Assignment 1

Exercise 1 *(Ride-sharing Matching Example).*

1. ***Generate Data:*** *(3pts) Simulate the positions of n riders and n drivers as random points in the unit square $[0,1]^2$. Use $n = 10$, $n = 30$, and $n = 50$. Compute the pairwise Euclidean distances between all riders and drivers. You can use* `scipy.spatial.distance.cdist`.

2. ***Implement Matching Methods:*** *(7pts) Implement a code for Greedy Matching, Optimal Matching, and Random Matching as discussed in class. You can use the Hungarian algorithm* `scipy.optimize.linear_sum_assignment` *to minimize total distance.*

3. ***Simulation:*** *(5pts) For $n = 10$, $n = 30$, $n = 50$, run 1000 simulations. In each simulation, generate new positions for the riders and drivers, and calculate the total distance for all three methods.*

   Plot the distribution of total distances for each method and n. Print the average total distance for each method.

4. ***Network Effects:*** *(5pts) Analyze whether the average optimal matching distance increases by a factor of 3 when moving from $n = 10$ to $n = 30$. Compare this to the random matching. Why might the scaling differ between these methods?*

5. ***Endogeneity of*** n *(5pts). Now consider that n (the number of riders and drivers) is endogenous, meaning the platform can decide to wait and collect more participants before performing the matching.*

   What is the trade-off from increasing n?

   Model this trade-off and introduce a new objective for the platform that accounts for this trade-off.

Exercise 2 *(Quality Selection) Consider the quality selection example we discussed in class and assume a platform is offering two menus of price-quality pairs: Menu 1: $\{(p_L, q_L) = (1.5, 1), (p_H, q_H) = (4, 2)\}$ and Menu 2: $\{(p_H, q_H) = (4, 2)\}$.*

Each buyer has price-quality sensitivity $1 \leq v < \infty$ that is distributed according to a density function $f(v)$. A buyer with sensitivity v has utility $vq - p$ from price-quality pair (p, q) and 0 from not buying (you can think that $(p_0, q_0) = (0, 0)$ is another price-quality possible pair). Buyers choose the price-quality pair that maximize their utility.

The platform's revenue from Menu 1 is $p_L D_{L,1} + p_H D_{H,1}$ where $D_{L,1}, D_{H,1}$ are the mass of buyers choosing L and H under Menu 1, respectively. The platform's revenue from Menu 2 is $p_H D_{H,2}$ where $D_{H,2}$ is the mass of buyers choosing H under Menu 2.

*1. **Compute Demand.** (10pts) Consider two different density functions $f(m)$:*

$$f_1(v) = 0.5 \cdot v^{-1.5}, \quad f_2(v) = 3 \cdot v^{-4}.$$

Plot these functions for $v \in [1, 3]$. For each menu, compute the demand for each price-quality pair under both f_1 and f_2.

*2. **Revenue.** (10pts) Calculate and compare the platform's revenue for each menu under both density functions.*

Why the revenue comparison differs between f_1 and f_2? explain how it relates to the curvature of the plots you found in part (1).

*3. **Different Demand Model.** (5pts) Based on our class discussions, why is the demand model in this exercise more suitable for ridesharing where quality tiers include GrabCar Premium and JustGrab (as discussed in class) compared to a labor platform like Upwork, where quality distinctions refer to Top Rated sellers and standard sellers?*

Exercise 3 *(Surge Pricing Schemes). (10 pts)*

*1. **Trip Length:** (5pts) The length of a trip (e.g., in miles) is modeled as an exponential random variable:*

$$f(x; \lambda) = \lambda e^{-\lambda x}, \quad x \geq 0,$$

where x is the trip length, $\lambda > 0$ is the rate parameter and f is the PDF. Generate random trip lengths in Python sampled from an exponential distribution with a mean of 10 miles (i.e., $\lambda = 0.1$).

Plot a histogram of the simulated trip lengths to visualize the distribution. Explain this modeling assumption in terms of distribution of trips.

