Predicting Air Travel Fare Using the KNN Model

Xuchao Zhou

Why Air Travel?

Fluctuation

Air ticket price fluctuates a lot which sometimes causes confusion among customers. The price is highly dynamic and it is easy to buy a ticket at a higher price than you have to.

Information Asymmetry

Customers have much less information than airlines. In order to learn about what is an “appropriate price,” they need to spend a lot of time.

Customers will be willing to use a product that allows them to avoid high air travel prices at a very low cost.

Dataset Selection

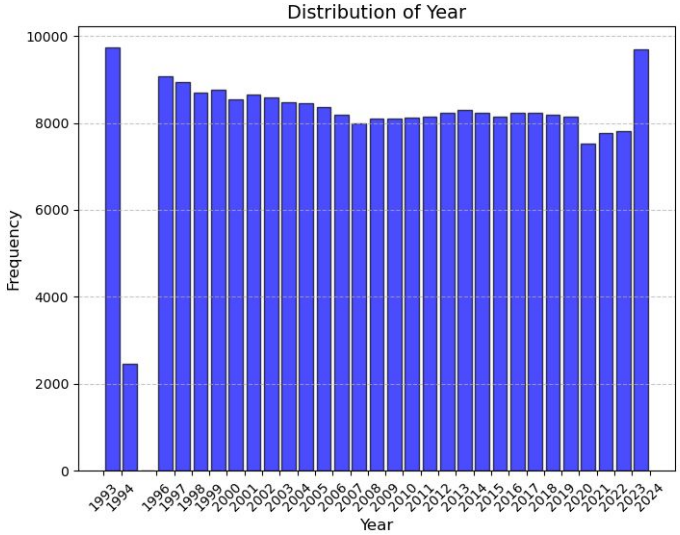
Dataset: US Airlines Routes and Fares 1993–2024 (Jikadara, 2024) dataset from Kaggle

- Contains information about air travel in the US from 1993 to 2024
- Enables analysis on the trend of air ticket prices, passenger flow, and carrier competition

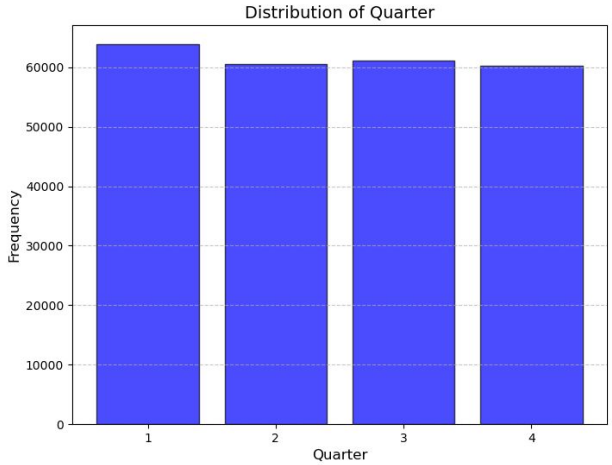
Feature Engineering

Wipe out unnecessary variables

- Make a new variable “route” by combining the departure & arrival airports
- citymarketid_1, citymarketid_2, city1, city2, airportid_1, airportid_2, Geocoded_City1, Geocoded_City2, large_ms, lf_ms are redundant
- tbl, carrier_lg, carrier_low, tbl1apk are irrelevant
- Selected predictors – Year, quarter, route, nsmiles, passengers

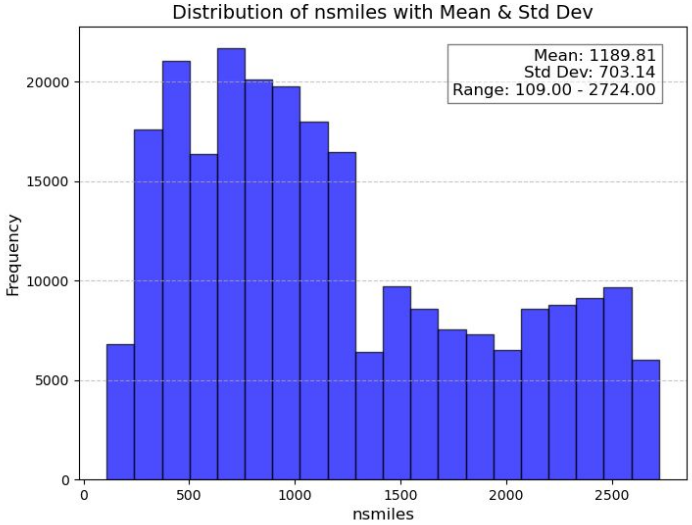


- Year – numerical, represents the year when the data was collected
- Most evenly distributed among years 1993–2024
- Mean and standard deviation not applicable
- No need for encoding or transformation

