Week 1:

In “utils.h” and “utils.cpp”, we write two functions this week to utilize for the future graph. The first function is “readEntry”, and it is used for reading data from different .dat file. We use a vector to contain the data, and push the data back every time we need it. The second function which is very helpful for our project is “getDistance”. This function converts inputs that contain two latitude and longitude and return the distance between two points. We plan to use this function to change latitude and longitude in the data file to actual distance. In this case, it would be easier to compare.

Making use of the readEntry function in “utils.cpp”, we developed “airlines.cpp”, “airports.cpp”, “planes.cpp”, and “routes.cpp” to convert the respective .dat files into maps. The values of the maps can be accessed through the “getXXID” function. Lastly, we used “airlineFlow.cpp” to turn all the maps and vectors into graphs.

Week2:

We updated airlineFlow a bit, which used “graph.h” and “edge.h” in lab_ml to produce a graph for all the airlines and airports.

Implemented Dijkstra’s algorithm in “Dijkstra.cpp”. It would find the shortest path from the source airport to the destination airport. Updated the makefile, which is mostly based on the provided codes in the cs225 folder. We can achieve “make” and “make test” easily like what we did previously in other assignments. Created test cases for all the functions.

Implemented BFS and DFS traversal algorithm in “airflow.h” and “airflow.cpp”. It could have an output that shows we go over all of the airports in the graph. BFS and DFS are useful to traverse every vertex on the graph, which is our airport.

Implemented pagerank algorithm in “pagerank.h”, “pagerank.cpp”, and improved "utils.cpp". This algorithm is to be used to find the frequencies of airlines to different airports. For example, a popular airport would have a larger frequency than smaller airports.

Implemented disjoint sets method in “dsets.h” and “dsets.cpp”. From disjoint sets, we find that some airports are disconnected to other airports on the graph. We can list out all disconnected parts from using disjoint sets.

Draw the airlines to show the path on the map, and give it color to make it more obvious..