A Study on Domestic Air Travel in the US in 2008

Thu Nguyen

Contents

The coding was done in R and can be found on GitHub here. The report was prepared in LaTeX, and was last updated on September 30, 2019.

Below are some of the interesting findings and results from the study, further findings and results will be covered as we go along in the chapters.

1. Exploratory Data Analysis

1. The busiest air routes were mostly short-distance, many of which could be completed in 5 hours or less (section 2).

2. Flight delay was more than just sometimes, in fact more than 1 would be delayed for every 3 flights. In particular, if you fly Southwest Airlines in 2008, expect more delays than on-times (section 3.1).

2. Machine Learning: Classification

1. Flight delay was common, but flight cancellation was rare, at less than 2% (section 3.2).

2. Using only information available to regular consumers at the time of booking, our models could correctly point out over 65 for every 100 flights that would indeed be cancelled (section 9).

3. Network Analysis

1. ATL (Atlanta, GA) and ORD (Chicago, IL) were indisputably the most important airports for air travel in the US (section 12.1).

2. Delta Airlines offered the most choices for intra-airline travel, however, expect to pass by ATL (Atlanta, GA) at some point (section 13.2).

3. With simple and readily available information, we could recreate the air travel network offered by Southwest Airlines, and other airlines (section 14).

1. Exploratory Data Analysis

Objectives:

1. to explore the market of domestic air travel in the US in 2008, from the perspectives of air routes, airports, airlines, and the time of travel
2. to investigate the commonality of flight delay and flight cancel, and how severe they could be

1 Data

The data was obtained from Statistical Computing: a record of all US domestic flights in 2008, including all 50 states and Washington DC, and the territories Puerto Rico and Virgin Islands.

To help with manipulating and plotting data, packages `tidyverse, ggplot2`, together with other common packages were used.

2 Overall Statistics

In this section, we will explore the market of domestic air traveling in the US in 2008. We will look at the overall size, the main competitors and related statistics from the perspectives of air routes, airports, airlines, and the time of travel.

2.1 Air Routes

First up is air routes. Table 1 gives us a look at the 30 busiest air routes, ordered by the number of total flights in both directions.

	Airport 1	City, State	Airport 2	City, State	Number of Serving Airlines	Flights $A_1 \to A_2$	Flights $A_2 \to A_1$	Number of Total Flights	Difference	Ratio	Market Share (%)
1	SFO	San Francisco, CA	LAX	Los Angeles, CA	6	13,788	13,390	27,178	398	0.015	0.388
2	OGG	Kahului, HI	HNL	Honolulu, HI	3	12,383	12,014	24,397	369	0.015	0.348
3	LGA	New York, NY	BOS	Boston, MA	3	12,035	12,029	24,064	6	0.000	0.343
4	LAX	Los Angeles, CA	LAS	Las Vegas, NV	10	11,773	11,729	23,502	44	0.002	0.335
5	LAX	Los Angeles, CA	SAN	San Diego, CA	3	11,257	11,224	22,481	33	0.001	0.321
6	LGA	New York, NY	DCA	Arlington, VA	3	11,063	11,102	22,165	39	0.002	0.316
7	LGA	New York, NY	ORD	Chicago, IL	2	10,862	10,770	21,632	92	0.004	0.309
8	HNL	Honolulu, HI	LIH	Lihue, HI	3	10,769	10,407	21,176	362	0.017	0.302
9	LGA	New York, NY	ATL	Atlanta, GA	4	10,507	10,506	21,013	1	0.000	0.300
10	LAS	Las Vegas, NV	PHX	Phoenix, AZ	3	10,626	10,337	20,963	289	0.014	0.299
11	LAX	Los Angeles, CA	PHX	Phoenix, AZ	6	9,897	9,992	19,889	95	0.005	0.284
12	ATL	Atlanta, GA	DFW	Dallas-Fort Worth, TX	7	9,847	9,849	19,696	2	0.000	0.281
13	DAL	Dallas, TX	HOU	Houston, TX	1	9,790	9,766	19,556	24	0.001	0.279
14	ATL	Atlanta, GA	MCO	Orlando, FL	2	9,613	9,611	19,224	2	0.000	0.274
15	ORD	Chicago, IL	MSP	Minneapolis, MN	4	9,688	9,356	19,044	332	0.017	0.272
16	SLC	Salt Lake City, UT	DEN	Denver, CO	6	9,269	8,905	18,174	364	0.020	0.259
17	LAX	Los Angeles, CA	SJC	San Jose, CA	4	8,908	8,939	17,847	31	0.002	0.255
18	BOS	Boston, MA	DCA	Arlington, VA	5	8,899	8,929	17,828	30	0.002	0.254
19	HNL	Honolulu, HI	KOA	Kailua/Kona, HI	3	8,745	9,038	17,783	293	0.016	0.254
20	LAX	Los Angeles, CA	DEN	Denver, CO	6	8,894	8,811	17,705	83	0.005	0.253
21	PHX	Phoenix, AZ	DEN	Denver, CO	5	8,402	8,391	16,793	11	0.001	0.240
22	DFW	Dallas-Fort Worth, TX	DEN	Denver, CO	4	8,193	8,268	16,461	75	0.005	0.235
23	LAS	Las Vegas, NV	DEN	Denver, CO	3	8,165	8,147	16,312	18	0.001	0.233
24	ORD	Chicago, IL	DFW	Dallas-Fort Worth, TX	3	8,093	8,165	16,258	72	0.004	0.232
25	LAX	Los Angeles, CA	JFK	New York, NY	3	8,058	8,078	16,136	20	0.001	0.230
26	ATL	Atlanta, GA	EWR	Newark, NJ	7	8,028	8,060	16,088	32	0.002	0.230
27	ATL	Atlanta, GA	FLL	Fort Lauderdale, FL	2	7,665	7,666	15,331	1	0.000	0.219
28	SAN	San Diego, CA	PHX	Phoenix, AZ	3	7,581	7,609	15,190	28	0.002	0.217
29	LAX	Los Angeles, CA	OAK	Oakland, CA	2	7,583	7,578	15,161	5	0.000	0.216
30	ORD	Chicago, IL	DTW	Detroit, MI	7	7,602	7,553	15,155	49	0.003	0.216

Table 1: The busiest air routes, ordered by the total number of flights in both directions in 2008. Additionally, Difference is the difference between the number of flights in each direction, Ratio is Difference over the total number of flights, and Market Share is the number of flights per routes over the total number of flights in the year 2008.

Some observations can be made:

1. The top 5 busiest routes were all short-distance routes, with the exception of OGG (Kahului, HI) - HNL (Honolulu, HI), all of which could be completed by cars in less than 6 hours.

2. Among the 10 busiest routes, LGA (New York, NY) featured in 4 routes, and LAX (Los Angeles, CA) in 3 routes, topping the charts.

3. Interestingly, even if the routes were highly popular, they were not served by many airlines, as shown in the Number of Serving Airlines column, which could support the argument for oligopoly among those popular routes.

4. At the 25^{th} position, LAX (Los Angeles, CA) - JFK (New York, NY) was the busiest coast-to-coast flight in 2008, served by 3 airlines, and accounted for 0.23% of all flights.

5. Hawaii is an interesting state. Having 3 routes in the top 30, they, however, were all intra-state routes, while the busiest route connecting Hawaii to the rest of the US was not in the list.

Alternatively, figure 1 provides a visualization of the busiest routes. The routes are color-coded according to the number of airlines serving the routes: blue is if there were more than 6, brown is if more than 3, and black is if 3 or less. Further discussion of the structure of air routes can be found in chapter 3: Network Analysis.

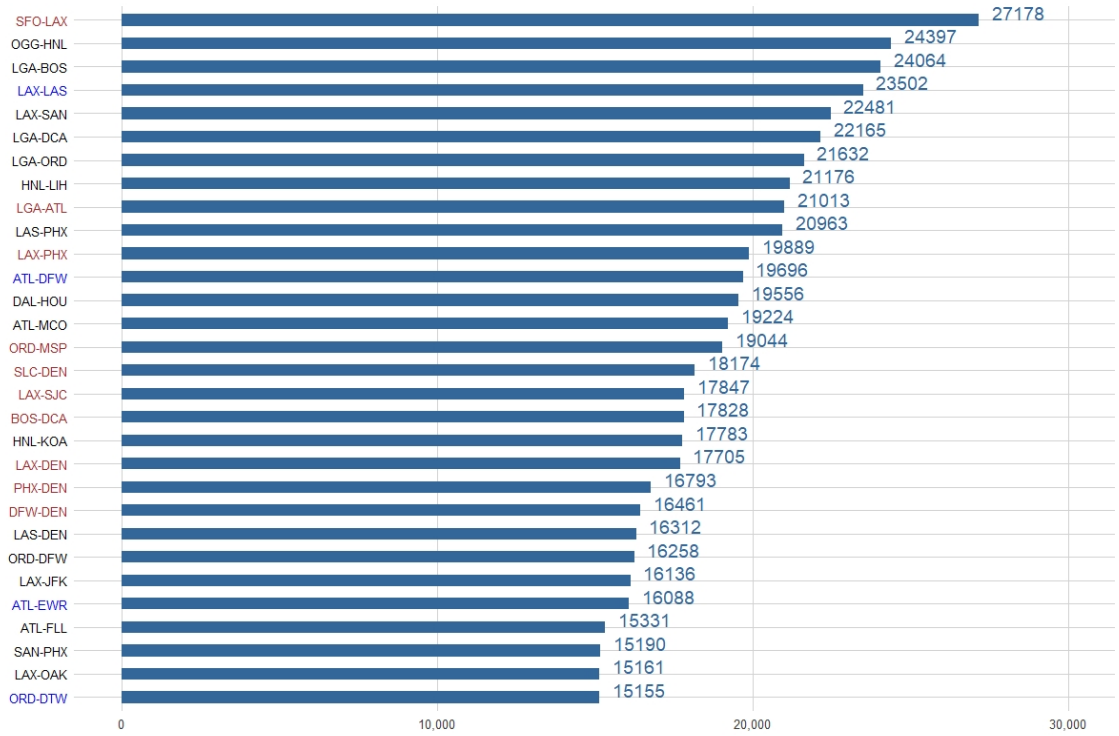

Figure 1: Plot of the busiest air routes, color-coded by the number of airlines serving the routes: blue: more than 6, brown: more than 3, and black: 3 or less.

As seen from table 1, the differences in the flight direction among the those routes were negligible, as indicated by the Difference and Ratio columns. This roughly translates to for every 1 flight in this direction, we could expect another flight in the opposite direction, completing the round trip. However, table 2 shows how certain routes could have a considerable unbalance between directions.

Some observations can be made:

1. At one extreme, DCA (Arlington, VA) - PHL (Philadelphia, PA) had an almost unbalance ratio of 1, only 5 flights were from DCA to PHL, while there were 584 flights in the other direction.

2. The unbalance of route OGG (Kahului, HI) - LIH (Lihue, HI) might suggest the popularity of sea travel between the islands, and among attractions in Hawaii in general.

	Airport 1	City, State	Airport 2	City, State	Flights $A_1 \to A_2$	Flights $A_2 \to A_1$	Number of Total Flights	Difference	Ratio
1	DCA	Arlington, VA	PHL	Philadelphia, PA	5	584	589	579	0.983
2	KOA	Kailua/Kona, HI	SEA	Seattle, WA	411	45	456	366	0.803
3	OGG	Kahului, HI	LIH	Lihue, HI	60	426	486	366	0.753
4	CLE	Cleveland, OH	IND	Indianapolis, IN	142	635	777	493	0.634
5	DCA	Arlington, VA	BUF	Buffalo, NY	632	312	944	320	0.339
6	ANC	Anchorage, AK	OTZ	Kotzebue, AK	725	361	1086	364	0.335
7	OME	Nome, AK	OTZ	Kotzebue, AK	361	725	1086	364	0.335
8	ANC	Anchorage, AK	OME	Nome, AK	365	729	1094	364	0.333
9	SJU	San Juan, PR	STT	Charlotte Amalie, VI	245	478	723	233	0.322
10	PHL	Philadelphia, PA	BUF	Buffalo, NY	365	685	1050	320	0.305

Table 2: Air routes with the highest unbalance in the numbers of flights per directions, ordered by Ratio, where airports are restricted to those with at least 366 flights, or an average of 1 flight a day (since 2008 was a leap year).

2.2 Airports

Next is a look at the airports. Table 3 lists the 28 busiest airports, ordered by the number of departing flights in 2008. With the market shares of over 1%, they were the large hubs, as classified by FAA:

- large hub: handling over 1% of total annual passenger boardings
- medium hub: handling less than 1% and more than .25%
- small hub: handling less than .25% and more than .05%
- non-hub: handling less than .05%

	Airport	City, State	Number of Neighboring Airports	Number of Serving Airlines	Number of Departing Flights	Market Share (%)
1	ATL	Atlanta, GA	173	15	414,513	5.91
2	ORD	Chicago, IL	149	15	350,380	5.00
3	DFW	Dallas-Fort Worth, TX	134	16	281,281	4.01
4	DEN	Denver, CO	127	16	241,443	3.44
5	LAX	Los Angeles, CA	90	15	215,608	3.08
6	PHX	Phoenix, AZ	88	15	199,408	2.84
7	IAH	Houston, TX	114	13	185,172	2.64
8	LAS	Las Vegas, NV	91	17	172,876	2.47
9	DTW	Detroit, MI	118	16	161,989	2.31
10	SFO	San Francisco, CA	74	15	140,587	2.01
11	EWR	Newark, NJ	92	16	138,506	1.98
12	SLC	Salt Lake City, UT	114	13	139,088	1.98
13	MCO	Orlando, FL	89	15	130,872	1.87
14	MSP	Minneapolis, MN	126	15	130,289	1.86
15	CLT	Charlotte, NC	82	15	126,045	1.80
16	LGA	New York, NY	60	15	119,135	1.70
17	JFK	New York, NY	68	11	118,804	1.69
18	BOS	Boston, MA	63	15	117,915	1.68
19	SEA	Seattle, WA	56	14	109,069	1.56
20	BWI	Baltimore, MD	65	15	104,074	1.48
21	PHL	Philadelphia, PA	61	16	100,499	1.43
22	SAN	San Diego, CA	54	17	93,775	1.34
23	CVG	Covington, KY	113	9	91,265	1.30
24	MDW	Chicago, IL	54	9	87,619	1.25
25	DCA	Arlington, VA	52	14	86,662	1.24
26	MEM	Memphis, TN	79	12	80,966	1.16
27	TPA	Tampa, FL	62	14	78,179	1.12
28	IAD	Chantilly, VA	71	14	76,031	1.08

Table 3: The busiest airports, ordered by the total number of departing flights.

This gives some observations:

1. Despite featuring frequently on the list of busy routes, LGA (New York, NY) was only the 16^{th} busiest airport, and offering direct flights to only 60 airports, about one third that of ATL (Atlanta, GA). This might be explained by the fact that New York, NY were served by several major airports, including LGA, JFK, and EWR (Newark, NJ).

2. Interesting, Hawaii did not have any large hub airports, even though it had the second busiest air routes, and a total 3 of the busiest routes.

3. There appears strong positive relation ship between the number of neighboring airports, of serving airlines, and of departing flights, as seen from table 4.

	Number of Neighboring Airports	Number of Serving Airlines	Number of Departing Flights
Number of Neighboring Airports	1.00	0.77	0.94
Number of Serving Airlines		1.00	0.65
Number of Departing Flights			1.00

Table 4: Correlation matrix among the busiest airports' statistics.

Figure 2 is a map of the location of those airports. It appears that a good portion of those were situated along the North-East and South-West corridors, two of the most populous and urbanized areas of the US. Meanwhile, the other airports scattered around the other areas, some of which were at the geographic center of the contiguous US, which suggest the role of transit hubs among those airports (such as DEN - Denver, CO).

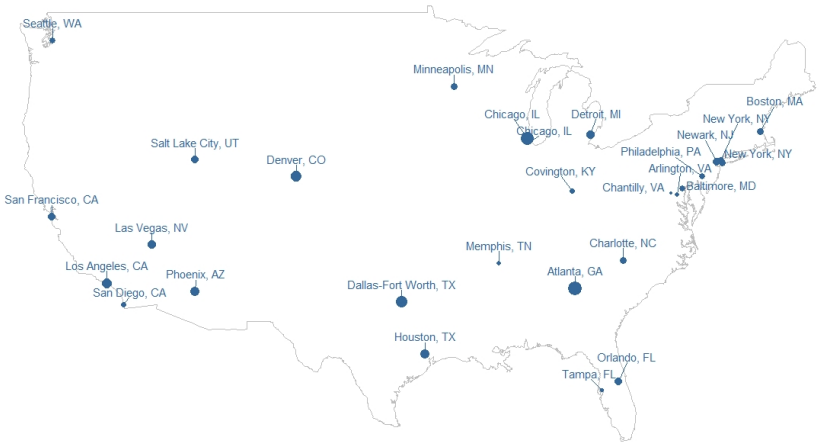

Figure 2: Map of the busiest airports in 2008, where the sizes of the airports are proportional to the number of their departing flights.

