DM&SL Project Food

Sibo Ding

Spring 2024

Estimate and Predict my Food Pattern in Austin Using Data Mining and Machine Learning

Abstract

I want to understand the driving factors of my life and food patterns during my study at The University of Texas at Austin. I use data mining and machine learning to build a real-life data science project from scratch. I get 0.76 predictive accuracy with KNN and CatBoost models.

Introduction

I started to record my everyday meals in July 2022 by a very incidental chance. Since then, I spent most of my time in Hong Kong until I moved to Austin in July 2023. From my experience in Hong Kong, I was wondering whether people tend to eat better (to relax or to compensate) or simpler (to save time) when they are busy. However, my life and food patterns in Hong Kong were too complicated and unpredictable to verify this hypothesis. Considering the feasibility, I decide to estimate and predict my life and food patterns in Austin.

I admit this estimation and prediction is not very extendable, as my life pattern is likely to change in one or two years after my study at The University of Texas at Austin. But it is still fun to know the driving factors of my life and food patterns during this time. I select this topic as it allows me to build a real-life data science project from scratch, literally mining data and building models.

When recording meals, there are potential discrepancies and biases due to my discretion. For example, if I have a brunch (at 10:00) and an afternoon tea (at 16:00), sometimes I may record them as breakfast and lunch, but I may also record them as lunch and dinner. Another discrepancy is the vague distinction between snacks and meals. If I consider 10 g popcorn as a snack, should I consider 11 g as a meal? If so, then what about 10.1 g, 10.11 g, or 10 g rice, etc.? Beyond the discrepancies, I am not very confident in the predictive accuracy for two additional reasons. First, the data set is small. Second, although my life in Austin is simple due to some constraints, the data is from a real human with certain flexibility and unpredictability.

Methods

Data mining

I keep `date` when I am in Austin: after Jul 4, 2023 (inclusive), exclude the Thanksgiving holiday (from Nov 20 to Nov 26, both inclusive) and winter vacation (from Dec 12, 2023 to Jan 11, 2024, both inclusive). The initial data looks like this:

date	dow	breakfast	lunch	dinner
7/4/2023	Tue	NA	SouthCloud Ramen	NA
7/5/2023	Wed	MA Econ orientation	MA Econ orientation	NA
7/6/2023	Thu	Home	Wendy's	NA
7/7/2023	Fri	NA	Home	China Family
7/8/2023	Sat	NA	Home	Home
7/9/2023	Sun	Home	Home	NA

During this time, I am studying at UT Austin, so my life pattern heavily depends on the school calendar. Thus, I create a categotical variable `semester`: it is *summer* when `date` is before Aug 14 (inclusive), *fall* when `date` is after Aug 15 and before Dec 11 (both inclusive), and *spring* otherwise.

For the same reason, I create a categorical variable `week_of_sem`, where the first week of a semester is 1, the second is 2, etc. Every week starts on Monday or the first day of a semester if that day is not a Monday. I set non-school days as 0, including spring break and days before or after each semester.

The variation in `breakfast` is close to zero as I eat at home most of the time. To extract useful information, I convert `breakfast` to a binary variable `breakfast_or_not`, because having breakfast may indicate going out, and its food pattern may be different from staying at home.

Sport is an important part of my life. Visiting sports facilities may signal certain life patterns, though patterns may differ between on-campus gyms and off-campus fields. I obtain my visiting records of sports facilities from UT Recreational Sports. I create a binary variable `gym_or_not`, indicating whether I visit sports facilities on a day.

I convert the data frame from wide format to long format.

I am interested in estimating and predicting my food pattern. For ease of implementation, I create a categorical variable `food_class` representing three `food` categories: *home*, *canteen* (including *J2 Dining*, *Jester City Limits*, and *Kins Dining*), and *other*.

Previous meals have impacts on the choice of the next meal. On one hand, I may get bored with previous meals (diminishing marginal return). On the other hand, I may be reluctant or constrained to change life and food patterns. Therefore, I create four

lagging variables of `food_class`. I drop the first four observations and the first four observations after winter break.

Here is the data after all processing:

food_class	food_class_l1	food_class_l2	food_class_l3	food_class_l4	meal	semester	week_of_sem	dow	breakfast_or_not	gym_or_not
other	home	other	other	other	dinner	summer	0	Fri	0	0
home	other	home	other	other	lunch	summer	0	Sat	0	0
home	home	other	home	other	dinner	summer	0	Sat	0	0
home	home	home	other	home	lunch	summer	0	Sun	1	0
other	home	home	home	other	lunch	summer	1	Mon	1	0
other	other	home	home	home	lunch	summer	1	Tue	1	0

