PDF Report - Task 1

1) Overview - Data Extract and Cleaning

I copied the snapshot, renamed as "Copied Snapshot - Transportation Databases (Linux)", to retrieve the dataset. I created a new EBS volume from the snapshot, and a new EC2 instance to be attached to the EBS volume. After mounting EBS to your EC2 instance, I copied files from EBS to my local machine.

Once I had the entire airline_ontime directory on my local, I unzipped all the CSV files, expecting and finding 240 files, ignoring 2008_11 and 2008_12 zips. I then moved all CSV to one directory and converted them to utf-8. From here, I used Jupyter for a data cleaning script. This was a two-part optimization because a) smaller file sizes needed to be uploaded to S3, saving on storage costs, and b) before doing this, I ran into issues with many of the columns were null even though they were in the CSVs in S3. It seems that these were float integers, and I had to go back to convert those columns.

After the script completed, I compressed each cleaned csv into their .gz, and upserted all the .gz to S3. From there, I used AWS Glue to clean, organize, and view the data by creating and running a crawler in AWS Glue. After the crawler completed, I cleaned and updated the table schema to confirm the wanted columns and data types. Using AWS Athena, I validated my cleaned data in S3. I expected and found 116,753,952 rows and also validated the data by querying a few rows and seeing that the stored data looked correct.

2) Overview - System Integration

Used AWS EMR with Hive and DynamoDB. I launched an EMR cluster with the default, 3 nodes (2 master and 1 slave). My SSH first timed out when I tried SSHing directly into the master node after it was up and running. I resolved this by adding SSH to port 22 in the AWS Security Group.From there I set up a Hive external table using S3 as the location from which I would import the airline data into DynamoDB. S3 initially imported 116,754,192 records, vs 116,753,952 rows found from Athena before. This is a 240 row difference, for the CSV headers. I then created the DynamoDB tables for Group 2 and Group 3.2. I did spend some time tinkering with the partition and sort keys and learning how they worked for DynamoDB, but eventually went back to approaching the problems and optimizing my queries.

3) Approaches and Algorithms

I started by testing out my queries in Athena and validating I got the expected solution there. From there, I could copy and paste those queries into the HIVE CLI to run on the EMR Hive cluster when they were ready. From there, it was a simple matter of adopting the query to create an external table and insert the data into DynamoDB. For each question, I created separate DynamoDB tables and ran the HIVE queries to insert data in their respective tables.

For Question 3.1, I ran the Hive query needed for the data points, then added those results to a Hive table which was subsequent exported to a file in S3. I then converted that file from S3 to a CSV file. From there, I created the distribution graphs needed to answer 3.1. For 3.2, it took some time to figure out how I wanted to do this. I first approached this from the perspective where I would have just one table and do one massive query with JOIN/UNION/WHERE/GROUPBY/etc. However, that would be time and performance intensive. Thus, I decided to break the problem up by using three tables: one for the first leg, one for the second leg, and one for the complete flight that selected the necessary fields from the first two tables. For importing the result for this question, I only imported results from the third table.

4) Results

Only results are included here. Full Hive Queries & image proof can be provided if asked. *My results are rounded to the hundredths place to mirror the example solutions

Group 1

1.1) Rank the top 10 most popular airports by numbers of flights to/from the airport. ORD: 12449354 ATL: 11540422 DFW: 10799303 LAX: 7723596 PHX: 6585534 DEN: 6273787 DTW: 5636622 IAH: 5480734 MSP: 5199213 SFO: 5171023	1.3) Weekday: Average delay in minutes Saturday / 6: 4.30 Tuesday / 2: 5.99 Sunday /7 : 6.61 Monday / 1: 6.72 Wednesday / 3: 7.20 Thursday / 4: 9.09 Friday / 5: 9.72

Group 2

2.1) For each airport X, rank the top-10 carriers in decreasing order of on-time departure performance from X.

