IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

By Alex Cerda
10/27/24

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
    - Data Collection
    - Data Wrangling and Analysis
    - Interactive Maps

- Summary of all results
    - Data Analysis and Visualization

  -> Best model?

Introduction

- Project background and context:
  - Will SpaceX rocket Falcon 9 will land successfully?
  - Note: SpaceX reuses of first stage of rocket
- Natural questions we want to find answers:
  - Which factors influences the landing?
  - -> Its precise impact?
  - => What are the best conditions for best results?

Section 1

Methodology

Methodology

- Data collection methodology:

  -> SpaceX API, Wikipedia

- Perform data wrangling

  -> One hot encoding, drop irrelevant (i.e. uncorrelated) columns

- Perform exploratory data analysis (EDA) using visualization and SQL

  -> Scatter/Bar/Pie charts for visual pattern recognition

- Perform interactive visual analytics using Folium and Plotly Dash

  -> Via Folium and Plotly Dash

- Perform predictive analysis using classification models

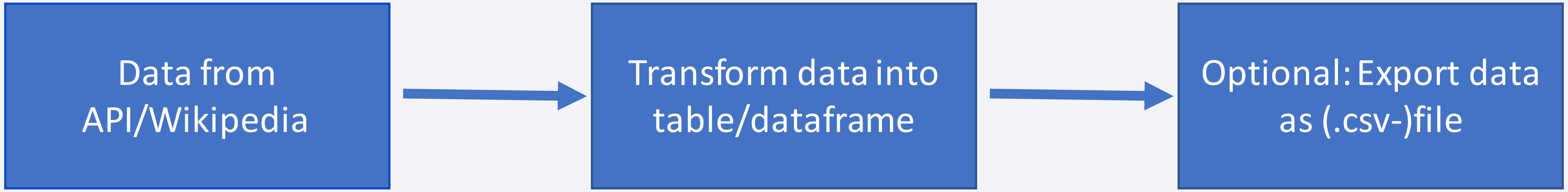
  -> Using Regression and Tree techniques

Data Collection

- What is to do?

  -> Gather and measure information on targeted variables

Pipeline:

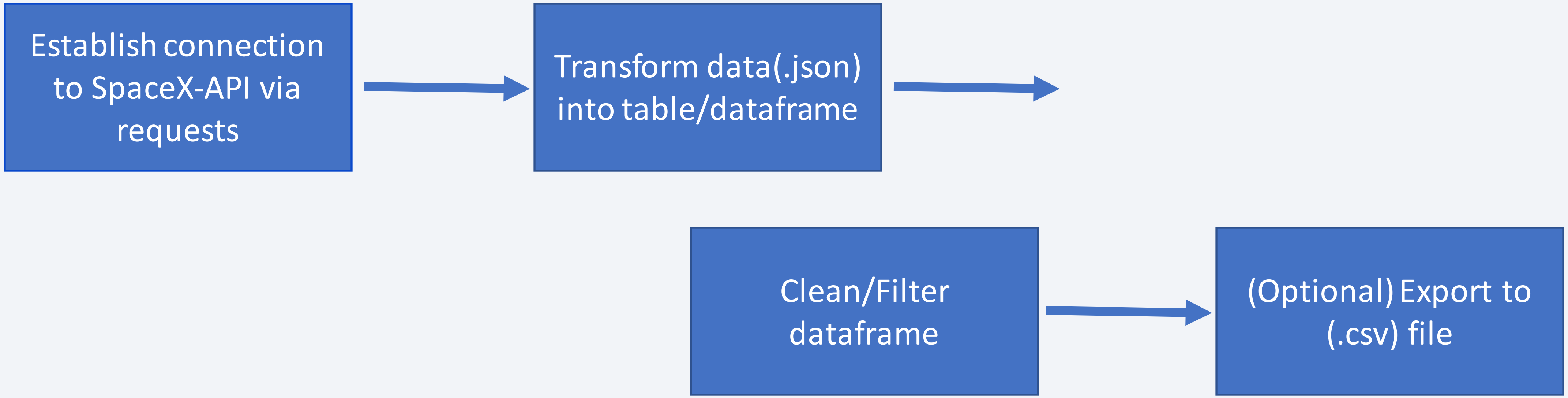


Data from API/Wikipedia

Transform data into table/dataframe

Optional: Export data as (.csv-)file

Data Collection – SpaceX API pipeline



Establish connection to SpaceX-API via requests

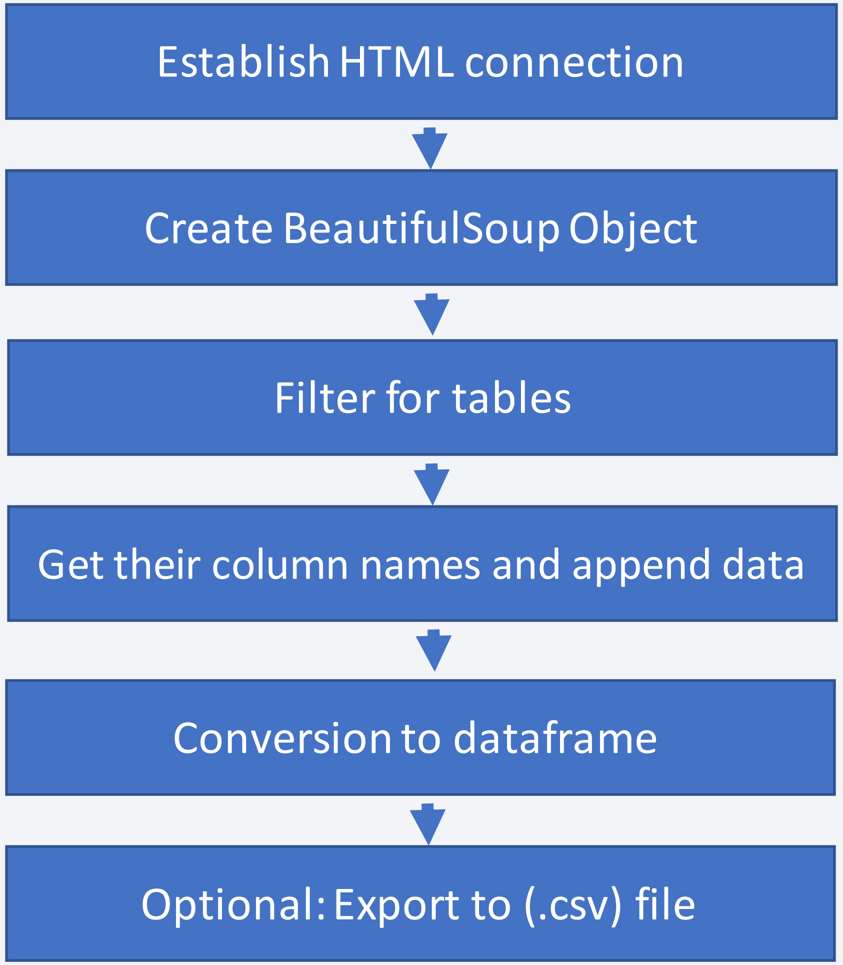
Transform data(.json) into table/dataframe

Clean/Filter dataframe

(Optional) Export to (.csv) file

File on GitHub

Data Collection – Webscraping pipeline



Establish HTML connection

Create BeautifulSoup Object

Filter for tables

Get their column names and append data

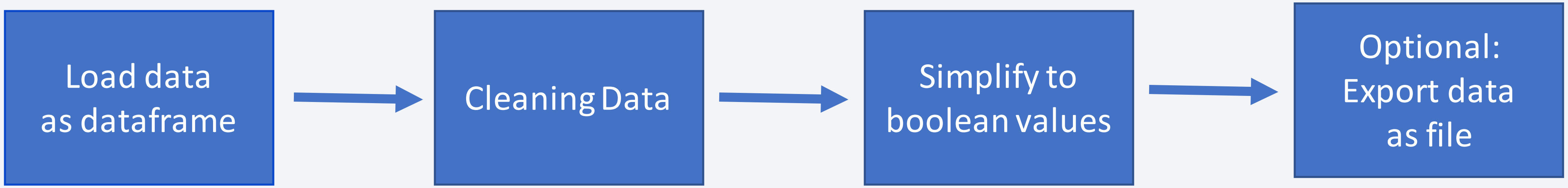
Conversion to dataframe

Optional: Export to (.csv) file

File on GitHub

Data Wrangling

- What is to do?
  - Cleaning and polishing possible messy/complex data sets for better handling
- Pipeline:



Load data as dataframe

Cleaning Data

Simplify to boolean values

Optional: Export data as file

[File on GitHub]

EDA with Data Visualization

- What is to do?
  - Create visuals and collection optical insights
- Here:
  - Payload mass/Flight number vs Launch site/Orbit type/Flight number as scatterplot for type of dependency
  - Success rate vs orbit type as bar chart for impact of variable
  - Launch Success vs Year for trend observation

File on GitHub

EDA with SQL

- What is to do?
  - Heuristically guessing/querying/questioning in the database what might have happened
- We have questioned as follows:
  - General overview over available landsides, in particular whose five entries who start with 'CCA'
  - Number of successful/failed mission outcome
    -> List for failed ones in 2015
    -> List first successful landing outcome in drone ship
  - Specify, count and rank outcomes between ~2010 and ~2017
  - Average Payload per booster version 'F9v1.1'/Total for boosters carried by NASA (CRS)
  - Booster version with maximal payload
  - Names of boosters with successful ground pad and certain payload

File on GitHub

Build an Interactive Map with Folium

What has been done?

Added Map(Simple)/Icons/Circles markers, and Lines

Why? To enhance/summarize visual insights as success and failure for each landing site

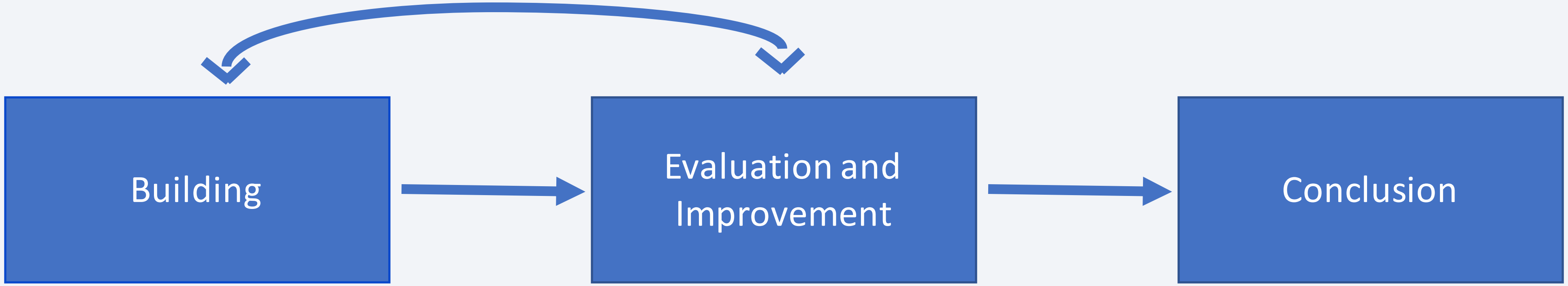
File on GitHub

Build a Dashboard with Plotly Dash

- What has been done?
  - Selection of Launch Site
  - Pie charts for relation percentage based on launch site
  - Scatter graphs for correlation between payload and success based on launch site
- Why? To visualize the effectiveness of launch site and payload mass

File on GitHub

Predictive Analysis (Classification)



- Standardize/transform data
- Split into test/training sets
- Using training set initialize different ML algorithm

- Check for accuracy via
  - R-score
  - Confusion matrix (true/false vs land./not land.)

=> Search for best
- Score
-> Parameters for ML

- Depending on accuary: Choose best model!

