Taxi Price Prediction

Kaggle Competition — Step-by-Step Submission Guide

https://www.kaggle.com/t/3e8bc07027f24cf6b4f44d54291e293f

Metric: MAE (lower = better)

About this competition: The Taxi Price Prediction challenge on Kaggle asks participants to predict taxi fare amounts using trip features such as pickup/drop-off coordinates, passenger count, and datetime. It is an excellent end-to-end regression exercise for all skill levels. Submissions are scored using Mean Absolute Error (MAE) — the lower your score, the higher you rank on the public leaderboard. Multiple submissions per day are allowed. You work in your own modeling tool, then submit a simple two-column CSV file through the Kaggle website.

REAL-WORLD USE CASE — TAXI FARE ESTIMATION

A taxi operator wants to estimate fares before a trip begins. A regression model uses pickup and drop-off locations, passenger count, and trip time to predict the expected fare. Reliable estimates help customers plan costs and help operators assess pricing accuracy.

TASK OBJECTIVE & WHY IT MATTERS

Build a regression model using historical trips, generate fare predictions for unseen trips, and submit them in Kaggle's required format. This exercise connects data preparation, model training, prediction, and performance evaluation to a practical transportation decision.

UNDERSTANDING THE ERROR — MAE EXAMPLE

For classroom interpretation, Mean Absolute Error (MAE) shows the average fare prediction error in dollars. Example: actual fares $10, $20, $30; predicted fares $12, $18, $35. MAE = (|10−12| + |20−18| + |30−35|) ÷ 3 = (2 + 2 + 5) ÷ 3 = $3. A $3 MAE means predictions differ from actual fares by $3 on average. Kaggle uses MAE to score submissions: lower MAE means a smaller average prediction error and a better leaderboard result.

REAL-WORLD APPLICATION — HOW RIDE-SHARING APPS USE FARE ESTIMATES

Think of ride-sharing services such as Uber or Lyft: when a rider enters a pickup point and destination, the app can show an estimated or upfront price before they book. A simple fare-prediction model illustrates one part of a much larger pricing system. Actual ride-sharing prices may also account for demand, available drivers, service type, fees and other business rules.

ILLUSTRATIVE SYSTEM WORKFLOW

1. Rider enters pickup and destination → 2. App gathers trip features (such as distance, time and traffic) → 3. Backend pricing service applies a predictive model and pricing rules → 4. App displays the fare estimate → 5. Rider decides whether to book.

How this connects to your Kaggle task: you train a regression model on historical trip records, evaluate error using MAE (lower is better), and produce fare predictions for unseen trips. In a deployed app, the model could be called through an API to return an estimate; the Kaggle exercise itself does not build a live booking or pricing system.

OTHER RIDE-SHARING & TRANSPORTATION USE CASES

The same regression idea can estimate ride duration, delivery charges, airport-transfer costs, fuel expense, or driver operating cost. These are example applications—not claims about the internal algorithms of specific platforms.

TRANSFER THE SAME SKILL TO YOUR PROFESSIONAL DOMAIN

Construction / Project Management — Estimate project costs using area, labour hours and material quantities.

Engineering — Predict concrete strength from mix composition and curing age.

Procurement / Supply Chain — Estimate material or delivery costs from quantity, supplier and distance.

Equipment Maintenance — Predict maintenance expense from machine age and operating hours.

Facilities / Energy — Estimate monthly energy consumption using temperature and occupancy.

Logistics — Predict delivery cost or journey duration using distance, load and route conditions.

REFLECTION: Identify one numerical value you could predict in your own profession. What historical inputs would you need, and what decision could the estimate support?

PART 1 — Get Set Up on Kaggle & Download the Data

1 **Create a free Kaggle account**

Go to kaggle.com and click Sign Up (free). Verify your email address. Then go to Account Settings → Phone and verify your phone number — Kaggle requires this before your first submission.

2 **Join the competition**

Navigate to https://www.kaggle.com/t/3e8bc07027f24cf6b4f44d54291e293f . Click the blue Join Competition button and accept the rules.

3 **Download the data files**
Click the Data tab on the competition page. Download the zip file and extract it. You will get three files described in the table below.

File	What it is & what to do with it
`train.csv`	Labeled training data — includes the fare_amount target column. Import into your modeling tool and build your regression model here.
`test.csv`	Unseen scoring data — no fare_amount column. Run your trained model on this file to generate one predicted fare per row.
`sample_submission.csv`	Template showing the required format: exactly two columns — key and fare_amount. Your final file must match this structure.

PART 2 — Train Your Model & Score Unseen Data

4 **Build your model using train.csv**
Load train.csv into your modeling tool (e.g. an MLOps platform, notebook, or any tool you prefer). Use fare_amount as the target variable. Engineer features (e.g. trip distance, hour of day, day of week). Train a regression model — Random Forest, Gradient Boosting, or similar work well.

5 **Score test.csv with your trained model**
Load test.csv into the same modeling tool. Apply your trained model to generate a predicted fare_amount for every row. Export the result as a CSV. This output file will contain your predictions alongside the original columns.

PART 3 — Format Your File & Submit to Kaggle

6 **Trim your output to exactly 2 columns**
Open your scored test output file. Keep only two columns: key (the unique row ID — copy it exactly as it appears in test.csv) and fare_amount (your predicted values). Delete all other columns. Save the result as submission.csv.

7 **Verify the format against sample_submission.csv**
Open sample_submission.csv side by side with your submission.csv. Both must have the same number of rows and identical column names: key and fare_amount (lowercase, exact). Every row in test.csv must have a prediction — no missing rows.

8 **Submit on Kaggle & check your score**
On the competition page, click Submit Predictions. Click Upload File and select your submission.csv. Add an optional note, then click Make Submission. After a few minutes, your MAE score appears under My Submissions. Check the Leaderboard tab to see how you rank.

Key reminders: Your file must have exactly the same number of rows as test.csv. Column names are case-sensitive (key and fare_amount must be lowercase). If a submission is rejected, re-check column names and row count against sample_submission.csv.