*Rounded to the hundredths place to mirror the example solutions

CMI (University of Illinois Willard Airport) (OH, 0.61) (US, 2.03) (TW, 4.12) (PI, 4.46) (DH, 6.03) (EV, 6.67) (MQ, 8.02)	BWI (Baltimore-Washington International Airport) (F9, 0.76) (PA (1), 4.76) (CO, 5.18) (YV, 5.50) (NW, 5.71) (AA, 6.00) (9E, 7.24) (US, 7.49) (DL, 7.68) (UA, 7.74)	MIA (Miami International Airport) (9E, -3.0) (EV, 1.20) (TZ, 1.78) (XE, 1.87) (PA (1), 4.20) (NW, 4.50) (US, 6.09) (UA, 6.87) (ML (1), 7.50) (FL, 8.57)
LAX (Los Angeles International Airport) (MQ, 2.41) (OO, 4.22) (FL, 4.73) (TZ, 4.76) (PS, 4.86) (NW, 5.12) (F9, 5.73) (HA, 5.81) (YV, 6.02) (US, 6.75)	IAH (George Bush Intercontinental Airport) (NW, 3.56) (PA (1), 3.98) (PI, 3.99) (US, 5.06) (F9, 5.55) (AA, 5.70) (TW, 6.05) (WN, 6.23) (OO, 6.59) (MQ, 6.71)	SFO (San Francisco International Airport) (TZ, 3.95) (MQ, 4.85) (F9, 5.16) (PA (1), 5.29) (NW, 5.76) (PS, 6.30) (DL, 6.56) (CO, 7.08) (US, 7.53) (TW, 7.79)

2.2) For each source airport X, rank the top-10 destination airports in decreasing order of on-time departure performance from X.

*Rounded to the hundredths place to mirror the example solutions

CMI (University of Illinois Willard Airport) (ABI, -7.0)	BWI (Baltimore-Washington International Airport)	MIA (Miami International Airport) (SHV, 0.0)

(PIT, 1.10) (CVG, 1.89) (DAY, 3.12) (STL, 3.98) (PIA, 4.59) (DFW, 5.94) (ATL, 6.67) (ORD, 8.19)	(SAV, -7.0) (MLB, 1.16) (DAB, 1.47) (SRQ, 1.59) (IAD, 1.79) (UCA, 3.65) (CHO, 3.74) (GSP, 4.20) (SJU, 4.44) (OAJ, 4.47)	(BUF, 1.0) (SAN, 1.71) (SLC, 2.5) (HOU, 2.91) (ISP, 3.65) (MEM, 3.75) (PSE, 3.98) (TLH, 4.26) (MCI, 4.61)
LAX (Los Angeles International Airport) (SDF, -16.0) (IDA, -7.0), DRO, -6.0) (RSW, -3.0) (LAX, -2.0) (BZN, -0.73) (MAF, 0.0) (PIH, 0.0) (IYK, 1.27) (MFE, 1.38)	IAH (George Bush Intercontinental Airport) (MSN, -2.0) (AGS, -0.62) (MLI, -0.5) (EFD, 1.89) (HOU, 2.17) (JAC, 2.57) (MTJ, 2.95) (RNO, 3.22) (BPT, 3.60) (VCT, 3.61)	SFO (San Francisco International Airport) (SDF, -10.0) (MSO, -4.0) (PIH, -3.0) (LGA, -1.76) (PIE, -1.34) (OAK, -0.81) (FAR, 0.0) (BNA, 2.43) (MEM, 3.30) (SCK, 4.0)

2.4) For each source-destination pair X-Y, determine the mean arrival delay (in minutes) for a flight from X to Y.