Figure 3 is a bar plot representing those airports. A quick look at it gives an impression of how busy ATL (Atlanta, GA) and ORD (Chicago, IL) were, even compared to the busiest airports. For example, compared to the 10^{th} busiest airport, SFO (San Francisco, CA), the two airports handled 3 times as many flights. Together, they were responsible for over 10% of all domestic flights in the US in 2008.

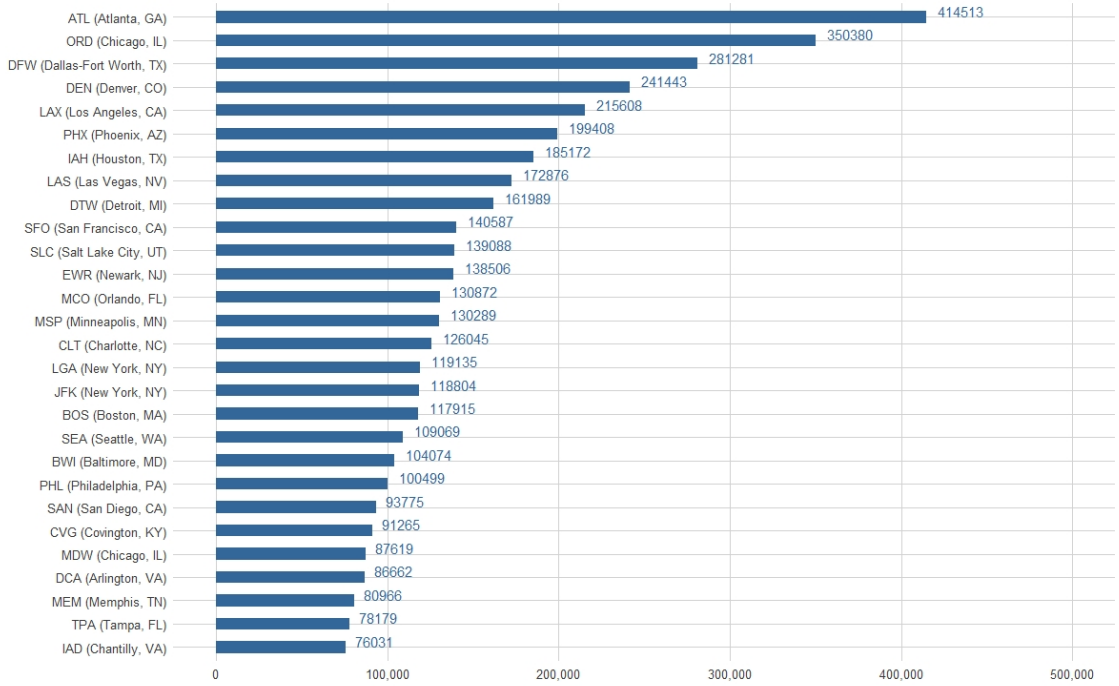

Figure 3: Plot of the busiest airports.

A more in-depth study of the connectivity of airports, their roles in the web of air travel, and their importance in airlines' planning of their air routes can be found in chapter 3: Network Analysis.

2.3 Airlines

What about the market among the airlines? From table 5, it is immediately clear that Southwest Airlines was indisputably the largest airline in the domestic air travel market in 2008. Having operated over 1.2 millions flights, that translates to an average of more than 2 airports sending off a Southwest Airlines flight every minute.

	Airline Codes	Airlines	Number of Flights	Market Share (%)
1	WN	Southwest Airlines	1,201,754	17.14
2	AA	American Airlines	604,885	8.63
3	OO	SkyWest Airlines	567,159	8.09
4	MQ	Frontier Airlines	490,693	7.00
5	US	US Airways	453,589	6.47
6	DL	Delta Airlines	451,931	6.45
7	UA	United Airlines	449,515	6.41
8	XE	ExpressJet	374,510	5.34
9	NW	Northwest Airlines	347,652	4.96
10	CO	Continental Airlines	298,455	4.26
11	EV	Envoy Air	280,575	4.00
12	9E	Pinnacle Airlines	262,208	3.74
13	FL	AirTran Airways	261,684	3.73
14	YV	Mesa Airlines	254,930	3.64
15	OH	Comair	197,607	2.82
16	B6	JetBlue Airways	196,091	2.80
17	AS	Alaska Airlines	151,102	2.16
18	F9	ExpressJet Airlines	95,762	1.37
19	HA	Hawaiian Airlines	61,826	0.88
20	AQ	Aloha Airlines	7,800	0.11

Table 5: The busiest airlines, ordered by the total number of flights in 2008.

As of 2019, the 4 biggest US airlines were Southwest Airlines, American Airlines, United Airlines, and Delta Air Lines (formerly Delta Airlines). However, the picture was quite different in 2008, as shown in figure 4.

Only Southwest Airlines and American Airlines were among the largest airlines in 2008, while United Airines and Delta Air Lines trailed behind. Although the differences were not that great in absolute terms, their ranks were low because of the presence of other airlines, some of which used

to be popular and large but are now non-existent, such as SkyWest Airlines (merged with Delta Air Lines) or US Airways (merged with American Airlines). Interestingly, of the 20 airlines in 2008, only 11 are still operating independently, while the other 9 have either merged with other airlines or ceased to operate altogether.

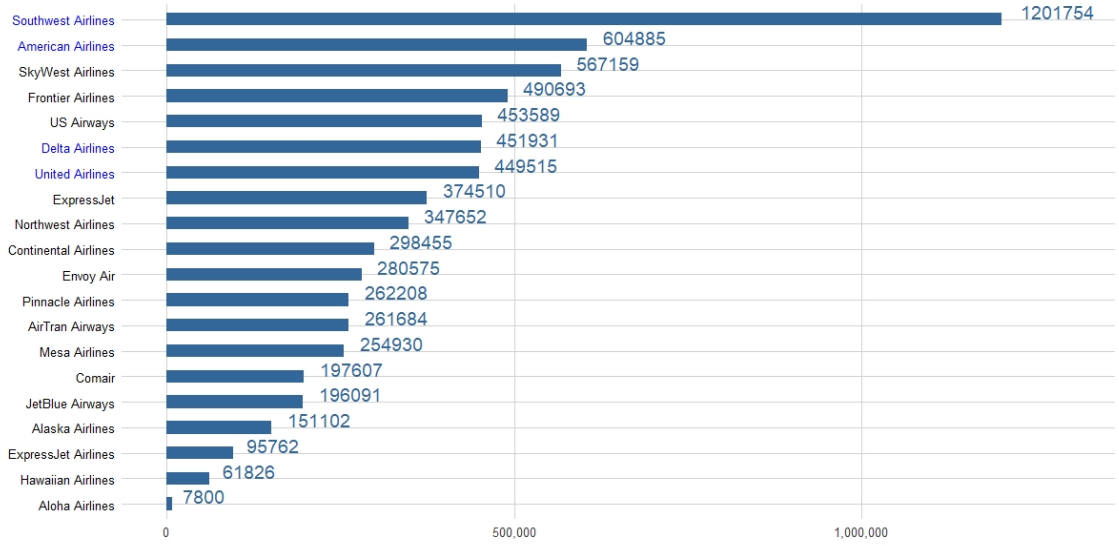

Figure 4: Plot of the busiest airlines.

A detailed examination of airlines' network structures, including the popular *hub-and-spoke* network structure, can be found in chapter 3: Network Analysis.

2.4 Time

Taking a break from the means of air travel, we will now look at when we travelled in 2008. Figure 5 is a scatter plot of the total number of domestic flights over every day in 2008. A quick reminder that 2008 was a leap year with 366 days. Together with tables 6 and 7, some observations are:

1. Weekday traveling was a lot more common than weekend traveling, by a significant margin.
2. The busiest days of week were consistently Wednesdays, followed by Mondays and Fridays.
3. After controling for the number of days in a month, it appears there was a dip in air travel after summer: the volume of air travel prior to August was considerably larger than that of after August.

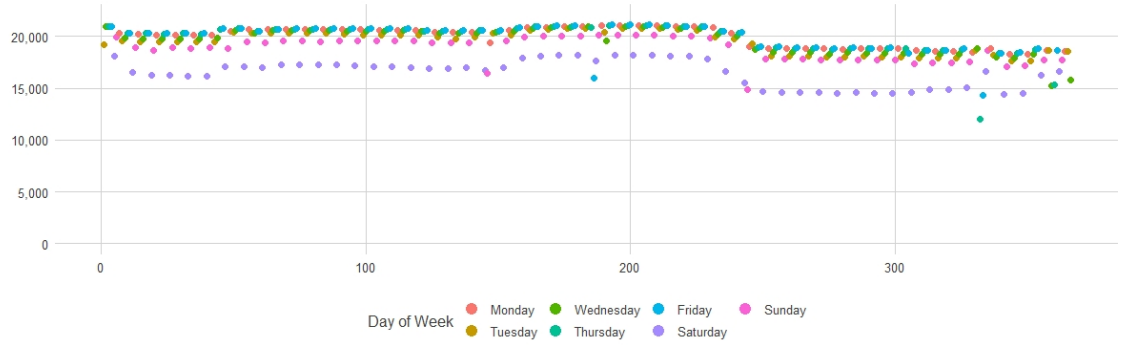

Figure 5: The distribution of domestic air travel over every day in 2008.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Average
Number of Flights	1,036,201	1,032,049	1,039,665	1,032,224	1,035,166	857,536	976,887	1,001,390

Table 6: The number of flights over days of the week in 2008.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Average
Number of Flights	605,765	569,236	616,090	598,126	606,293	608,665	627,931	612,279	540,908	556,205	523,272	544,958	584,144

Table 7: The number of flights per month in 2008.

3 Delays and Cancels

One of the most common inconveniences of air travel is arguably flight delay and/or flight cancel. Indeed, among US domestic air travel in 2008, it turned out the event of a flight delay was more than frequent, while that of a flight cancel was quite rare.

3.1 Delays

Table 8 gives a quick look at the statistics of flight delay. At almost 40%, it would be reasonable for us, passengers, to conservatively expect flight delay if we were taking a domestic flight in 2008.

	Total Number	Percentage (%)
Delays	2,700,974	38.53

Table 8: Statistics of flight delays in 2008.

3.1.1 Delays: Causes

Thanks to the Bureau of Transportation Statistics, most of those delays were classified into the 5 causes. Tables 9 and 10 give us a look at how common each cause was, and how severe the delay could be, in terms of delay duration.

	Carrier	Weather	National Air System	Security	Late Aircraft	Total
Number of Flights	670,622	99,985	928,031	6,202	699,418	2,404,258
Percentage (%)	9.57	1.43	13.24	0.09	9.98	34.31

Table 9: Statistics of causes of flight delays.

Duration (minutes)	Carrier	Weather	National Air System	Security	Late Aircraft	Total	Percentage (%)
[0,30]	445,348	56,510	689,233	5,312	362,358	1,558,761	22.24
(30,60]	118,658	20,171	141,870	629	172,377	453,705	6.47
(60,90]	47,390	9,376	44,933	155	76,032	177,886	2.54
(90,120]	23,781	5,184	22,505	56	38,573	90,099	1.29
(120,150]	13,122	3,107	12,241	26	21,528	50,024	0.71
(150,180]	7,597	1,894	7,034	10	12,111	28,646	0.41
(180,∞)	14,726	3,743	10,215	14	16,439	45,137	0.64

Table 10: Durations of flight delays per causes, where $[0, 30]$ is delayed for less than half an hour, $(30, 60]$ is between half an hour and an hour, and so on, and $(180, \infty)$ is more than 3 hours.

3.1.2 Delays: Airports

Table 11 provides us with a view at the state of flight delay with the focus on airports. At the median of 30.97%, at half of the airports, for every 10 flights, 3 flights would be delayed, which could go as high as almost 6 for every 10 flights.

	Minimum	1^{st} Quartile	Median	Mean	3^{rd} Quartile	Maximum
Percentage (%)	10.01	25.27	30.97	31.13	36.52	58.52

Table 11: Desciptive statistics of the rate of flight delays among the airports in 2008.

Figure 6 gives a visualization of the airports with the highest rates of flight cancels. Please note that only airports having at least 366 flights, or an average of 1 flight a day since 2008 was a leap year, were considered. Further details can be found in table 12.

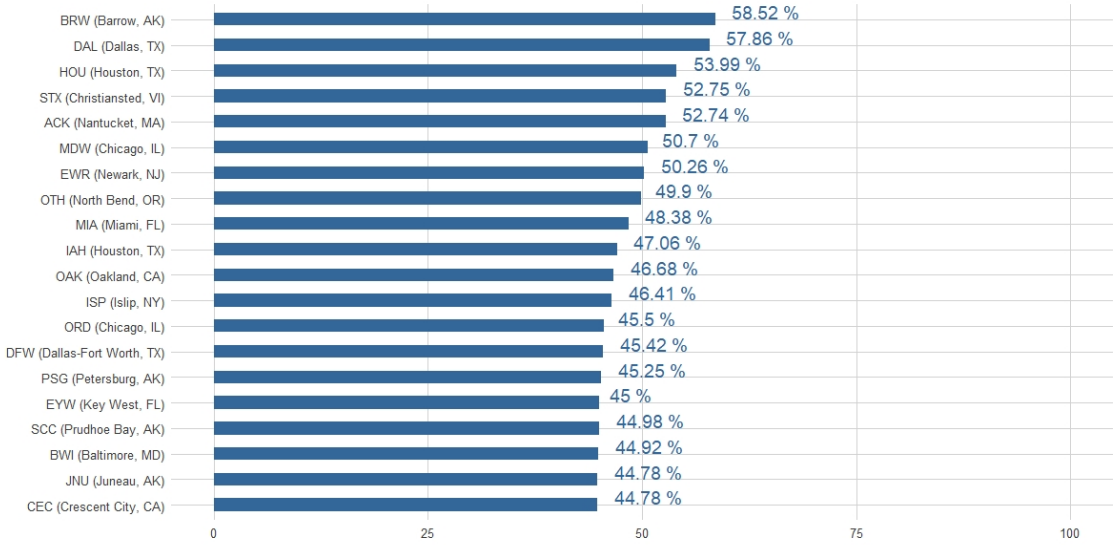

Figure 6: Plot of airports with the highest rates of flight delays in 2008, restricted to airports having at least 366 flights in 2008, or an average of 1 flight a day (since 2008 was a leap year).

	Airport	City, State	Number of Flight Delays	Number of Total Flights	Percentage (%)
1	BRW	Barrow, AK	426	728	58.52
2	DAL	Dallas, TX	31,205	53,928	57.86
3	HOU	Houston, TX	30,199	55,933	53.99
4	STX	Christiansted, VI	230	436	52.75
5	ACK	Nantucket, MA	241	457	52.74
6	MDW	Chicago, IL	44,426	87,619	50.70
7	EWR	Newark, NJ	69,612	138,506	50.26
8	OTH	North Bend, OR	257	515	49.90
9	MIA	Miami, FL	30,263	62,559	48.38
10	IAH	Houston, TX	87,139	185,172	47.06
11	OAK	Oakland, CA	29,189	62,535	46.68
12	ISP	Islip, NY	4,726	10,183	46.41
13	ORD	Chicago, IL	159,427	350,380	45.50
14	DFW	Dallas-Fort Worth, TX	127,749	281,281	45.42
15	PSG	Petersburg, AK	329	727	45.25
16	EYW	Key West, FL	454	1,009	45.00
17	SCC	Prudhoe Bay, AK	327	727	44.98
18	BWI	Baltimore, MD	46,748	104,074	44.92
19	CEC	Crescent City, CA	476	1,063	44.78
20	JNU	Juneau, AK	1,975	4,410	44.78

Table 12: Airports with the highest rates of flight delays in 2008, restricted to airports having at least 366 flights in 2008, or an average of 1 flight a day (since 2008 was a leap year).

Alternatively, we can zoom in the busiest airports and see how they fared. Table 13 gives the statistics for flight delay. A quick observation is that barring DTW (Detroit, MI), all the busiest airports had the flight rates of over 40%, which is more than the third quartile from table 11. In other words, all of the busiest airports had some of the highest rates of flight delays in 2008.