File on GitHub

Results

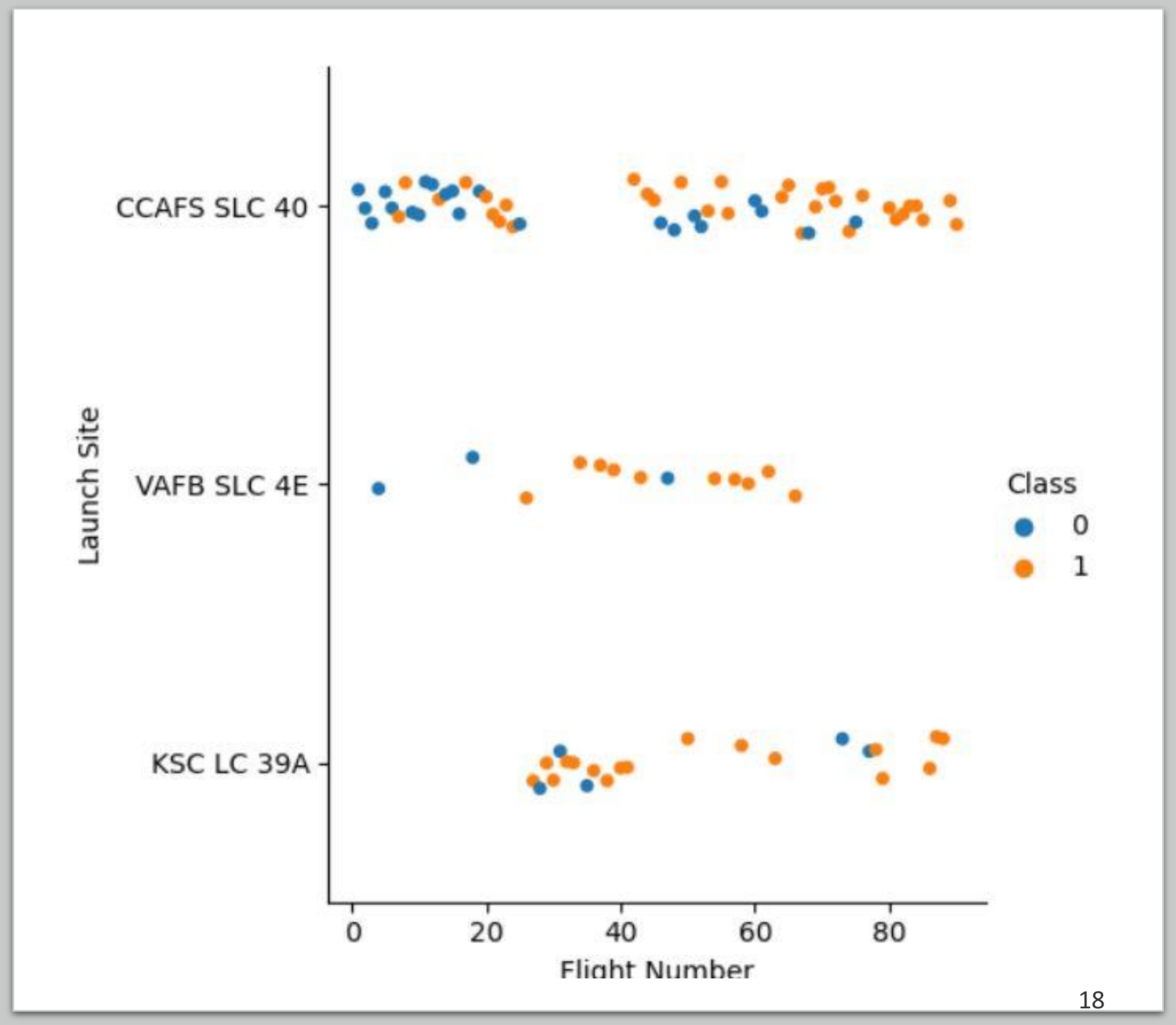
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

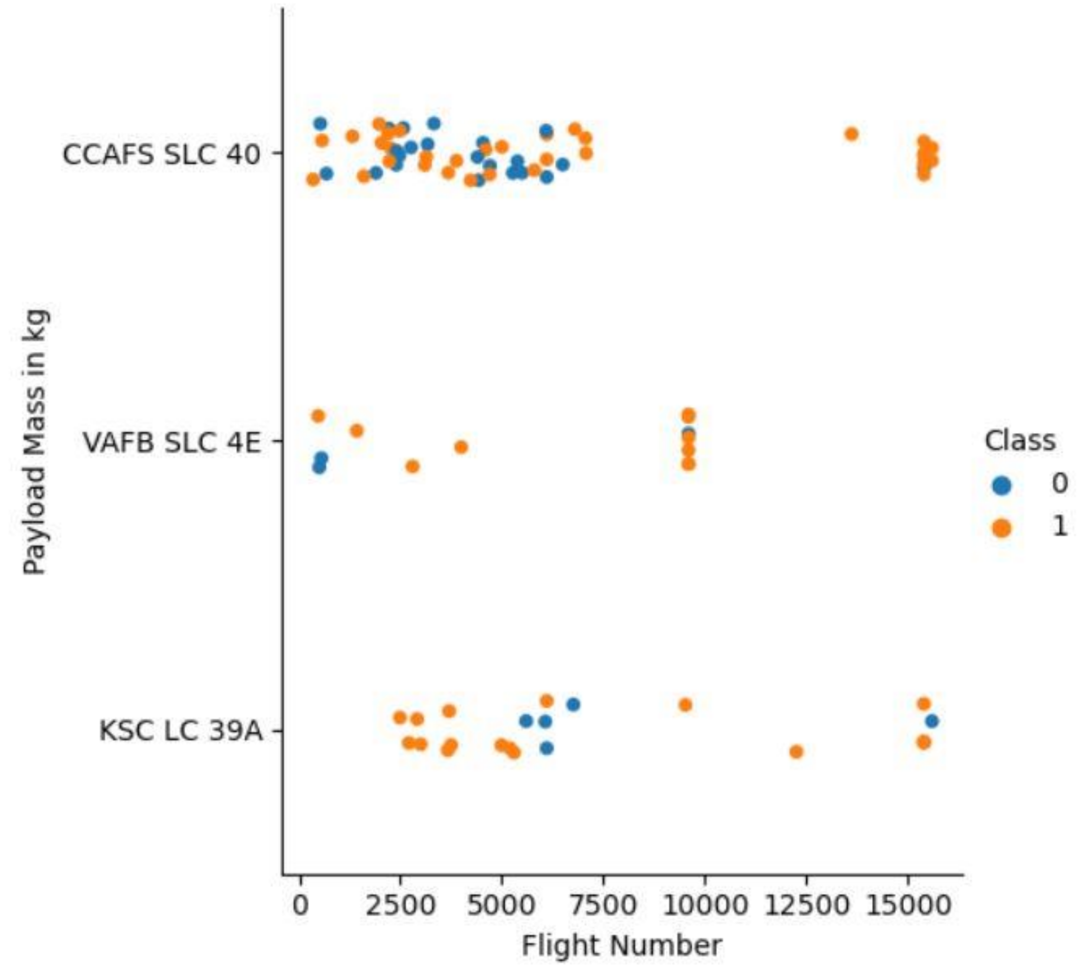
Flight Number vs. Launch Site

- First ~25 Starts, in particular. at CCAFS SLC 40 were mostly failures
- Starts at KSC LC 39A gave them then information for success
- VAFB SLC 4E with most successful launches (relatively)



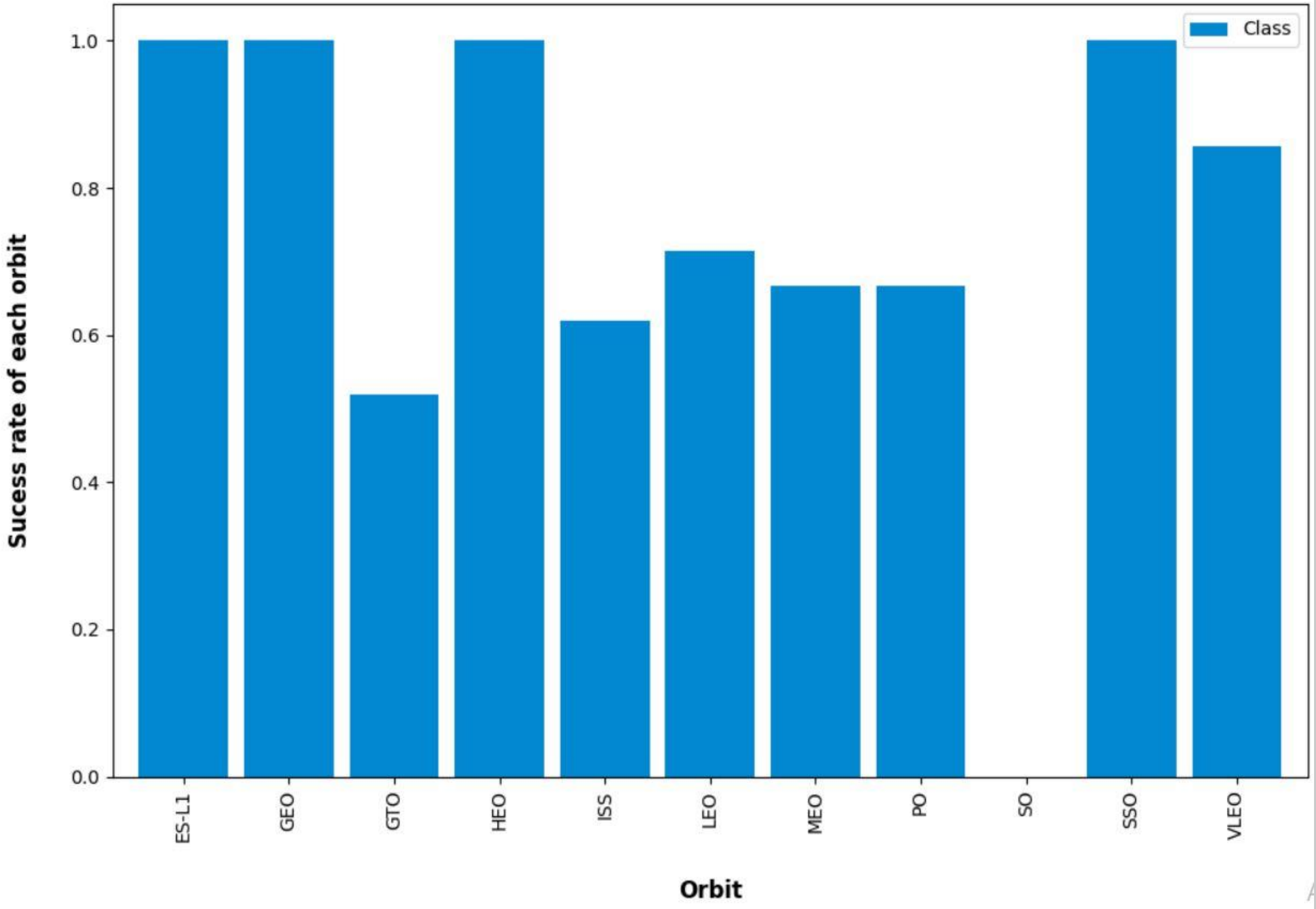
Payload vs. Launch Site

- After successfully establish a launch with low mass ~7500kg,
- Launches with higher payload masses were almost always successful



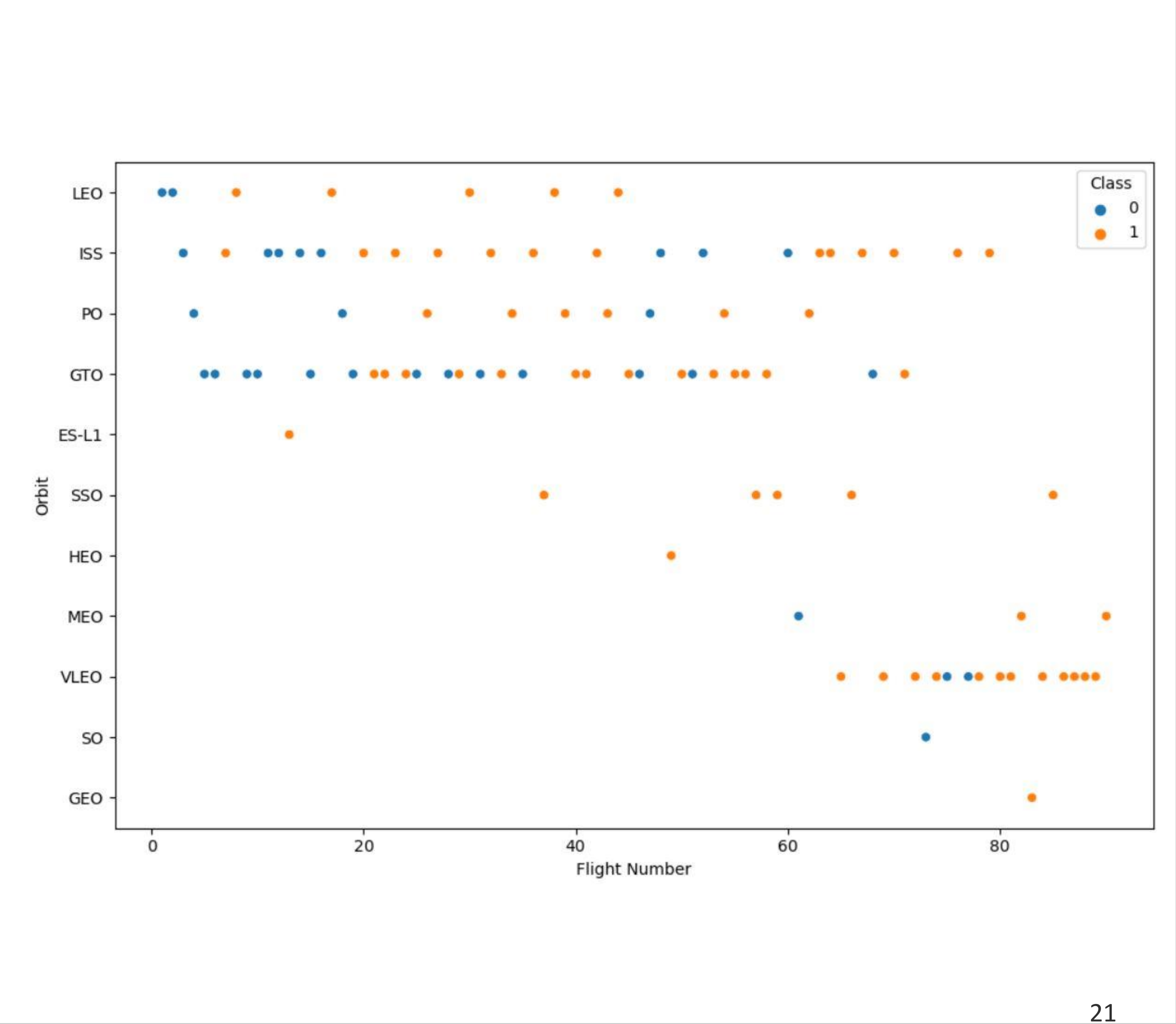
Success Rate vs. Orbit Type

- Launches onto SO were disastrous (only one launch)
- ES-L1, GEO, HEO and SSO with success only



