# Analysis Of flight and airport operational data

ABHISHEK GUPTA[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: abhigupt@iu.edu

*project: https://github.com/abhishek8gupta/projects/tree/master/flight-report , July 25, 2017*

**We have lot of operational public data available for various categories like weather, transportation, agriculture, economics, genetics, biology, government etc but very few ways to analyze and present the data. This project aims at analyzing public data from aiports and airlines and make it available for querying as well as some visualization of the data. This analysis will reveal interesting metrics like passenger flow on particular days from a terminal or airline traffic on a airport on a given day or what airlines operate from a airport etc. These reports will help someone external or may be the airline itself to operate better.**

**Keywords:** ETL, CSV, LAMP

Report: https://github.com/abhishek8gupta/projects/tree/master/flight-report/docs/report.pdf
Code: https://github.com/abhishek8gupta/projects/tree/master/flight-report

## CONTENTS

## 1. INTRODUCTION

This project intends to use publicly available data sources to provide meaningful insights, along with a proof of concept to build a entire pipeline to process any such data. The data available from the public sources [1] is in different formats and may not be very easy to parse and load, since this data may have been exported from a proprietary system. There is always a challenge to extract and clean the data. Event after cleaning, the data may not be in a state to get loaded directly to a database, there are special characters, data type mismatch and other issues which may cause load failures. Hence, we have to carefully skip such failures and continue. Ideally the ETL tools are smart enough to provide all these capabilities. But the the purpose of this project, I have used simple python script to demonstrate an ETL workflow. Overall, this project walks you through all stages to view and analyze one such datasource. I have chosen to use airport public data [1] for SFO and LAX airports, and chosen a small dataset to demonstrate the feature. However, this whole workflow can be extended towards a completely automated end to end solution.

| module | Purpose |
|---|---|
| python script | etl script |
| dll script | create SQL script |
| loader script | load data to mysql |
| UI | php files to visualize the data |

**Table 1.** Project Modules

## 2. DESIGN

### 2.1. technology stack

For running this application, I have chosen LAMP combination i.e. Linux, apache, Mysql and PHP. This is quite stable combination and has been used in production by number of people due to variety of reason like easy install and configure, integrate, troubleshoot and lot of community support. Along with these I have chosen python to do ETL kind of workflow. Again python is widely adopted and has modules to support variety of use cases. I am using xml tree module to do xml parsing along with other standard modules.

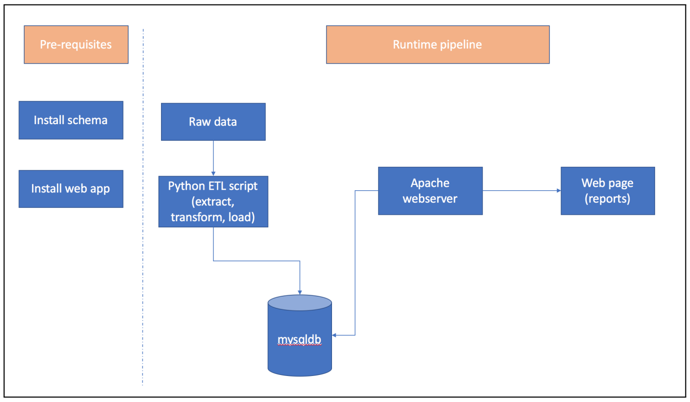


**Fig. 1.** ETL and Data Pipeline.

Figure 1 shows the overall data pipeline for the project. The data pipeline has three important stages:

- Extract: Extract primarily is responsible for downloading the raw files and understanding the schema of the data. Assuming we have downloaded these files using wget or http, this python script reads each of the files from input folder and parses. It skips erroneous records. Erroneous record can be present do to variety of reasons or could have been introduced while exporting the data. Ideally, the tool should log these records to a file, where use can manually fix the errors and replay this file. This parser also generates the header for each file which is helpful creating the schema from the output file. The output file is a comma separated file(csv format) which can readily be loaded into mysql.

- Transform: Transform part of the process deals with mapping xml to a CSV format. This requires understanding the inputfile write the parsing logic to construct the output csv file. Here, the assumption is that the input file format will not change per file and will remain same throughout this process. If the input file format changes, then this parsing logic has to acomodate this change.

- Load: Once we have output files generated, we load these files into mysql using the infile syntax. Load data script [2] reads the input files generated by transform stage and writes to mysql. There are challenges in load time where the input format and data type may not match with mysql format [2] and datatype. Hence, we have to use conditions on those columns and provide a way to load the data into these columns. For example the date format export file is mmddyyyy, however load script doesn't understand the change in format. Hence it provides a step in the script to transform this data while loading.

