CS545 - Flight Cost Analysis

Andrey Karnauch
Rojae Johnson
Cai John
Matthew Kramer
Matt Anderson

Project Overview

- Determine most cost effective airport near a user to fly out of
    - Using historical prices
    - Incorporating gas costs to drive to each airport
    - Allow user to specify driving radius

Motivation

- Many sites already exist for finding cheap, current tickets
    - Have to manually enter separate ORIGIN airports for comparison
    - Do not take into account driving distance and gas costs
- Closest service to our project is Faredetective

Dataset - Bureau of Transportation Statistics

- Provided data for each flight:
  - One CSV for flight information
  - One CSV for ticket information
  - Joined using common ID
- Missing data:
  - Flight date(s)
  - Live Ticket Data
  - Not all airports

DEPARTMENT OF TRANSPORTATION UNITED STATES OF AMERICA

Development Platform

- Google Cloud Compute Engine Instance
  - Familiar with from Practice0
  - Packages and development uniform for everyone
- Google Cloud Storage Bucket
  - One point for all members to view program outputs
  - Stored all cleaned data
- Google Cloud SQL
  - Provided CSVs fit relational DB model
  - Communicate with our GC instance

Compute Engine

Cloud Storage

Cloud SQL

Data Retrieval and Cleaning

- Retrieval Scripts
  - **Download** all flight data from 2015-2018
  - **Store** data on GC instance
- Cleaning Script
  - Loaded into pandas dataframe and **merged** on common ID
  - **Dropped** unnecessary columns
  - **Dropped** rows that would skew averages
  - **Inserted** into GC SQL

Cloud SQL Data Storage

- 400,000 flights from each quarter of each year
  - 2015 - 2nd quarter of 2018
- ~6 million total entries
- Queries take some time
  - Record count
  - Communication b/w instance and database

Data Analysis

Done in modular way:

1. Run simple queries on GC SQL database
2. Incorporate python libraries to do basic statistics
   a. Average cost per ticket, yearly trends, etc.
3. Incorporate APIs
   a. Geopy, matplotlib, Google Maps

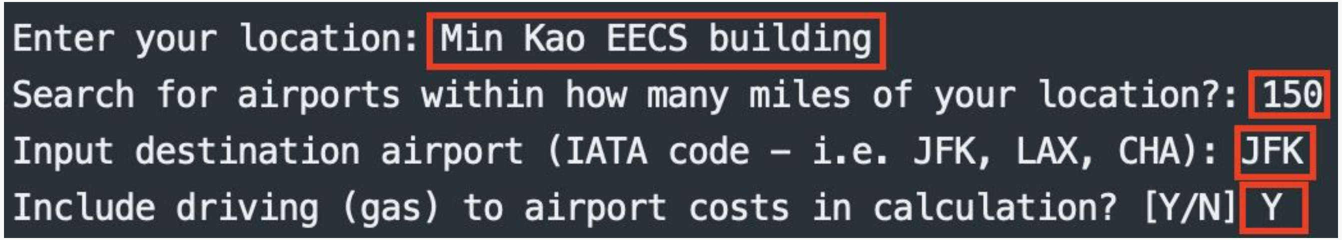
Results

- Function to perform main goals
  - Show user average ticket prices for all airports in area
  - Option to include gas costs in averages
  - Show ticket price trends over each quarter
- Additional Functionality
  - Option to show price averages based on destination or generalized

Graph Generation

- Created using matplotlib
- Pulls the data from Cloud SQL DB for generation
- Stores resulting graphic on GCloud Bucket