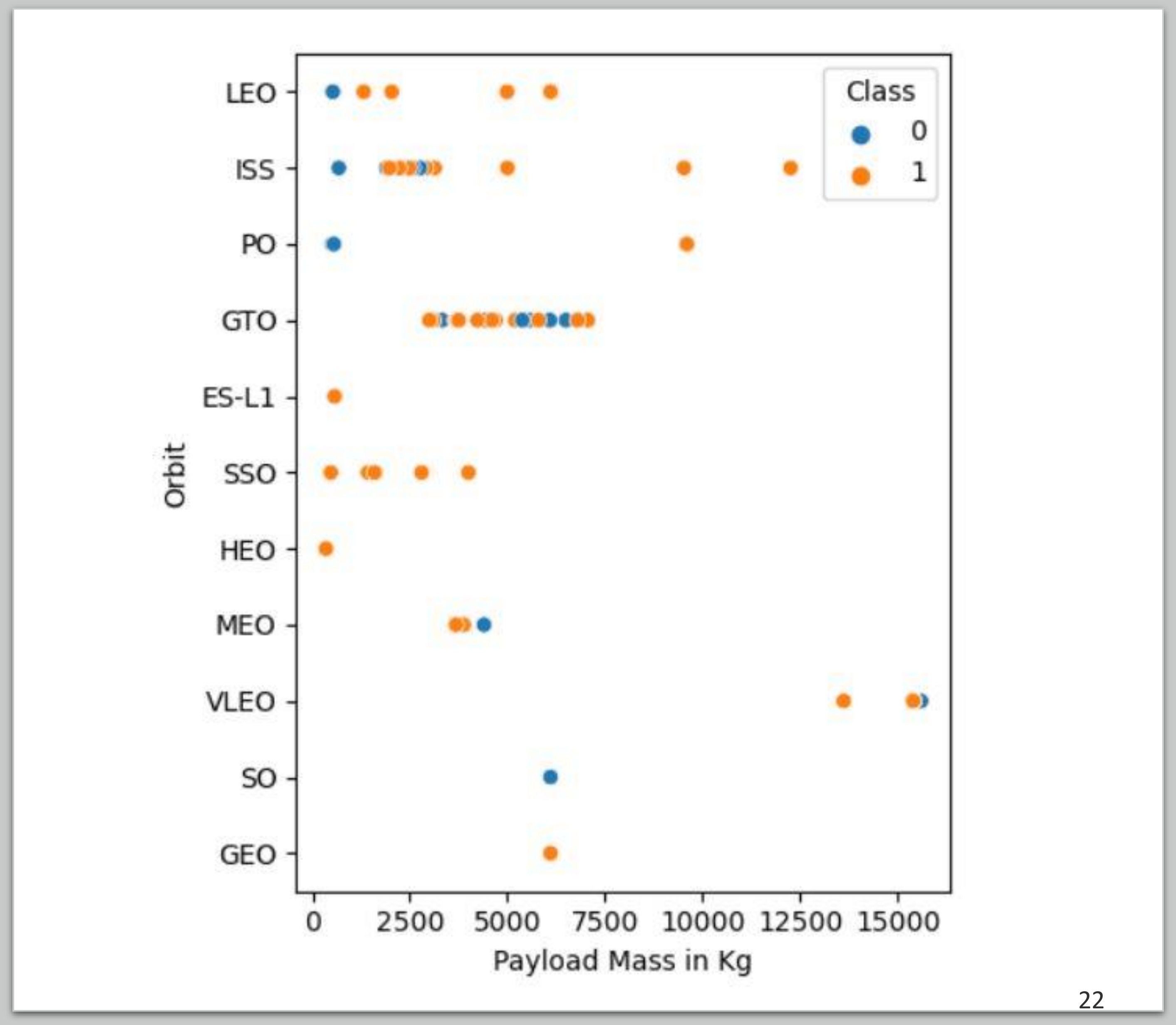
Flight Number vs. Orbit Type

- At the beginning, testing on four selected orbits
- ISS and GTO gave them a breakthrough after flight number ~ 20
- After flight number ~ 60 also focused on other orbits, in particular on VLEO



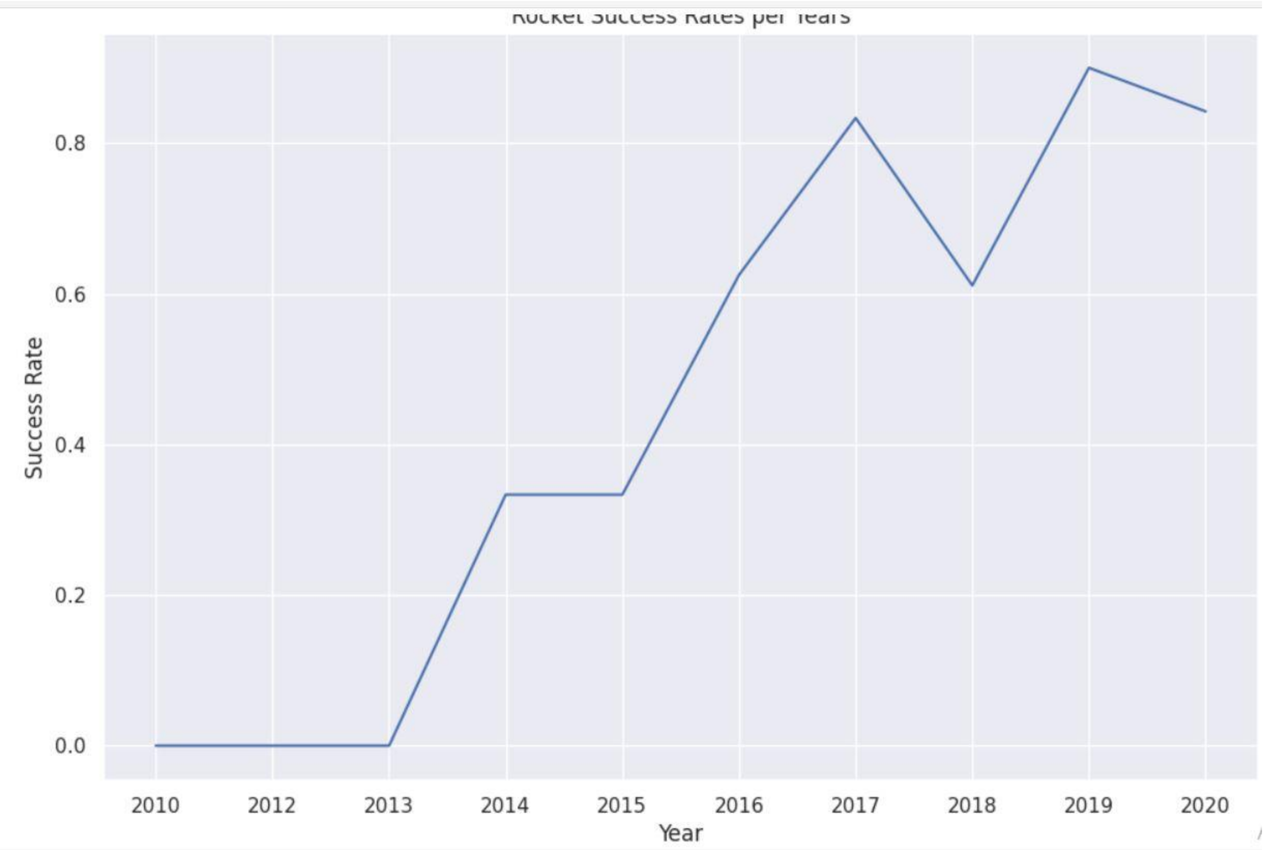
Payload vs. Orbit Type

- Started mostly with ~2500-7500kg
- Launches with higher payload > 7500kg rather successful



Launch Success Yearly Trend

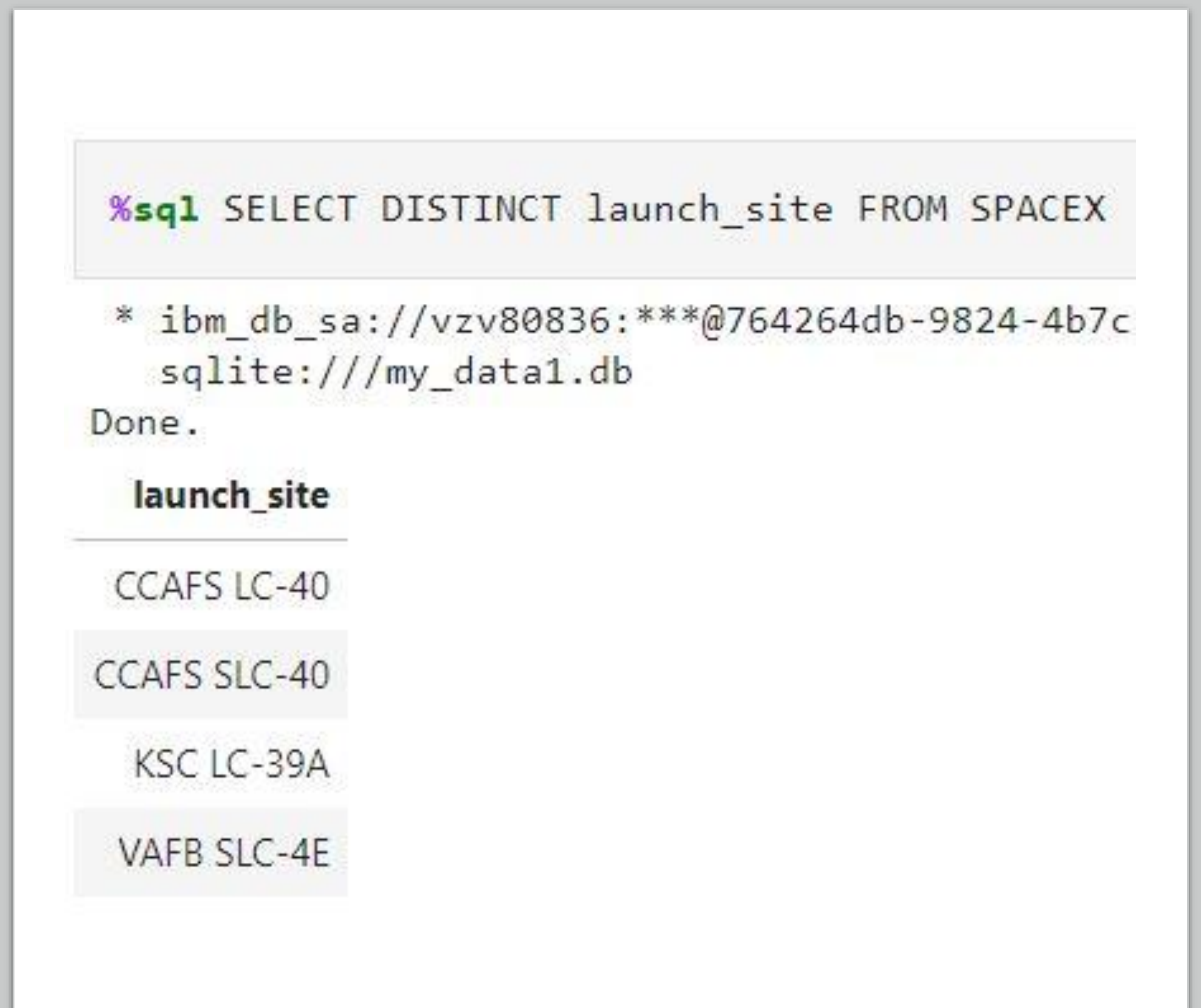
- Since 2013 mostly improving
- Since 2017 mostly >80% except for
- Dip in Success in 2018



