Project Proposal

Kevin He, Anubhav Jaidka

1 Overview

In this project, we are predicting the time it takes to complete an online food delivery using historic data collected from an online application. The data contains information on the driver, road and weather conditions, and details about the delivery. We will work hands-on with this data to clean, impute, and train and test the model using continuous optimization methods.

2 Motivation

Our project is applied as we will be exploring a range of continuous optimization methods on our dataset.

The food delivery business is a rapidly growing sector with most restaurants offering delivery services through various food delivery applications. The estimated delivery time in the app provides a promise to the customer of when the delivery will be completed and estimates the time it takes for the order to be prepared and delivered. Ensuring this estimate is correct is important as hungry customers expect the delivery to be completed by the estimated time and it may create a poor experience if the estimate is inaccurate. This problem is interesting as we can learn how how companies including Uber Eats, Door Dash, and FanTuan use continuous optimization techniques to calculate the delivery time estimate.

3 Data

We are using a food delivery data set from Kaggle. There is work needed to clean and impute the data as it contains many null values. We will also need to perform feature engineering to better extract information about the data for our model.

Below is the link attached to the website.
https://www.kaggle.com/code/gauravmalik26/food-delivery/data

4 Methods

Since the target variable is the time taken for an online food delivery to be completed, we will use a range of regression models. We will attempt to fit the data with linear and quadratic regression with and without L1 and L2 regularization, random forests, and K-NN regression. This is a variety of models from fitting linear lines to creating forests of decision trees.

We plan to use Python for this project in a Jupyter Notebook and create a continuous optimization pipeline for the data. The models will be trained using pre-existing open-source libraries. We also plan to use a train/validate/test split of our data to use for parameter turning for the parametric models and to evaluate model performance.