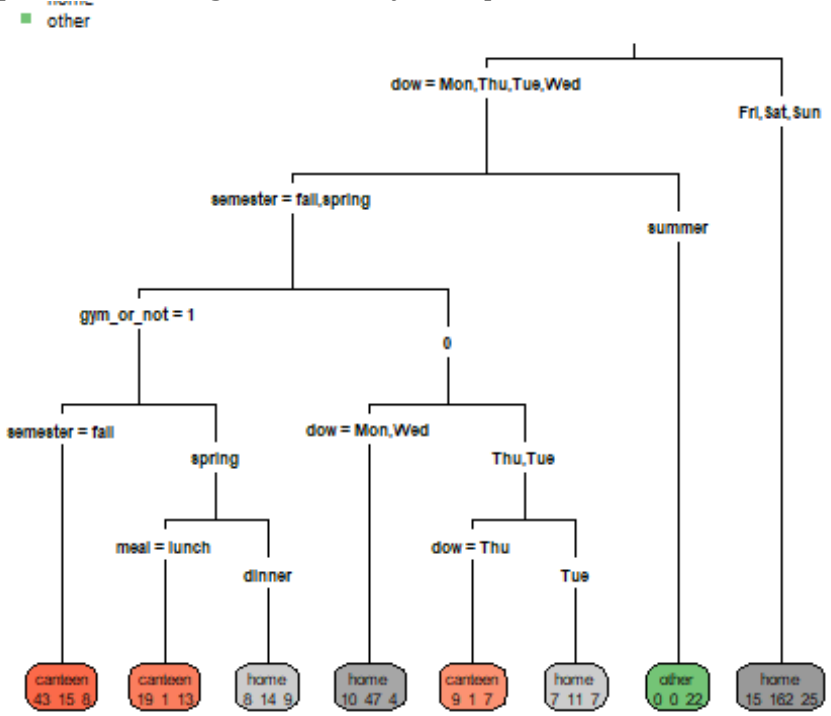
The outcome variable (y variable) `food_class` is categorical. Before any analysis, here is the number of observations in each category:

food_class	count
canteen	111
home	251
other	95

Driving factors of my food pattern

I plot a classification tree to understand the driving factors of my food pattern. For ease of interpretation, I exclude `week_of_sem` and all lags of `food_class`. There are other methods to find out important features, such as stepwise selection, variable importance plot, etc. But they are from a prediction perspective so I do not use them

to explain the driving factors of my food pattern.



In each leaf node, three numbers represent the number of *canteen*, *home*, and *other* respectively.

The first split is workdays (Mon to Thu) vs. weekends (Fri to Sun). I ate mostly at home on weekends. From an information gain perspective, the weekend node is quite pure after only one split.

The other leaf nodes are all on workdays. In the summer semester, I always ate others, because I went to school for class every day and had not perceived canteens yet.

In spring and fall semesters, going to gyms or not distinguished my life pattern. If I went to gyms in the fall semester, I would likely eat at canteens. If I did not go to gyms on Monday and Wednesday, I would probably eat at home.

The other leaf nodes are not very pure or dominant.

Predict my food pattern using classification models

I do not scale or normalize the data, as all features (x variables) are categorical.

I set 80% of the observations as training data, and 20% as test data. I fit classification models with training data, predict outcomes on test data, and compare the predicted outcomes to the actual outcomes.

I try 6 models: logistic regression, lasso, Naive Bayes, KNN, random forest, and CatBoost.

I include all features and most of their interactions in logistic regression. The reason for including interactions is, for example, a lunch on Monday may differ from one on Saturday, depending on my class schedule. For easy computation, I omit four interactions: `food_class_l4 * week_of_sem`, `semester * dow`, `week_of_sem * dow`, and `semester * week_of_sem`. I use softmax function to handle three classes in the outcome variable.

Logistic regression with too many features may result in overfitting. Thus, I use lasso to regularize the above model. I use 10-fold cross validation in the training data to find the optimal regularization parameter λ.

Naive Bayes assumes every feature is independent of all other features, conditional on the class labels of the outcome variable. This assumption contradicts the assumption of interactions in the logistic regression section above. However, it is still worth a try to fit Naive Bayes with all features.

KNN measures "distances" between features, which is not strictly appropriate for this data set with categorical features since the distances between categories are not clear. However, it is still worth a try to fit KNN with all features. I use 10-fold cross validation in the training data to find the optimal number of neighbors k.

I include all features in random forest.

CatBoost is a gradient boosting model for handling categorical features. I include all features in CatBoost. To select (tune) optimal parameters, boosting models are gradient-free and expensive to evaluate, and Bayesian optimization is a common measure. However, my computer is too old to run Bayesian optimization, so I manually tune the CatBoost model.

Results

Overall accuracies of classification models

model	overall_accuracy
Logistic regression	0.7033
Lasso regularization	0.7473
Naive Bayes	0.7033
KNN	0.7582
Random forest	0.7473
CatBoost	0.7582

Overall accuracy measures the fraction of accurate predictions among outcomes in test data. KNN and CatBoost have the highest accuracy 0.7582. The corresponding confusion matrices are in the appendix.

From the summary table in Data mining section, there are 251 *home* (54.92%). The predictive accuracy 0.7582 is improved a lot compared to guessing all observations as the mode *home*.

Conclusion

During my first-year at The University of Texas at Austin, my life and food patterns heavily depend on whether a day is workday or weekend. Also, my patterns at the summer semester were different as I just started to acclimate. Furthermore, going to gyms or not implicates my patterns.

KNN and CatBoost have 0.76 predictive accuracy. Meaning I have 3/4 confidence to know my food selection each morning. This accuracy is improved a lot compared to guessing all observations as the mode *home*.

Throughout this project, I have a better understanding of my food and life patterns. I admit this topic is not very interesting or extendable with deep business insights. However, I have a better understanding of how to explore (or mine) data to solve a real-life data science problem.

Appendix

Below are confusion matrices of predictive models. In a confusion matrix, each column is an original class, each row is a predicted class.

Logistic regression:

	canteen	home	other
canteen	15	5	5
home	5	37	2
other	2	8	12

Lasso regularization:

	canteen	home	other
canteen	14	4	5
home	4	43	3
other	4	3	11

Naive Bayes:

	canteen	home	other
canteen	16	5	4
home	2	37	4
other	4	8	11

KNN:

	canteen	home	other
canteen	17	1	8
home	3	48	7
other	2	1	4

Random forest:

	canteen	home	other
canteen	15	2	5
home	4	44	5
other	3	4	9

CatBoost:

	canteen	home	other
canteen	17	3	8
home	5	47	6
other	0	0	5