Predicting meal-choices based on menus, questionnaire and eye-tracking

Machine Learning - Report Assignment

Daan Wesselman

Student number: 85229912

Open Universiteit

1. INTRODUCTION

Restaurant owners often aim to control consumer's choices on menus by menu engineering. This is defined as organising info on menus in such a way that customers choose what menu engineers want them to choose. Although menus are often based on some form of menu engineering, empirical proof for these tactics are weak or lacking [Yang, 2012]. More empirical research would help menu engineers in their aim to make more profit, or customers to be more aware of marketing tricks that try to seduce them spending more money.

Even without menu engineering, defining the right prices is a complicated task. As explained in [Kimes et al., 2012] and [Lewis and Shoemaker, 1997], prices can be too expensive or too cheap. Typically, customers tend to choose for products that are in an acceptable range rather than the cheapest or most expensive, as illustrated in Figure 1.

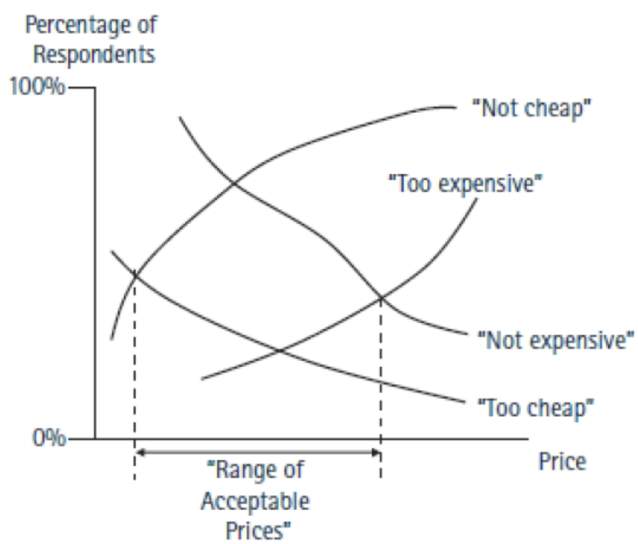

Figure 1: Example from [Kimes et al., 2012] that shows how complicated it is to define the right price.

Menu engineering can be done in several ways, which are often linked to some psychological effects [Carmin and Norkus, 1990]. For instance, the primary effect describes the phenomenon that the first piece of information (and sometimes also the last piece) is often remembered in more detail than information provided in the middle [Andersson and Nelander, 2021]. In terms of menus, this implies that it might help to position the desired meal on the top or bottom of the menu [Gallups, 1987]. Other strategies that aim to use the primary effect are to highlight certain meals [Hug, 1997] or to add attractive pictures of desired choices [Hou et al., 2017].

Other strategies are more related to the price [Kienzler and Kowalkowski, 2017]. The decoy effect describes the phenomenon that a third option affects the final choice [Ariely, 2010]. For instance, if one can choose for a small salad of 5 dollar and a large salad of 10 dollar, customers tend to choose for the small salad. However, when a middle salad of 9 dollars is added, customers tend to choose for the large salad. Another strategy is to leave out dollar or euro signs, which increases the average amount of money spent by customers.

The aforementioned strategies are often not strongly based on empirical studies. Three empirical studies that use eye-tracking can be found with variable outcomes. [Yang, 2012] describes that menu engineers use the sweet spots where customers tend to focus on. These sweet spots are typically assumed to be in the left corner, at the bottom or somewhere in the middle. However, this study concludes that the only sweet spot might be in the left corner, while other sweet spots are not found at all. Another frequently used strategy of getting the attention of customers is using colors. Unfortunately, [Smith et al., 2019] suggests this hardly affects the focus of customers. To my knowledge, [Mora et al., 2023] is the only eye-tracking study that is able to use this eye-tracking information to better predict the choices of customers by using the eye fixation duration. It is however not clear whether this is caused by menu engineering or simply by personal preferences of dishes.

For the assignment of the Machine Learning course, I will use a menu-dataset for which participants were asked to make choices for 30 different menus. During this experiment, their eye movements were registered with eye-tracking and afterward they were asked to fill in a questionnaire. I will combine and use these datasets to try to predict the choices of the participants with machine learning. To this end, I will use several methods and algorithms that we learned during the course, and additional methods that I found online.

Section two describes the goals of this assignment. Section three explains the datasets. The report continues with the methodologies in Section four. The results are shown and evaluated in Section five. The report ends with the conclusions and discussion in Section six.

