soup. urse/Aftemoon.tea/Dessert/Vegetable.course/Supper/Lun /Breakfast/Cocktails/Drink/Buffet/Starter/Condiment/Ca pes/Pasta.course/Fish.Course

- Prepare dataset

Before we run the models, we have to prepare complete and correct dataset! For dealing with missing values (NA), we came up with different ways for diverse variables.

1. Nutrition (sugar, fat…): Delete 344 records (rows) because their nutrient contents are all NA.
2. step: Replace NA by average (165.6)
3. fibre: Replace NA by average (3.3)
4. sugar: Replace NA by average (14.6)
5. serving: Replace NA manually by url

For deleting some variables (see gray rows in Table.1), we have different reasons for diverse variables.

1. comment: we delete this variable, because we cannot get it when predicting.
2. ratingnumber: we delete this variable, because we cannot get it when predicting.
3. cookmethod: we delete this variable, because there are too many NA.
4. published_duration: not use because we cannot get this predictor when predictingAdditionally, we derived a variable, “total time” (prepare time + cook time)

Besides, we adjust the categories for country. We retained only 4 categories: “america”, “europe”, “asia”, “other” according to Table. A.1

Moreover, we binned “serving” because we think there exist same characteristics in same groups of binned serving according to the following:

[1] : 1 people(single)
[2] : 2 people(couple)
[3] : 3-10 people (family)
[4] : many people (we give the meaning that this recipe provides with elastic quantity; in other words, it can be cooked for many or few people)

Table.2 *Change 44 Countries into 4 Regions (values = number of recipes)*

Europe	italian	910	american	american	317	other	jewish	2
	english	141		cajun-creole	31		moroccan	179
	british	4062		brazilian	25		african	5
	swiss	10		chilean	1		north-african	18
	french	614		cuban	1		mediterranean	306
	greek	113		caribbean	58		australian	47
	irish	34		latin-american	14		tunisian	2
	scottish	26		mexican	225			
	german	28		balinese	3			
	hungarian	5		southern-soul	3			
	spanish	183	asia	chinese	204			
	danish	4		asian	261			
	swedish	29		korean	18			
	scandinavian	32		japanese	56			
	portugese	6		indian	372			
	belgian	3		thai	172			
	eastern-european	21		middle-eastern	197			
	austrian	1		turkish	13			