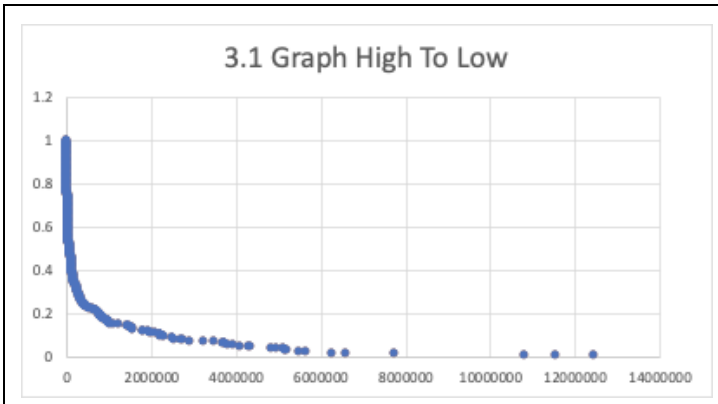
- CMI → ORD: 10.14
- IND → CMH: 2.90
- DFW → IAH: 7.65
- LAX → SFO: 9.59
- JFK → LAX: 6.64
- ATL → PHX: 9.02

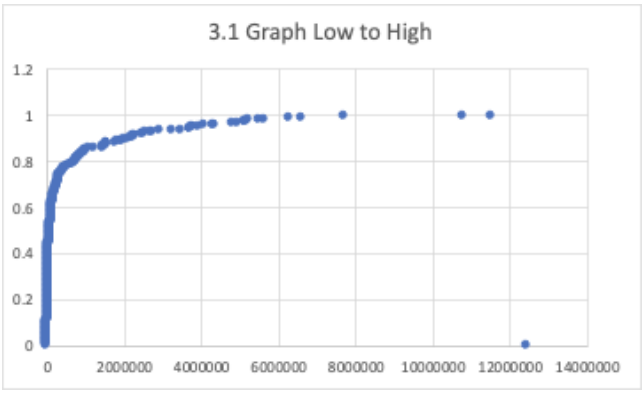
Group 3

3.1) Does the popularity distribution of airports follow a Zipf distribution? If not, what distribution does it follow?

No, when plotting the result of the above query, we can see the overall shape follows a log-normal distribution, and not more of a straight line, which a power-law (Zipfian) distribution would be more similar to. In the attached images, we have the frequency distribution on the y-axis and the popularity/number of flights on the x-axis. One was plotted with popularity from highest to lowest while the other is plotted from lowest to highest.

Discrepancy Explanation: This is expected to differ from the provided example solution because I plotted my popularity distribution as a classic CDF in excel, not CCDF as the example solution does. This means that the "CCDF illustrates the fraction of airports with popularity above a given value." while the CDF illustrates the fraction of airports with popularity below a given value. Thus, the chart is similar to the provided example solution in its curvature and distribution, but different in how the distribution is calculated and thus plotted. You may notice the graph could be seen as a mirror of the provided sample solution.





3.2) I factored in departure and arrival delay, not just arrival delay, because while the passenger may have an arrival delay of X, this does not include any departure delays of Y.

While the example solutions are using arrival delay, the prompt itself for Question 3.2 does not specify arrival delay as the only measure of delay, so using total delay would still fall within the requirements (Capstone Project Overview) for "as little delay as possible".

Most of the queries aligned with the example solutions, except in 1 and 5, where the optimal flight found was not the same as the one provided in the query sample solutions, although they were very close in total delay as the sample solutions. Optimal flights are boxed in red; total_delay is the name of the last column (it's cut off in the images).

Reference for Requirement + Explanation for allowed solution discrepancy

- c) Tom wants to arrive at each destination with as little delay as possible. You can assume you know the actual delay of each flight.
- Instructors on #45: "we tolerate inconsistencies of results (within 10%) comparing to the example solutions."