	Airport	City, State	Number of Flight Delays	Number of Total Flights	Percentage (%)
1	ATL	Atlanta, GA	175,017	414,513	42.22
2	ORD	Chicago, IL	159,427	350,380	45.50
3	DFW	Dallas-Fort Worth, TX	127,749	281,281	45.42
4	DEN	Denver, CO	104,414	241,443	43.25
5	LAX	Los Angeles, CA	87,258	215,608	40.47
6	PHX	Phoenix, AZ	82,915	199,408	41.58
7	IAH	Houston, TX	87,139	185,172	47.06
8	LAS	Las Vegas, NV	76,240	172,876	44.10
9	DTW	Detroit, MI	59,837	161,989	36.94
10	SFO	San Francisco, CA	58,200	140,587	41.40

Table 13: Rates of flight delays among the 10 busiest airports in 2008.

3.1.3 Delays: Airlines

What about flight delay among the airlines? Figure 7 is a visualization of the flight delay rates, ordered by airlines with the highest (worst) rates of flight delay. For a quick comparison, airlines in blue are the US biggest airlines as of 2019. Some observations are:

1. At over 50%, passengers flying Southwest Airlines or Continental Airlines could even expect more flight delay than not.

2. The average rate seemed to be around 36%, or a little more than 1 every 3 flights.

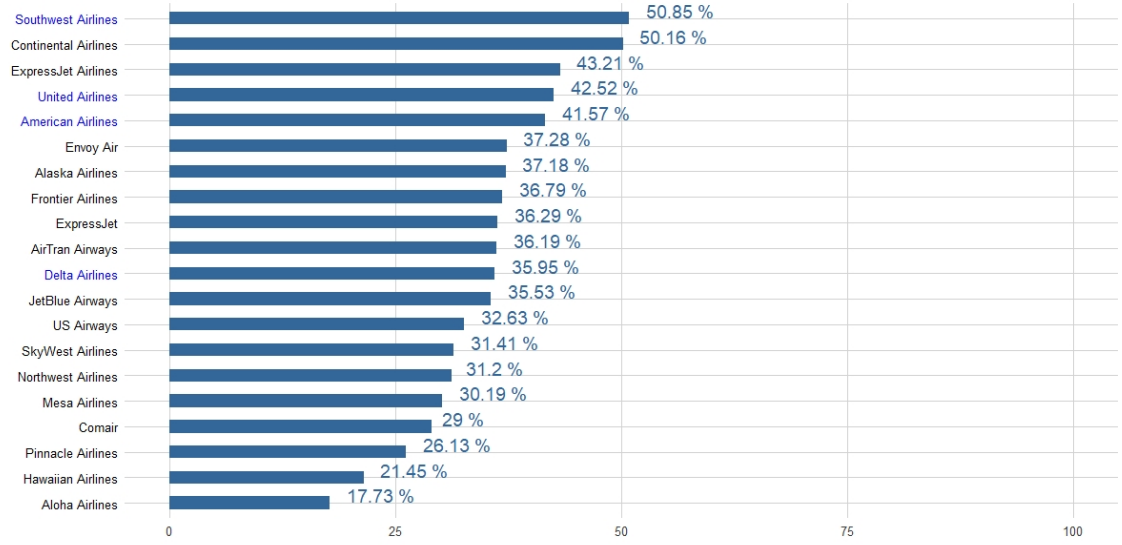

Figure 7: Flight delay among the airlines in 2008.

Table 14 gives a detailed look at the statistics, ordered by the rate of flight delay (percentage). It appears there was no strong relationship between how big an airline was and how bad the delay could be.

	Airline Code	Airlines	Number of Delayed Flights	Number of Flights	Percentage (%)
1	WN	Southwest Airlines	611,149	1,201,754	50.85
2	CO	Continental Airlines	149,707	298,455	50.16
3	F9	ExpressJet Airlines	41,381	95,762	43.21
4	UA	United Airlines	191,149	449,515	42.52
5	AA	American Airlines	251,464	604,885	41.57
6	EV	Envoy Air	104,610	280,575	37.28
7	AS	Alaska Airlines	56,182	151,102	37.18
8	MQ	Frontier Airlines	180,515	490,693	36.79
9	XE	ExpressJet	135,902	374,510	36.29
10	FL	AirTran Airways	94,706	261,684	36.19
11	DL	Delta Airlines	162,467	451,931	35.95
12	B6	JetBlue Airways	69,675	196,091	35.53
13	US	US Airways	148,024	453,589	32.63
14	OO	SkyWest Airlines	178,164	567,159	31.41
15	NW	Northwest Airlines	108,465	347,652	31.20
16	YV	Mesa Airlines	76,957	254,930	30.19
17	OH	Comair	57,302	197,607	29.00
18	9E	Pinnacle Airlines	68,511	262,208	26.13
19	HA	Hawaiian Airlines	13,261	61,826	21.45
20	AQ	Aloha Airlines	1,383	7,800	17.73

Table 14: Rates of flight delays among the airlines in 2008.

3.2 Cancels

Arguably, the only thing worse than a long flight delay is a flight cancel, if not a flight cancel after a flight delay. Fortunately, at only 1.96% (table 15), the chance of a flight cancel was indeed rare.

	Total Number	Percentage (%)
Cancels	137,434	1.96

Table 15: Statistics of flight cancels in 2008.

3.2.1 Cancels: Airports

When it comes to flight cancellation, there appear differences between the busiest airports and the less busy ones.

Figure 8 and table 16 consist of the airports with the highest rates of flight cancel, which could reach as high as 8% or 9%, effectively 1 every 11 or 12 flights.

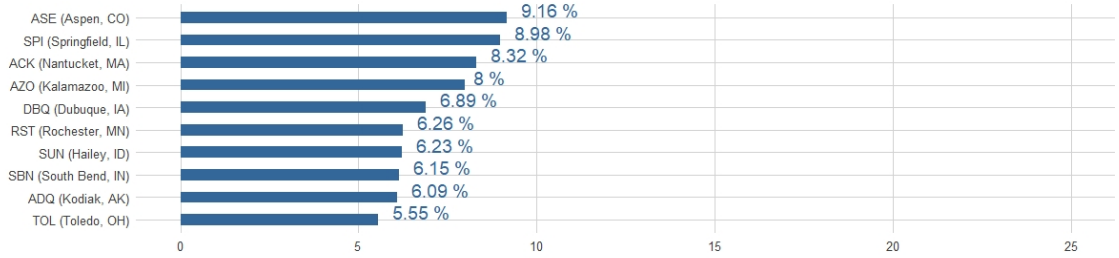

Figure 8: Plot of airports with the highest rates of flight cancels in 2008, restricted to airports having at least 366 flights in 2008, or an average of 1 flight a day (since 2008 was a leap year).

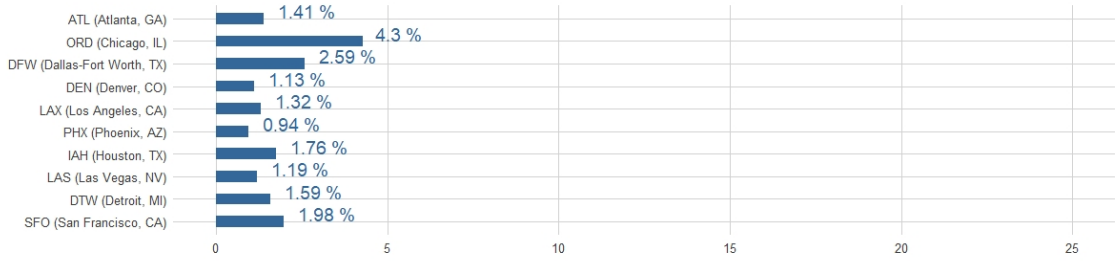

Figure 9: Plot of the rates of flight delays among the busiest airports, ordered by airports with the highest number of departing flights.

	Airport	City, State	Number of Flight Cancels	Number of Total Flights	Percentage (%)
1	ASE	Aspen, CO	486	5,307	9.16
2	SPI	Springfield, IL	110	1,225	8.98
3	ACK	Nantucket, MA	38	457	8.32
4	AZO	Kalamazoo, MI	314	3,924	8.00
5	DBQ	Dubuque, IA	93	1,349	6.89
6	RST	Rochester, MN	210	3,353	6.26
7	SUN	Hailey, ID	179	2,871	6.23
8	SBN	South Bend, IN	327	5,319	6.15
9	ADQ	Kodiak, AK	43	706	6.09
10	TOL	Toledo, OH	82	1,478	5.55
11	LSE	La Crosse, WI	111	2,005	5.54
12	FLG	Flagstaff, AZ	102	1,871	5.45
13	MBS	Saginaw, MI	158	2,918	5.41
14	CWA	Mosinee, WI	138	2,563	5.38
15	PIA	Peoria, IL	300	5,581	5.38
16	ATW	Appleton, WI	311	5,873	5.30
17	HHH	Hilton Head Island, SC	44	836	5.26
18	CMI	Champaign/Urbana, IL	143	2,774	5.16
19	OME	Nome, AK	55	1,090	5.05
20	CID	Cedar Rapids, IA	460	9,377	4.91

Table 16: Airports with the highest rates of flight cancels in 2008, restricted to airports having at least 366 flights in 2008, or an average of 1 flight a day (since 2008 was a leap year).

On the other hand, from figure 9 and table 17, which show the flight cancellation statistics among the busiest airports, some observations can be made:

1. With the exception of ORD (Chicago, IL) and DFW (Dallas-Fort Worth, TX), the busy airports seemed to operate quite well, and thus had lower rates of flight cancellation.

2. PHX (Phoenix, AZ), in particular, had the rate lower than 1%, or less than 1 every 100 flights.

	Airport	City, State	Number of Flight Cancels	Number of Total Flights	Percentage (%)
1	ATL	Atlanta, GA	5,830	414,513	1.41
2	ORD	Chicago, IL	15,050	350,380	4.30
3	DFW	Dallas-Fort Worth, TX	7,272	281,281	2.59
4	DEN	Denver, CO	2,725	241,443	1.13
5	LAX	Los Angeles, CA	2,838	215,608	1.32
6	PHX	Phoenix, AZ	1,875	199,408	0.94
7	IAH	Houston, TX	3,261	185,172	1.76
8	LAS	Las Vegas, NV	2,057	172,876	1.19
9	DTW	Detroit, MI	2,583	161,989	1.59
10	SFO	San Francisco, CA	2,790	140,587	1.98

Table 17: Rates of flight cancels among the 10 busiest airports in 2008.

3.2.2 Cancels: Airlines

In terms of flight cancellation among airlines, there also appear differences compared to that of flight delay, as can be seen from figure 10 and table 18:

1. There appears little relationship between flight delay and flight cancel at the level of airlines: having a high rate in one measure did not necessarily translate to a high rate in the other measure.

2. The same weak relationship goes for between the rate of flight cancel and how big the airline was.

3. Southwest Airlines is an interesting case: whereas they had the highest rate of flight delay, they had competitively low rate of flight cancel, at only 1.03%, meaning while their passengers should expect frequent flight delay, they could expect to be able to get on the flights nonetheless.

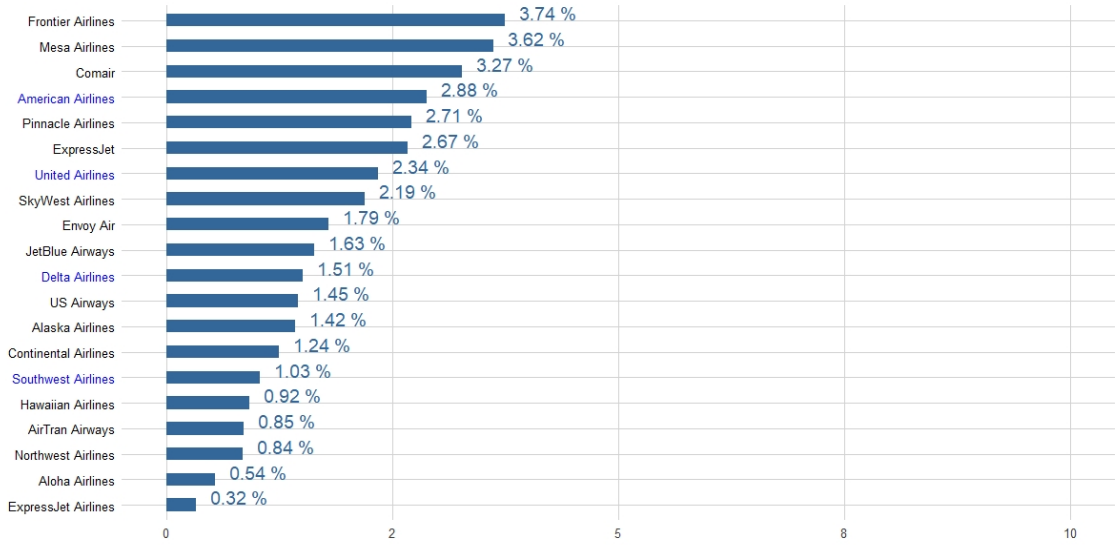

Figure 10: Flight cancel among the airlines in 2008.

	Airline Code	Airlines	Number of Cacncelled Flights	Number of Flights	Percentage (%)
1	MQ	Frontier Airlines	18,331	490,693	3.74
2	YV	Mesa Airlines	9,219	254,930	3.62
3	OH	Comair	6,462	197,607	3.27
4	AA	American Airlines	17,440	604,885	2.88
5	9E	Pinnacle Airlines	7,100	262,208	2.71
6	XE	ExpressJet	9,992	374,510	2.67
7	UA	United Airlines	10,541	449,515	2.34
8	OO	SkyWest Airlines	12,436	567,159	2.19
9	EV	Envoy Air	5,026	280,575	1.79
10	B6	JetBlue Airways	3,205	196,091	1.63
11	DL	Delta Airlines	6,813	451,931	1.51
12	US	US Airways	6,582	453,589	1.45
13	AS	Alaska Airlines	2,139	151,102	1.42
14	CO	Continental Airlines	3,702	298,455	1.24
15	WN	Southwest Airlines	12,389	1,201,754	1.03
16	HA	Hawaiian Airlines	570	61,826	0.92
17	FL	AirTran Airways	2,236	261,684	0.85
18	NW	Northwest Airlines	2,906	347,652	0.84
19	AQ	Aloha Airlines	42	7,800	0.54
20	F9	ExpressJet Airlines	303	95,762	0.32

Table 18: Rates of flight cancel among the airlines in 2008.

2. Machine Learning

Objectives:

1. to build a classification model predicting if a flight will be cancelled given information available to consumers at the time of booking
2. to handle highly imbalance and huge data set
3. to evaluate and analyze built models for future improvements

4 Motivation

Born out of personal and other people's inconvenience experience during the event of last minute flight cancellation, and the extent to which that could affect subsequent travel plans, my goal is to build a predictive model that helps predict if a flight would be cancelled, given only information available to us, regular consumers, at the time of purchasing. As shown in part 1, the chance of a flight being cancelled is indeed very small, around 1.96%. However, what matters is the ability to correctly predict if a flight would be cancelled, or the sensitivity measure.

5 Data

The data was obtained from Statistical Computing: a record of all US domestic flights in 2008. The data was then narrowed down to consist of flights from and to all US states and Puerto Rico. Supplementary information such as the airport's city's population demographic and city area was taken from the 2010 US Census Survey.

After preliminary investigation of the data set, and considering what types of information are usually available to a regular consumer at a time of booking, either explicitly or implicitly, table 20 lists all the features, or variables, to be used for building predictive models.

Features	Explanation
Carrier	a 12-level factor feature for airlines operating in 2008, including Southwest Airlines, American Airlines, United Airlines, and Delta Air Lines, among others
CRS Elapsed Time	a numeric feature for the scheduled flight duration, in minutes
CRS Departure Time	a 4-level factor feature, partitioning scheduled departure time into 6-hour blocks: $0-6, 6-12, 12-18$, and $18-24$ hours
CRS Arrival Time	similar to *CRS Departure Time*, but for scheduled arrival time
Distance	a numeric feature for the flight distance, in miles
Day	a 7-level factor for the day of week of the flight
Month	a 12-level factor for the month of the flight
Holiday	a binary feature for if the day of the flight is a US national holiday
Origin	a 4-level factor for the hub level of the departure airport, according to FAA: • large hub: handling over 1% of total annual passenger boardings • medium hub: handling less than 1% and more than .25% • small hub: handling less than .25% and more than .05% • non-hub: handling less than .05%
Destination	similar to *Origin*, but for the destination airport
Origin Capital	a binary feature for if the departure airport serves the state capital
Destination Capital	a binary feature for if the destination airport serves the state capital

Table 20: Features to be used in building predictive models.