Appendix B
Data Exploration

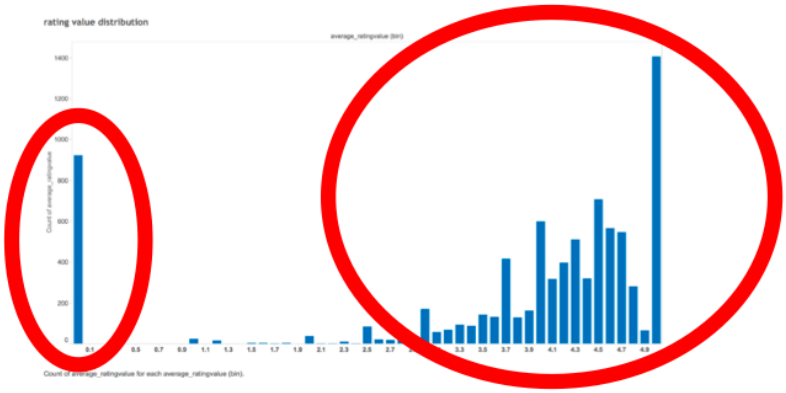

Figure B.1 *Average Rating Value Distribution*

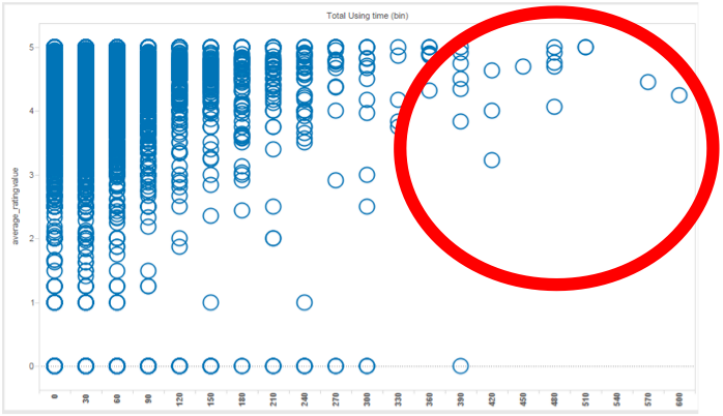

Figure B.2 *Total Using time (Cook time +Prepare time) vs. AVG Rating value*

Figure B.3 *Ingredient vs. Average Rating Value*

Appendix C

Data Mining Solution

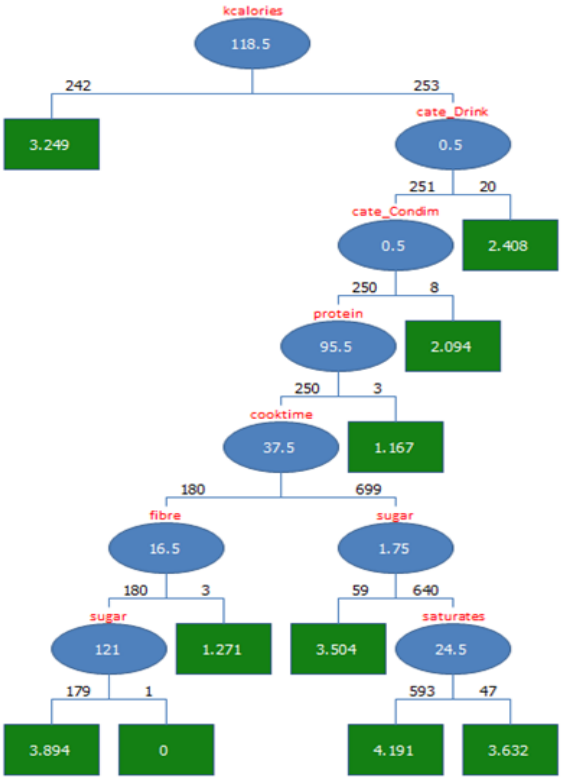

Figure C.1 *Full Tree*

Variable Selection

Subset Link	#Coeffs	RSS	Cp	R?	Adjusted R?	Probabili
Choose Subset	1	1197.5498	34552.5425	-13.5782	-13.5782	
Choose Subset	2	219.672	4324.8472	-1.6741	-1.6752	
Choose Subset	3	181.1578	3136.2362	-1.2053	-1.2071	
Choose Subset	4	123.4563	1354.4759	-0.5029	-0.5047	
Choose Subset	5	116.8644	1152.6964	-0.4226	-0.4249	
Choose Subset	6	115.8343	1122.8533	-0.4101	-0.413	
Choose Subset	7	114.7643	1091.7748	-0.3971	-0.4005	
Choose Subset	8	113.5367	1055.8242	-0.3821	-0.386	
Choose Subset	9	112.561	1027.6633	-0.3702	-0.3747	
Choose Subset	10	111.8447	1007.5194	-0.3615	-0.3665	
Choose Subset	11	111.3198	993.2928	-0.3551	-0.3606	
Choose Subset	12	110.8495	980.7547	-0.3494	-0.3554	
Choose Subset	13	109.2946	934.6865	-0.3305	-0.337	
Choose Subset	14	80.5617	48.4495	0.0193	0.0141	0.0
Choose Subset	15	80.2122	39.6447	0.0236	0.018	0.0
Choose Subset	16	79.911	32.334	0.0272	0.0213	0.0
Choose Subset	17	79.6862	27.3861	0.03	0.0236	0.0
Choose Subset	18	79.5456	25.0401	0.0317	0.025	0.1
Choose Subset	17	79.5533	23.2768	0.0316	0.0253	0.1
Choose Subset	16	79.5639	21.603	0.0314	0.0255	0.1
Choose Subset	15	79.5812	20.1407	0.0312	0.0257	0.2
Choose Subset	14	79.6591	20.5461	0.0303	0.0252	0.1