*2. **Pricing Schemes:** (12 pts). In additive surge pricing, the payout for a trip is calculated as:*

$$\textit{Payout} = \textit{Base Fare} + (\textit{Surge Multiplier} - 1) \times \textit{Surge Bonus},$$

where: Base Fare $= c + r \cdot x$, c is a fixed fee, r is the per-unit cost, and x is the trip length. The *surge bonus is a fixed amount added during surge periods.*

In multiplicative surge pricing, the payout for a trip is calculated as:

$$\textit{Payout} = \textit{Base Fare} \times \textit{Surge Multiplier}.$$

Implement both pricing schemes in Python. Generate simulated data with 1,000 trips. Use the following parameters: Fixed Fee (c): 2.50*. Per-Unit Cost (r):* 0.75*. Surge Multipliers: Randomly sampled from {1.0, 1.5, 2.0, 2.5}, with probabilities {0.5, 0.3, 0.15, 0.05}. Surge Bonus:* 5.00*.*

Plot histograms of payouts under both pricing schemes and compare their distributions. In particular, calculate the mean and variance of payouts for each pricing scheme. Which pricing scheme has higher variance in payouts? Why?

3. ***Driver Strategic Behavior.*** *(8 pts). Answer the following questions:*

a) In which pricing scheme drivers are more likely to ***cherry-pick*** *rides (i.e., selectively accept longer trips)? Explain based on the from part (2).*

b) Given your findings, why do you think Uber transitioned from multiplicative to additive surge pricing? Explain.

Exercise 4 *(Spatial Pricing).*

Consider a simplified ride-sharing platform with two zones (Zone A and Zone B) as discussed in class.

1. ***Optimize Prices.*** *(10 pts) Assume the supply is constant in each zone and the demand in each zone is a linear function of the price. In particular, The demand and supply details are as follows:*

$$\textit{Demand in Zone A: } D_A(p) = \max(0, 100 - 10p), \quad S_A = 45$$

$$\textit{Demand in Zone B: } D_B(p) = \max(0, 50 - 8p), \quad S_B = 35$$

Write a Python program to implement the profit function for a ride-sharing platform. The profit is computed as the total revenue across all zones, factoring in price, demand, and supply constraints.

Compute the optimal spatial prices (p_A for Zone A and p_B for Zone B) that maximize profit.

Compute the optimal uniform price (p) that maximizes profit when the same price is applied in both zones.

Plot the profit under uniform pricing as a function of the price.

*2. **Estimate Demand and Optimize Prices**. (10 pts) In practice, we don't know the demand function but have a dataset that includes historical observations of demand and feature vectors for each zone. We decided to employ a two-step "estimate and optimize" approach:*

***Estimate:** We use a machine learning model to predict demand in each zone based on the given feature vectors.*

***Optimization.** Using the estimated demand functions, find the optimal prices that maximize total profit.*

The first step is done below and employs the RandomForest algorithm to model and estimate the demand.

Use

```
data = pd.read_csv
```

to read the

```
    historical_data_with_demand_v2
```

from the Assignment 1 files.

a) Explain this estimation step (if you didn't take any machine learning class consult with ChatGPT).

```
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from scipy.optimize import minimize
import matplotlib.pyplot as plt



features_A = ["price_A", "market_sentiment", "zone_specific_A", "competitor_prices_A"]
```

```
features_B = ["price_B", "market_sentiment", "zone_specific_B", "competitor_prices_B"]


X_A = data_sample[features_A]
y_A = data_sample["demand_A"]


X_B = data_sample[features_B]
y_B = data_sample["demand_B"]


# Train-test split
X_train_A, X_test_A, y_train_A, y_test_A = train_test_split(X_A, y_A, test_size=0.2,
    ↪ random_state=42)
X_train_B, X_test_B, y_train_B, y_test_B = train_test_split(X_B, y_B, test_size=0.2,
    ↪ random_state=42)


# Random Forest Regressor
rf_A = RandomForestRegressor(n_estimators=100, random_state=42)
rf_B = RandomForestRegressor(n_estimators=100, random_state=42)


rf_A.fit(X_train_A, y_train_A)
rf_B.fit(X_train_B, y_train_B)
```

b) Using the estimated demand functions from part (a), find the optimal prices that maximize total profit. Assume supply is fixed: $S_A = 45$ and $S_B = 45$ and that the current features are

$$\texttt{market\_sentiment = 1.0}, \quad \texttt{zone\_specific\_A = 2.1}, \quad \texttt{zone\_specific\_B = 1.0}$$

$$\texttt{competitor\_prices\_A = 4.0}, \quad \texttt{competitor\_prices\_B = 6.0}$$

Compute the optimal prices for:

- ***Spatial Pricing:** Separate prices for Zone A (p_A) and Zone B (p_B).*
- ***Uniform Pricing:** A single price p for both zones.*

Plot the profit under uniform pricing as a function of p and print the optimal prices (x-axis is

p and y-axis profits).

3. ***Challenges.*** *(5 pts) Provide (in words) three challenges to scale the approach of previous part to many zones.*