Flight Prices Prediction Milestone 1

Anh Nguyen, Fiona Xu, Sadichchha Maharjan

Previous solutions

- Many solutions with traditional machine learning algorithms (KNN, Random Forest, Decision Tree)

- Challenges: not good at capturing intricate patterns and rapid changes in market, dynamic pricing

Previous solutions

- Can Deep Learning add value? Better at capturing temporal dependencies in flight data? Better at generalizing?

Deep-Learning-Powered GRU Model for Flight Ticket Fare Forecasting

by Worku Abebe Degife and Bor-Shen Lin *

Datasets

- Main dataset: Almost 6 million entries, US flights between 2022-04-17 and 2022-11-11

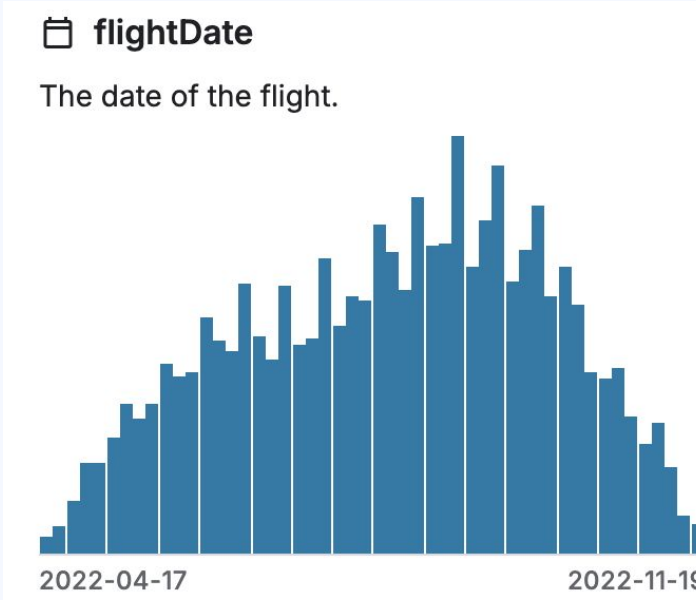
- Additional datasets:
  + https://www.kaggle.com/datasets/iamavyukt/goibibo-flight-data (~300,000 entries, Indian flights, July-August 2023)
  + https://www.kaggle.com/datasets/darjand/domestic-german-air-fares (~63,000 entries, German flights, October 2019-April 2020)

Data Visualization

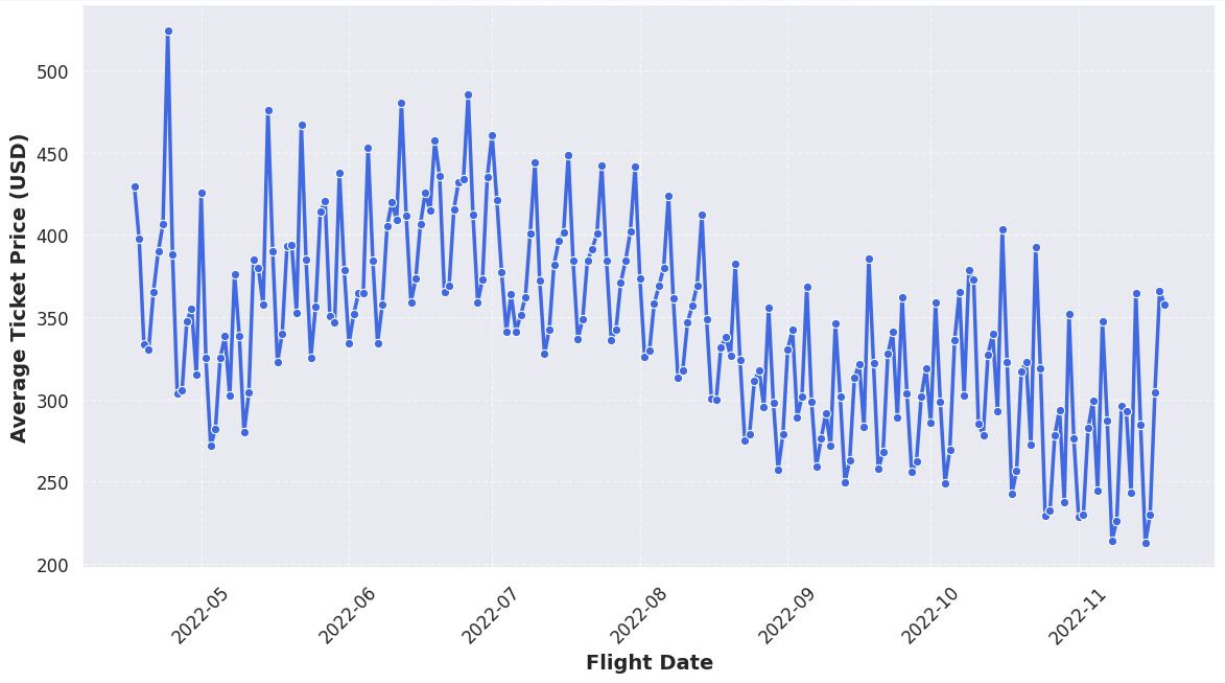
Dataset Overview

- dataset contains flight ticket prices from Expedia between **April 17, 2022 – November 11, 2022**
- includes one-way flights from major U.S. airports like ATL, JFK, LAX, SFO, and more