To help with building models, packages `caret, e1071, rpart, kernlab, klaR`, together with other common packages were used.

6 Challenges

Upon preliminary investigation, the current data set presents two challenges that would require special attention:

1. Huge data set: at over $1,700,000$ observations.
2. Highly imbalance data set: where the positives (cancelled flights) make up of only 1.96% of all observations.

6.1 Huge data set

Whereas it is generally desirable to have more data, when faced with building models on personal computers, one needs to take into account the complexity of the algorithms, and the computational costs. Thus, building a model utilizing all the available data would not be feasible.

In tackling this, I divided algorithms into groups, depending on the computational costs:

- *Faster algorithms*: logistic regression, linear discriminant analysis, quadratic discriminant analysis, and flexible discriminant analysis.
- *Slower algorithms (1)*: neural networks, random forest, stochastic gradient boosting, and k-nearest neighbors (which is faster than the other within-group algorithms, but can become considerably slow given larger data set to be classified into the *faster algorithms* group).
- *Slower algorithms (2)*: support vector machines: linear kernel and radial kernel.

Each of these algorithms might use different data set for the purpose of training and evaluating models. Details can be found in section 7.2. Additionally, the underlying ideas of all the algorithms are discussed in section 8.

6.2 Highly imbalance data set

At the ratio of almost $50:1$, a simple guess of always predicting negative (a flight not being cancelled) can easily give us an accuracy of over 98%. However, such a guess would completely overlook the event that a flight would in fact be cancelled.

In tackling this, I planned to employ the 3 different techniques:

- *Alternate Cut-offs*: instead of .5, the threshold will be chosen as the point "closest" to the top-left corner on the ROC curve, where "closest" is the Euclidean distance.
- *Resampling*: the idea is to rebalance the distribution of the 2 classes, either by *up-sampling* or *down-sampling*, given the already huge data set, a *down-sampling* would be sufficient here.
- *Cost-sensitive training*: the idea is to associate wrong predictions (whether a false positive or a false negative) with some pre-defined costs, the ratio of which can emphasize the importance of correctly predicting a flight cancellation.

7 Methodology

Since *Alternate Cut-offs* and *Resampling* techniques are similar (the differences are in how to choose the cut-offs and the data used for training, all else being equivalent), I will employ the same algorithms for these 2 approaches: the *faster algorithms* and *slower algorithms (1)*, as introduced in section 6.1.

The third technique, *Cost-sensitive training*, being fundamentally different from the other approaches, will make use of the *slower algorithms (2)*, hereafter to be referred to as *cost-sensitive algorithms*.

Each model will be trained using the `caret` package. For parameter tuning, a 5-fold cross-validation is used, after which models are evaluated accoring to the perfomance metrics from section 7.1. For details on how training and evaluation data sets are selected, please refer to section 7.2.

Additionally, a test set is held out, to be tested on by the best algorithms per approach. The result on test set can be found in section 9.4.

7.1 Performance metrics

Predicting if a flight would be cancelled is, at heart, a classification problem, where the predicted variable is binary: 1 if predicted cancelled, and 0 otherwise. As such, in evaluating models, it would be helpful to refer to the confusion matrix:

		Actual event	
		Yes	No
Prediction	Yes	TP	FP
	No	FN	TN

Table 21: Confusion matrix, where a "Yes" event is the event of a flight being cancelled.

where

- TP: the number of true positives: predicted cancelled flight is in fact cancelled
- FP: the number of false positives: predicted cancelled flight is not cancelled
- TN: the number of true negatives: predicted not cancelled flight is in fact not cancelled
- FN: the number of false negatives: predicted not cancelled flight is cancelled
- N: the number of total observations

Table 23 gives a summary of 6 metrics that are relevant to the problem of classification and imbalance data set, based on the statistics from the confusion matrix above. Depending on the techniques and algorithms, some metrics will be given higher weights than the others in comparing algorithms.

Metric	Explanation
AUC	the area under the curve of the ROC curve
Accuracy	the ratio of correctly labelled events, $$Accuracy = \frac{TP + TN}{N}$$
Sensitivity	the ratio of correctly predicted positive events, $$Sensitivity = \frac{TP}{TP + FN}$$
Specificity	the ratio of correctly predicted negative events, $$Specificity = \frac{TN}{TN + FP}$$
κ	the overall accuracy normalized by the imbalance of the classes in the data, $$\kappa = \frac{\text{Accuracy} - \text{Random Accuracy}}{1 - \text{Random Accuracy}}$$ where $$\text{Random Accuracy} = \frac{(TN + FP)(TN + FN) + FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2}$$
F_1	harmonic mean of *sensitivity* and *specificity*, $$F_1 = 2 \times \frac{Sensitivity \times Specificity}{Sensitivity + Specificity}$$

Table 23: Performance metrics.

7.2 Approaches to Imbalance Data

7.2.1 Alternate Cut-offs

The idea behind *alternate cut-offs* is to look for a cut-off point that would balance out the *sensitivity* and *accuracy*: trading some *accuracy* for supposedly much greater improvements in *sensitivity*, in order to correctly predict the rare event of flight cancellation.

Here, both *faster algorithms* and *slower algorithms* follow the same procedure. The difference is in the size of the training and evaluation sets:

- *faster algorithms*: 100,000-observation training and 50,000-observation evaluation sets;
- *slower algorithms*: 10,000-observation training and 5,000-observation evaluation sets.

Each algorithm will repeatedly build models based on appropriately sized training and evaluation sets 10 times, where each training and evaluation set is sampled so that they are representative of the entire data via the *Goodness-of-fit* test (discussion of which can be found in appendix 15). In evaluating the modes, the cut-offs are set to be the points closest to the top-left corner in the ROC curve, as illustrated in 11. The performance metrics from all models are then averaged. The model "closest" to the average is then chosen to be the representative, to be used in comparing algorithms. In other words, each algorithm is evaluated with 10-fold cross-validation to determine the performance.

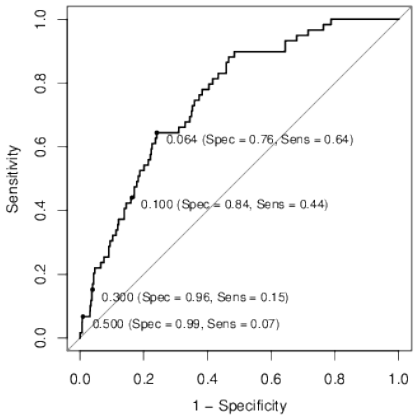

Figure 11: Alternate cut-offs. Adapted from 'Applied Predictive Modeling,' by Max Kuhn and Kjell Johnson, 2013, New York, NY: Springer.

To determine the representative models for each algorithm, suppose each model has the associated performance metrics

$$m^{(i)} = \left(m^{(i)}_{AUC},\ m^{(i)}_{Acc},\ m^{(i)}_{Sen},\ m^{(i)}_{Spe},\ m^{(i)}_{\kappa},\ m^{(i)}_{F_1}\right),$$

then the average metrics for that algorithm is

$$\overline{m} = \frac{1}{10}\sum_{i=1}^{10} m_{(i)}$$

The distance between each model and the average is defined as

$$dist(m^{(i)}, \overline{m}) = \sqrt{3(m^{(i)}_{Sen} - \overline{m}_{Sen})^2 + 2(m^{(i)}_{AUC} - \overline{m}_{AUC})^2 + (m^{(i)}_{F_1} - \overline{m}_{F_1})^2} \tag{1}$$

which is, in effect, the Euclidean distance, or l^2-norm, with the distances along the *accuracy, specificity*, and κ replaced by 2 times the distances along *sensitivity*, and 1 time that along AUC. The model with the smallest distance is chosen to represent the algorithm.

In summary, the procedure is:

- For each algorithm:
  1. For $i = 1, 2, \ldots, 10$:
     1.1. Find appropriately sized and representative training and evaluation subsets.
     1.2. Train the model, and evaluate based on the cut-off being the point closest to the top-left corner in the ROC curve.
     1.3. Evaluate the model, and return the associated performance metrics.
  2. Average all the 10 performance metrics.
  3. Calculate the distance between each model and the average, according to formula 1.
  4. Return the model with the smallest distance to represent the algorithm.

7.2.2 Resampling

The idea is to resample the data such that the distributions of classes are balanced; in this case, we aim for a more balanced distribution of *cancelled* and *not cancelled* flights. Of the 3 major ways: *up-sampling, down-sampling* and *SMOTE* (which is a hybrid of both *up-* and *down-sampling*), thanks to the already huge data set to begin with, just a *down-sampling* would be sufficient.

Given 135,808 number of cancelled flights in the current data set (after holding out the common test set), after a random *down-sampling*, the resulting resampled data set has 271,616 observations, which can be handled well by the *faster algorithms* from section 7.2.1.

Thus, in training models based on *resampling* approach, *faster algorithms* will use all the resampled data set for training and evaluating models, whereas *slower algorithms* will follow the same 10-fold cross-validation on the algorithms' performances procedure in section 7.2.1 with respect to the new resampled data, keeping the threshold fixed at .5.

7.2.3 Cost-sensitive training

This third approach is based on the idea that different mistakes will have different costs: mistakes deemed more undesirable will carry higher costs (or weights) relative to the less costly mistakes.

Support vector machines (SVM) is an algorithm that allows for the specification of asymmetrical costs, referred to as case weights in this context. SVM will be trained with *linear* and *radial* kernels, coupled with a range of costs (case weights, or simply weights).

Since the algorithms are quite computational heavy, they will follow the same 10-fold cross-validation on the algorithms' performances procedure used previously for *slower algorithms* in secion 7.2.1. More details about the algorithms are discussed in section 8.

A note on the technical detail regarding the SVM algorithms: a weighted SVM model does not return class probability, and thus the metric AUC is not available. As such, in calculating the distance of each model from the average, the formula is

$$dist(m^{(i)}, \overline{m}) = \sqrt{3(m^{(i)}_{Sen} - \overline{m}_{Sen})^2 + (m^{(i)}_{Spe} - \overline{m}_{Spe})^2 + (m^{(i)}_{\kappa} - \overline{m}_{\kappa})^2 + (m^{(i)}_{F_1} - \overline{m}_{F_1})^2} \quad (2)$$

8 Algorithms

Table 24 is my attempt to consolidate the underlying ideas behind the algorithms in my own words. The references are the books *Applied Predictive Modeling* by Max Kuhn and Kjell Johnson, *An Introduction to Statistical Learning, with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, and *The Elements of Statistical Learning* by Trevor Hastie, Robert Tibshirani, and Jerome Friedman, among other resources.

Please note that the explanations hereafter refer to the case of a binary classification. Some algorithms (such as *neural networks* and *linear* and *quadratic discriminant analysis*) extend naturally to the case of multi-class classification, while others (*logistic regression* and *support vector machines*) are less so.

For illustration, let $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_p^{(i)})$ be a column vector of length p, where each $x_1^{(i)}, \ldots, x_p^{(i)}$ is the feature of the training point i, and the class $y^{(i)}$.

Algorithms

logistic regression

- Models the variable based on the assumption of a Bernoulli distribution with the probability parameter p such that:

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = \beta_0 + \beta X$$

  where the right hand side is a linear combination of predictors and a constant; taking exponential gives the logistic function

$$p = \frac{e^{\beta_0 + \beta X}}{1 + e^{\beta_0 + \beta X}} \tag{3}$$

  giving p between 0 and 1, to be compared against some threshold p_0.

parameters for tuning: none

linear discriminant analysis (LDA)

- The idea is to model the distribution of features X per response class, and to then use Bayes' rule to flip into the probability of certain response class given features X.
- LDA assumes that for each data point the distribution of the features given the class of the data point follows a multi-variate normal distribution with equal covariance matrix for all classes, in particular, for all class i,

  $$X|Y = i \sim \mathcal{N}(\mu_i, \Sigma)$$

  where a multi-variate normal distribution means that each feature follows a 1-dimensional normal distribution and any linear combination of the features is also normally distributed.
- Under said assumption, Bayes' rule can then be used to find out the probability of the class for the current data point given current variables via

  $$\mathbb{P}(Y = i|X) = \frac{\mathbb{P}(X|Y = i)\ \mathbb{P}(Y = i)}{\mathbb{P}(X)}.$$

parameters for tuning: none

---

quadratic discriminant analysis

- An extension of *linear discriminant analysis*, where the assumption of multi-variate normal distribution is relaxed to allow for different covariance matrices, in particular, for all class i,

  $$X|Y = i \sim \mathcal{N}(\mu_i, \Sigma_i)$$

  and that Σ_i are not necessarily equal to each other.

parameters for tuning: none

---

flexible discriminant analysis

- Another extension of *LDA*, to allow for non-linear decision boundary.
- In doing so, a common choice is through the hinge functions from Multivariate Adaptive Regression Splines (MARS) approach, which allows for flexible decision boundary.

parameters for tuning:

- *degree*: degree of the hinge function, fixed at 1
- *nprune*: number of variables, in range $\{2, 4, 6, 8, 10\}$

---

neural networks

- The idea is to combine variables into derived variables, and then model the target random variable as a function of those derived variables.
- At the fundamental level, a neural network architecture consists of an input layer, a hidden layer (which may be made up by more than 1 layer, each of which consists of a number of variables, or neurons), and an output layer.
- At the first step, each neuron in the hidden layer can be activated by some linear combination of input variables (there are many choices for activation functions, a common one is the *sigmoid* function, similar formula 3).
- At each subsequent step, the prior layer becomes the input layer, feeding into the next layer, until the target layer, the class with the highest probability will be chosen.

parameters for tuning:

- *size*: number of hidden units, in range $\{1, 4, 7, 10\}$
- *decay*: decay coefficient, in range $\{.1, 1, 2\}$
- number of layers fixed at 1, by default

---

random forest

- A variation of *bagging* (short for *bootstrap aggregation*), with a twist.
- The idea is a more stable and less variable prediction can be obtained if the trees are less correlated, since otherwise, if there exist some variables imposing great influence, the resulting trees will heavily feature those variables at the expense of others.
- In implementing random forests, at each split in the tree, only a number of variables is considered; this decorrelates the trees, preventing the case of certain variables being selected more than desired; this results in the average of the trees being less variable and more stable.
- Similar to *bagging* and other algorithms under the *tree-based method* family, after n trees are built, a new data point will be predicted according to the majority vote from the n trees.

parameters for tuning:

- *m.try*: number of variables to used for building trees, in range $\{10, 20, 30, 40, 50\}$
- *n.trees*: number of trees built, fixed at 500, by default

---

stochastic gradient boosting

- A variation of *bagging*, with the focus on improving the performance of weak learners (trees whose performances are only slightly better than random chance).
- The idea is it is beneficial to quickly build trees (simpler trees at smaller depths), evaluate the performances, look for where the trees under-performm the most and modify the trees accordingly; such succession of building and modifying trees would supposedly improve the performance of the models.

parameters for tuning:

- *interaction.depth*: depth of trees, in range $\{4, 7, 10\}$
- *shrinkage*: shrinkage coefficient, in range $\{.001, .01, .1\}$
- *n.trees*: number of trees built, fixed at 1000
- *n.minobsinnode*: minimum number of terminal nodes, fixed at 10

***k*-nearest neighbors (*k*-NN)**

- Different from the other algorithms, k-NN algorithm is memory-based, and requires no model fitting; instead, to predict a new data point, the algorithm calculates the distances[1] from that point to all existing data points in the training set,

$$dist(X^{(i)}, X) = \sqrt{(x_1^{(i)} - x_1)^2 + \cdots + (x_p^{(i)} - x_p)^2}$$

  where X is the column vector of features from the new data point.
- The classes associated with the k smallest distances are recorded, the majority of which will be the class of that new data point.

parameters for tuning:

- k: number of nearest neighbors, in range $\{1, 5, 9, 13, 17\}$

---

support vector machines (SVM): linear kernel

- SVM is an extension of *support vector classifier* (SVC): enlarging the feature space using *kernels* to accomodate complex boundary between the classes.
- Firstly, SVC aims to find the *best* separating hyperplane: one that maximizes the margins between the 2 classes, where the hyperplane, which divides the p-dimensional space into 2 halves, is defined as:

  $$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0$$

  where $(\beta_0, \ldots, \beta_p)$ are coefficients to be determined.
- As such, if given the data point $X(i)$ defined above and the true class of $y_i \in \{-1, 1\}$, one can quickly see that such the model has correctly labelled the point if:

  $$y_i \ (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq 0$$

  where
  - y_i is the true class, defined as 1 or -1
  - $(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$ is the predicted class: a positive sign corresponds to the class of 1, and -1 otherwise.