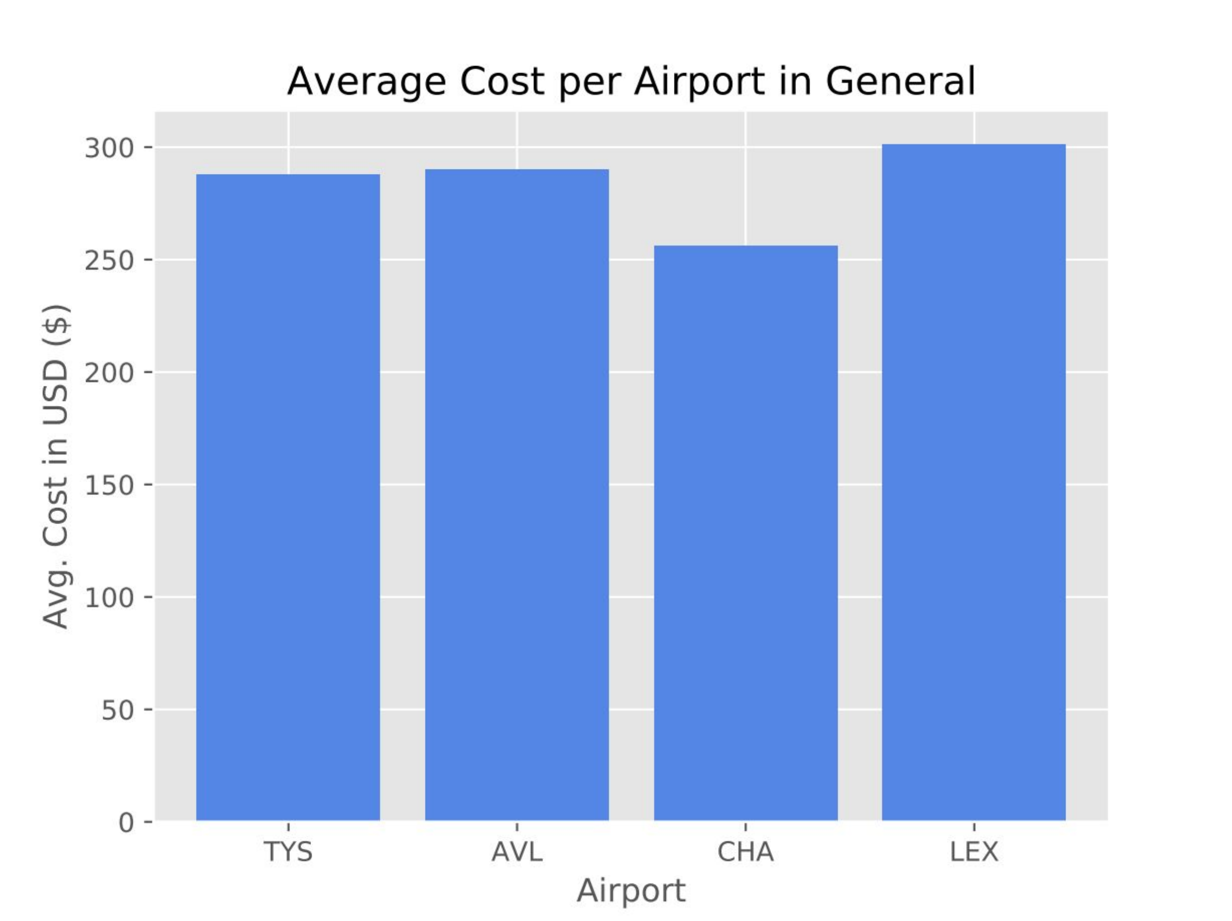
Example Input



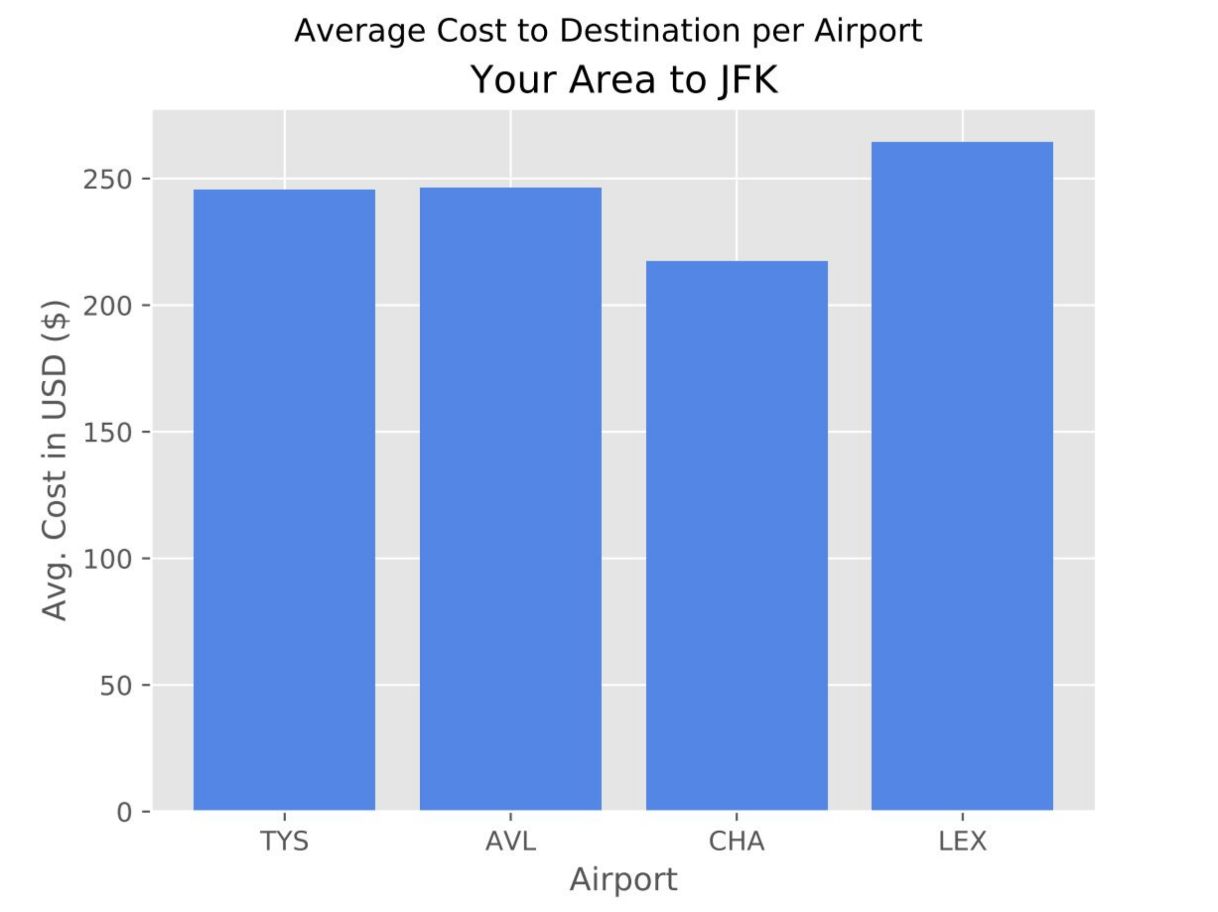
```
Enter your location: Min Kao EECS building
Search for airports within how many miles of your location?: 150
Input destination airport (IATA code - i.e. JFK, LAX, CHA): JFK
Include driving (gas) to airport costs in calculation? [Y/N] Y
```

- Location: **Min Kao EECS Building**
- Search for airports within: **150 miles**
- Destination airport code: **JFK**
- Include gas prices to average costs in calculation: **Y**