EDA with SQL

All Launch Site Names

- Find the names of the unique launch sites
- Key word DISTINCT does the job



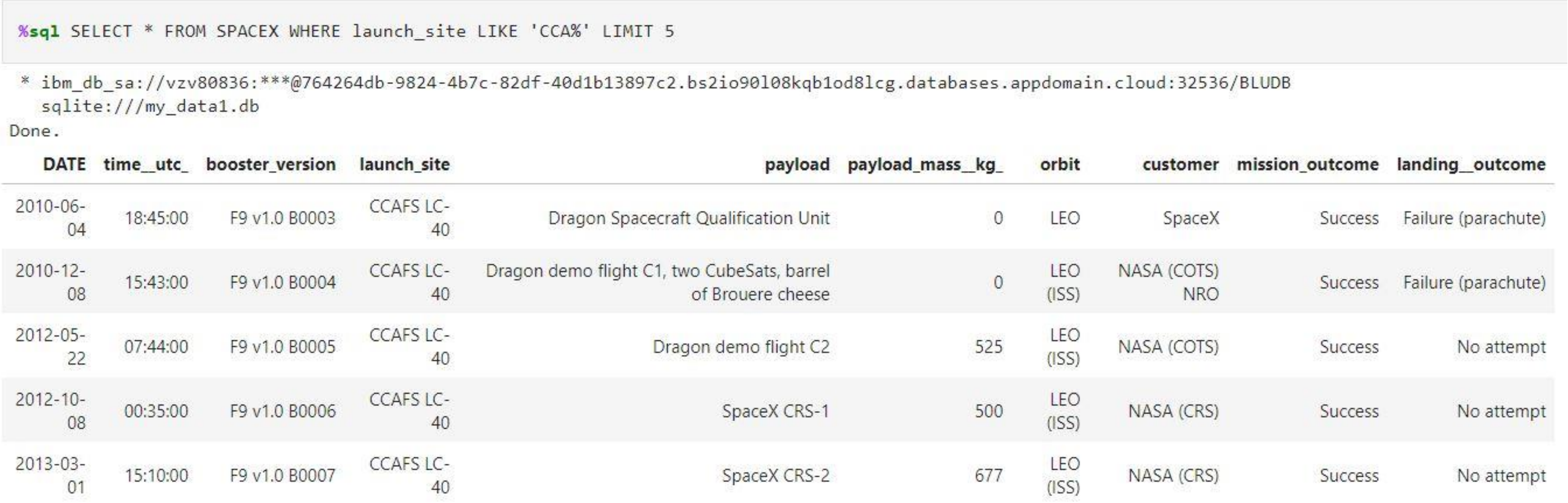
```
%sql SELECT DISTINCT launch_site FROM SPACEX
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c
   sqlite:///my_data1.db
Done.
```

```
   launch_site
  CCAFS LC-40
 CCAFS SLC-40
   KSC LC-39A
  VAFB SLC-4E
```

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with \`CCA\`
- Make use of the WHERE keyword for the condition and use % as a wildcard for an arbitrary ending



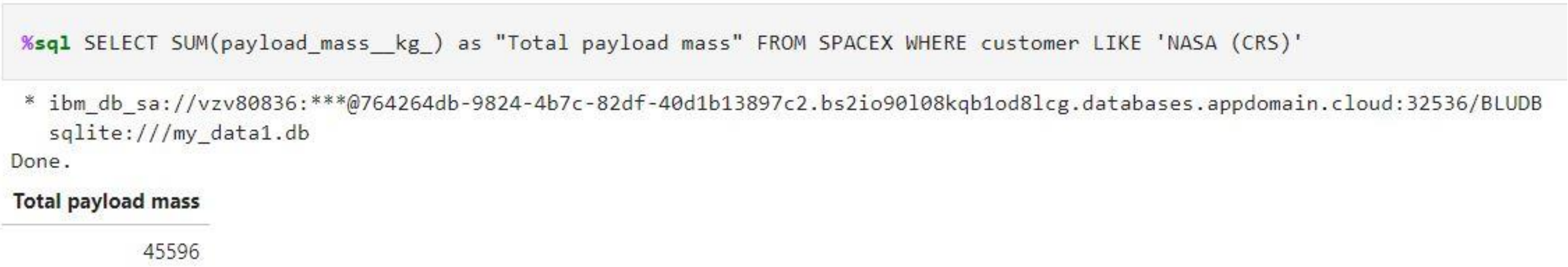
```
%sql SELECT * FROM SPACEX WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Use the function SUM to create a new column and thus, the total payload



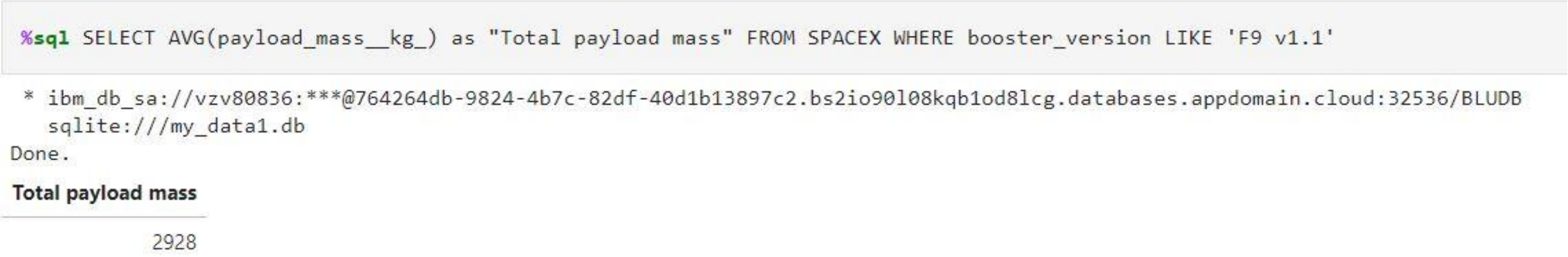
```
%sql SELECT SUM(payload_mass__kg_) as "Total payload mass" FROM SPACEX WHERE customer LIKE 'NASA (CRS)'
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

Total payload mass
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Use the function AVG to create a new column and thus, the desired average



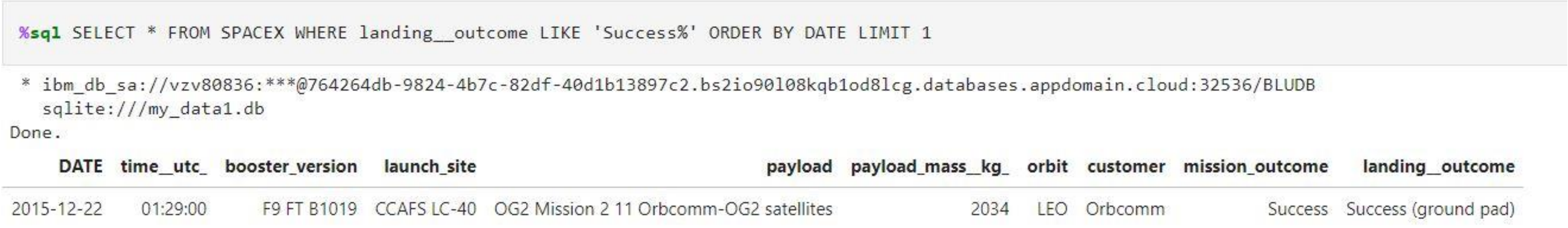
```
%sql SELECT AVG(payload_mass__kg_) as "Total payload mass" FROM SPACEX WHERE booster_version LIKE 'F9 v1.1'
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

Total payload mass
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Use ORDER to get a descending sequence of dates and therefore, the first entry gives the desired result, implemented via LIMIT 1



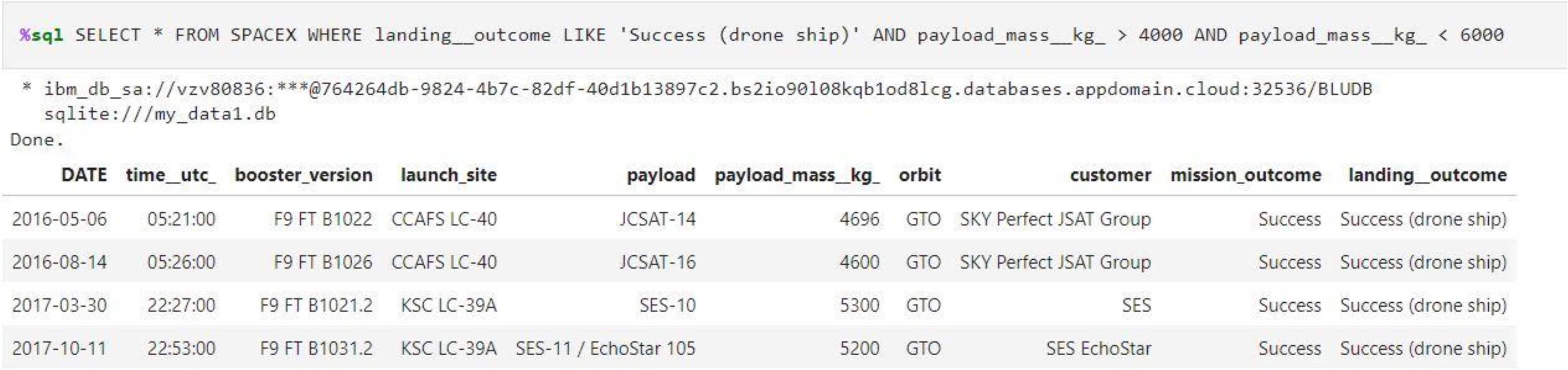
```
%sql SELECT * FROM SPACEX WHERE landing__outcome LIKE 'Success%' ORDER BY DATE LIMIT 1
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- A combination of two conditions in the WHERE clause is realized via the AND keyword



