CPSC 5310 – Machine Learning
Seattle University – Winter 2026 Quarter

PROJECT PROBLEM STATEMENT

DeepDispatch

Photo by Stig Husby on Unsplash

Prepared by:

Ben Tran, Rizvan Ahmed Rafsan

DeepDispatch

OPTIMIZING TAXI FLEET ALLOCATION VIA INTELLIGENT DEMAND PREDICTION

1. Team Members:

1. Ben Tran
2. Rizvan Ahmed Rafsan

2. Selected Dataset

NYC Yellow Taxi Trip Data (On Kaggle)

3. Problem statement

We are trying to address the inefficiency of taxi fleet allocation caused by the "spatial mismatch" between supply (drivers) and demand (passengers). Drivers often circle empty streets, wasting fuel because they rely on intuition rather than data to find passengers. Specifically, drivers and fleet operators struggle with **revenue loss** (missed opportunities during demand surges), **increased operational costs** (wasted fuel and possible wear-and-tear from "dead mileage" without passengers). To resolve this, we propose DeepDispatch: an end-to-end intelligent dispatch system.

> *The goal: Replace reactive driver intuition with a spatiotemporal model of future demand density to optimize fleet repositioning and enable a utilization-maximizing dispatch strategy 1 hour in advance.*

As illustrated in the system architecture in Figure 1, the model ingests historical trip data to generate a "Future Demand Heatmap." This output is fed into a dispatch logic algorithm, which sends proactive repositioning alerts to a driver's mobile app, allowing them to move to high-demand zones before the surge occurs.

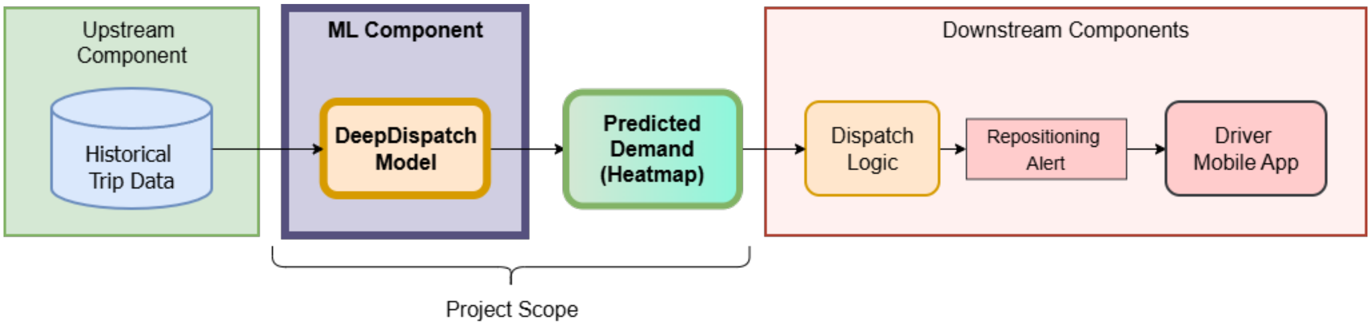

Figure 1: System architecture of DeepDispatch: an end-to-end intelligent dispatch system

Machine Learning Pipeline

To achieve this, we will implement a four-stage data processing and modeling pipeline. This transforms raw GPS logs into actionable predictions:

1. **Clustering (Unsupervised):** We will first group the pickup coordinates into functional "Demand Clusters" (for example, specific neighborhoods or transit hubs) using an appropriate clustering algorithm.

2. **Temporal Aggregation & Sequence Generation:** Once regions are defined, we will aggregate trip counts per region into hourly time buckets. We will then transform this data into time-series sequences (for example, demand at $t - 24\ h$, $t - 23\ h$...) to capture daily and weekly seasonality patterns.
3. **Deep Learning Forecasting (Supervised):** We will treat these sequences as a forecasting problem. We intend to employ Sequence Models (such as LSTMs or Transformers) to predict the exact number of pickups for the next hour ($t + 1$) based on historical trends.
4. **Evaluation & Baseline Comparison:** The model's performance will be evaluated against a baseline (e.g., a simple moving average) using metrics like RMSE or MAE to quantify the reduction in prediction error and theoretical revenue increase.

4. First Look at the Data

Our analysis is driven by the NYC Yellow Taxi Trip Data as mentioned above, comprising approximately 47 million trip records. Each row represents a single taxi journey, capturing the complete lifecycle of a ride from passenger pickup to drop-off. The data is characterized by high granularity and high volume, making it an ideal candidate for deep learning applications but also presenting significant challenges regarding *memory management* and *noise*.

Possible Feature Usage

Input Features

1. Spatial Features: `pickup_latitude` & `pickup_longitude`: Since these are coordinates, we will use them to generate Region IDs to define our demand zones.
2. Temporal Features (to be derived from `tpep_pickup_datetime`): Converted into cyclical features to capture rush hour, weekend patterns, etc.
3. Sequence Features: To get a “Lagged Demand” we will determine the number of pickups in the zone for the previous $t - 1, t - 2, t - 3, \ldots\ t - 24$ hours.

Target Variable (Derived)

- `demand_volume`: We want to aggregate the raw trip information to calculate the total count of pickups per "Cluster Zone" per hour. Using the `pickup_latitude` & `pickup_longitude` we will use a clustering algorithm to assign every trip a Region ID. Then grouping the data by Region ID and `tpep_pickup_datetime` to determine the volume.
- `revenue_potential`: Instead of simply counting trip volume, we can also aggregate the sum of total_amount per cluster per hour. This will allow the model to predict the financial value of a zone rather than just the activity level. Which will prioritize high-value long-distance trips over short trips.

5. Preparation Plan

a. **Data Preprocessing:** From an initial exploration, the dataset seems to have a significant amount of **outliers/noise**. So, informed data cleaning might be necessary. Also, the features available in the dataset need to be transformed before data modeling stage. So, extensive preprocessing will be required before working with the dataset and creating the **pipeline**. Additionally, raw timestamps (`tpep_pickup_datetime`) needs to be converted into datetime objects to extract cyclical features before the modeling stage.
b. **Exploratory Analysis (EDA):** We will conduct visual analysis to confirm the **spatiotemporal** nature of the task. This might include plotting **Time-Series Decomposition graphs** to verify

daily and weekly seasonality (for example, rush hour peaks) and generating Geospatial Heatmaps to visualize high-density pickup zones.

c. **Spatial Discretization:** Since the dataset uses GPS coordinates rather than fixed zones, we will use a clustering algorithm to group these points into functional "Demand Regions." This effectively discretizes the spatial data, allowing us to treat the problem as a time-series forecast for specific areas. We are planning to explore different clustering algorithms (staring with K-means or K-means++) to find the appropriate clustering algorithm for this task.

d. **Baseline Definition:** Before building complex Deep Learning models, we want to establish a baseline. This is an important step in the pipeline to measure whether our advanced models actually provide any added value.

6. Challenges

- The dataset contains around 47 million of rows. Processing this requires efficient code and memory optimization, or it will be extremely slow and computationally expensive.
- The raw data is very noisy. Without aggressive outlier removal, models will try to learn from GPS errors, leading to poor generalization. So, extensive data cleaning and preprocessing will be required.
- GPS coordinates are continuous, so we must transform them into meaningful spatial regions using clustering before analysis.
- For exploratory data analysis, using the full dataset will be slow, so we are planning to use a random sample from the dataset.
- Taxi demand has multiple overlapping seasonalities (hourly, daily, weekly). Capturing all of these (e.g., the "Friday night rush" vs. "Monday morning rush") is difficult.