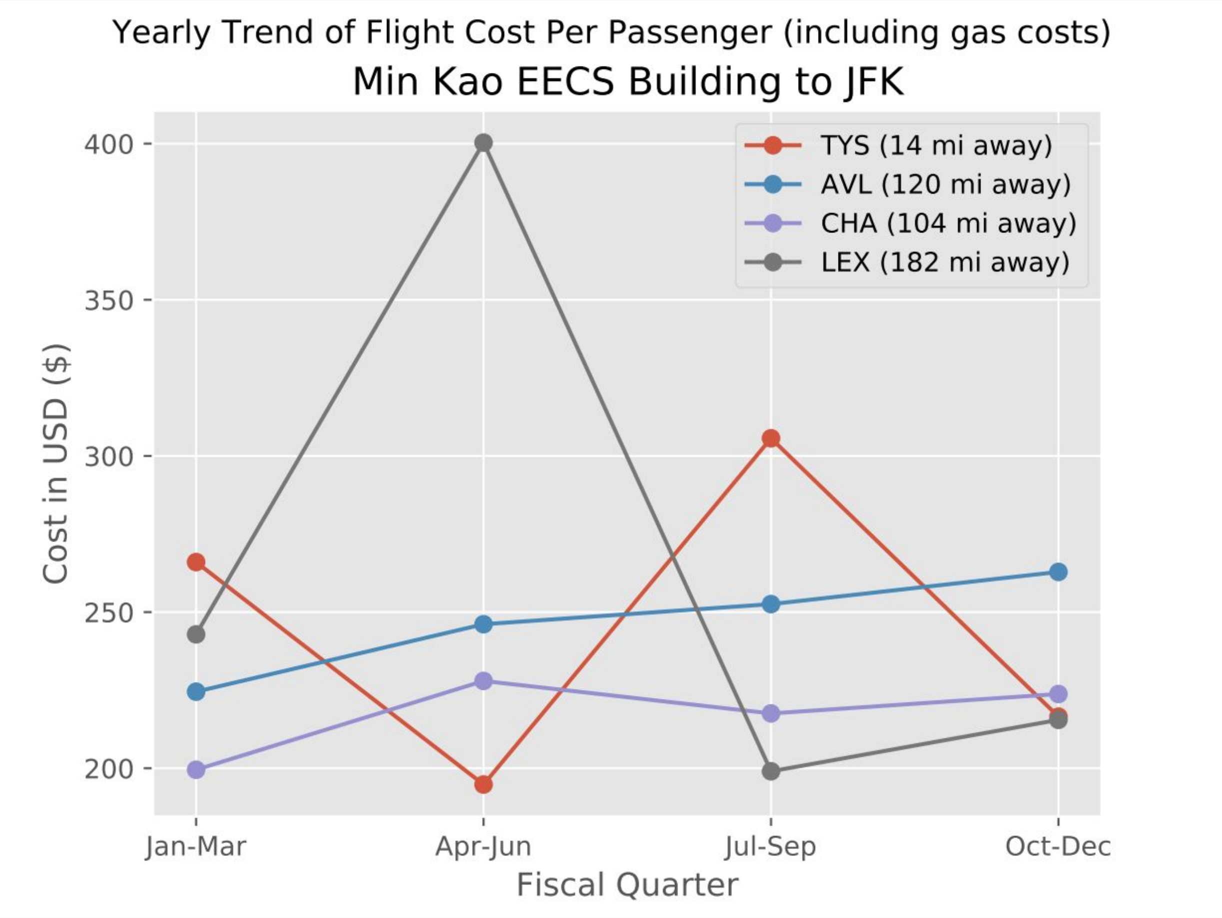
Example Output - General



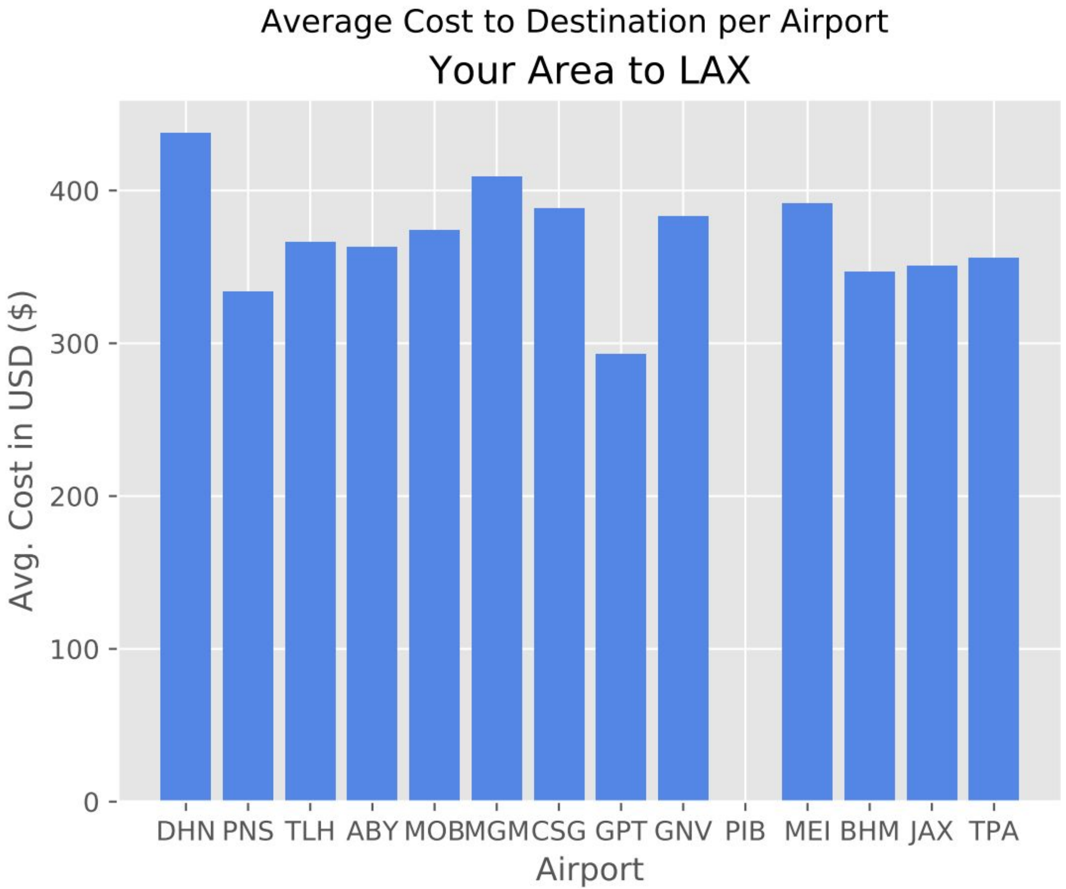
Example Output - Destination Specific



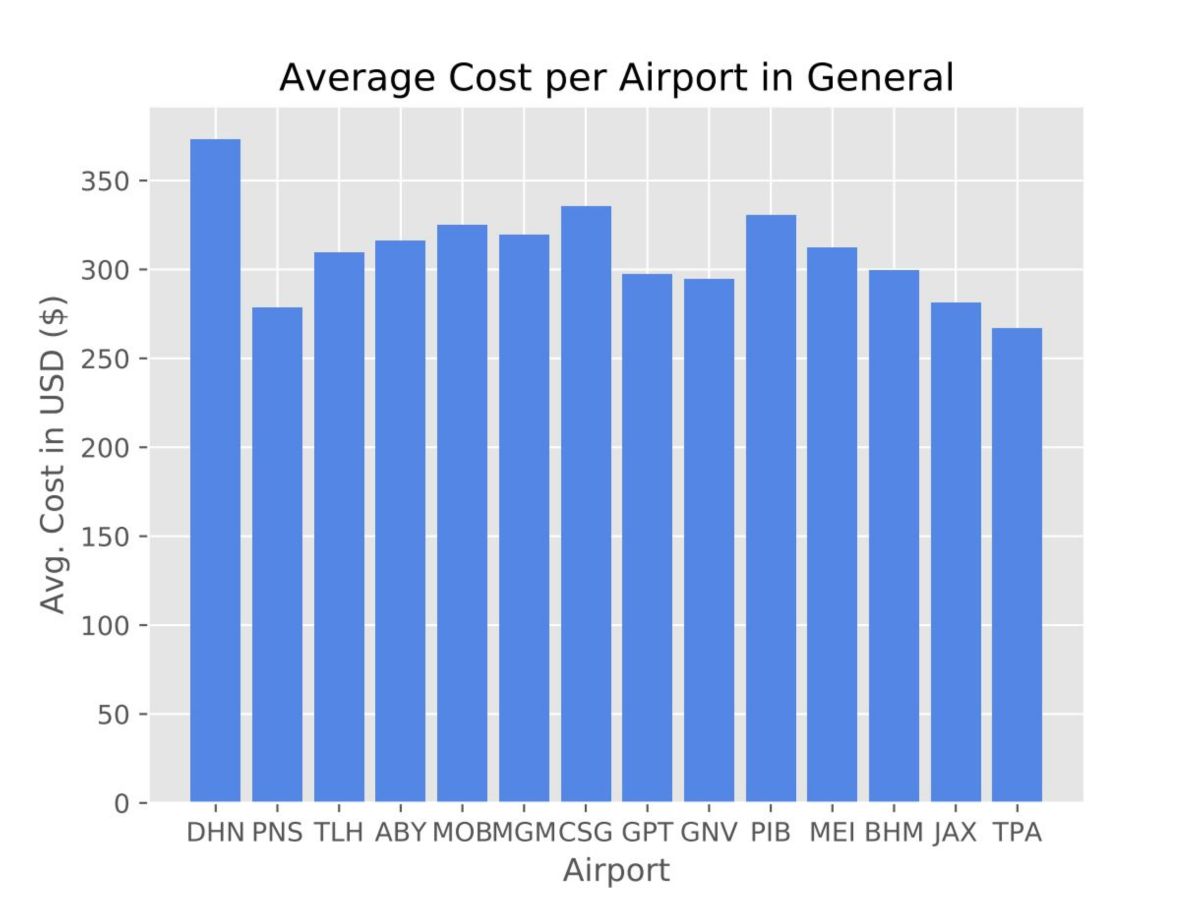
Example Output - Destination Specific



Dataset Shortcomings



Dataset Shortcomings



Future Work

- Live Scraping
- GUI rather than CLI
- Database optimization (indexing, etc.)
- Increase amount of data used

Thank you!

Questions?