- Secondly, in enlarging the feature space from SVC to SVM, each data point undergoes the transformation

  $$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(x, x_i)$$

  where $K(x, x_i)$ is the *kernel* function, measuring the similarity between observations.
- Thus, SVM with *linear* kernel is one with the kernel

  $$K(x_i, x_j) = \sum_{k=1}^{p} x_{ik} x_{jk}.$$

- Thirdly, in the context of *cost-sensitive training*, weights are assigned to each classes, throughout the model building process, the weight for the *notcancelled* class is kept at 1, while that of the *cancelled* class varies in range $\{1, 10, 25, 50, 75, 100\}$.

parameters for tuning:

- C: the tolerance how much mis-classification is allowed, in range $\{.0625, .125, .25, .5, 1, 2\}$
- *case weights*: technically not a parameter for tuning, but will be manually specified for each iteration, in range $\{1, 10, 25, 50, 75, 100\}$

SVM: radial kernel

- Similarly, SVM with *radial* kernel is an example of a SVM with non-linear decision boundary; of which the kernel is

$$K(x_i, x_j) = \exp\left\{\gamma \sum_{k=1}^{p} (x_{ik} - x_{jk})^2\right\}$$

- Additionally, in the case of SVM with *radial* kernel, the weights range over $\{1, 50, 100\}$.

parameters for tuning:

- C: as above, in range $\{.0625, .125, .25, .5, 1, 2\}$
- *case weights*: as above, in range $\{1, 50, 100\}$

Table 24: Overview of the algorithms.

[1] unless otherwise specified, all distances are defined as Euclidean distances

9 Results

9.1 Alternate Cut-offs

Following the procedure as outlined in section 7.2.1 under the *alternate cut-offs* approach, table 25 gives the performance metric and the threshold averages for each algorithm in no particular order.

Algorithm	AUC	Accuracy	Sensitivity	Specificity	κ	F_1	Threshold
Logistic Regression	0.713	0.662	0.656	0.662	0.035	0.071	0.021
Linear Discriminant Analysis	0.706	0.657	0.657	0.657	0.034	0.070	0.020
Quadratic Discriminant Analysis	0.705	0.664	0.649	0.664	0.035	0.071	0.056
Flexible Discriminant Analysis	0.632	0.625	0.581	0.626	0.021	0.057	0.018
Neural Networks	0.659	0.628	0.628	0.628	0.027	0.064	0.019
Random Forests	0.590	0.612	0.537	0.613	0.016	0.054	0.014
Gradient Boosting	0.647	0.620	0.617	0.620	0.025	0.063	0.017
k-NN	0.439	0.727	0.377	0.734	0.017	0.054	0.044

Table 25: Averages of performance metrics and threshold for each algorithm under the *alternate cut-offs* approach.

Additionally, figure 12 gives a visual plot of the 95% confidence interval for the 6 performance metrics based on the averages and standard errors, and how the algorithms compare againts the others. Some observations are:

1. *Logistic regression, linear discriminant analysis*, and *quadratic discriminant analysis* all have comparable performance for the 6 metrics, which collectively are considerably better than that of the other algorithms.
2. It appears that, overall, *faster algorithms* have better performance than *slower algorithms* in almost all metrics.
3. k-NN algorithm is an interesting case: while k-NN gives some of the best accuracy, k-NN trails behind in sensitivity significantly, probably due to the fact that the data is highly imbalance.

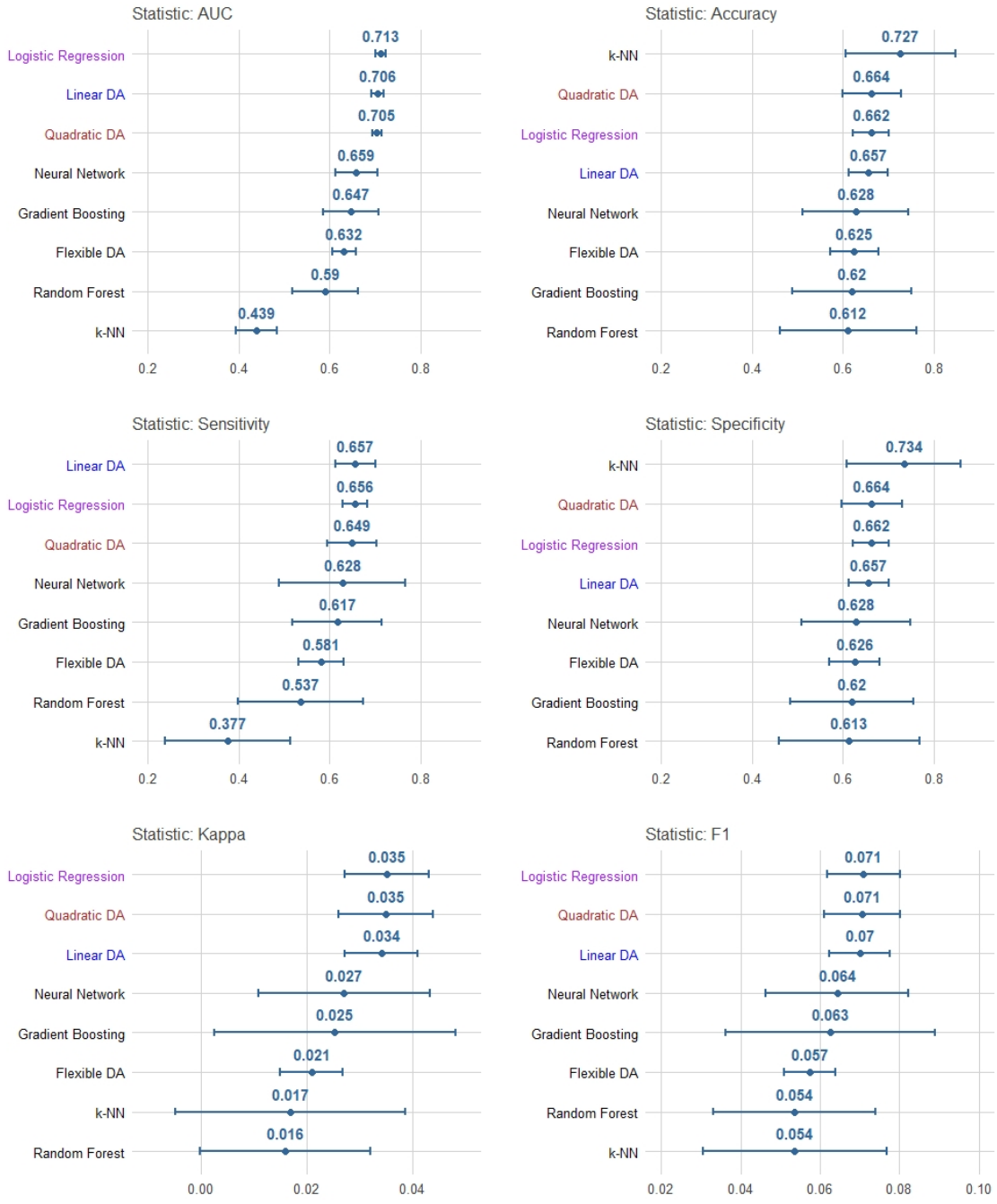

Figure 12: Performance metrics for each algorithm under the *alternate cut-offs* approach.

Since the 3 algorithms *logistic regression, linear discriminant analysis*, and *quadratic discriminant analysis* consistently have better performance metrics, they will be selected to represent the *alternate cut-offs* approach. Recall the distance formula 1 in selecting the model most representative of the algorithm, table 26 gives the summary of the representative models for the top 3 algorithms.

Algorithm	AUC	Accuracy	Sensitivity	Specificity	κ	F_1	Threshold
Logistic Regression	0.709	0.654	0.654	0.654	0.034	0.072	0.021
Linear Discriminant Analysis	0.711	0.670	0.657	0.671	0.036	0.071	0.020
Quadratic Discriminant Analysis	0.700	0.658	0.653	0.658	0.034	0.069	0.059

Table 26: Representative models from the top 3 algorithms under the *alternate cut-offs* approach.

For illustration of how the 3 algorithms compare to each other, figure 13 gives a visual plot with respect to the performance metrics. A quick observation is that *quadratic discriminant analysis* allows for a significantly higher threshold of .059 compared to the other 2 algorithms, otherwise all the metrics are comparable.

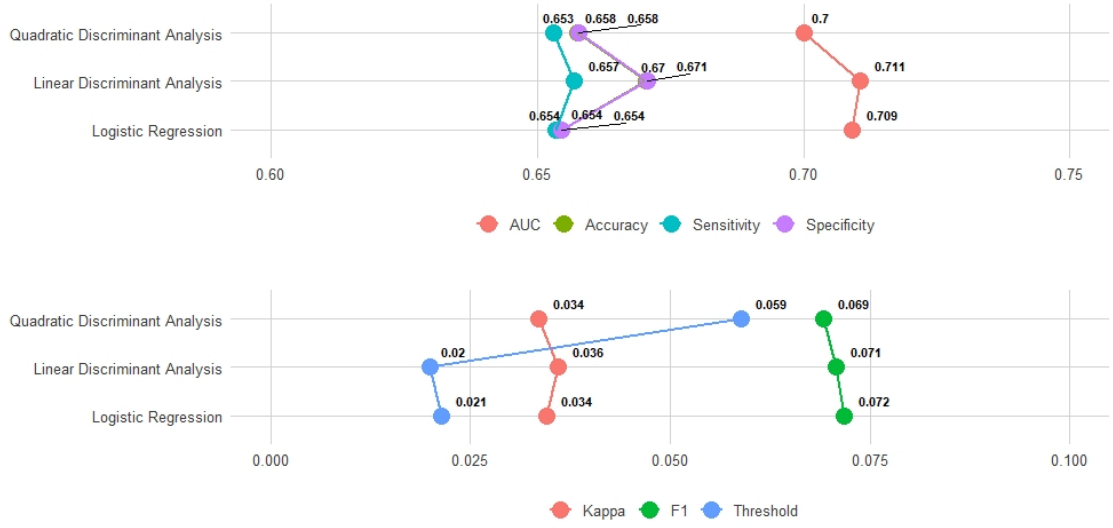

Figure 13: Representative models from the top 3 algorithms under the *alternate cut-offs* approach.

9.2 Resampling

Similar to above, table 27 gives the averages of the performance metrics for each algorithm under the *resampling* approach.

Algorithm	AUC	Accuracy	Sensitivity	Specificity	κ	F_1
Logistic Regression	0.712	0.654	0.680	0.629	0.309	0.663
Linear Discriminant Analysis	0.710	0.653	0.684	0.623	0.307	0.664
Quadratic Discriminant Analysis	0.706	0.651	0.722	0.580	0.302	0.674
Flexible Discriminant Analysis	0.675	0.628	0.706	0.551	0.257	0.655
Neural Networks	0.725	0.668	0.684	0.652	0.336	0.673
Random Forests	0.724	0.663	0.693	0.633	0.327	0.674
Gradient Boosting	0.734	0.673	0.688	0.658	0.346	0.678
k-NN	0.691	0.638	0.728	0.547	0.275	0.668

Table 27: Averages of performance metrics and threshold for each algorithm under the *resampling* approach.

A visual plot of those performance metrics can be seen in figure 14. Some observations are:

1. Unlike under the *alternate cut-offs* approach, here *neural networks, random forests*, and *stochastic gradient boosting* consistently have considerably better performance statistics than the other algorithms.

2. k-NN is still an interesting case: performing well under the sensitivity metric but trailing behind at all the other metrics.

3. *Quadratic discriminant analysis* is also an interesting case: having very competitive sensitivity and F_1 metric, while the other metrics are behind but not by a large margin, like the case of k-NN.

As such, representative models from the *neural networks, random forests, stochastic gradient boosting*, and *quadratic discriminant analysis* will be selected to represent the *resampling* approach. Table 32 gives a summary of the performance of the representative models per algorithm, while figure 15 gives a visual look at how the models compare to each other.

Algorithm	AUC	Accuracy	Sensitivity	Specificity	κ	F_1
Gradient Boosting	0.734	0.676	0.682	0.671	0.353	0.679
Neural Network	0.728	0.667	0.683	0.651	0.334	0.675
Quadratic Discriminant Analysis	0.706	0.651	0.722	0.580	0.302	0.674
Random Forest	0.724	0.667	0.693	0.640	0.333	0.678

Table 28: Representative models from the top 4 algorithms under the *resampling* approach.

Figure 14: Performance metrics for each algorithm under the *resampling* approach.

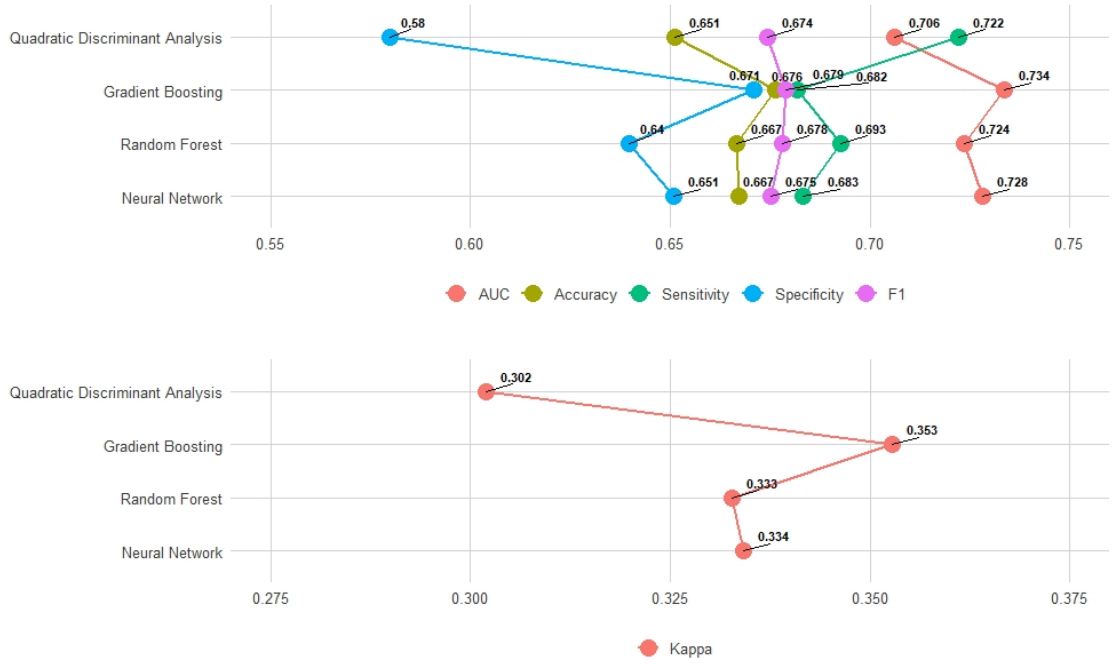

Figure 15: Representative models from the top 4 algorithms under the *resampling* approach.

Additionally, table 29 gives a summary of summary of parameters associated with the models.

Algorithm	Parameters
Quadratic Discriminant Analysis	none
Neural Network	$size = 7, decay = 2$
Random Forest	$m.try = 10$
Gradient Boosting	$n.trees = 1000$, $interaction.depth = 10$, $shrinkage = 0.01$, $n.minobsinnode = 10$

Table 29: Representative models from the top 4 algorithms under the *resampling* approach.

9.3 Cost-sensitive training

Recall that *cost-sensitive training* approach assigns costs to each mis-classifications, be them false positives or false negatives. Table 30 gives the averages of the performance metrics (with the exception of AUC) for the two major algorithms: *support vector machines* (SVM) with linear kernel and radial kernel, over the range of different weights, denoted by `Weights`, for mis-classifying a false negative: incorrectly predicting that the flight would not be cancelled.

For example, when *weight* is 100, the cost of mis-classifying a false negative is 100 times more expensive than that of a false positive. Note that when *weight* is 1, it is the trivial case that both types of error carry the same costs.

Algorithm	Weights	Accuracy	Sensitivity	Specificity	κ	F_1
SVM: Linear	100	0.311	0.855	0.300	0.009	0.048
	75	0.451	0.770	0.444	0.015	0.053
	50	0.624	0.616	0.624	0.025	0.062
	25	0.889	0.210	0.903	0.037	0.069
SVM: Radial	100	0.959	0.054	0.977	0.029	0.048
	50	0.959	0.054	0.977	0.029	0.048
	1	0.979	0.014	0.998	0.021	0.025

Table 30: Averages of performance metrics for each algorithm with different case weights for the class "cancelled", denoted by `Weights`, with the weights of class "notcancelled" being fixed at 1, under the *cost-sensitive training* approach.