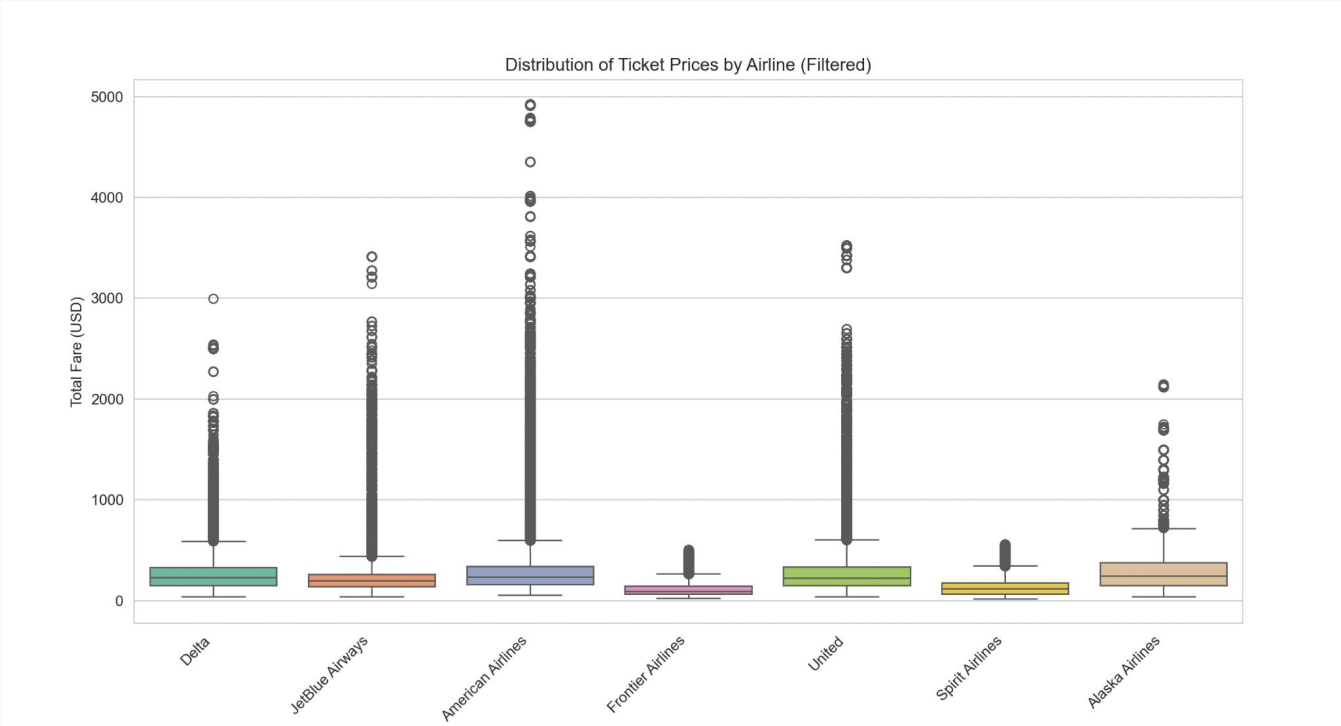
- Total Entries: **5,999,739** flight ticket records
- Total Fields: **27** data attributes (destination, airline, flight duration, ticket price)



Trend of Ticket Prices Over Time

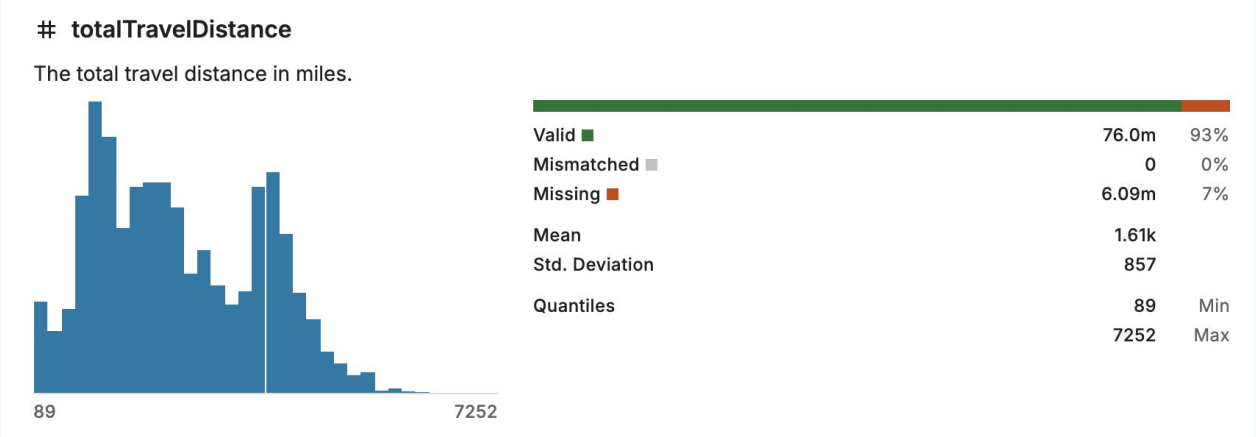


Distribution of Ticket Prices



Data Cleansing

- **Handling Missing Data**
  - Entries with missing data dropped



Challenges

Dataset is too large to process, do not have enough memory to run the data. Even in data preprocessing

Running everything takes 10+ minutes

Solution: use PySpark library for to work in distributed computing environment

Data Preparation

Feature Engineering

Converting string attribute to numerical variable
One-hot-encoding and binary encoding for categorical variables(eg. 16 airports)
Take out irrelevant/repeating columns such as ids, duplicates of cabin info
Create new columns: weekend indicator, holiday indicator

Splitting the Dataset

Train, validation, test
70%, 15%, 15%

Create Training Batches

Prepare mini batches for training

Future Challenges

Find a baseline to evaluate model performance.
using traditional ML
choose right model for prediction: time series forecasting (ARIMA, RNNs)

Deal with outliers:
Want to use flight datasets on other geographic locations and dates: we might meet problems in **generalizing** the model in non US countries and different dates.

Overfitting

Thank you for listening!
Questions or Suggestions?