2. Goal

The goals for this assignment are as follows:

- Find out how 30 different menus can be combined to obtain a general Machine Learning model that aims to predict customers choices.

- Investigate whether menu characteristics can be used to predict customers choices.
- Investigate whether the questionnaires and/or eye-tracking data improves these predictions.

3. Data analysis

3.1. Experiment description

25 participants were asked to choose a meal from each of the 30 different menus in an experimental setting, by clicking on meals with a mouse. Simultaneously, their eye-movements were registered with eye-tracking. The order in which the participants saw the menus varied to prevent the specific order to play a role in the general trends. Afterwards, participants were asked to fill in a questionnaire. For instance, they were asked whether they were hungry during the experiment.

3.2. Dataset 1: Experiment

Dataset 1 contains the combined, pre-processed data of the experiment. Per case, meaning one menu for one participant, the following steps are performed:

- The raw eye-tracking data are converted to fixations on regions of interest (ROI). These are specific locations at the menu were participants focused on (more technically: time periods with relatively little eye-movement are identified and assigned to ROIs).
- For every fixation, one instance is added to the dataset. Thus, every case consists of several instances.

The following features that will be used are in this dataset:

- Duration per fixation/instance
- Which participant
- Which menu
- which ROI (feature name is 'Name')
- Category of the ROI. This can be Menu_Item, Title, Picture, etcetera.

- Selection. This is the final choice and therefore the same for every instance per case.
- Selection_cat. This is the category of the selection, which are the same options as for the categories of the ROIs.

In total there are 61707 instances. 49177 of them have 'Menu_Item' as selection category. Sometimes however, participants clicked on price, image, logos or background.

3.3. Dataset 2: Questionnaire

All 25 participants filled in a questionnaire after the experiment. Thus, there are 25 instances. The features that will be used for this assignment are:

- Are you vegetarian?
- Were you hungry during the experiment?
- Are you trying to lose weight?

3.4. Dataset 3: Menu information

I created an additional dataset to obtain menu information. The following steps are performed:

- A list of all menu items per menu is made with the notebook 'Create_info_menus.ipynb'. This is done by finding the unique objects in the 'Name' and 'selection' features of dataset 1. This list is exported to an Excel file.
- This Excel file is manually completed by remaining menu items that do not occur in dataset 1 but are nevertheless on the menu.
- All prices (if available) and number of calories (if available) are added as new feature for all menu items.
- In addition, also the position on the menu is added as new feature. '1' means on top of the menu, '2' means second highest, etcetera.

In total, there are 345 instances (or menu items), which means menus contain on average 11.5 menu items. Menu 1 and 2 are shown in Figures 2 and 3 respectively.

Figure 2: Menu 1

Figure 3: Menu 2

4. Methodology and implementation

4.1. Feature engineering

The machine learning models will be trained in three different rounds. First, only menu information is taken into account. Next, three features from the questionnaire are added. Finally, eye-tracking data is added in two different ways.

4.1.1. Menu information

Make aggregated dataset First, a new aggregated dataset is created that will be used for model training. It is aggregated, because every case is reduced to only one instance. Initially, it contains the features that shows the participant and menu, and the target variable (both selection and selection_cat).

Menu 4 is removed from the aggregated dataset, since it is judged as too complicated. In addition, some typo's are corrected manually to make sure that the menu items will be added to this dataset succesfully. Finally, the instances that do not have 'Menu_Item' as selection_cat are removed, since they are hard to interpret.

Convert target to (string) numbers To develop a general model that aims to predict menu choices, independent of the specific meals on the menus, the menus must be somehow combined in the dataset. Otherwise, the target would consists of approximately 300 labels where only a small part exists on one specific menu. To this end, the target 'selection' is converted to a random number. For instance, if a menu consists of 10 meals, they get a random number between 1 and 10. Since the menu with the largest number of meals consists of 26 meals, 26 is the highest target number. These numbers are saved as object instead of integers, to make sure that the target is categorical instead of numerical.

This approach means that cases also have features that in reality do not exist. For instance, for a feature concerning meal 10, when there are only 8 meals on a menu. How I deal with this will be explained below.

Add menu information Firstly, price information is added. I chose not to use the absolute value of the price, assuming that price is especially important in the context of other prices on the menu. Instead, prices are converted to categories, which are 'Most expensive', 'Second most expensive', 'Middle', 'Second cheapest' and 'Cheapest'. In addition, they get label 'Unknown' if prices are not indicated. Finally, they get label 'No_meal' when a particular target number is larger than the number of meals on the menu. Thus, with this approach 26 features are added: Price_cat_dish01, Price_cat_dish02, etcetera.