As usual, figure 16 gives a visual look at how the different algorithms with different case weights compare againts the others. Some observations can be made:

1. there is a clear trend that the higher the weights, the more emphasis the models place on the cancelled flights, and the higher chance of correctly predicting ones; however, that comes at a great trade-off in accuracy.

2. with the exception of SVM with linear kernel and weight of 50, all other algorithms have quite extreme *sensitivity* and *accuracy, specificity* metrics: an algorithm is either good at one metric and quite bad at the others and vice versa.

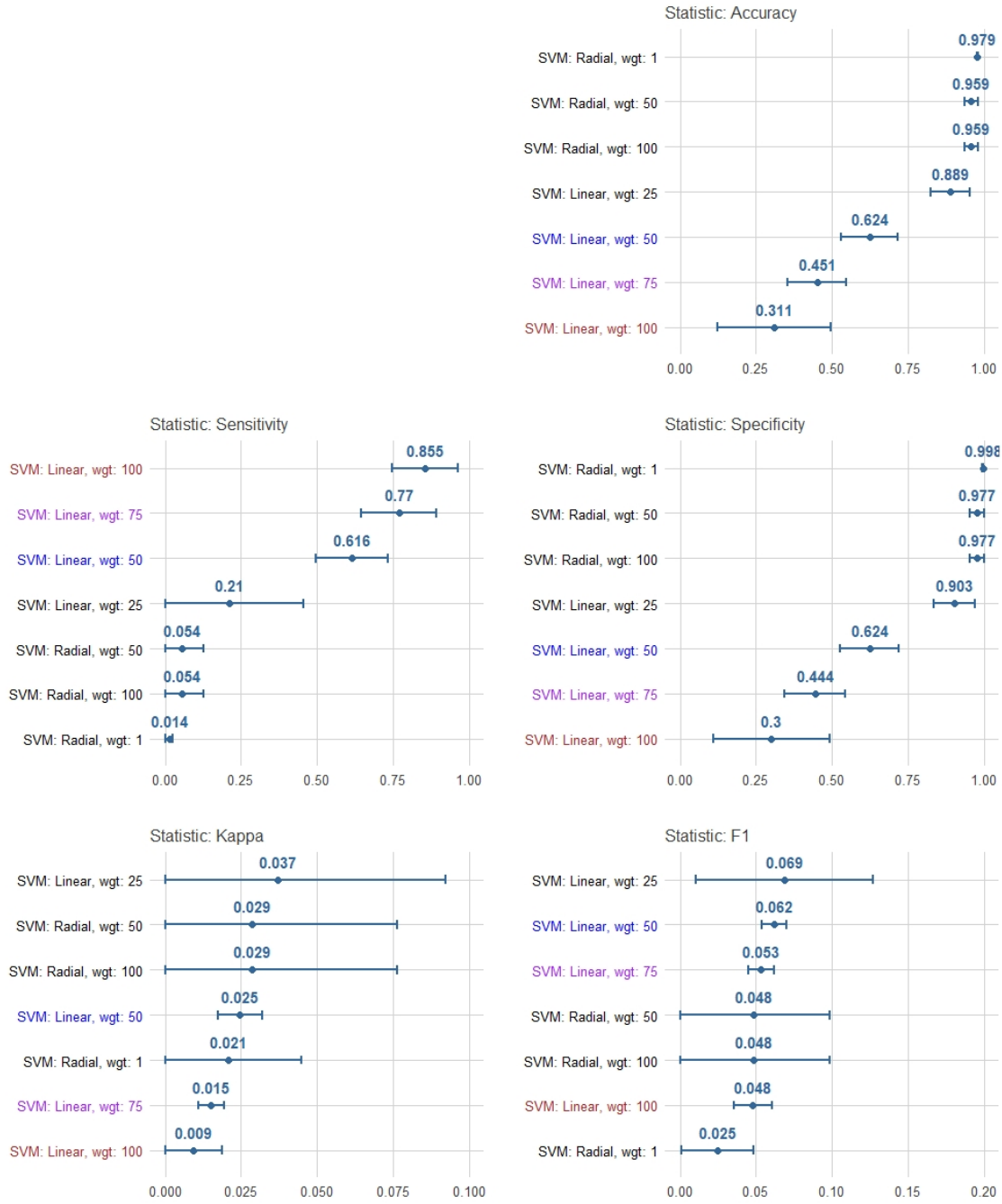

Figure 16: Performance metrics for each algorithm and their respective case weights for the "cancelled" class under the *cost-sensitive training* approach.

Thus, after much consideration, the algorithm SVM with linear kernel and weights of $100, 75$, and 50 will be selected to represent this approach. Table 31 and figure 17 give further information on the representative models for each of the weights. Recall that in this case, since the AUC is not available, the distance formula is:

$$dist(m_i, \overline{m}) = \sqrt{3(m_i^{Sen} - \overline{m}^{Sen})^2 + (m_i^{Spe} - \overline{m}^{Spe})^2 + (m_i^{\kappa} - \overline{m}^{\kappa})^2 + (m_i^{F_1} - \overline{m}^{F_1})^2} \quad (4)$$

Algorithm	Weights	Accuracy	Sensitivity	Specificity	κ	F_1
SVM: Linear	100	0.286	0.844	0.273	0.007	0.052
SVM: Linear	75	0.432	0.760	0.425	0.013	0.053
SVM: Linear	50	0.610	0.615	0.610	0.023	0.062

Table 31: Representative models from the top 3 algorithms under the *cost-sensitive training* approach.

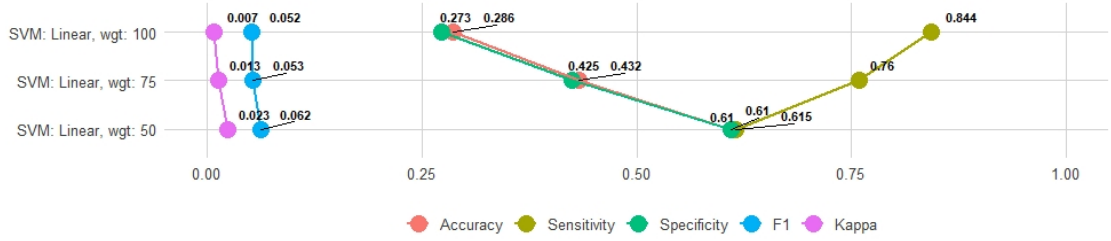

Figure 17: Representative models from the top 3 algorithms under the *cost-sensitive training* approach.

9.4 Test set evaluation

Recall that prior to building any model, a test set had been held out, intended to tested upon by the top performing algorithms from each approach. Table 9.4 provides the final evaluation statistics on that test set.

Approach	Algorithm	AUC	Accuracy	Sensitivity	Specificity	κ	F_1	Threshold
Alternate Cut-offs	Logistic Regression	0.708	0.609	0.709	0.607	0.029	0.064	0.019
	Linear Discriminant Analysis	0.705	0.651	0.653	0.651	0.031	0.066	0.019
	Quadratic Discriminant Analysis	0.701	0.645	0.659	0.645	0.031	0.066	0.048
Resampling	Quadratic Discriminant Analysis	0.708	0.580	0.724	0.577	0.026	0.061	0.500
	Neural Network	0.722	0.655	0.690	0.654	0.036	0.070	0.500
	Random Forest	0.717	0.643	0.667	0.643	0.031	0.066	0.500
	Gradient Boosting	0.730	0.668	0.661	0.668	0.035	0.070	0.500
Cost-sensitive Training	SVM: Linear; Weight = 50		0.632	0.604	0.632	0.023	0.058	
	SVM: Linear; Weight = 75		0.447	0.764	0.441	0.014	0.050	
	SVM: Linear; Weight = 100		0.282	0.872	0.271	0.007	0.044	

Table 32: Representative models from the top 4 algorithms under the *resampling* approach.

To give a visualization of how the metrics stack up againts the others for each algorithm, figure 18 gives a plot of *Accuracy-Sensitivity*, 2 of the most relevant metrics given the setting. Some observations can be made:

1. There appears no single algorithm that considerably outperforms in terms of accuracy and sensitivity.
2. There is a clear inverse relationship between accuracy and sensitivity among the algorithms, indicating the trade-off between the two.
3. In terms of approaches, it can be argued that *resampling* gives slightly better results than the other 2 approaches, and *cost-sensitive training* gives slightly worse results.
4. However, if one looks at the complete range (in figure 19), most algorithms cluster together, with no algorithms or approaches considerably standing out.

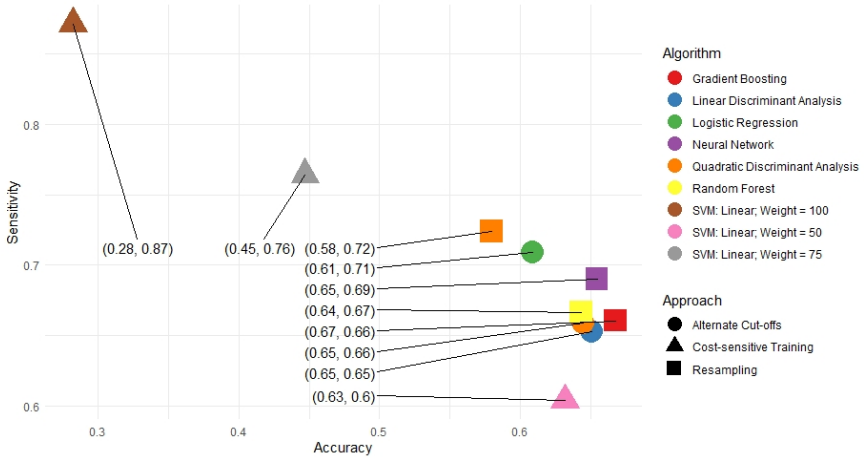

Figure 18: Accuracy and sensitivity measures on the test set from the top performing algorithms.

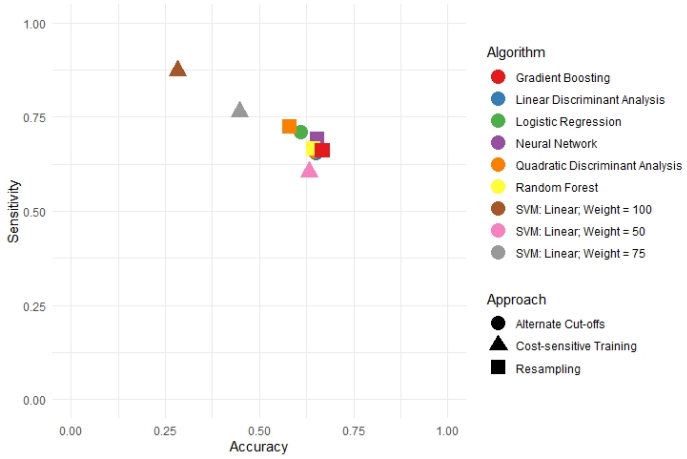

Figure 19: Accuracy and sensitivity measures on the test set from the top performing algorithms, shown in complete range.

10 Conclusion

Since all models were built with fairly simple formulas: variables were individual, there were no interaction terms between variables and not many higher degree polynomials of variables were used, there remains much freedom to twist the above models in more complicated ways, in order to improve the performance. Additionally, while the goal is to use only information available to regular consumers at the time of booking, besides the variables as indicated in table 20, other types of information can be implicitly available, such as the number of flights (same route, same airline) there are in the last 1 or 2 hours, or the schedule arriving time of flight going in the opposite direction (the idea concerns the turn-around schedule, a slight delay at the start of the day may trigger a domino effect, sometimes causing cancellation among later flights) and so on. More over, given more computing power, one can allow for wider range of parameter tuning and more extensive cross-validation (10-fold versus 5-fold, which was used).

Thus, further twisting of models can be expected to improve the performance. As such, there appears reasons that the result in section 9.4 provides a new benchmark: at the very least, our predictive models can be expected to correctly point out that a flight will be cancelled more than 65% of the time, with the overall accuracy of, accidentally, also 65%.

3. Network Analysis

Objectives:

1. to explore airport connectivity within the contiguous US air travel, and the hierarchy and grouping of airports' roles in the network;
2. to explore the network structures behind airlines' planning of flight routes;
3. to model an airline network via random graph theory

11 Data

The data was obtained from Statistical Computing: a record of all US domestic flights in 2008. For analysis in sections 12 and 13, the data was narrowed down to consist of flights from and to the 48 contiguous states and Washington DC. For analysis in section 14, the data was narrowed down further to only flights operated by Southwest Airlines. Supplementary information such as the airport's city's population demographic and city area was taken from the 2010 US Census Survey.

In analyzing the data, I used the R packages `igraph, statnet, sna, ergm`, and `gof`, together with other common packages for plotting and data analysis.

12 US Air Transportation Network

12.1 Descriptive Statistics

At the fundamental level, a *graph*, or a *network*, is a collection of vertices and edges: let G be a graph, denoted $G = (V, E)$, where V and E are the sets of vertices and edges respectively, and that G has n vertices and m edges, or $|V| = n, |E| = m$.

In modeling the network, let airports be vertices, and there is an edge between 2 vertices if there was at least 1 flight in each direction between the 2 respective airports. This gives a simple, undirected, and connected graph with 273 vertices and $2,392$ edges.

12.1.1 Network Statistics

Diameter is the length of the "longest shortest path", which can be either unweighted or weighted with parameters of choice. As we can see from table 33, we could start from any airport and have a choice to get to any other airports with at most 3 flights, or $9,672$ miles in total distance.

Related to *diameter* is **average path length**, which, at 2.23, is the expected number of flights one would need to take to get from a randomly chosen airport to some other airport.

Density is the ratio of the number of edges over that of all possible edges:

$$den(G) = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$$

A *density* of 6.44% (from table 33) indicates that the network was quite sparse: only a small number of edges were realized.

Diameter	Average path length	Density (%)
3	2.23	6.44

Table 33: Descriptive Statistics of the flight network over contiguous US in 2008.

12.1.2 Vertex Centrality

Given our network, it is natural to explore the importance of airports to the network. One option is via the concept of vertex centrality, of which there are a number of measures, each measure gives a perspective at the dynamic interactions between the airports.

Degree centrality is the number of neighbors each vertex has. An airport having higher degrees offers directed flights to a higher number of destinations. Table 34 shows that from Atlanta, GA, we could get to 168 other airports with direct flights, and from Chicago to 144 other airports.

	Airport	City, State	Degree		Airport	City, State	Degree
1	ATL	Atlanta, GA	168	6	DTW	Detroit, MI	115
2	ORD	Chicago, IL	144	7	CVG	Covington, KY	112
3	DFW	Dallas-Fort Worth, TX	130	8	IAH	Houston, TX	109
4	MSP	Minneapolis, MN	121	9	SLC	Salt Lake City, UT	106
5	DEN	Denver, CO	117	10	LAS	Las Vegas, NV	89

Table 34: Degree Centrality.

Closeness centrality is related to the average distance from a vertex to all other vertices, for a vertex u:

$$C_u = \frac{n}{\sum_v d_{uv}}$$

where d_{uv} is the shortest distance between vertices u and v. An airport with higher *closeness* centrality would imply that starting from that airport, it would take smaller numbers of flights to get to other airports. For example, at $C_{ATL} = .723$ and a degree of 168, it would take an average of 2.24 flights to get from Atlanta, GA, to any other airports, the lowest among all airports.

	Airport	City, State	Closeness		Airport	City, State	Closeness
1	ATL	Atlanta, GA	0.723	6	DTW	Detroit, MI	0.634
2	ORD	Chicago, IL	0.680	7	CVG	Covington, KY	0.630
3	DFW	Dallas-Fort Worth, TX	0.657	8	IAH	Houston, TX	0.625
4	MSP	Minneapolis, MN	0.643	9	SLC	Salt Lake City, UT	0.621
5	DEN	Denver, CO	0.637	10	LAS	Las Vegas, NV	0.598

Table 35: Closeness Centrality.

Betweenness centrality measures the extent to which a vertex is on the shortest paths between other vertices, for a vertex u:

$$B_u = \sum_{vw} \frac{n^u_{vw}}{g_{vw}}$$

where n^u_{vw} is the number of shortest paths between v and w that pass through u, and g_{vw} is the number of shortest paths between v and w, summing over all pairs of vertices not u. A larger *betweenness* value would indicate that such vertex is located at the "bottleneck" along the flow paths. Such vertex would imply that the respective airport being a transit hub, such as Atlanta, GA, collecting and distributing flights across other airports.