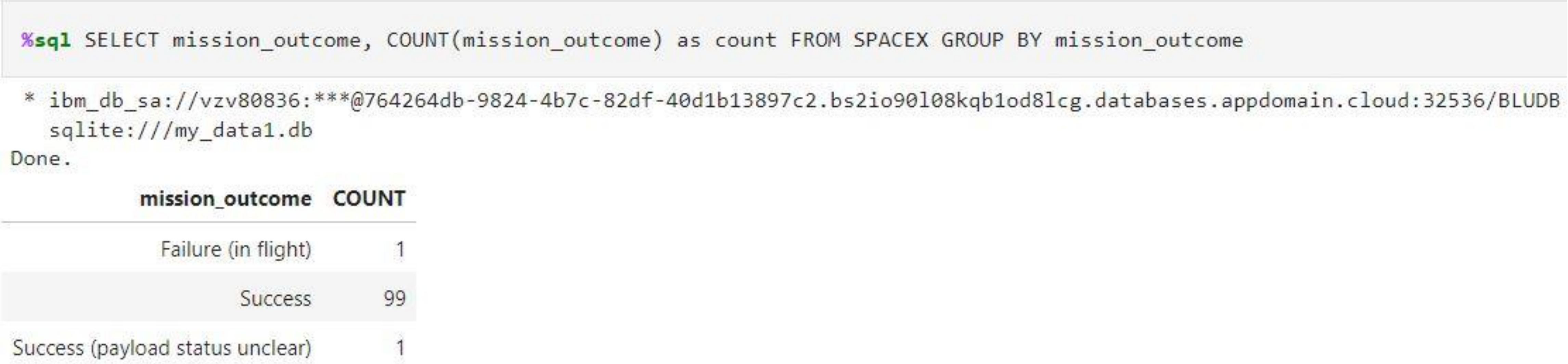
```
%sql SELECT * FROM SPACEX WHERE landing__outcome LIKE 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Using the function COUNT one can count the amount of entries But, we would like to do so with a grouping of result which is done via the GROUP BY keyword



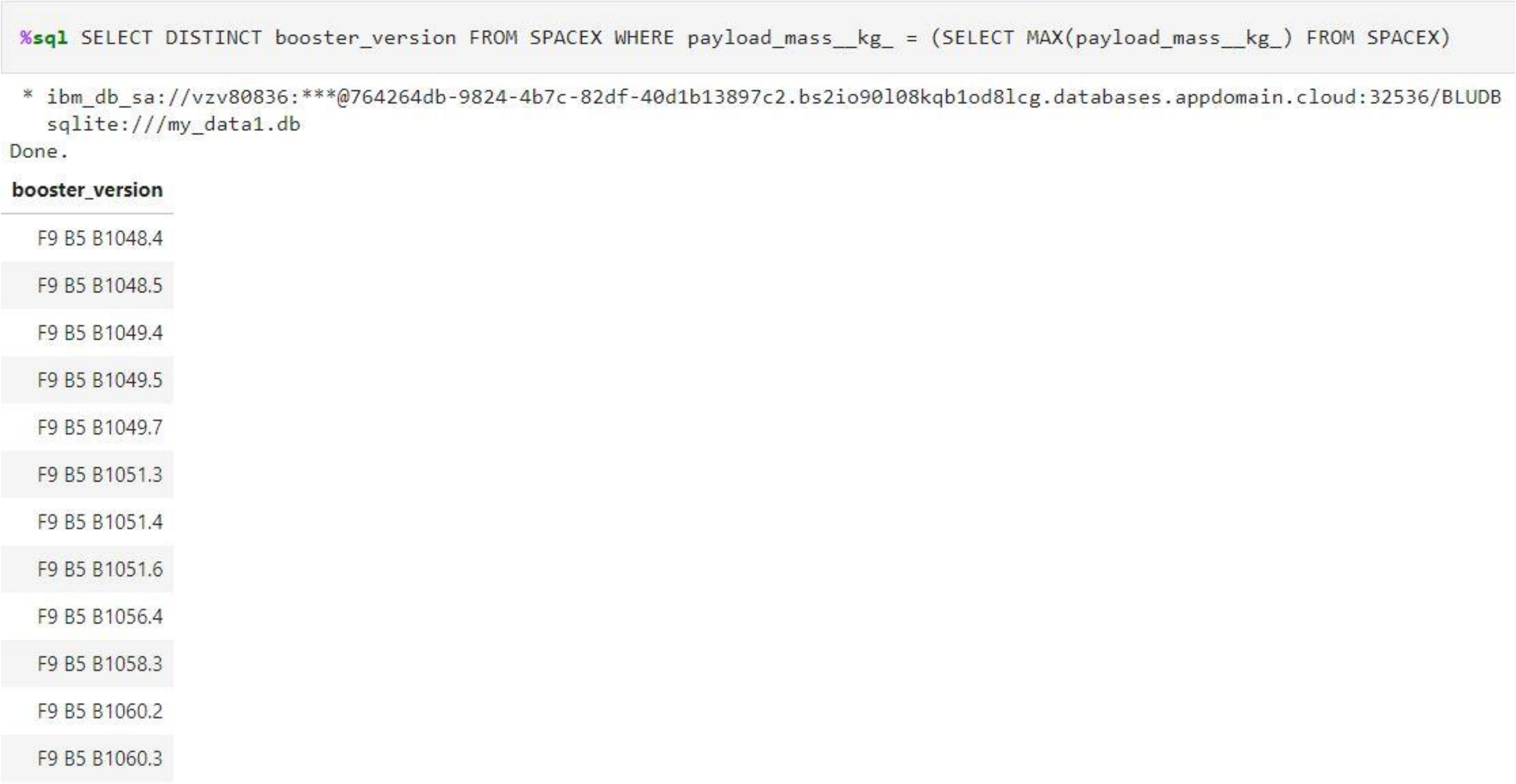
```
%sql SELECT mission_outcome, COUNT(mission_outcome) as count FROM SPACEX GROUP BY mission_outcome
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used a second query to find the maximal payload in which we used the MAX function. With this information, we can employ it in the condition part.



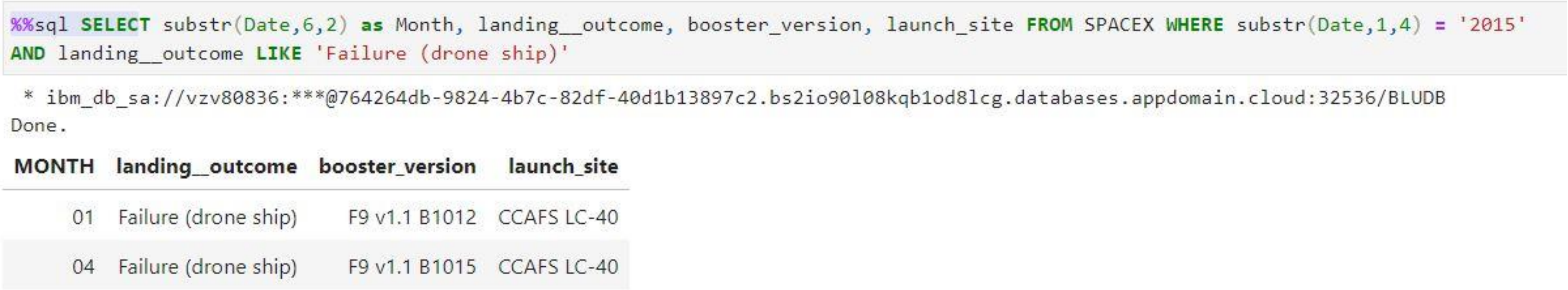
```
%sql SELECT DISTINCT booster_version FROM SPACEX WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEX)
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
   sqlite:///my_data1.db
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Here we use a specialty of SQLite, the substr function, i.e. it gives a substring based on an index and some length. We apply it on the date variable to extract year and month



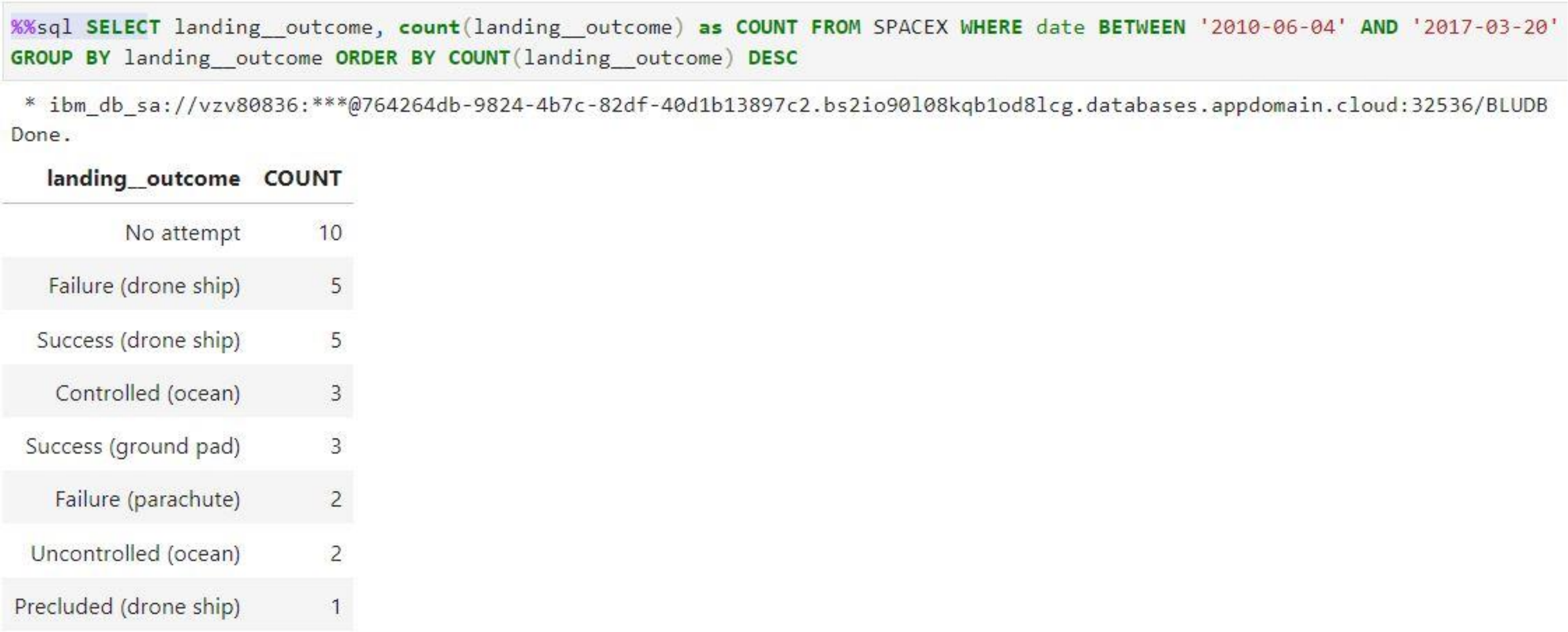
```
%%sql SELECT substr(Date,6,2) as Month, landing__outcome, booster_version, launch_site FROM SPACEX WHERE substr(Date,1,4) = '2015'
AND landing__outcome LIKE 'Failure (drone ship)'
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

MONTH	landing__outcome	booster_version	launch_site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Used the DESC keyword to get a descending list which is ordered by the amount of landing outcomes while satisfying all other conditions