The same approach is done with the number of calories or weight. Since this is indicated only a few times I made less categories, which are 'Most Calories', 'Not most calories', 'Unknown' and 'No_meal'.

In contrast, the position on the menu is added as numerical value, since this is uniform for all menus. '1' stands for the position on top of the menu, etcetera.

Convert target to dummy features Dummy encoding is applied to the target. This means that the target consists of 26 columns, which are 'target_01', 'target_02', etcetera.

Thus, the choice of which machine learning methods to use is restricted to methods that can handle multi-column targets.

Features: Dummy encoding versus target encoding For converting the features into numerical values, two approaches are investigated. Firstly, dummy encoding is applied. Although this works fine for features with few labels, it gets more tricky when there are many features with several labels. For instance, Price_cat_dish01 will be split in Price_cat_dish_01_Cheapest, Price_cat_dish_01_Middle, Price_cat_dish_01_MostExpensive, Price_cat_dish_01_SecondCheapest, Price_cat_dish_01_SecondMostExpensive, Price_cat_dish_01_Unkno and Price_cat_dish_01_NoMeal. This would lead to 26*7 = 182 features for price only, risking the curse of dimensionality.

An alternative is to use target encoding. Here, the categorical features are transformed to numerical values, based on correlations between the feature-labels and the target. In this way, the numerical values have meaning and dummy encoding is no longer required. Target encoding is a popular method for features with many labels, however, it is more complicated for multi-column targets. As explained here, target encoding should be done for each target-column. Thus, having 26 features for price and 26 target-columns, this would lead to 26*26 target-encoded price features, much more than for dummy encoding.

The solution might be to assume that the feature Price_cat_dish01 only influences target-column target_01. In that case, Price_cat_dish01 is target-encoded based on target_01, Price_cat_dish02 on target_02, etcetera. This leads to 26 price features, much less than for dummy encoding.

Random forest modelling shows for menu-info only (without questionnaire and eye-tracking data) that the latter target-encoding method gives slightly better results than dummy encoding, while the number of features is much lower. Hence, this approach is used for all models described in this assignment.

4.1.2. Questionnaire

For the second round of modelling, the three feature from the questionnaire that are mentioned before are added. These are labeled 0 or 1, hence only three features are added.

4.1.3. Eye-tracking

For the third round of modelling, eye-tracking data is added. This is (apart from each other) done in two different ways.

Number of fixations Firstly, the number of fixations per ROI is added as feature. In other words, when a participant looked 5 times at a certain meal, the number 5 is added to this instance. More in detail, 26 features are added to the dataset for all target labels.

Fixation duration Secondly, the same approach is used for the total duration that participants looked at the menu items. Similarly, 26 features are added to the dataset.

4.2. Machine Learning methods

4.2.1. Splitting data in train and test

I use the Scikit learn module in Python to train the models. Initially, data was split in test and train data with the normal train-test module. This led to the promising result that models were able to predict the right choice of the participants roughly two times better than with random guessing, using only menu information.

However, this is not only caused by menu information but also by potential similarities between participants. Lets assume that for menu X, participant A is in the train set and likes hot wings. Participant B is in the test set and also likes hot wings. Consequently, the model might correctly predict the choice of participant B, based on participant A and not based on menu characteristics. Hence, this approach might not be able to find the predicting power of the menu characteristics.

Instead, data is split per menu. For instance, menus 1-29 are in the training set and menu 30 is the test set, to investigate whether choices for menu 30 can be predicted without any data of menu 30 being in the test set. This approach is repeated for every menu in the test set.

4.2.2. Predict versus predict_proba

For this assignment, the 'predict_proba' method is used, which generates probabilities instead of hard predictions. This is necessary, because with too many target columns, the method 'predict' would only give zeros since for all targets, probabilities will be closer to zero than to one.

As a side mark, results show very low probabilities, which are close to zero, for target numbers that do not exist on the specific menu, i.e. where feature have labels 'No_Meal'. This suggests that my approach in how to combine 30 different menus into 1 dataset works.

4.2.3. Used algorithms

Two Machine Learning methods are used, which are a Random Forest Classifier and a Multi-Layer Perceptron classifier. Both methods can handle multi-column targets and are able to generate output with 'predict_proba'. The hyperparameters of the Random Forest are not changed, thus, default settings are used. For the MLP classifier, only the parameter 'max_iter' is adapted due to convergence warnings with the initial value of 200. To this end, the value is set to 1000 or even 2000 to prevent these warnings.