	Airport	City, State	Betweenness		Airport	City, State	Betweenness
1	ATL	Atlanta, GA	8321	6	DTW	Detroit, MI	2654
2	SLC	Salt Lake City, UT	4656	7	DEN	Denver, CO	2486
3	DFW	Dallas-Fort Worth, TX	4209	8	IAH	Houston, TX	2300
4	MSP	Minneapolis, MN	4120	9	LAX	Los Angeles, CA	1826
5	ORD	Chicago, IL	3433	10	SFO	San Francisco, CA	1762

Table 36: Betweenness Centrality.

Eigenvector centrality measures the centrality of not just a vertex but also its neighbors, as such a higher *eigenvector* would indicate that the vertex itself and its neighbors are both important in the graph.

	Airport	City, State	Eigenvector		Airport	City, State	Eigenvector
1	ATL	Atlanta, GA	1.000	6	MSP	Minneapolis, MN	0.862
2	ORD	Chicago, IL	0.965	7	IAH	Houston, TX	0.858
3	DFW	Dallas-Fort Worth, TX	0.913	8	DTW	Detroit, MI	0.854
4	DEN	Denver, CO	0.874	9	LAS	Las Vegas, NV	0.815
5	CVG	Covington, KY	0.868	10	EWR	Newark, NJ	0.804

Table 37: Eigenvector Centrality.

Table 38 summarizes the ranking of airports' importance in the network based on different centrality measures, and figure 20 shows the locations of those airports. Some observations are:

1. The list has 13 airports, implying that the list of the most central airports is very consistent across different measures: airports important in 1 measure are likely to be important in other too.
2. The airports were spread out quite evenly over the contiguous US: about half of which were in the geographically central area, while the rest were along the 2 coasts.
3. ATL (Atlanta, GA) consistently ranks top in all 4 centrality measures, followed by ORD (Chicago, IL) and DFW (Dallas-Fort Worth, TX).
4. It was surprising to find that there was only 1 airport (EWR - Newark, NJ) in the busy North-East corridor (where major cities such as New York, Washington DC, and Boston are), and no airports in the North-West area (which is where Seattle is.)
5. Possibly due to the central geographic location, SLC (Salt Lake City, UT) lies in a large portion of shortest domestic flights, ranked second in *betweenness centrality*, despite having not-so-high scores in other centrality measures.

	Degree	Closeness	Betweenness	Eigenvector
1	ATL (Atlanta, GA)	ATL (Atlanta, GA)	ATL (Atlanta, GA)	ATL (Atlanta, GA)
2	ORD (Chicago, IL)	ORD (Chicago, IL)	SLC (Salt Lake City, UT)	ORD (Chicago, IL)
3	DFW (Dallas-Fort Worth, TX)	DFW (Dallas-Fort Worth, TX)	DFW (Dallas-Fort Worth, TX)	DFW (Dallas-Fort Worth, TX)
4	MSP (Minneapolis, MN)	MSP (Minneapolis, MN)	MSP (Minneapolis, MN)	DEN (Denver, CO)
5	DEN (Denver, CO)	DEN (Denver, CO)	ORD (Chicago, IL)	CVG (Covington, KY)
6	DTW (Detroit, MI)	DTW (Detroit, MI)	DTW (Detroit, MI)	MSP (Minneapolis, MN)
7	CVG (Covington, KY)	CVG (Covington, KY)	DEN (Denver, CO)	IAH (Houston, TX)
8	IAH (Houston, TX)	IAH (Houston, TX)	IAH (Houston, TX)	DTW (Detroit, MI)
9	SLC (Salt Lake City, UT)	SLC (Salt Lake City, UT)	LAX (Los Angeles, CA)	LAS (Las Vegas, NV)
10	LAS (Las Vegas, NV)	LAS (Las Vegas, NV)	SFO (San Francisco, CA)	EWR (Newark, NJ)

Table 38: Airport ranking based on centrality measures.

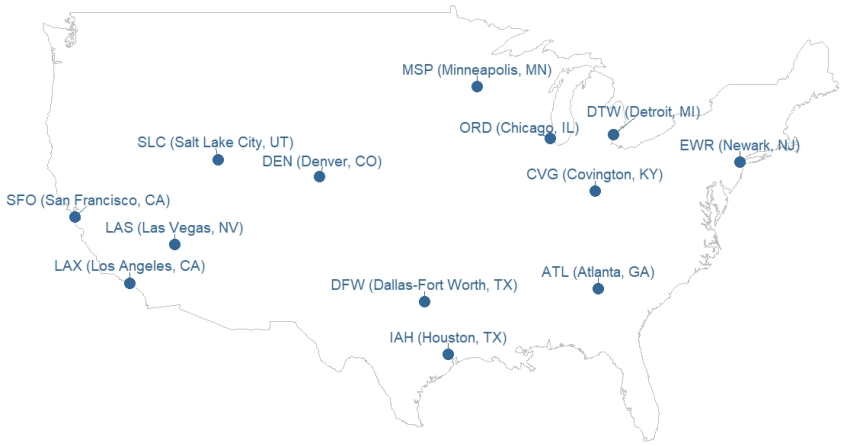

Figure 20: Map of most central airports.

12.2 Small-world and Preferential Attachment Properties

12.2.1 Small-world Property

A frequently observed pattern from real world networks is the *small-world* property: a network with high level of clustering and low average path length.

Clustering measures the tendency of if 2 vertices are connected by an edge if they have a common neighbor. Here, *clustering*, or *transitivity*, measures the tendency of if airports A and B are connected by a direct flight if they are both directly connected to an airport C. Thus, a *small-world* network offers consumers more choices for direct flights between airports.

To investigate if our flight network has the *small-world* property, an option is through Monte Carlo simulation and compare the observed statistics against what would have been expected under classical random graphs of the same size.

Attributes	Observed	μ_{MC}	σ_{MC}	Remarks
Clustering Coefficient	0.385	0.064	0.002	$> 3\sigma_{MC}$
Average Path Length	2.228	2.239	0.002	$< 3\sigma_{MC}$

Table 39: Monte Carlo simulation to access *small-world* property, where μ_{MC} and σ_{MC} are the mean and standard deviation simulation's statistics.

From table 39, the observed clustering coefficient is statistically significantly higher than that would have been produced by a random graph, while the average path length is significantly lower. Even though the absolute observed statistics are not very high or low, this suggests evidence that our network does exhibit the *small-world* property.

12.2.2 Preferential Attachment Property

Preferential attachment property seeks to reproduce the frequently observed principle of "the rich get richer": a vertex of high degree will continue to be connected to other vertices at a faster rate than a vertex of lower degree. A consequence of that is the network's degree distribution will closely follow the power-law: given a vertex u with degree d_u:

$$\mathbb{P}(d_u = k) \propto k^{-\alpha}$$

for some positive number α and large positive integer k, and where $\propto$ is proportional to. This gives rise to network where a majority of vertices have small degrees and a few vertices with very high degrees. Numerous studies have shown that biological networks (gene interaction, protein-protein interaction, ...) tend to have α less than 2, while non-biological networks (internet graph, transportation graph, ...) have α between 2 and 3: the higher the α, the faster the decay rate, or the smaller the number of high-degree vertices.

Figure 21 shows the observed degree distribution from our flight network in black and a power law fit in blue. Some remarks and observations are:

1. In fitting the data, the degrees were limited to 5 and above.
2. Starting at around degree 30, the power law function starts to fit the observed distribution particularly well.
3. The fitted line has $\alpha \approx 1.71$, giving $\mathbb{P}(d_u = k) \approx k^{-1.71}$, for $k \geq 30$.
4. At $\alpha \approx 1.71$, the power is quite low for its type, a consequence of this is our network's degree distribution decays slower than what have been observed from other transportation networks, and that there would be a few more airports with high number of direct flights to other airports.

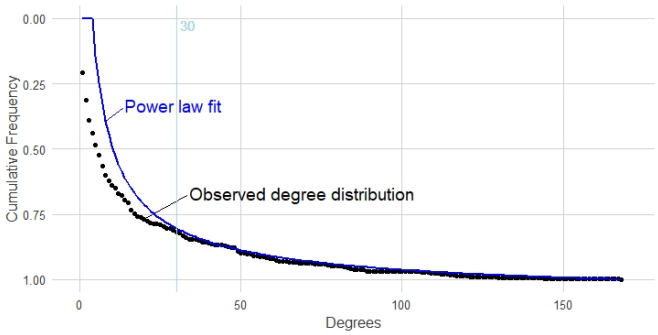

Figure 21: Degree distribution of flight network.

12.3 Graph Partitioning

It is often of our interest to find some inherent structures among the vertices, in the form of community clusters, via the method of graph partitioning or community detection. Table 40 summarizes the different algorithms on our network where the edge weights were defined to be the number of flights for that particular route in 2008, and their modularity (which is often used as the criteria to compare the algorithms - the higher the modularity, the better the partition):

Algorithms	Modularity	Number of clusters
Fast Greedy	0.289	3
InfoMAP	0.000	1
Louvain	0.291	4
Spinglass	0.000	6
Leading Eigenvector	0.257	4
Walktrap	0.262	65

Table 40: Graph paritioning algorithms.

Except from the InfoMAP and Spinglass algorithms, the other 4 algorithms had quite similar modularity, and that there was no algorithm that partitioned our network particularly better than others.

Figure 22 gives an example of the partition produced by the Louvain algorithm. There appears evidience that geography played a strong role in how the airports were grouped together into partitions:

- Group 1 (purple): airports from the western half of the contiguous US,
- Group 2 (blue): airports from the north-east area,
- Group 3 (red): airports from the south of the US,
- Group 4 (green): airports mostly located along the eastern coast.

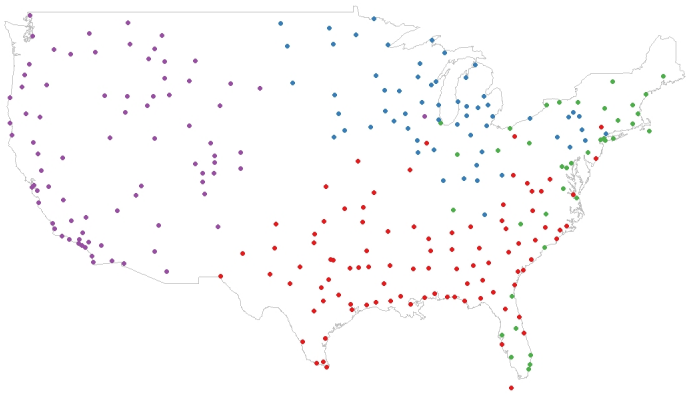

Figure 22: Network paritioning based on Louvain algorithm.

13 Airlines' Networks

13.1 Network Structure

Perhaps the defining characteristic about airlines' flight networks is the *hub-and-spoke* structure, in which a selected few airports, thanks to a variety of reasons, are designed to be the central transit hubs, collecting and distributing flights across smaller airports.

Figure 23 shows the network structures among the 4 largest US airlines, with the vertex size being in proportion to the number of departing flights. A quick observation is that the *hub-and-spoke* structure is strongly visible in how American Airlines, United Airlines, and Delta Air Lines designed their US domestic flight routes in 2008, while the distinction between hub and spoke airports is not as clear-cut in the Southwest Airlines' network.

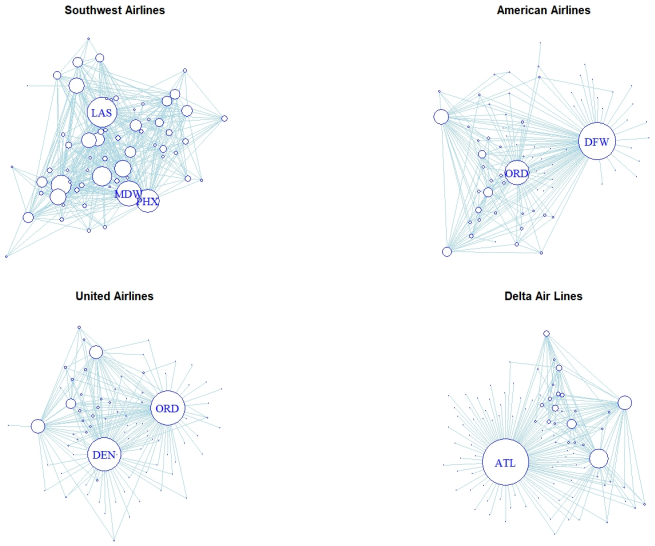

Figure 23: Network structures among top 4 airlines.

Figure 24 gives a look at how the networks are located geographically:

1. Southwest Airlines' biggest share of the air travel market is visible through how their vertex sizes are consistently bigger than that of other airlines.
2. Southwest Airlines, American Airlines, and United Airlines network coverage spread throughout the US, while Delta Air Lines appears to cover mostly areas along the 2 coasts.
3. There is a considerable gap in the central north area (Plains and Rock Mountain regions) not being extensively serviced by the 4 airlines.
4. There is little overlapping of airport hubs among the airlines networks: an airline's major hub is unlikely to be a major hub for another airline.

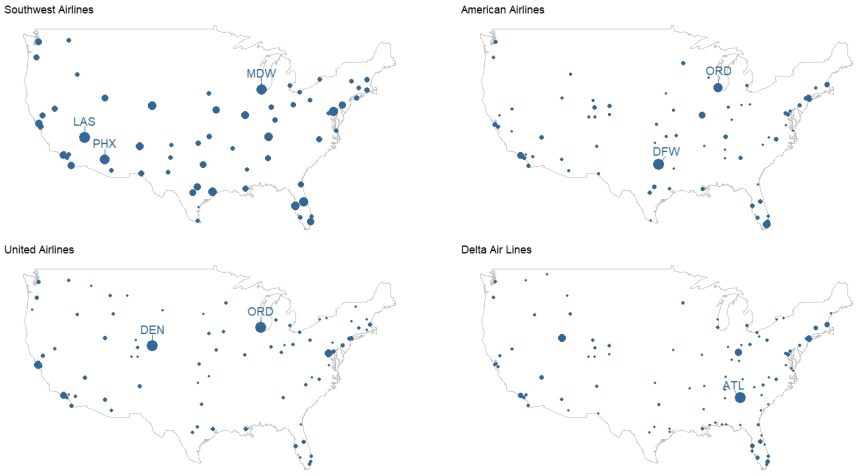

Figure 24: Maps of the geographic locations of top 4 airlines' networks.

13.2 Descriptive Statistics

Similar to the different descriptive statistics measures explored in section 12.1, we can apply those to the current airline networks. Starting with the size of the respective networks from table 41, there are a number of interesting observations:

1. At 95 airports (around 35% of all airports in contiguous US), Delta Air Lines gave customers the most number of choice for intra-airline air travel, although they most likely would have to pass through ATL (Atlanta, GA) along the way.
2. At 449 routes across 64 airports, the Southwest Airlines network was significantly denser than that of other airlines (at least 3 times denser).

Airline	Airports	Routes	Density (%)
Southwest	64	449	22.27
American	75	209	7.53
United	82	192	5.78
Delta	95	207	4.64

Table 41: Airline networks' desciptive statistics.

Tables 42, 43, 44, and 45 give a look at the most central airports for Southwest Airlines, American Airlines, United Airlines, and Delta Air Lines respectively. Interestingly, from table 42 of Southwest Airlines: there is a positive correlation between *degree* and *eigenvector* measures, and between *closeness* and *betweenness* measures, while that between the 2 groups is not as strong: airports having high degrees might not necessarily lie on the shortest paths between other airports.

	Degree	Closeness	Betweenness	Eigenvector
1	LAS (Las Vegas, NV)	FLL (Fort Lauderdale, FL)	FLL (Fort Lauderdale, FL)	LAS (Las Vegas, NV)
2	MDW (Chicago, IL)	DEN (Denver, CO)	DEN (Denver, CO)	PHX (Phoenix, AZ)
3	PHX (Phoenix, AZ)	LAS (Las Vegas, NV)	MCO (Orlando, FL)	OAK (Oakland, CA)
4	BWI (Baltimore, MD)	TUL (Tulsa, OK)	LAS (Las Vegas, NV)	MDW (Chicago, IL)
5	MCO (Orlando, FL)	MCO (Orlando, FL)	TUL (Tulsa, OK)	LAX (Los Angeles, CA)

Table 42: Southwest Airlines.

	Degree	Closeness	Betweenness	Eigenvector
1	DFW (Dallas-Fort Worth, TX)	ORD (Chicago, IL)	ORD (Chicago, IL)	DFW (Dallas-Fort Worth, TX)
2	ORD (Chicago, IL)	MIA (Miami, FL)	MIA (Miami, FL)	ORD (Chicago, IL)
3	MIA (Miami, FL)	XNA (Fayetteville/Springdale/Rogers, AR)	XNA (Fayetteville/Springdale/Rogers, AR)	MIA (Miami, FL)
4	STL (St. Louis, MO)	FLL (Fort Lauderdale, FL)	DFW (Dallas-Fort Worth, TX)	LGA (New York, NY)
5	LAX (Los Angeles, CA)	GUC (Gunnison, CO)	MTJ (Montrose, CO)	LAX (Los Angeles, CA)

Table 43: American Airlines.