Table C.1 *"Stepwise" Variable Selection in Linear Regression*

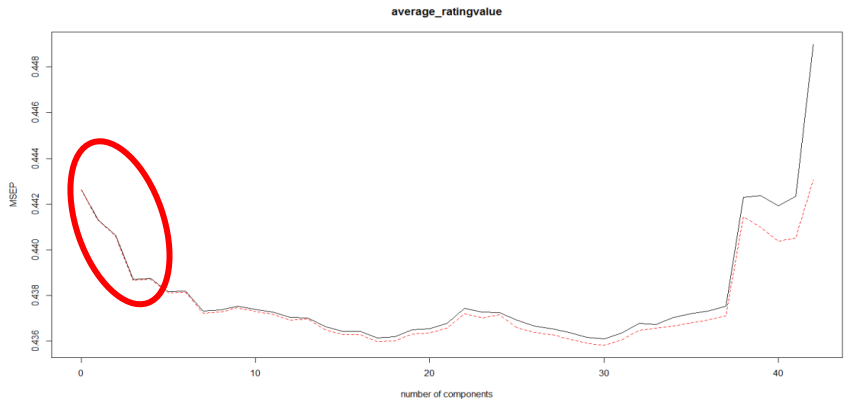

Figure C.2 *Screeplot of PCR*

- R codes for Principal Component Regression

```
##############################################################################################
recipe_remove0=read.csv('I:/BADM Final/bbc recipes without zero average rating(0107).csv',header=T)
recipe_remove0=recipe_remove0[,-c(1:3)]
###spilt data
set.seed(1234)
train=sample(1:nrow(recipe_remove0),nrow(recipe_remove0)*0.7,rep=F)
test=(1:nrow(recipe_remove0))[-train]
#####PCR#####
library (pls)
set.seed(1234)
pcr.fit=pcr(average_ratingvalue~.,data=recipe_remove0,subset=train ,scale=TRUE,validation ="CV")
validationplot(pcr.fit ,val.type="MSEP")
summary(pcr.fit)
###predict
pcr.pred=predict (pcr.fit,recipe_remove0[test,-1], ncomp =2)
sqrt(sum((pcr.pred -recipe_remove0[test ,1])^2)/length(test))#rmse=0.6608985
#########################################################################
recipe_with_0=read.csv('I:/BADM Final/bbc recipes with zero average rating.csv',header=T)
recipe_with_0=recipe_with_0[,-c(1:3)]
###spilt data
set.seed(1234)
train=sample(1:nrow(recipe_with_0),nrow(recipe_with_0)*0.7,rep=F)
test=(1:nrow(recipe_with_0))[-train]
#####PCR#####
```

```
set.seed(1234)
pcr.fit=pcr(average_ratingvalue~.,data=recipe_with_0,subset=train ,scale=TRUE,validation ="CV")
validationplot(pcr.fit ,val.type="MSEP")
summary(pcr.fit)
###predict
pcr.pred=predict (pcr.fit,recipe_with_0[test,-1], ncomp =2)
sqrt(sum((pcr.pred -recipe_with_0[test ,1])^2)/length(test))#rmse=1.506129
```