4.3. Validation metrics

The resulting probabilities from 'predict_proba' are used by following these steps:

- Apply the model for all menus in the test test (29 runs).
- Repeat this 20 times with different random states (20*29 = 580 runs).
- For every run, take the predicted probability of the menu item that was chosen.
- Use this array to test its significant against other results, for instance, another model or with versus without eye-tracking.

When I compare results against random guessing, the mean probability of guessing correctly (corrected for the number of menu items in all menus) is used. This is 10.8%.

5. Evaluation and results

The results are summarized in Table 1.

Random guess	Model	Prediction	Relative improvement
10.76%	Menu, RF	11.41%	6.04%
10.76%	Menu, MLP	12.74%	18.40%
10.76%	Menu+Q, RF	11.02%	2.42%
10.76%	Menu+Q, MLP	13.45%	25.00%
10.76%	Menu+Q+EY_v1, RF	15.50%	44.54%
10.76%	Menu+Q+EY_v1, MLP	19.59%	82.06%
10.76%	Menu+Q+EY_v2, RF	10.99%	2.14%
10.76%	Menu+Q+EY_v2, MLP	8.78%	-18.04%

Table 1: Mean predicted probabilities for all situations, both for random forests (RF) and multi-layer perceptron (MLP). Q means questionnaire, EY_v1 is eye-tracking data regarding number of fixations, EY_v2 is eye-tracking data concerning fixation duration. All outcomes are in the last columns compared to random guessing.

5.1. Menu information

The mean predicted probability of the choices is increased with 6.04% with menu information only compared to random guessing, using RF. This is significant ($p=0.026$ or 0.013 for two-sided and one-sided comparison). By using MLP, it is increased with 18.40% ($p<0.001$). The different results for both methods are visualised in Figure 4.

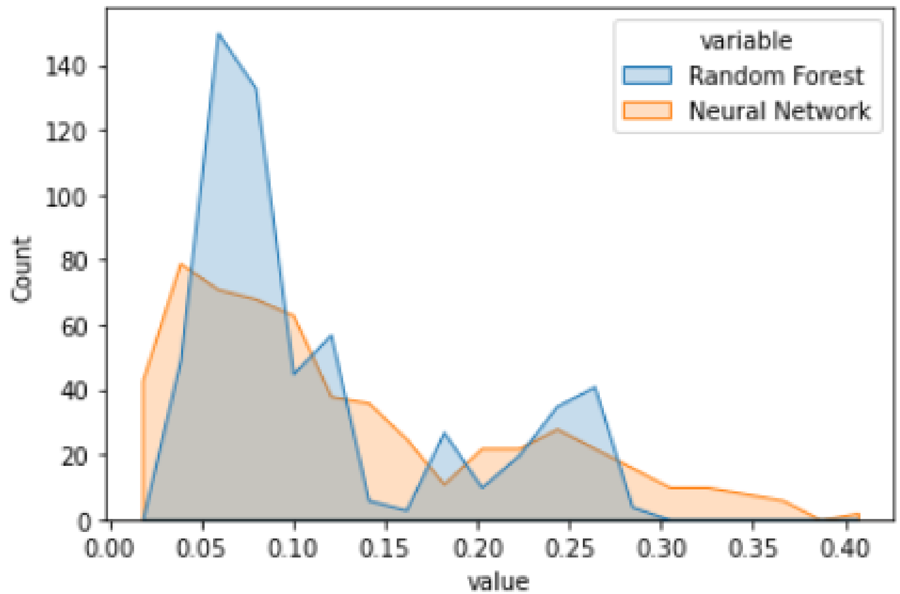

Figure 4: Histograms of RF and MLP predictions, using menu information only.

5.2. Questionnaire

For RF, the prediction becomes worse when the questionnaire data is added, compared to only using menu information. For MLP however, the mean predicted probability of the choices increases with 25.00%, compared to random guessing. Adding the questionnaire info significantly improves the prediction of the MLP, compared to only using menu information (p=0.002 or 0.001 for two-sided and one-sided comparison). These two conditions are visualised in Figure 5.

