Data Science Workflow – Flight Fare Prediction

Author: Anwesha Das
Created with Pi

CONTENTS

Feature engineering means creating new, more meaningful inputs for the model. Some useful approaches include:

1. Understanding the Problem

Goal: Predict the ticket price for a flight.

Why: Helps airlines set prices and customers plan better.

Question: "Given details of a flight (airline, source, destination, date, time, etc.), what will be its fare?"

2. Collecting Data

Ways to get data:

Airline websites scrape prices.	**Travel portals (e.g., MakeMyTrip, Skyscanner)** collect flight info.
Public datasets Kaggle or government portals.	**Company's own booking records** if working with airlines.

Collected data usually has:

Airline name	Date of journey
Departure and arrival time	Duration of flight
Source & destination	Number of stops
Ticket price	

3. Preprocessing Data

1

Remove missing values (e.g., if price is not available).

2

Handle wrong entries (e.g., negative duration).

3

Standardize date and time format.

Feature engineering means creating new, more meaningful inputs for the model. Some useful approaches include:

Date & Time Features

- Extract day, month, weekday, and weekend/holiday flags.

- Convert departure/arrival times into categories (morning, afternoon, evening, night).

Duration Features

- Convert total flight duration into minutes.

Categorical Encoding

- Encode airline names, source, and destination into numerical form (label encoding or one-hot encoding).

- Represent number of stops as integers (non-stop = 0, one-stop = 1, etc.).

Scaling & Transformation

- Normalize or standardize features like duration and price.
- Apply log transformation to reduce skewness in ticket prices.

5. Exploratory Data Analysis (EDA)

Find patterns and trends in the data. Examples:

1

Prices may be higher for certain airlines.

2

Evening flights may cost more than morning flights.

3

Non-stop flights usually cost more than 1-stop flights.

4

Fares may increase during holiday seasons.

Use graphs and charts to understand these trends.

6. Model Building

Split data into train set (to learn) and test set (to check accuracy).

Choose models:

- Linear Regression: simple and fast.
- Random Forest / XGBoost: handles complex relations well.

Train the model with features → predict fares.

Evaluate performance using MAE, RMSE, R^2 (error metrics).

7. Deployment

1
Put the trained model into a system (website or app).

2
Example: A user enters details → Model predicts fare instantly.

3
Use tools like Flask, Django, or FastAPI to connect model with website.

4
Update model regularly with new data (as prices keep changing).

Thank You