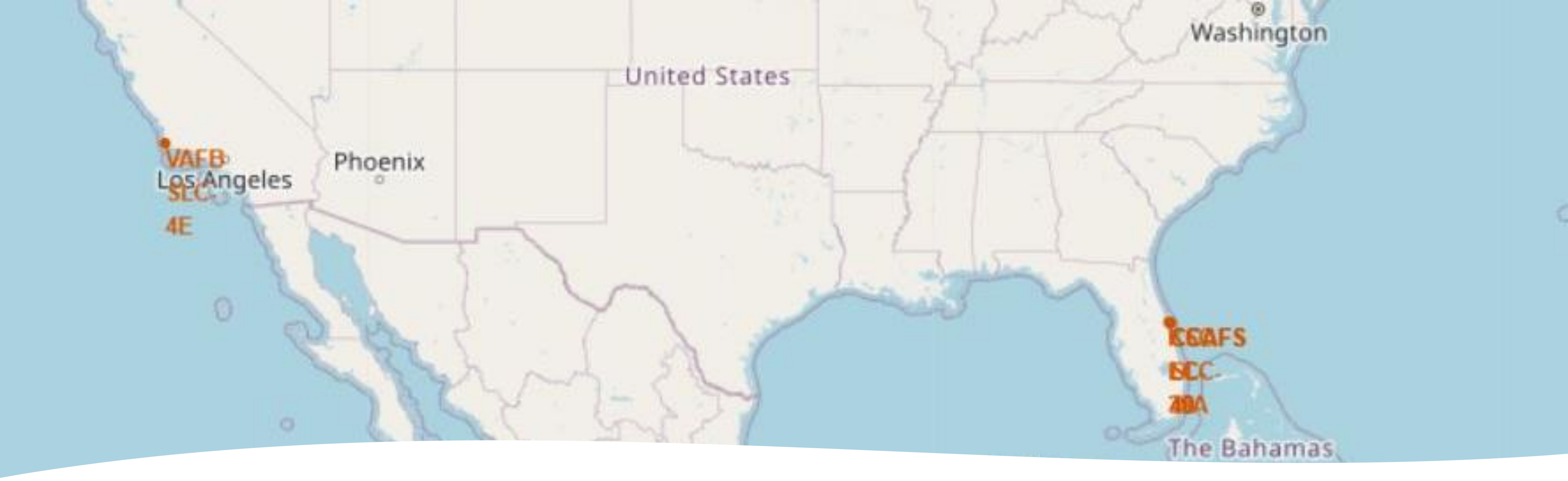
```
%%sql SELECT landing__outcome, count(landing__outcome) as COUNT FROM SPACEX WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome ORDER BY COUNT(landing__outcome) DESC
```

```
 * ibm_db_sa://vzv80836:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

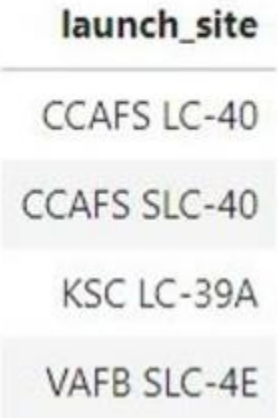
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis



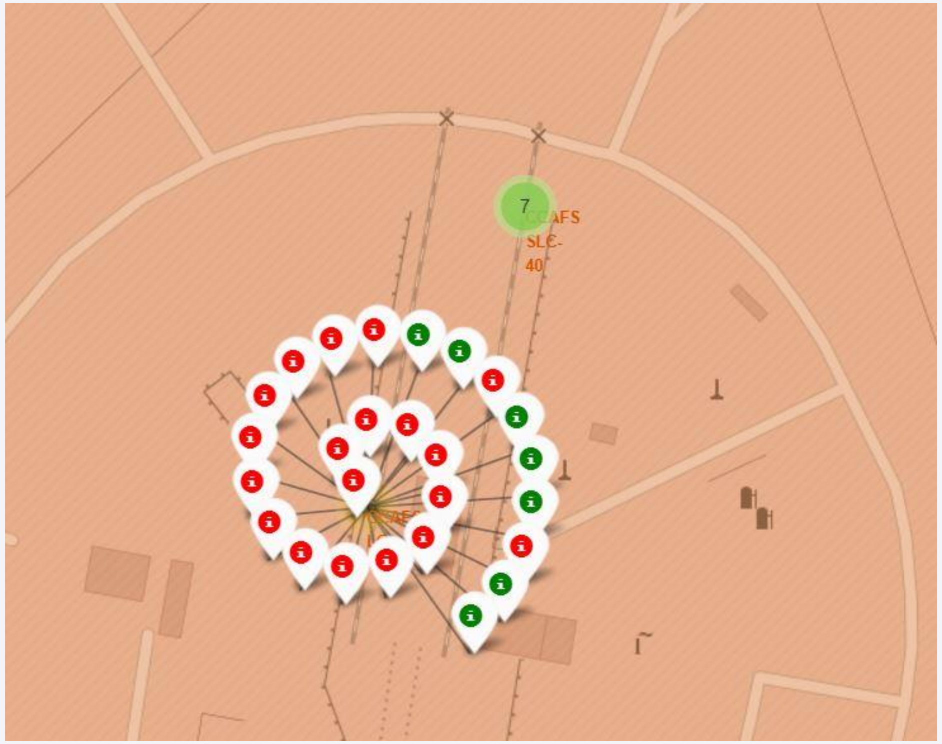
Launch Sites on Map

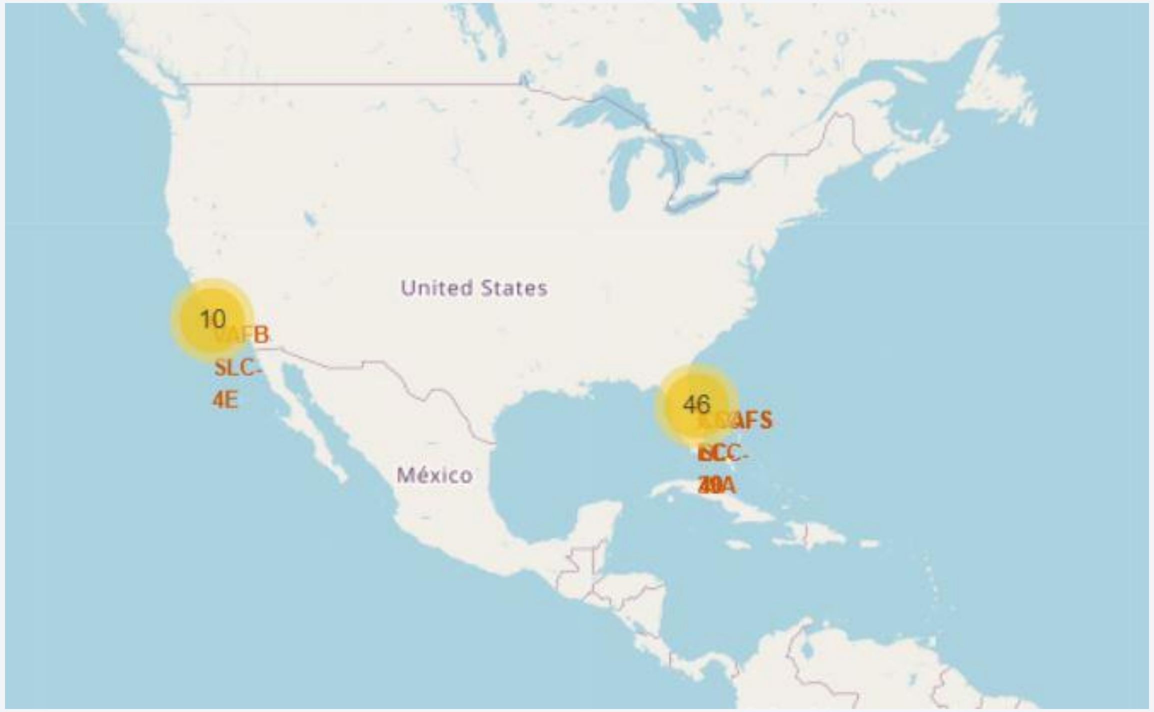


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

- Launch Sites are located on the east and west coast of the US
  -> precisely in California and Florida

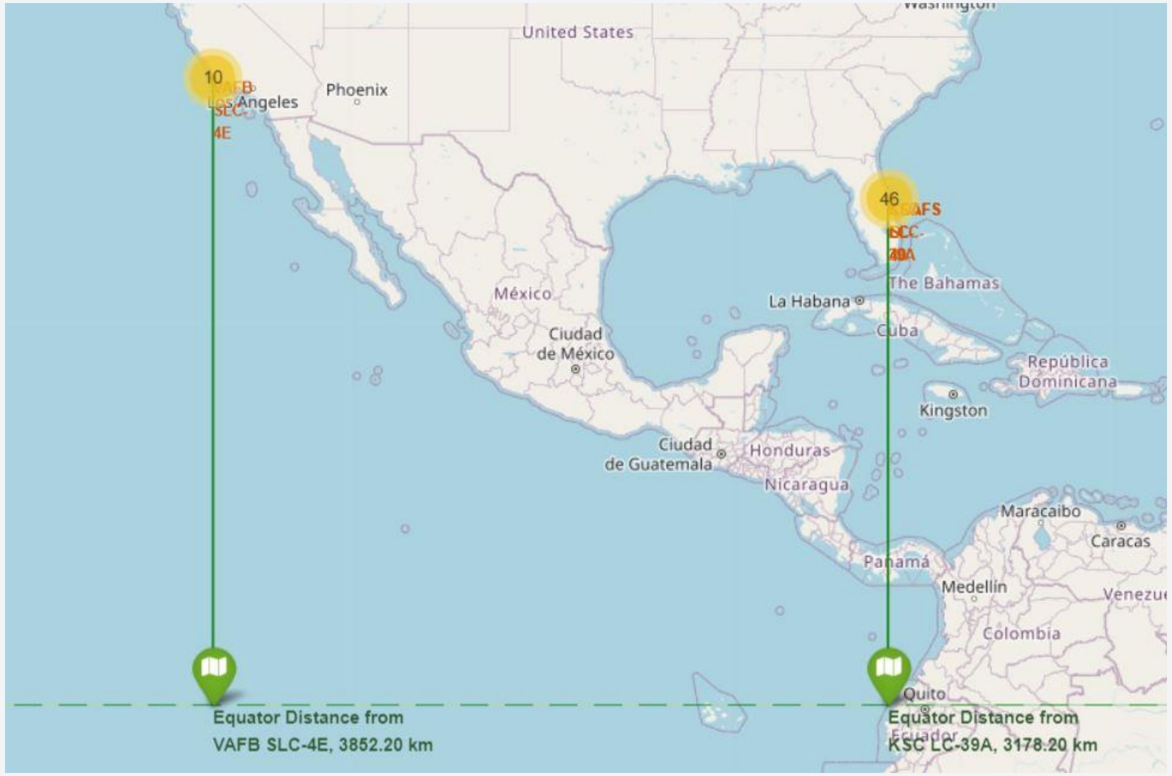
Success on Launch Site

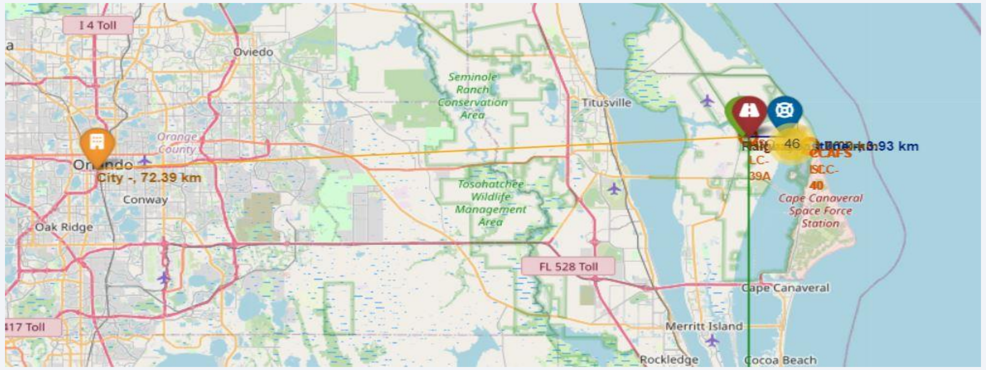


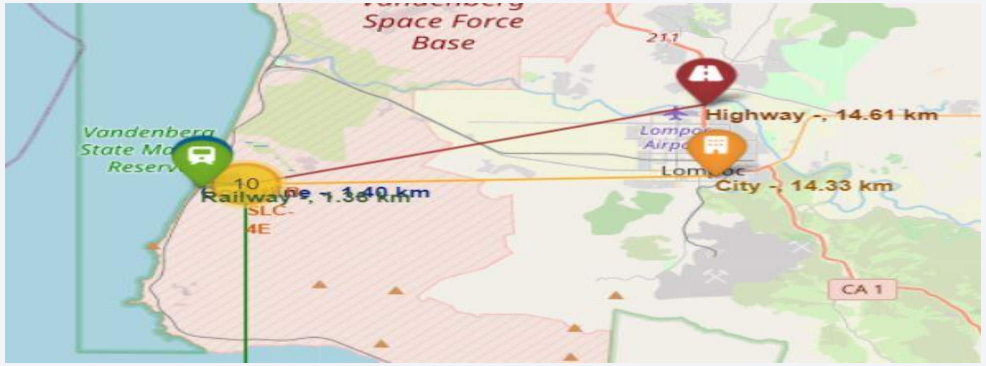


- Added Markers to indicate success and failure at each launch site.
- Due to their size they are summarized in larger zooming views

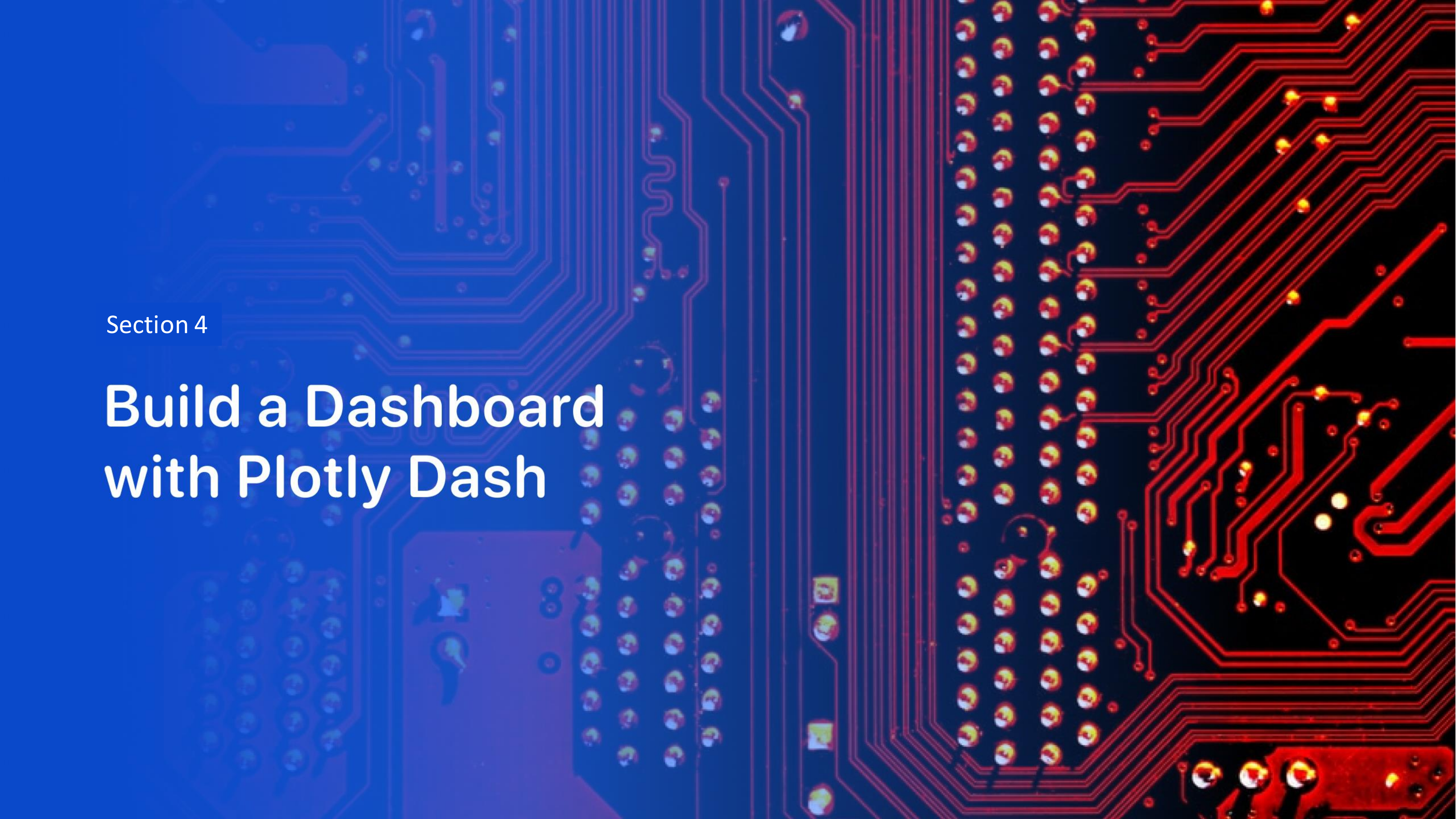
Distances from launch sites to equator and environment