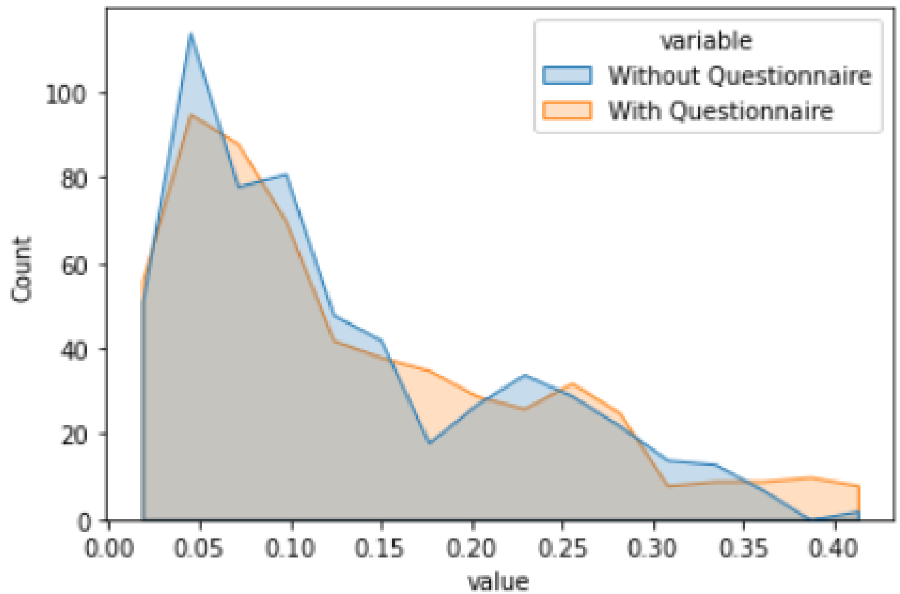

Figure 5: Histograms of MLP predictions, using menu information only and using menu information and questionnaire info.

5.3. Eye tracking

When eye-tracking is added in terms of how often is looked at each menu item (number of fixations), the predictions significantly improve when using RF with 44.54%, compared to random guessing. In addition, it is a significant improvement compared to only using menu and questionnaire info ($p<0.001$). This is even better for the MLP method. Here, the mean predicted probability of the choices is 19.59%. This is 82.06% better than random guessing and again significantly better than only using menu and questionnaire info ($p<0.001$). This is visualised in Figure 6.

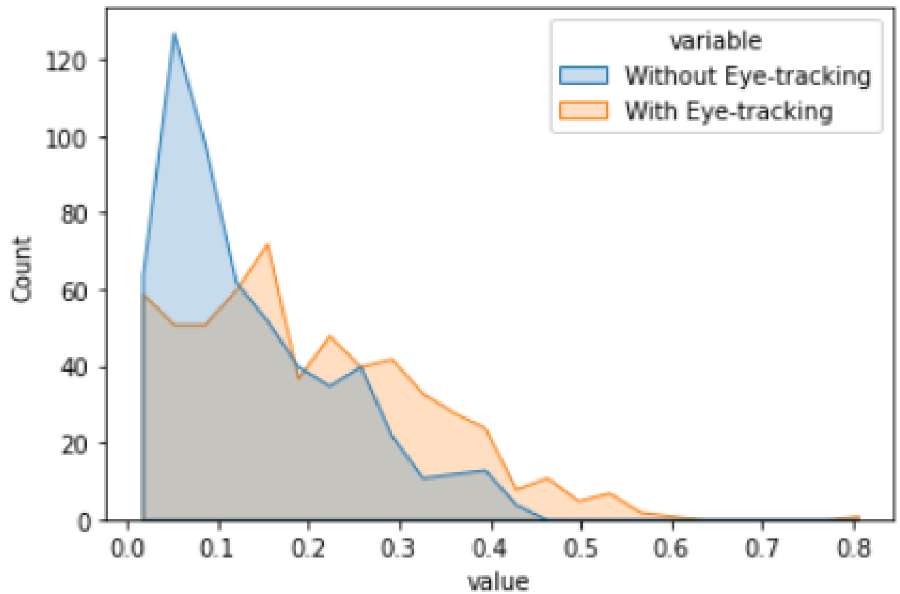

Figure 6: Histograms of MLP predictions, using menu and questionnaire information, versus using information from menu, questionnaire and eye-tracking (number of fixations).

Using eye-tracking in the form of how long is looked at each menu item (fixation duration) performs much worse, both for RF and MLP. For MLP, results are even worse than random guessing. Most likely, this does not mean that this feature has no predictive power. More likely, technical problems with the methodology cause this result. For instance, it might help to normalize this features since magnitudes of the numbers are much larger than the other features. This is a topic for further research.