### 2.2. install configure mysql

Mysql [2] can be installed on separate machine or can run on the same machine as the webserver. We will need to create the database and schema to create the tables. The table creation is manual process and derived from the out CSV file. The table creation script flight-report/ddl/ddl.sql can be executed from mysql client CLI or can be executed from some other client tool like mysql workbench. There is a dataloader script which is also manually created, this script can be used to load the CSV files in the database. This loader script uses mysql infile syntax to bulk load the tables. We can add some logic to the loader script to transform the incoming data in case its malformed or need to in desired form. For example the date syntax from CSV file may not be correct and need to be transformed before loaded to mysql.

### 2.3. install configure apache web server

Apache web server [3] is freely available under apache license. I have setup apache on my local mac for this project. I have made few changes to standard install like configuring my site by enabling userdir site modules. Once apache is up and running you can deploy you web pages in the userdir folder and then you can access the site using the syntax *http://localhost/ abhigupt/<site-folder>*, here the site folder is where I deploy all my PHP and html pages. I ran into one issue where PHP modules were not enabled by default on apache server [4] and had to run a test to view phpinfo.php script using apache server. PHP modules can be enabled in /etc/apache2/httpd.conf by uncommenting *LoadModule php5* line.

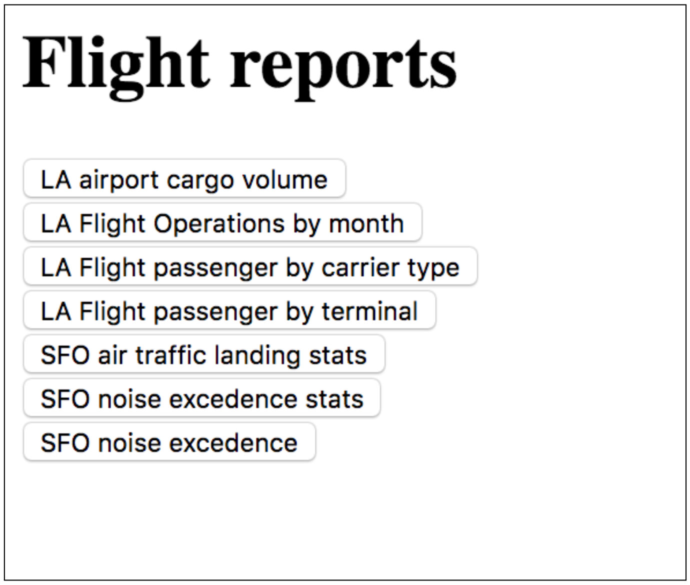
### 2.4. running solution

To run the solution, you need to install/configre mysql and apache server. Further create database and schema, deploy the web app. These tasks are one time tasks. Further, running ETL on input files can be done periodically or can be scheduled using a cron job etc.

These all stapes can very well automated and can work based on a sche

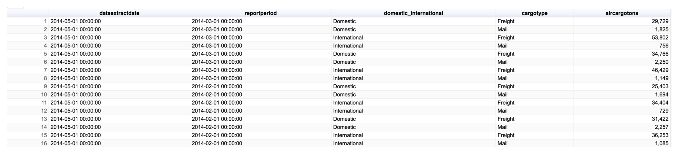
## 3. REPORTS

I generated primarily tabular report[5] but more reporting can be done on this raw data. Some of the same reports are shown below. Figure 2 shows landing page and buttons to view the existing report.


Flight reports

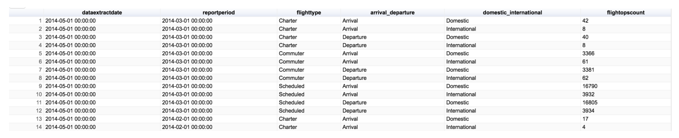
LA airport cargo volume
LA Flight Operations by month
LA Flight passenger by carrier type
LA Flight passenger by terminal
SFO air traffic landing stats
SFO noise exceedence stats
SFO noise excedence

**Fig. 2.** Application dashboard



| | dataextractdate | reportperiod | domestic_international | cargotype | aircargotons |
|---|---|---|---|---|---|
| 1 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Domestic | Freight | 29,729 |
| 2 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Domestic | Mail | 1,825 |
| 3 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | International | Freight | 53,802 |
| 4 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | International | Mail | 756 |
| 5 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Domestic | Freight | 34,766 |
| 6 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Domestic | Mail | 2,250 |
| 7 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | International | Freight | 46,429 |
| 8 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | International | Mail | 1,149 |
| 9 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Domestic | Freight | 25,403 |
| 10 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Domestic | Mail | 1,694 |
| 11 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | International | Freight | 34,404 |
| 12 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | International | Mail | 729 |
| 13 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Domestic | Freight | 31,422 |
| 14 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Domestic | Mail | 2,257 |
| 15 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | International | Freight | 36,253 |
| 16 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | International | Mail | 1,085 |

