Travel Assistant: Find your next travel destination,Estimating Flight Costs & demand.

Anika Ahmed | Math 37700 | Spring / 2025

Table of Contents

Project Overview

Problem Statement

> Flight prices are notoriously inconsistent—fluctuating due to demand, time of booking, seasonality, and route popularity. These variations create challenges for budget-conscious travelers.

Proposed Solution

> This project uses regression modeling to predict passenger demand on various U.S. flight routes based on features like fare price, historical passenger counts, seasonality (quarter), and route identifiers. Travelever will be able to **predict price**, **evaluate demand** of their flight, The workflow includes data cleaning, one-hot encoding of city pair routes, exploratory data analysis (EDA), and the application of baseline (linear regression) and advanced models (Random Forest, decision tree, and logistic regression). Visualizations such as scatter plots, feature importance charts, and predicted vs. actual comparisons help interpret model performance and identify the most popular predicted routes. Final outputs include both a predictive model and actionable insights for route-level demand.

Impact

- Travelers planning budget-friendly trips
- Airlines optimizing fare strategies
- Travel agencies and booking platforms
- Developers building travel planning tools

Data Description

- **Source:** Department of Traportation
- **Raw Files:** two Located under `data/raw/` 1.(`Consumer_Airfare_Report__Table_3_-_City-Pair_Markets_With_A_Substantial_Increase_In_Average_Fare_20250421.csv`)

2. (`Consumer_Airfare_Report__Table_4_-_City-Pair_Markets_With_A_Substantial_Increase_In_Average_Fare_20250421.csv`)

- **Processed Files:** two datasets are join into one `data/processed/` under the name `(data_clean.csv)`

Raw Data Dictionary

Column Name	Description	Data Type
Year	Year of observation	Integer
Quarter	Quarter of the year (1-4)	Integer
City Pair	Origin and destination cities	String
citymarketid_1	City market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market	Integer
citymarketid_2	City market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market	Integer
city1	City1 is used to consolidate airports serving the same city market	String
city2	City2 is used to consolidate airports serving the same city market	String
cur_passengers	Current year passengers	Float
cur_fare	Current year fare	Float
ly_fare	Last year fare	Float
ly_passengers	Last year passengers	Float
amount_change	Fare amount difference between current year and previous year	Float
percent_change	Fare percentage difference between current and previous year	Float
amount_change_pax	Passenger number difference between current year and previous year	Float
percent_change_pax	Passenger percentage difference between current and previous year	Float

File name	Description
`data\raw\Consumer_Airfare_Report__Table_3_-_City-Pair_Markets_With_A_Substantial_Increase_In_Average_Fare_20250421.csv`	Original full dataset
`data\raw\Consumer_Airfare_Report__Table_4_-_City-Pair_Markets_With_A_Substantial_Decrease_In_Average_Fare_20250421.csv`	Original full dataset

File name	Description
data/Processed/data_clean.csv	Cleaned & feature-engineered training set

Project Structure

```
CapstoneProject/
├── README.md
├── .gitignore
├── requirements.txt
│
├── data/
│   ├── raw/                # Immutable source data
│   └── processed/          # Cleaned, feature-engineered data
│
├── notebooks/              # Jupyter notebooks
│   ├── 1_data_cleaning.ipynb
│   ├── 2_eda.ipynb
│   ├── 3_baseline_models.ipynb
│   ├── 4_Popular_destination_model_3.ipynb
||  └── 5_Price_Prediction_model_1.ipynb
│
│
├── src/ (OPTIONAL)         #  Reusable Python modules
│   ├── __init__.py
│   ├── preprocessing.py
│   ├── features.py
│   ├── modeling.py
│   └── evaluation.py
│
├── slides/                 # Slide deck PDF & source
│   ├── Capstone_Final_Presentation.pdf
│   └── Capstone_Final_Presentation.pptx
│
├── outputs/                # Model artifacts & figures
│   ├── models/
│   └── figures/
│
└──
```

Project Flow

```
Raw Data Ingestion  →  Data Cleaning
Data Cleaning       →  Feature Engineering
```

```
Feature Engineering →  Exploratory Data Analysis
EDA                 →  Baseline Modeling
Baseline Modeling   →  Advanced Modeling & Optimization
Advanced Modeling   →  Model Evaluation & Interpretation
Evaluation          →  Deployment & Reporting
```

1. **Raw Data Ingestion** – load source files and validate schema.
2. **Data Cleaning** – handle missing values, remove duplicates, standardize formats.
3. **Feature Engineering** – create new variables, encode categoricals, scale numerics.
4. **Exploratory Data Analysis** – generate summary statistics and key visualizations (e.g. histograms, boxplots, scatterplots).
5. **Baseline Modeling** – simple models (e.g., linear regression, logistic regression, decision tree, ARIMA) to set performance benchmarks.
6. **Advanced Modeling & Optimization** – hyperparameter tuning, ensembles, or complex architectures (e.g. Random Forests, SARIMAX, Simple Neural Networks, LSTM, CNN, XGBoost etc.).
7. **Model Evaluation & Interpretation** – compare metrics (e.g. MSE, MAE, Accuracy), confusion matrix, plot ROC / precision-recall.
8. **Deployment & Reporting** – save final model in `outputs/models/`, create figures in `outputs/figures/`, assemble slide deck.

Notebooks & Modules

Path	Purpose
`notebooks/1_data_prep.ipynb`	Data ingestion, cleaning, and saving processed files
`notebooks/2_eda.ipynb`	Visual and statistical exploration of cleaned data
`notebooks/3_baseline_models.ipynb`	Fit and evaluate initial benchmark models
`notebooks/4_final_modeling.ipynb`	Advanced modeling, optimization, and final evaluation
`src/preprocessing.py`	Functions for loading and cleaning data
`src/features.py`	Feature-engineering utilities
`src/modeling.py`	Model training, tuning, and persistence
`src/evaluation.py`	Metric calculators and plotting helpers

Results & Deliverables

- **Key Figures** – located in `outputs/figures/` (e.g., feature importance, ROC curve)
- **Metrics Summary** – documented in notebooks and slide deck

---

Slide Deck

Location: `D:\spring 25\appiled stat\anika Sprint 1\slides\Travel.pdf`
Five slides covering: overview, data & preprocessing, key insights, model results, demo/design & next steps.

License

This project is licensed under the **MIT License** – see `LICENSE` for details.