	Degree	Closeness	Betweenness	Eigenvector
1	ORD (Chicago, IL)	ORD (Chicago, IL)	ORD (Chicago, IL)	ORD (Chicago, IL)
2	DEN (Denver, CO)	SFO (San Francisco, CA)	DEN (Denver, CO)	DEN (Denver, CO)
3	SFO (San Francisco, CA)	LAX (Los Angeles, CA)	ABQ (Albuquerque, NM)	SFO (San Francisco, CA)
4	IAD (Chantilly, VA)	ABQ (Albuquerque, NM)	PSP (Palm Springs, CA)	LAX (Los Angeles, CA)
5	LAX (Los Angeles, CA)	SLC (Salt Lake City, UT)	OAK (Oakland, CA)	IAD (Chantilly, VA)

Table 44: United Airlines.

	Degree	Closeness	Betweenness	Eigenvector
1	ATL (Atlanta, GA)	ATL (Atlanta, GA)	ATL (Atlanta, GA)	ATL (Atlanta, GA)
2	SLC (Salt Lake City, UT)	GUC (Gunnison, CO)	HDN (Hayden, CO)	LGA (New York, NY)
3	CVG (Covington, KY)	MTJ (Montrose, CO)	CVG (Covington, KY)	BOS (Boston, MA)
4	JFK (New York, NY)	FCA (Kalispell, MT)	SLC (Salt Lake City, UT)	DCA (Arlington, VA)
5	LAX (Los Angeles, CA)	HDN (Hayden, CO)	LGA (New York, NY)	MCO (Orlando, FL)

Table 45: Delta Air Lines.

14 Southwest Airlines Network

Being the airline enjoying the biggest market share of domestic air travel in 2008, and their dense and rather unconventional network structure as illustrated in figures 23 and 24, Southwest Airlines presents an interesting case study. In this section, we are going to apply some theoretical framework from graph theory, specifically:

1. *Link prediction*: predicting if Southwest Airlines served a direct flight between 2 airports.
2. *Network modeling*: fitting the network based on the classical random graph theory.

14.1 Link Prediction

It is often of our interest to look for interactions between vertices in the network, which leads to the problem of link prediction: given a graph $G = (V, E)$ and some vertex attributes, we would like to predict if there is an edge in the network between some vertices u and v.

14.1.1 Scoring Function Method

Among several methods is the *scoring* method: given some vertices u, v, let $s(u, v)$ be the associated score, after which all scores are ordered. In determining whether $s(u, v)$ indicates edge $\{u, v\}$ in G, we can either:

- compare against a threshold: all scores above the threshold indicate edge presence; or
- take the top n scores as positive edge predictions

There are a number of choices for the scoring functions, some of which are:

1. *Common neighbors*: the more common neighbors u and v have, the higher chance there is an edge between them:
$$s_{cn}(u, v) = |N(u) \cap N(v)|$$
where $N(u)$ is the set of vertices neighboring u;
2. *Jaccard coefficient*: a variation of *common neighbors*, instead of the number of common neighbors, we look at the portion of neighbors of u and v that are neighbors to both u and v:
$$s_{Ja}(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$
3. *Preferential attachment*: following the same idea as above, the likelihood of an edge is proportional to the multiplicity of the number of neighbors u and v already have:
$$s_{pa}(u, v) = |N(u)| \times |N(v)|$$

4. *Katz score*: looks at the ensemble of all paths between u and v:

$$s_{Ka}(u,v) = \sum_{l=1}^{\infty} \beta^l \mid p_{u,v}^{<l>} \mid$$

where $p_{u,v}^{<l>} = \{\text{paths of length } l \text{ between } u \text{ and } v\}$, $\mid p_{u,v}^{<l>} \mid$ is the number of such paths, and β is a parameter of choice: the smaller β is, the higher the influence shorter paths have over longer paths.

Table 46 summarizes how models trained on different scoring functions performed in terms of AUC (area under the curve) and maximum attainable accuracy. A quick observation suggests that *Katz score* gave some of the better performance statistics.

Scoring Functions		AUC	$\max_{Accuracy}$
Common Neighbors		0.826	0.827
Jaccard Coefficient		0.661	0.778
Preferential Attachment		0.890	0.862
Katz score	$\beta = 0.01$	0.973	0.922
	0.10	0.985	0.947
	0.25	0.987	0.947
	0.50	0.984	0.946
	0.75	0.807	0.921
	1.00	0.871	0.866

Table 46: Scoring function models' AUC and maximum Accuracy.

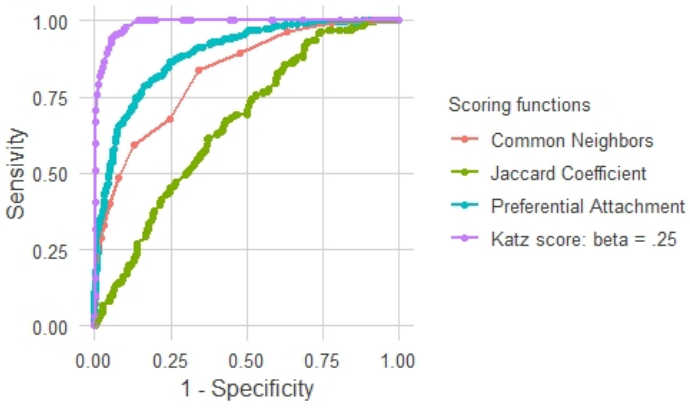

Figure 25: ROC curves of different scoring functions' models.

Figure 25 shows the graph of the ROC curves from the 4 different scoring functions, with $\beta = .25$ for the *Katz score* function. The figure suggests a clear order of performance in terms of ROC curves and AUC, with the *Katz score* option giving the best performance.

14.1.2 Probabilistic Classification Model

The second class of methods lends itself to the classical problem of classification in the context of machine learning. *Probabilistic classification model* is thus a collection of all pairs of vertices and the vertices' attributes. Given our network, it is natural to let city demographic information as vertex attributes for each respective airports, and the status of whether Southwest Airlines served a direct flight between 2 airports as the binary predicted variable.

Table 47 summarizes the performance of a few common algorithms, each of which was selected with the best tuning parameters and was trained based on the following vertex attributes:

- the airport
- the city population, area, and population density
- the state, and whether the city is the state capital

Models	AUC	$\max_{Accuracy}$
Logistic Regression	0.855	0.833
Flexible Discriminant Analysis	0.780	0.836
Random Forest	0.905	0.854
Stochastic Gradient Boosting	0.894	0.849

Table 47: Probabilistic classification models' AUC and maximum Accuracy.

Similarly, figure 26 shows each model's ROC curve, after parameter tuning. The figure suggests that *random forest* and *stochastic gradient boosting* gave better result than *logistic regression* and *flexible discriminant analysis*. Comparing with table 46, it appears that the *Katz score* scoring function method gave better performance overall, in terms of both the AUC and maximum accuracy.

However, recall that our Southwest Airlines network graph has a *density* of 22.27%. For the purpose of probabilistic classification, the data would be quite imbalance: the ratio of positive truth is only 22.27%, which is similar to the classification problem we had earlier in predicting if a flight would be cancelled, although the ratio is not as extreme (22.27% and 1.96%). Thus, in applying the *probabilistic classification model*, it would be necessary to apply the appropriate techniques when handling imbalance data, such as *alternative cut-offs, resampling,* and *cost-sensitive training*.

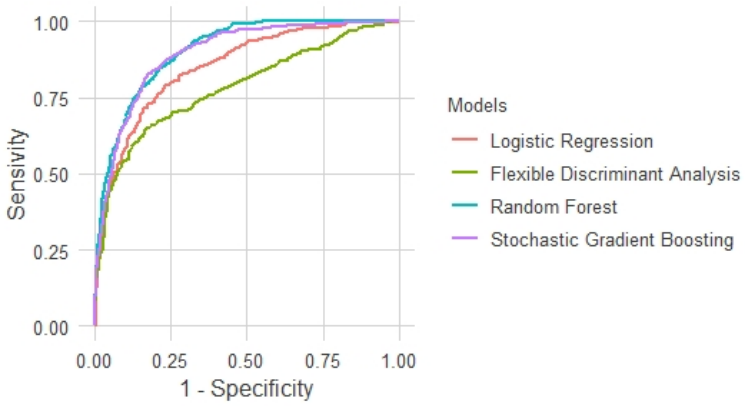

Figure 26: ROC curves of different probabilistic models.

14.2 Network Modeling

Closely related to *link prediction* is the problem of fitting our network with random graph models, of which the class of *exponential random graph models* is a popular choice.

Consider a random graph $G = (V, E)$. Let $\mathbf{Y} = [Y_{uv}]$ be the random adjacency matrix, and Y_{uv} be the binary random variable of if there is an edge between vertices u and v. Let $\mathbf{y} = [y_{uv}]$ be a particular realization of $\mathbf{Y}$, then:

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \left(\frac{1}{\kappa}\right) \exp\left\{\sum_{H} \theta_H \; g_H(\mathbf{y})\right\} \tag{5}$$

where

- each H is a configuration: a set of possible edges among a subset of vertices in G
- $g_H(\mathbf{y}) = \prod_{y_{uv} \in H} \; y_{uv} = 1$ if H occurs in $\mathbf{y}$, and 0 otherwise
- parameter θ_H indicates the dependency between all pairs of vertices, conditional upon the rest of the graph
- $\kappa = \kappa(\theta)$ is the normalization term

Formula 5 can be either narrowed by assuming the independency of edge status on all other edge status (which gives the classical Erdős-Renyi random graph, or Bernoulli random graph), or expanded by introducing statistics observed from the graph (such as vertex attributes, or the number of triangles and higher-order cliques).

Similar to the vertex attributes used in the probabilistic classification model from section 14.1.2, in fitting the network with exponential random graph, vertex attributes were defined as above, with an additional attribute of the type of hub the airport was, as defined by the FAA:

- large hub: handling more than 1% of total annual passenger boardings
- medium hub: handling less than 1% and more than .25%
- small hub: handling less than .25% and more than .05%
- non-hub: handling less than .05%

Table 48 gives a summary of how well each additional vertex attribute contributes to modeling the Southwest Airlines network, with AIC as a criteria for comparison: a good model would have a low AIC. Of all the models, the model based on the number of edges, the type of airport hub, the city state, and if the city is the capital state, and the city population was notably better than the others.

Model	Vertex attribtues	AIC
Baseline model	none	2140
Model 1	airport hub	1838
Model 2	airport state	1721
Model 3	city population and population density	2010
Model 4	city capital	2142
Model 5	city area	2106
Model 6	all attributes	1500
Model 7	hub, city state, city capital, city population	1499

Table 48: Exponential random graph modeling statistics.

To determine how well the model fits the Southwest Airlines network, a Monte Carlo simulation can be of help. Figure 27, obtained after running $50,000$ generations of on model 7 and compare them against the Southwest Airlines network, offers some observations:

1. With the exception of very high degrees, the fitted graph was able to capture most of the lower-degree distribution, as seen from the top-left figure.

2. The fitted model was also able to capture the other 3 statistics quite well: the observed statistics were all within the range of the simulation.

Thus, it appears that our simple random graph model of the number of edges, the type of airport hub, the city state, and if the city is the capital state, and the city population, was able to capture the actual network quite well. However, the model can certainly be improved, since the observed statistics frequently lie outside of the 95% confidence interval boxes as seen from figure 27, which could be explained by the fact that whether Southwest Airlines served a direct flight between any 2 airports depended much more than simply the airports' city demographic.

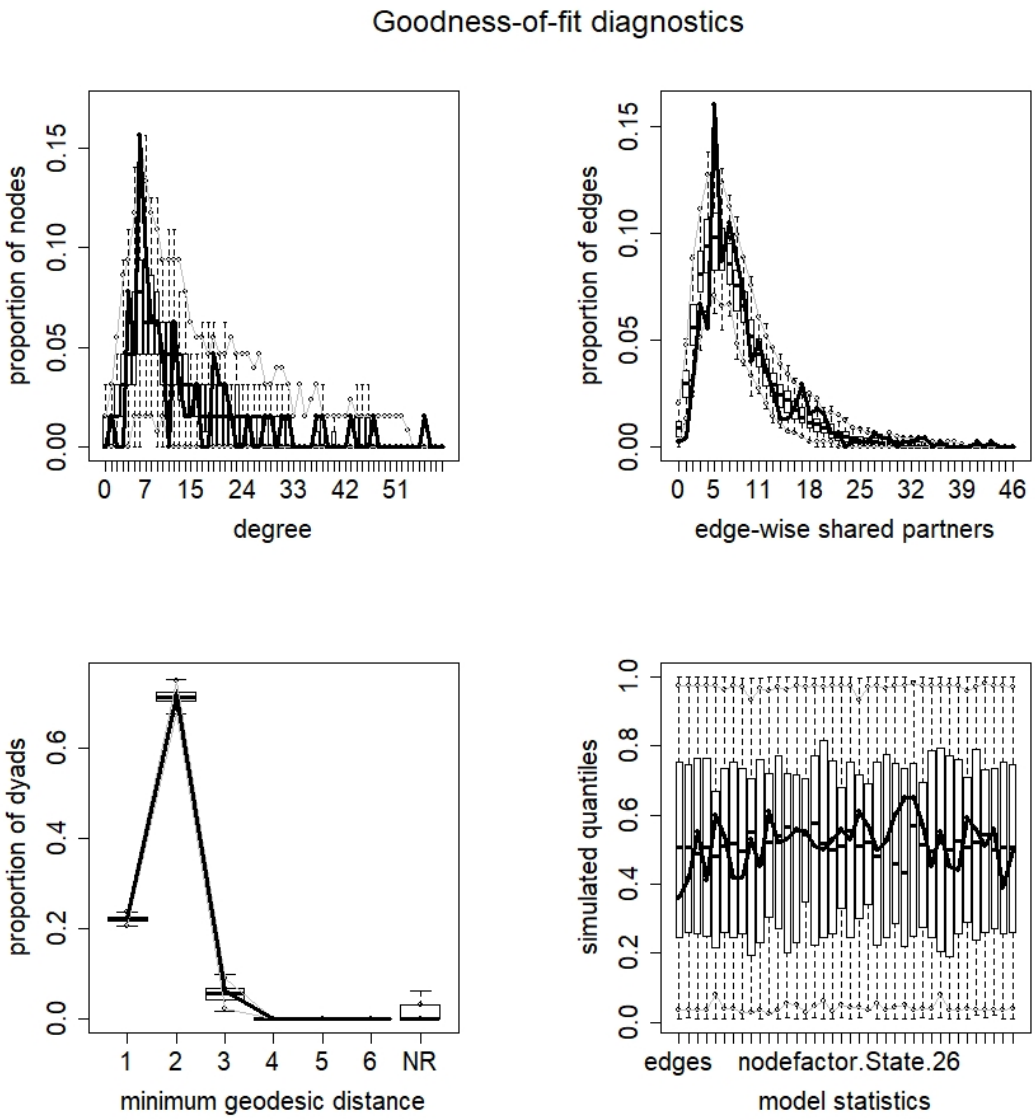

Figure 27: Goodness-of-fit diagnostic of model 7 based on 50,000 Monte Carlo simulations.

Appendix

15 Goodness-of-Fit Test

Given a data set, the *Goodness-of-Fit test* aims to test if a subset randomly sampled is representative of the entire data set. If the subset is representative, it is reasonable to expect the difference between the expected number for each variable drawn and the actual number is small.

Since most of the variables used in building models are discrete, and that continuous variables can be binned together to become discrete, the *Chi-squared* test variation was used in testing for representativeness.

For illustration, suppose the entire data set has n_0 observations, 1 discrete variable with k classes, and probability p_i for any observation to have class i (if there are variables, each variable follows the same procedure independently). Suppose we draw from the data set n samples, then the test statistic is defined as

$$Q = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

where

- $E_i = np_i$: the expected number of observations belonging to class i given sample size n
- O_i: the actual number of observations belonging to class i in the subset

Q is then compared with a *Chi-squared* distribution given $k - c$ degrees of freedom, where c is the number of estimated parameters, here c was taken to be 0.

For the purpose of finding representative subsets, a 95% confidence interval was used, meaning that if we repeated the sampling multiple times, the fraction of times the subset is actually representative of the data set would tend toward 95%.