6. Conclusions and Discussion

6.1. Model performance

In general, the MLP model performs better than the RF model. This suggests that the MLP model is better capable than the RF model to deal with the large number of features, relative to the number of instances. However, only default settings of the hyperparameters are used. A next step would be to fine-tune both models and investigate whether the optimal MLP model is still better than the optimal RF model. Since random forests require relative little tuning, my expectation is that the MLP model remains better than the RF model.

Some specific difficulties regarding the menus decrease the performance of the model. For instance, several menus contain sub-choices of one more general menu item. For instance, the surendra's curry in menu 15 consists of three sub-choices (Figure 7). Some participants clicked on the general menu item, which is in fact no real choice. These types of problems probably negatively affects the results.

Figure 7: Snapshot of menu 15.

6.2. Interpretation of the results

The results show that menu information, such as the prices and positions on the menu, significantly helps to predict the choices of the participants. This is in line with the studies of [Andersson and Nelander, 2021], [Gallups, 1987], [Hug, 1997], [Kienzler and Kowalkowski, 2017] and [Hou et al., 2017]. The contribution of this work is that empirical proves on this topic are scarce.

Eye-tracking (in the form of number of fixations) strongly increases the predictive power, which is in line with [Mora et al., 2023]. However, this information alone does not provide a full explanation. It remains unclear why participants looked more often at menu items. This can be caused by menu characteristics, or simply because they liked certain menu items more than others, or a combination of both.

For both menu information and eye-tracking, I suggest a different set-up for future research. To this end, I would use the same menu-items in different settings. For instance, I would create 10-20 different menus with the same items, but with different prices, positioning, etcetera. In this way, eye-tracking can be used to investigate whether participants look at items that they like most, or that this is affected by menu characteristics as well.

6.3. Ethical aspects

Knowledge of the effects of menu engineering, or marketing tricks in general, on customers buying behaviour is useful for both marketeers and customers. Companies probably often perform research on how to influence customers. Therefore, it is important that this type of research is also performed in scientific, open source environments. In this way, the knowledge is not only available to companies but also to customers. This might aid in their awareness of marketing tricks and prevent them for spending too much money.

Data and notebooks can be found at Github.

References

[Andersson and Nelander, 2021] Andersson, O. and Nelander, L. (2021). Nudge the lunch: A field experiment testing menu-primacy effects on lunch choices. *Games*, 12(2). 1, 15

[Ariely, 2010] Ariely, D. (2010). Predictably irrational: The hidden forces that shape our decisions. 2

[Carmin and Norkus, 1990] Carmin, J. and Norkus, G. (1990). Pricing strategies for menus: Magic or myth? *The Cornell Hotel and Restaurant Administration Quarterly*, 31(3):44–50. 1

[Gallups, 1987] Gallups (1987). Through the eyes of the customer. *The Gallup Monthly Report on Eating Out*, 7(3):1–9. 1, 15

[Hou et al., 2017] Hou, Y., Yang, W., and Sun, Y. (2017). Do pictures help? the effects of pictures and food names on menu evaluations. *International Journal on Hospitality Management*, 60:94–103. 1, 15

[Hug, 1997] Hug, R. (1997). Menu planning and merchandising. *McCutchan Pub Corp*. 1, 15

[Kienzler and Kowalkowski, 2017] Kienzler, M. and Kowalkowski, C. (2017). Pricing strategy: A review of 22 years of marketing research. *Journal of Business Research*, 78:101–110. 2, 15

[Kimes et al., 2012] Kimes, S., Phillips, R., and Summa, L. (2012). Pricing in restaurants. (754). 1

[Lewis and Shoemaker, 1997] Lewis, R. and Shoemaker, S. (1997). Price-sensitivity measurement: A tool for the hospitality industry. *Cornell Hospitality Quarterly*, 38(2). 1

[Mora et al., 2023] Mora, M., Romeo-Arroyo, E., Chaya, C., Gayoso, L., Larranagga-Ayastuy, E., and Vazquez-Araujo, L. (2023). Eating with the eyes? tracking food choice in restaurant's menu. *Food Quality and Preference*, 110. 2, 15

[Smith et al., 2019] Smith, J., Guliuzo, J., Benedict, J., and Chaparro, B. (2019). An ey-tracking analysis of a restaurant menu. *Proceedings of the human factors and ergonomics society*, pages 1522–1526. 2

[Yang, 2012] Yang, S. (2012). Eye movement on restaurant menus: A revisitation on gaze motion and consumer scanpaths. *International Journal on Hospitality Management*, 31:1021–1029. 1, 2