- Launch Sites are directly at the coasts but ~ 10-15km away from the next big city and next highway, safe enough in case of an accident
- Roughly 3000-4000km from the equator where the highest earth rotational speed of the earth is and thus, better launch conditions

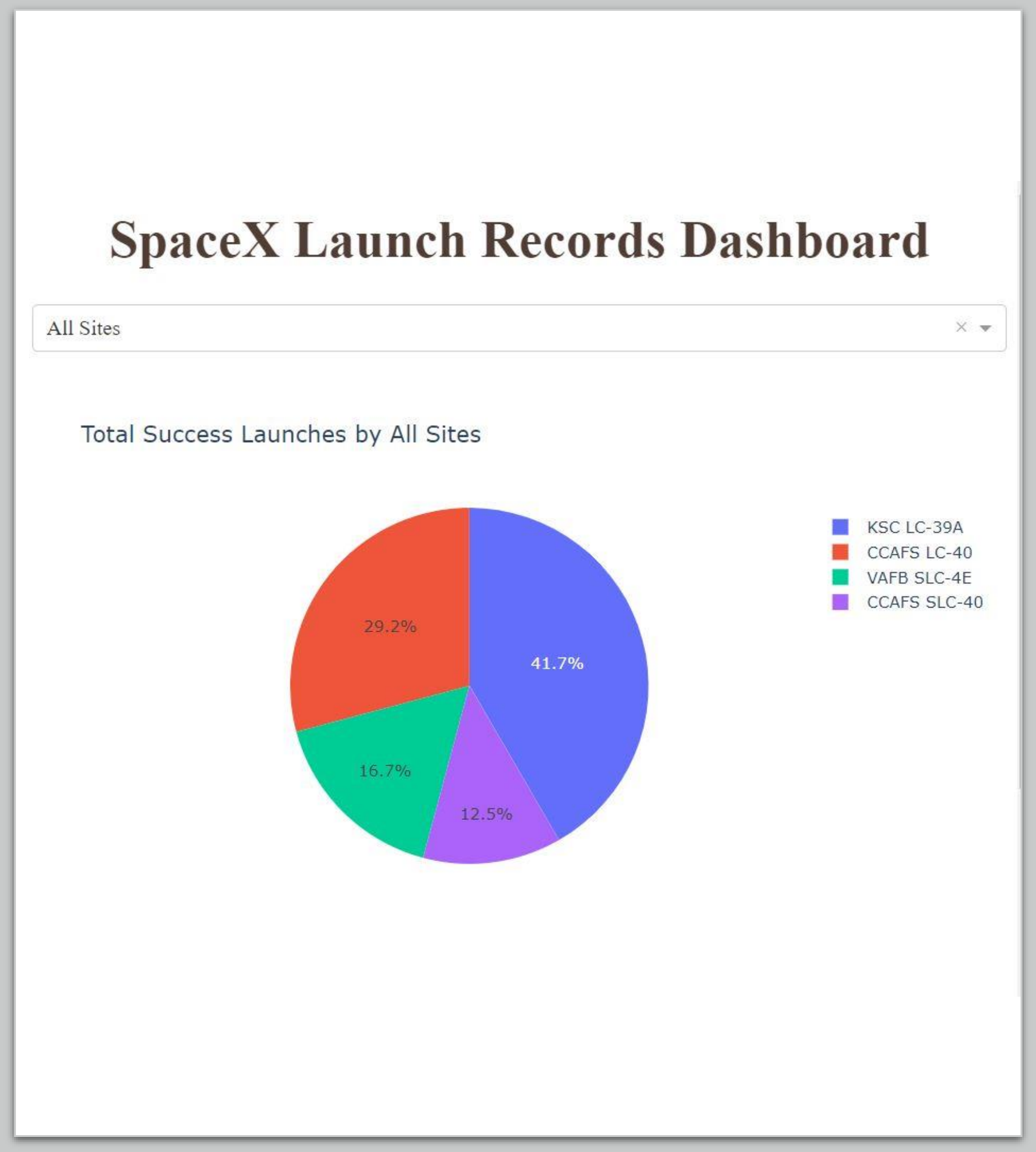


Section 4

Build a Dashboard with Plotly Dash

Launch Success: All Sites

- KSC-LC-39A has highest share of success
- CCAFS.SLC-40 has lowest share of success



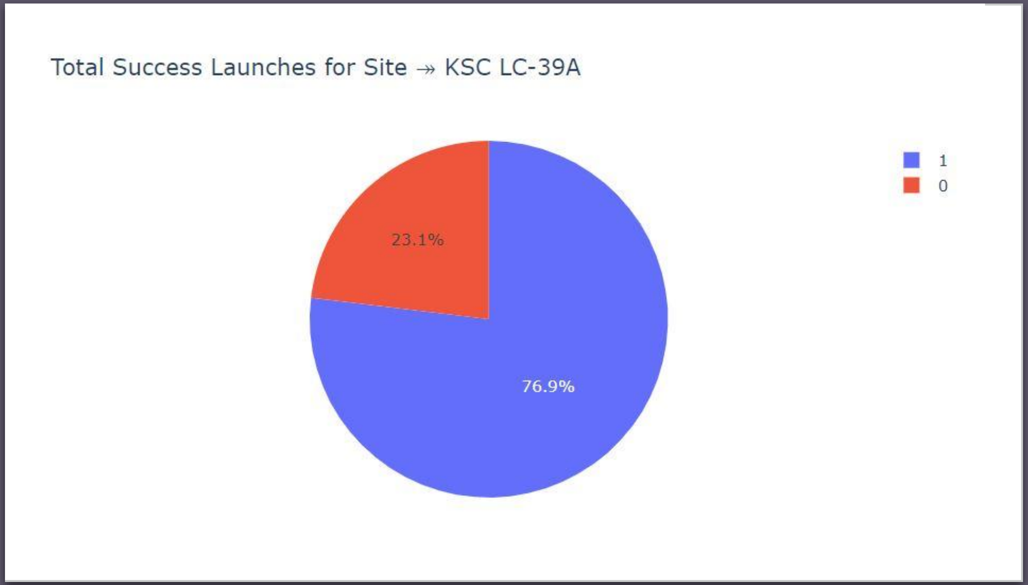
SpaceX Launch Records Dashboard

All Sites

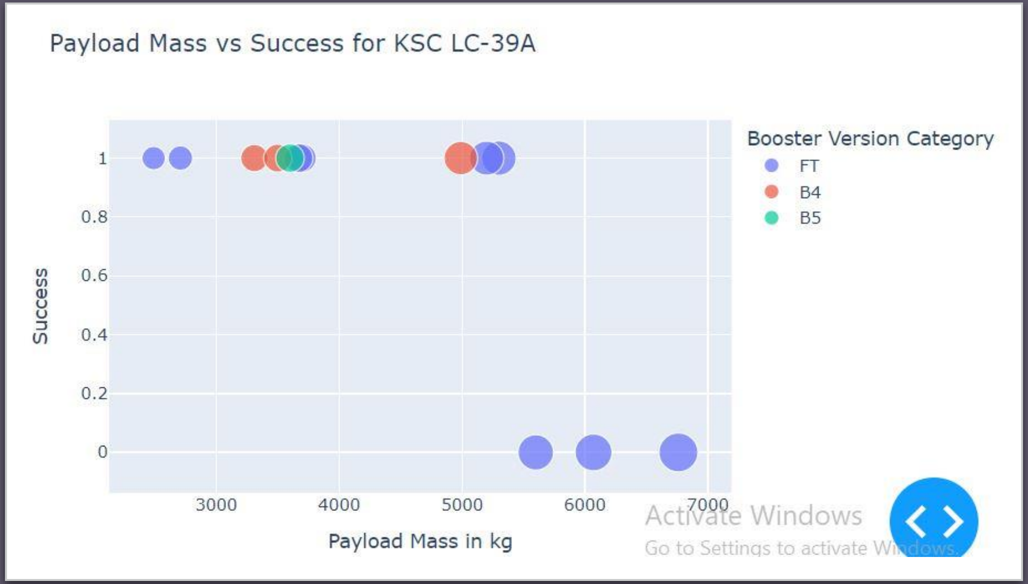
Total Success Launches by All Sites

KSC-LC-39A in Detail

- ~¾ Success rate
- Mixed success for FT boosters while overall best
- But, B4 and B5 version categories doing well



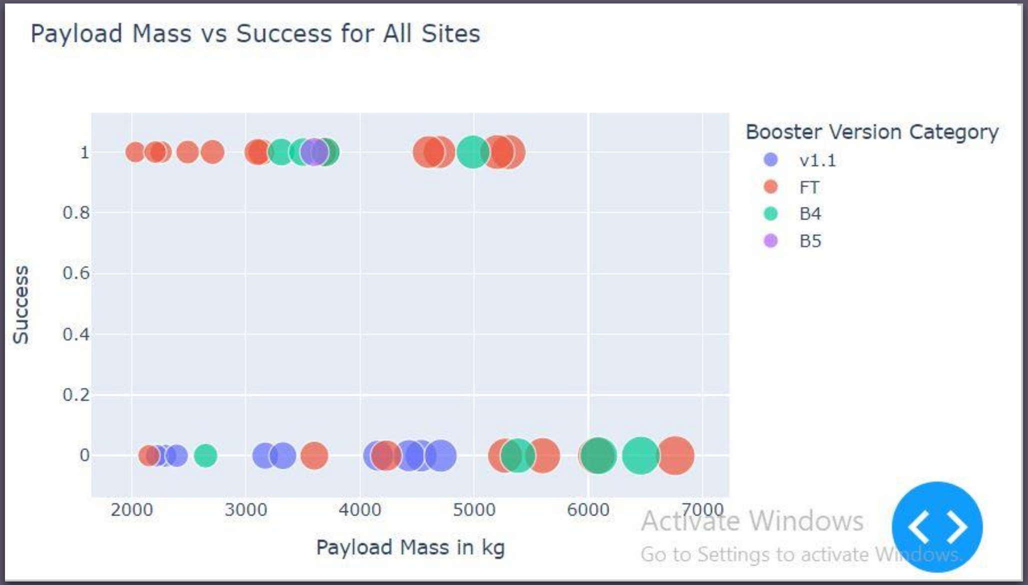
Total Success Launches for Site → KSC LC-39A

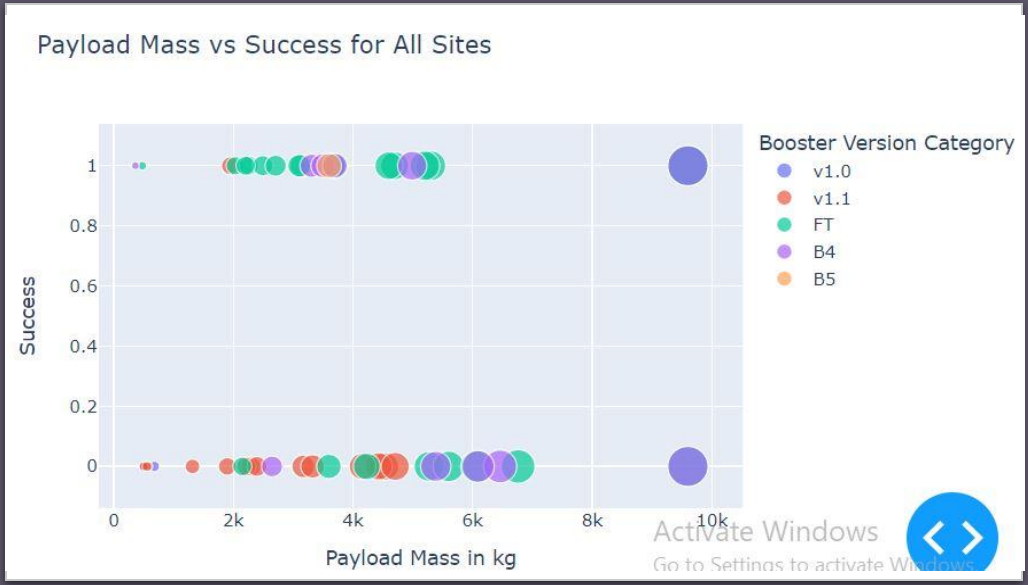


Payload Mass vs Success for KSC LC-39A

Payload vs Launch Outcome

- Generally, Booster category FT does well for lower payload masses
- For >5000kg rather failure