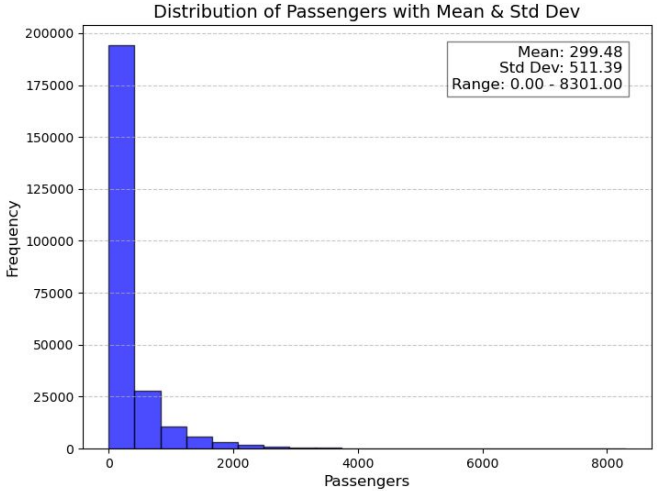


- Quarter - categorical, the quarter in the year when the data was collected
- Most evenly distributed among four categories
- Categorical variable, encoding needed
- One-hot encoding because no ranking among the four categories

- Route - categorical, the information about the origin and destination of the trip
- Important predictor of air ticket price
- Categorical variable, encoding needed
- Ideally one-hot encoding because there is no ranking on routes, but ordinal encoding is chosen since there are too many categories



- Nsmiles – numerical, represents the travel distance
- Right skewed and all positive, so a log transformation is a good choice to make it more suitable for ML training



- Passengers – numerical, represents the number of passengers on this flight
- Strongly right skewed, need log transformation.

Model training

Performance of four models

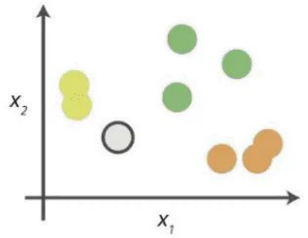
	Validation RMSE	Test RMSE	R-squared
KNN	49.17	49.16	0.5637
Random Forest	45.87	46.33	0.6125
Neural Network	51.42	51.66	0.5183
Decision Tree	55.41	55.84	0.4371

Decision on choice of model

- Select KNN
- A simple model with comparably high performance
- Do not deploy now
- Needs further training and tuning before deployment

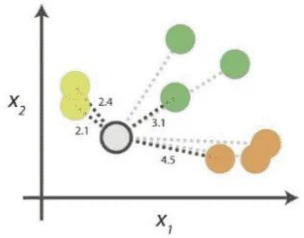
How does KNN work?

0. Look at the data



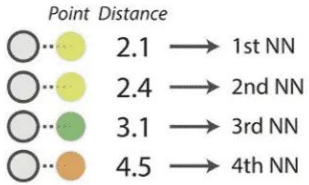
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



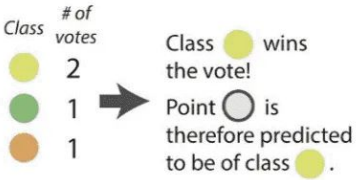
Start by calculating the distances between the grey point and all other points.

2. Find neighbours



Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels



Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

Source: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

Advantages of this solution

Deployment & MLOPS

How it works in production?

API-based model deployment on **AWS/Google Cloud** for real-time predictions.

Precomputed nearest neighbors to speed up lookup time.

Stored queries for tracking trends & monitoring performance.

Operational Monitoring

Continuous Model Evaluation: Regularly monitor the performance of the model using RMSE and R-squared values on new query data.

Drift Detection: Alerts if pricing trends change significantly.

Security & Fairness: GDPR-compliant, bias monitoring.

Thanks!

Any questions ?

You can find me at

- xuchaoz@andrew.cmu.edu