Airline Data Project Documentation

Avery Eddins

September 7, 2024

1 Introduction

1.1 Project Overview

The goal of this preliminary iteration of the project is to have data in a format useful for the overall goal of acquiring airline-business-relevant insights to causes of delays for U.S. domestic flights through upcoming analysis.

1.2 Target Audience

This work is intended to showcase Python and SQL proficiency. Future planned iterations of the project will include further use of SQL, as well as use of data analysis skills, to derive business-relevant insights into airline delay causes.

1.3 Data Information

Data download
Definitions of delay causes described in data

2 Installation and Setup

2.1 Prerequisites

The user must be able to run a Python script written in version 3.11. Pandas must be installed as well. This script uses Pandas 2.1.4. All other libraries used in the script are standard.

2.2 Installation instructions

This MVP comes with the Bureau of Transportation Statistics (BTS) comma-separated values (CSV) files organized into the directory structure the script is expecting: directory of directories, each containing the CSV files by year. This directory must be in the same directory as the Python script.

3 Usage

3.1 Core functionality

The script currently works with data from the years 2000 to 2023. Currently, the SQL database is one table, 'Flights', with the BTS data loaded in with fully null columns removed during the loading process. SQL was selected over Pandas, as the data is too large to work with it all loaded into memory. Pandas will likely be used to work with subsets of data obtained from queries to the SQL DB file.

The script largely comprises of a database class, which handles the operations with the database through the addData and closeConn methods. The addData method in turn depends on the checkDataPresent and getCols methods. This object-oriented style of managing the database was selected as it improved on readability and organization over a procedural version of the script.

The overall operation of the script is as follows:

- Initialize the database object with the database file name, opening a connection
- Initialize the list of CSV files to add to the SQL database
- Specify the directory that contains the data
- Loop through the files in the data directory to gather the full paths for the CSV files that need to be checked for or inserted
- Call the addData database method to add the files specified above to the SQL DB file with the overwrite parameter as either True or False, throwing out fully null columns in the loading process
- Close the connection

4 Future work

The data must be verified before analysis begins. The steps for that are as follows:

- Data types for columns are correct
- Missing values. Expect missing values for non-delayed flights in delay-related columns only.
- Outliers
- Observe distribution
- Check for duplicate rows

After that, relationships in the data should be identified through preliminary analysis:

- Correlations/dependencies
- Visualizations

From there, the direction of data analysis can be fully determined.

An additional feature could be to use the Python requests and os libraries to pull the CSVs from the BTS website directly, unzip them, organize them into the needed directory structure, load the data in, and then delete the CSV files once the SQL database (DB) file is constructed so that the user doesn't need to do anything to manage the data ahead of using the script.

The project could be expanded in the future to have another table with airplane or airline data to complement the delay data to determine, e.g., if older planes experience more delays or if airlines performing poorly financially have more delays.

More robustness could be included for interruptions to database procedures by automating closing the connection.

The functionality that checks the data being loaded into the database for fully null columns should also check which columns are already included in the database to keep the columns consistent across data currently loading in and data already loaded in.

It would likely be useful to look for improvements in efficiency for loading data into the table. Loading the majority of the data can take over an hour, and it would be good to speed this up so overwriting can be easier if some change is made to the loading process.

5 Changelog

This represents the first official version of the project, dated to September 7, 2024. Any changes to this version and following will be listed here.