**Fig. 3.** Sample report1



| | dataextractdate | reportperiod | flighttype | arrival_departure | domestic_international | flightopscount |
|---|---|---|---|---|---|---|
| 1 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Charter | Arrival | Domestic | 42 |
| 2 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Charter | Arrival | International | 8 |
| 3 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Charter | Departure | Domestic | 40 |
| 4 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Charter | Departure | International | 8 |
| 5 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Commuter | Arrival | Domestic | 3366 |
| 6 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Commuter | Arrival | International | 61 |
| 7 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Commuter | Departure | Domestic | 3381 |
| 8 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Commuter | Departure | International | 62 |
| 9 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Scheduled | Arrival | Domestic | 16790 |
| 10 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Scheduled | Arrival | International | 3932 |
| 11 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Scheduled | Departure | Domestic | 16805 |
| 12 | 2014-05-01 00:00:00 | 2014-03-01 00:00:00 | Scheduled | Departure | International | 3934 |
| 13 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Charter | Arrival | Domestic | 17 |
| 14 | 2014-05-01 00:00:00 | 2014-02-01 00:00:00 | Charter | Arrival | International | 4 |

**Fig. 4.** Sample report2

Figure 3 shows a sample report
Figure 4 shows a sample report

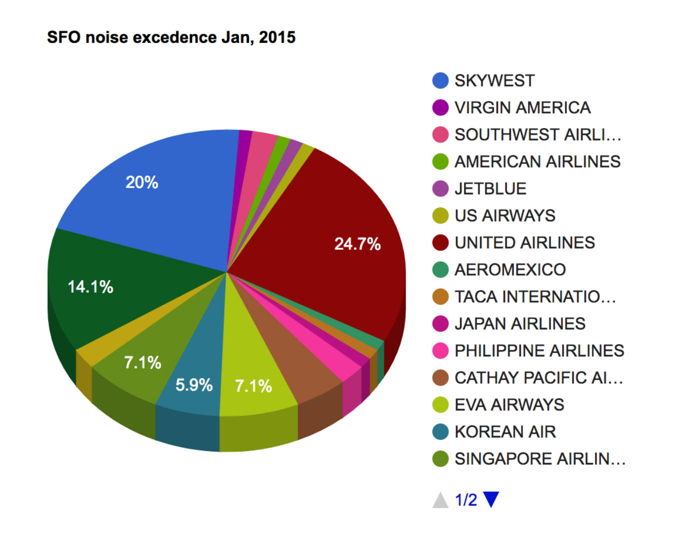


**Fig. 5.** Sample report pie chart

Figure 5 shows a sample report using pie chart

## 4. CHALLENGES

During this project I faced few challenges which I would like to highlight

- Apache server not loading php modules: to fix this issue, I had to enable php libraries in httpd.conf on apache server [4].
- Handling data load errors: handle these load errors in data loader script
- mysql [2] client not able to connect to mysql server: add hostname to ip address mapping in /etc/hosts file
- some of the php pages not loading due to memory errors: initialize the memory limit as -1 for the php page

Other major challenge was to understand the input data format and write the parser. The xml files downloaded from public website [1] we sometimes timing out due to bigger size.

## 5. DISCUSSION

The project is more like proof of concept on how we can utilize a public datasource [1] to extract and derive meaningful analytics and tries to touch on various aspects of data lifecycle management. Sometimes the challenges are finding the right dataset itself, since these public datasources [1] even though are said to be public but when you try to download a dataset the server times out or the format is such that you cannot decrypt unless you understand the format or there is a library available. However, if you are able to build a pipeline, then the format of the data hardly changes. The technology stack which I have used my not be ideal for processing this data since this data volume can be really huge. We need to use technologies which can scale horizontally as data increases. These data pipelines can be build more generically where you specify the input and output format and the pipeline will churn and spit the corresponding files. Also, the data loading part can be automated, where a cron job will download the file in a scheduled manner and upload it on a storage like amazon s3 bucket and consuming pipelines churn the data as soon as its available on s3.

## 6. CONCLUSION

Using this project I was able to visualize some of the public datasets [1] for airport and airline data. There can be more complex analytics that can be run this same dataset. Further, there can be correlations done with this dataset and another dataset for example whether. Correlations can be for example, if there are delays reported by airlines on a particular day was it due to bad whether. These kind of analytics will help take lot of benefit of this public data [1] where can even predict some of these events and take best course of action.

## 7. ACKNOWLEDGEMENT

We acknowledge our professor Ying Ding and associate instructors Yi Bu and Ajay Saini for helping us and guiding us throughout this course.

## REFERENCES

[1] "Data catalog - the home of the u.s. government's open data," Web Page, accessed: 2017-07-20. [Online]. Available: https://catalog.data.gov/dataset?res_format=XML&tags=airport&page=2
[2] Mysql, Inc., "Mysql documentation," Web Page, accessed 2017-07-15. [Online]. Available: https://dev.mysql.com/
[3] "Apache http server - documentation," Web Page, accessed 2017-07-15. [Online]. Available: https://httpd.apache.org/
[4] "Medium - how to set up apache in macos," Web Page, accessed: 2017-07-15. [Online]. Available: HowtoSetupApacheinmacOS
[5] "Google visualization - display live data on your site," Web Page, accessed: 2017-07-15. [Online]. Available: https://developers.google.com/