- For mid-size payloads (2k-7k) booster version v1.1 is miserable



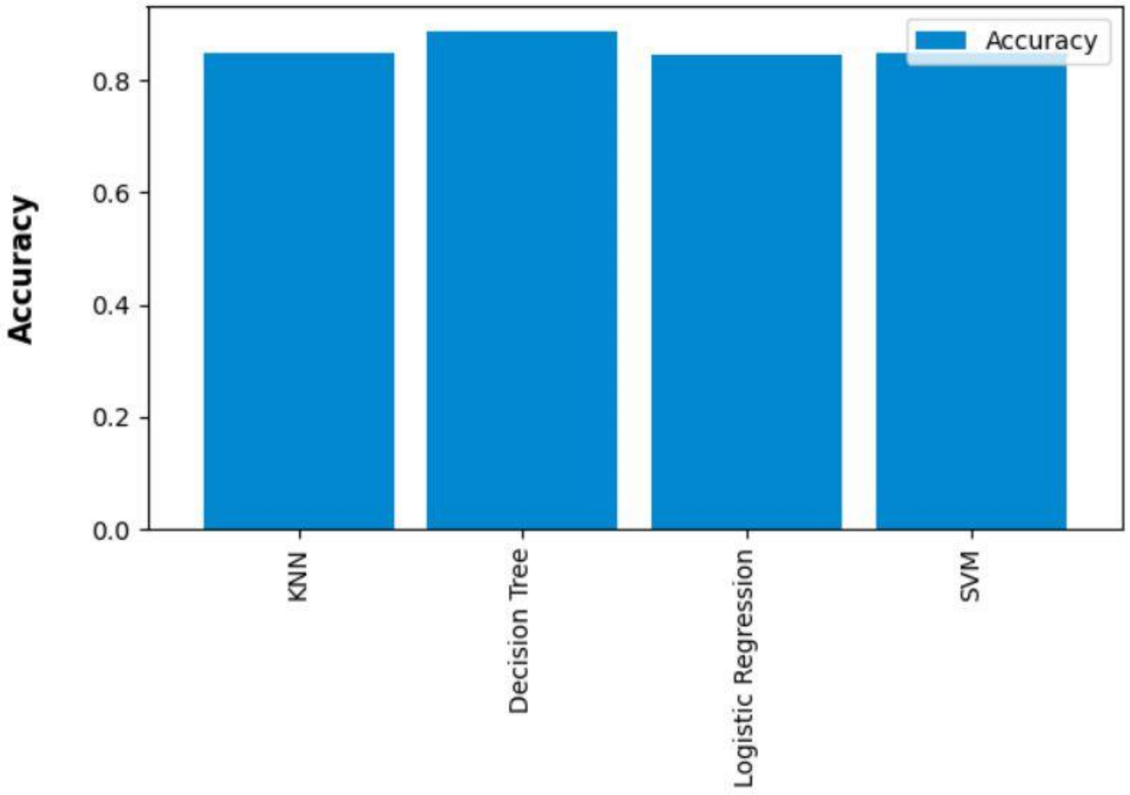


Section 5

Predictive Analysis (Classification)

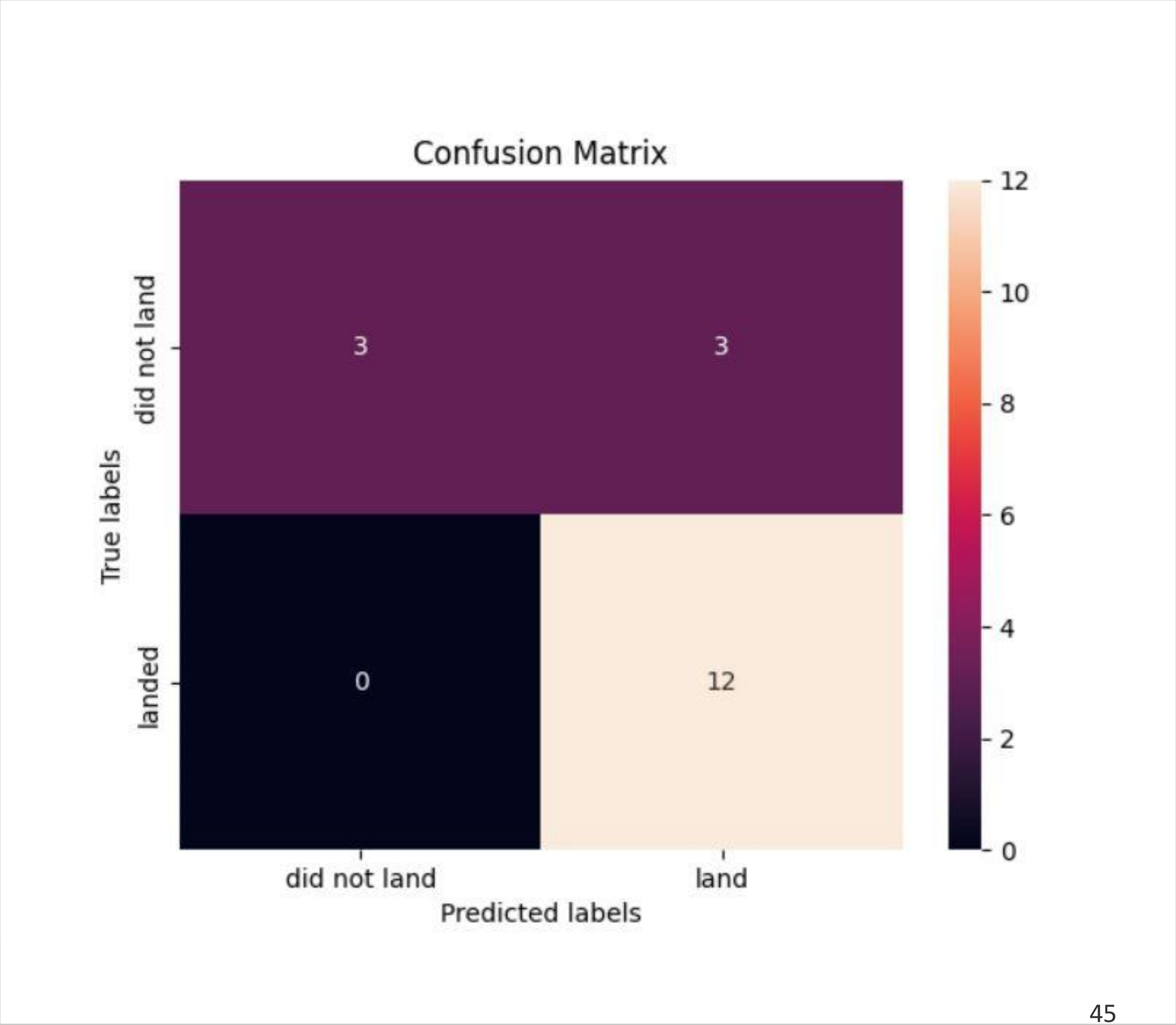
Classification Accuracy

- Decision Tree Model is the best with an accuracy with arround 91%
- Others perform only minorly worse with accuracy rates > 84%



Confusion Matrix for Decision Tree

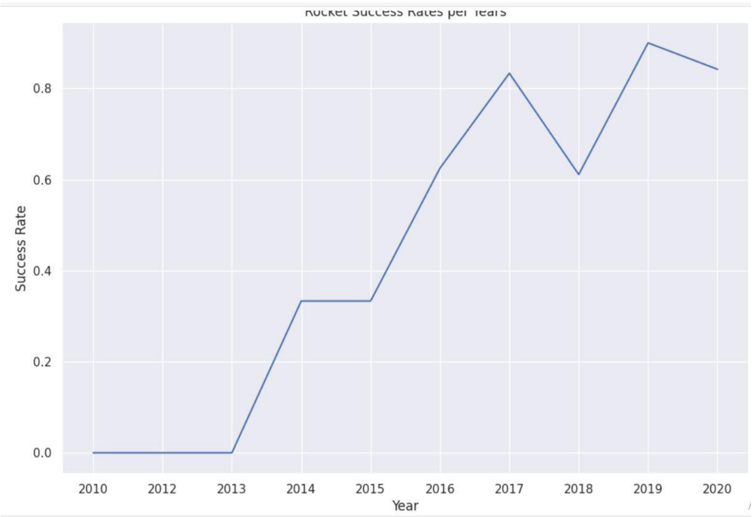
- True/false vs Land/Not Land Matrix
- Predicted landing outcomes for the test data=subset of original data
- Unfortunately, we have True/Not-Land outcomes
- But, overall 15/18 correct predictions

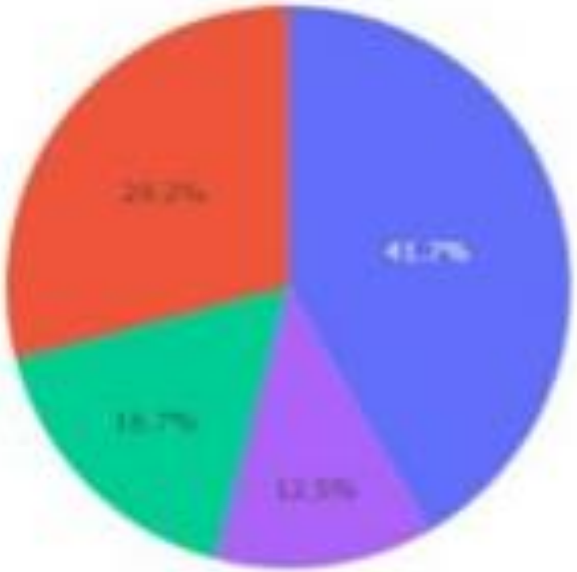


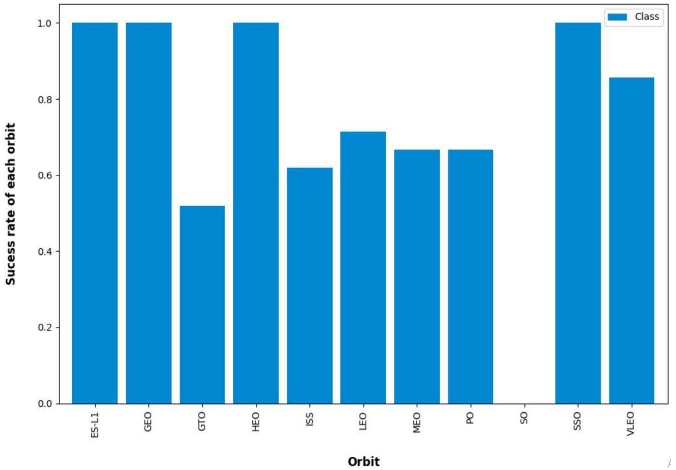
Conclusions

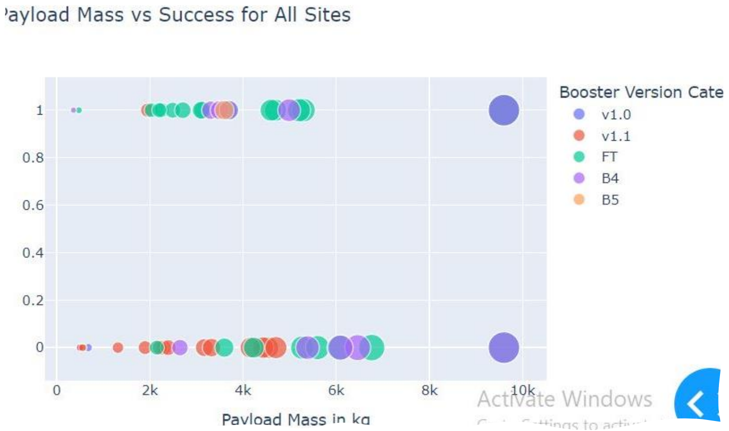
Conclusions

- Orbits ES-L1, GEO, HEO, SSO have highest success rates
- Success rate for launches increased over time
- KSC LC-39A had the most successful launches
- For higher payloads (>5000kg) rather failure
- Decision Tree is the best predictive Model









Thank you!