

Predicting Food Preparation time

Introduction

Food preparation time is a well-known issue for restaurants and one of the critical parts of a food delivery service. Among other aspects, assuming that the restaurant doesn't deliver its own orders, the food delivery platform needs that information to:

1. inform the customer upfront so they know how long they'll have to wait for the food and
2. decide when to dispatch the courier to pick-up the food.

Poor prediction makes the customer dissatisfied because of late delivery, makes either the courier arrive early at the restaurant and having to wait for the food to get ready or arrive late and picking-up a cold food. Worse than that, cold food may either have to be heated or thrown in the trash, generating loss to the restaurant and increasing delivery time to the customer.

In general, restaurants estimate food preparation time according to the cook's experience or simply averaging historical data. Unfortunately, these predictions are inaccurate because there are many scenarios and variables to consider. For that reason, one might use statistical or machine learning tools to obtain models that perform better than the restaurant guesses.

Data

We will provide a synthetic training dataset based on historical data from one restaurant. The participants will evaluate their approaches on the training set as they wish. Then, they will submit us their source code so we can run it on an evaluation environment prepared for the task. There are two CSV files as described next.

MenuItems.csv: contains the description of the menu items prepared by the restaurant. An actual order contains one or more menu items.

FoodPrepTimes.csv: contains historical food preparation times of the order prepared by the restaurant. Each row is an order placed to the restaurant. Each column is an independent variable (a feature) and the last column (FPT, food preparation time) is the dependent variable (the response) in minutes that has to be estimated. An order can have a maximum of ten menu items; when a dish has less than ten menu items, the remaining features are filled with NaN.

Evaluation of the technique

We evaluate the model's performance with the following metrics: R^2 , Mean Absolute error (MAE) and Root Mean Squared Error (RMSE). Our current baseline model has the following performance for the training set on a ten-fold cross-validation approach.

- R^2 : 0.6100377380269022
- MAE: 5.448831425639197
- RMSE: 6.978090339927755

Computation resources are unconstrained.

Final ranking

The techniques will be ranked according to the three metrics. MAE and RMSE must be minimized while R^2 must be maximized. The technique that has the best values wins. For instance, in the example below, the green cells have the best values and Team 1 is the winner. In case of ties, we will use Occam's razor and choose the simplest approach.

Team	R ²	MAE	RMSE
Team 1	0.62	5.44	6.96
Team 2	0.63	5.45	6.96
Team 3	0.64	5.45	6.97

Prizes

Grand Prize: \$300