1. CMI → ORD → LAX, 04/03/2008 First Leg Origin: CMI Destination: ORD Airline/Flight Number: MQ 4278 Sched Depart: 7:10 04/03/2008 Flight total delay: -14.0	Total Delay: -39 (2 routes possible) Second Leg: Origin: ORD Destination: LAX Airline/Flight Number: AA 1345 Sched Depart: 14:01 06/03/2008 Flight total delay: -25.0
2. JAX → DFW → CRP, 09/09/2008 Origin: JAX Destination: DFW Airline/Flight Number: AA 845 Sched Depart: 7:22 09/09/2008 Flight total delay: -2.0	Total Delay: -6 Origin: DFW Destination: CRP Airline/Flight Number: MQ 3627 Sched Depart: 16:48 11/09/2008 Flight total delay: -4.0
3. SLC → BFL → LAX, 01/04/2008 Origin: SLC Destination: BFL Airline/Flight Number: OO 3755 Sched Depart: 11:01 01/04/2008 Flight total delay: 13.0	Total Delay: 33 Origin: BFL Destination: LAX Airline/Flight Number: OO 5429 Sched Depart: 15:09 03/04/2008 Flight total delay: 20.0
4. LAX → SFO → PHX, 12/07/2008 Origin: LAX Destination: SFO Airline/Flight Number: WN 3534 Sched Depart: 6:50 12/07/2008	Total Delay: -41 Origin: SFO Destination: PHX Airline/Flight Number: US 412 Sched Depart: 19:16 14/07/2008 Flight total delay: -28.0

Flight total delay: -13.0	
5. DFW → ORD → DFW, 10/06/2008 Origin: DFW Destination: ORD Airline/Flight Number: UA 1104 Sched Depart: 6:58 10/06/2008 Flight total delay: -23.0	Total Delay: -31 Origin: ORD Destination: DFW Airline/Flight Number: OO6199 Sched Depart: 16:46 12/06/2008 Flight total delay: -8.0
6. LAX → ORD → JFK, 01/01/2008 Origin: LAX Destination: ORD Airline/Flight Number: UA 944 Sched Depart: 7:00 01/01/2008 Flight total delay: -4.0	Total Delay: -18 Origin: ORD Destination: JFK Airline/Flight Number: B6 918 Sched Depart: 18:53 03/01/2008 Flight total delay: -14.0

5) Optimizations

- Cleaning: Removed unused columns in data and cleaned via Python/Jupyter. This, along with gzipping, lowers the amount to be transferred, storage needed, and storage costs.
- DynamoDB: Maintained ordered and sorted results when inserting into DynamoDB using partition and sort keys. When I did not use a sort key, data ingestion took about an hour longer and was incomplete. By adding a sortkey, this optimizes the data integration process for DynamoDB to process the dataset faster and allow for sorting.
    - Partition: origin
    - Sort: averagedeparturedelay
- Data Architecture Optimization. Prior to the updated requirement to not require all the rows for Group 3.2, I was optimizing the data integration process by increasing the write throughput and increasing the EMR cluster size to speed up the data ingestion process.
    - SET dynamodb.throughput.write.percent=1.5;
    - SET mapreduce.job.maps = 20;
    - EMR Resize to 5 instead of default 3.

6) Opinion and Notes

- This was a great project, frustrating and challenging at times, having spent numerous hours figuring out how to do this and put it all together for the first time. Other technical opinions are in line with their response in their respective sections/queries/results.
- Youtube
    - For the Youtube video, it is recorded at 2x speed to fit in the original sub-5 minute requirement. You can use Youtube's playback speed to watch it at a slower pace.
    - There is no sound and that is expected. You can follow the Notes outline to understand what is being shown and more quickly than me speaking.
- Page Count: This page concludes my report, ending at 7 pages including table of contents and answers to all questions, not just minimum requirement.
    - Because the requirement for having a limit of 4-5 pages without query results was lifted, I added the table of contents + two dozen pages to add: expected grading of my Task 1 Submission according to the rubric, queries & proof of results, quick commands, and resources.
    - For the peer reviewers, you can skip the rest of the report, but I recommend you lightly review the rubric / expected grading section (next page) to see how I fulfilled all the requirements of the projects under the Excellent box.