Appendix D
Model Evaluation & Conclusion

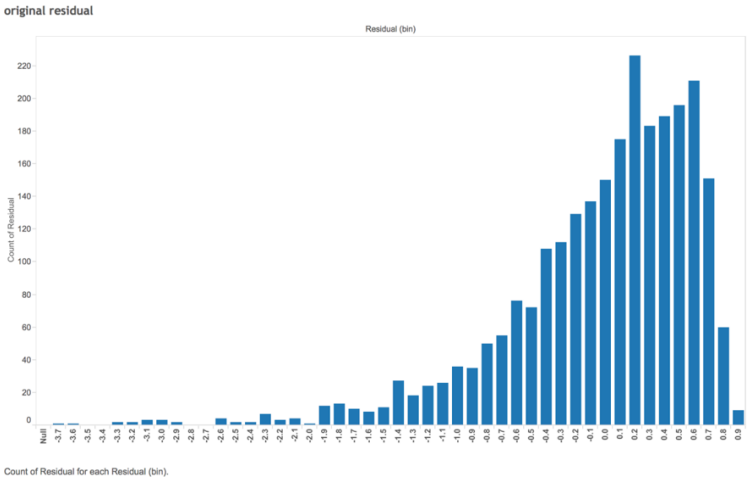

Figure D.1 *Histogram of Residuals from Original Model*

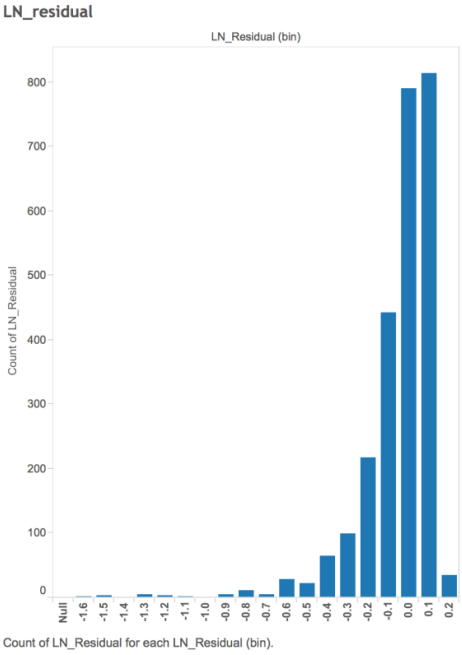

Figure D.2 *Histogram of Residuals from LogY Model*

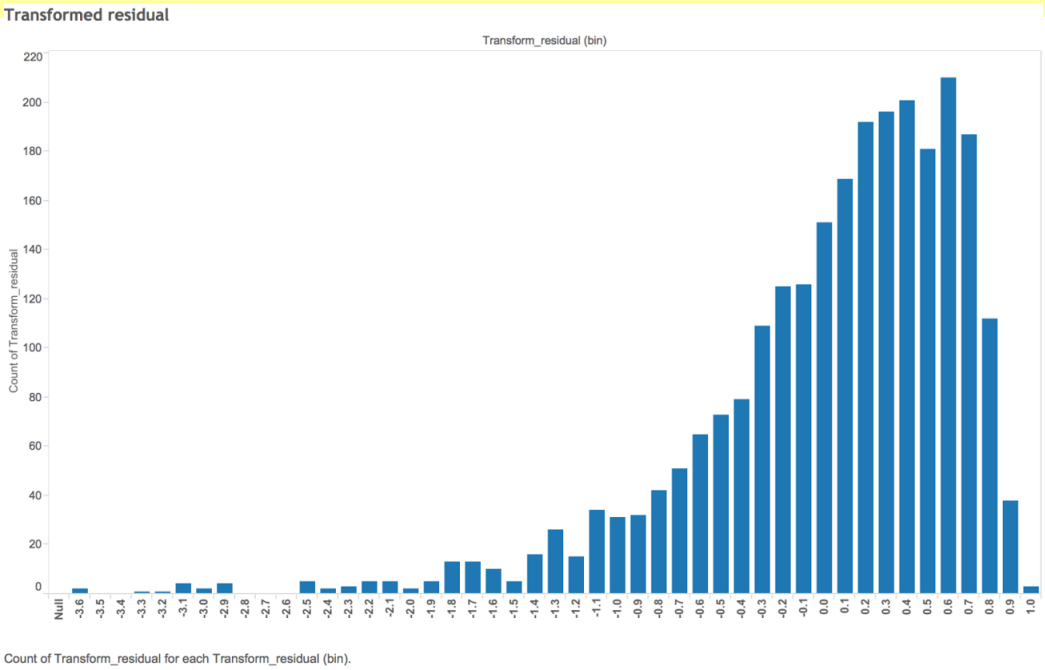

Figure D.3 *Histogram of Transformed-Residuals from LogY Model*

Regression Model

Input Variables	Coefficient	Std. Error	t-Statistic	P-Value	CI Lower	CI Upper	RSS Reduction
Intercept	4.23268	0.057102655	74.12409883	0	4.12071	4.34466	45480.6
saturates	0.00289	0.001749085	1.653735067	0.09831	-0.00054	0.00632	3.00608
salt	0.02035	0.011977373	1.699427094	0.08937	-0.00313	0.04384	0.00546
total_time	0.00086	0.000276594	3.113441923	0.00187	0.00032	0.0014	6.51486
cate_Dinner	-0.10653	0.028400099	-3.75097945	0.00018	-0.16222	-0.05084	7.90184
cate_Side.di	0.12083	0.037147827	3.252546606	0.00116	0.04798	0.19367	2.97768
cate_Aftern	-0.0977	0.045347988	-2.15443874	0.0313	-0.18662	-0.00878	1.61348
cate_Supper	-0.08404	0.032720637	-2.56850645	0.01027	-0.14821	-0.01988	2.6203
cate_Starter	0.40145	0.103865309	3.865148356	0.00011	0.19778	0.60513	6.31266
country_am	0.06988	0.068381604	1.021843048	0.30696	-0.06422	0.20397	0.41633
country_asia	-0.05537	0.060479828	-0.9155069	0.36002	-0.17397	0.06323	2.28417
country_eur	0.03465	0.052604697	0.658735035	0.51013	-0.0685	0.13781	0.18954
level_Mode	0.0933	0.040392016	2.309759909	0.02098	0.01409	0.1725	2.19066

Residual DF	2457
R?	0.03448
Adjusted R?	0.02977
Std. Error Estimate	0.6408
RSS	1008.9

Table D.1 *Summary Information of Our Best Model (Regression)*