IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kavin Yudhitia

12 Sept 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- SpaceX data is used to create prediction model of first-stage landing

- **Result**: Found the best Hyperparameter for the classification model to predict successful first-stage landing

Introduction

- Background:
  - SpaceX rocket launch cost is only ~40% of other provider’s cost. This is because SpaceX reuse their first-stage.
  - If we could predict if the first-stage will land, we would be able to predict the launch cost
- Business question:
  - How can we build a predictive model to predict successful first-stage landing?
  - What is the recommended predictive model?

Section 1

Methodology

Methodology

Executive Summary

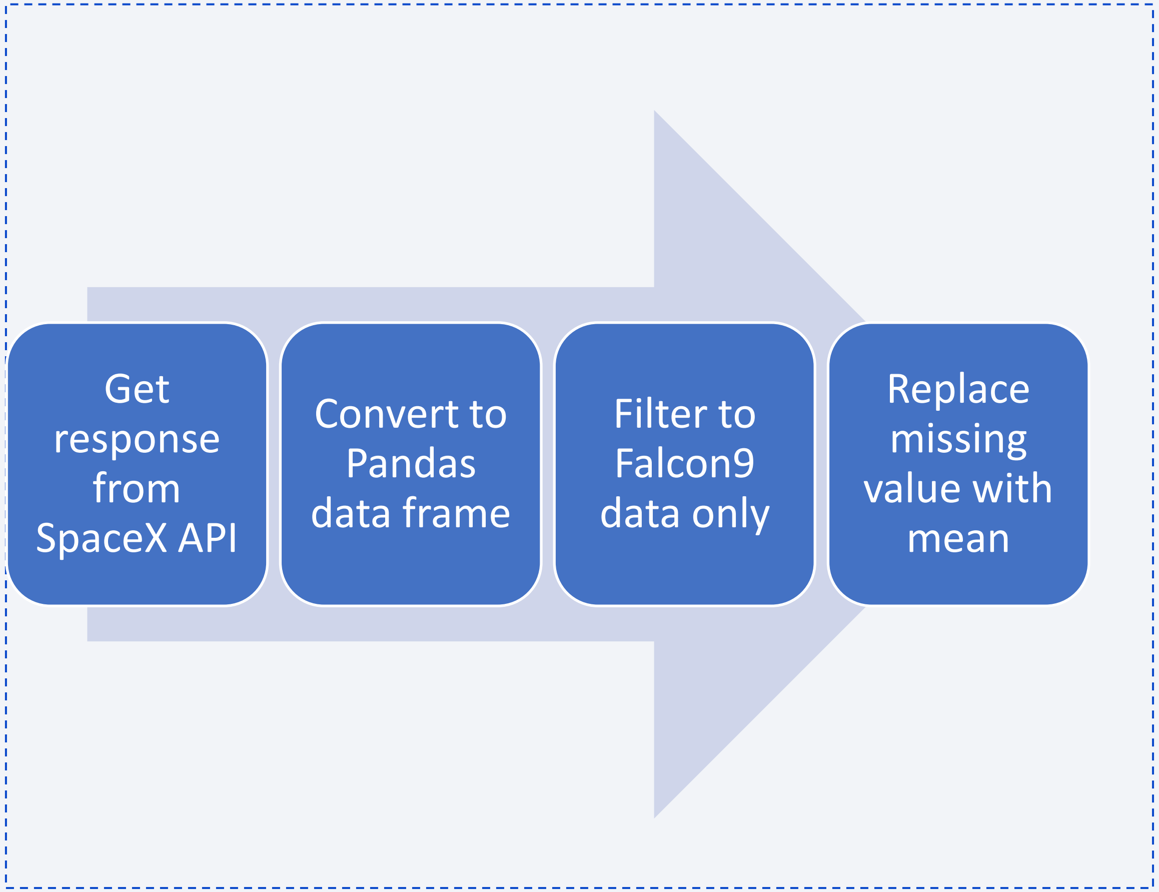
- Data collection methodology:
    - Data is collected from SpaceX API
- Perform data wrangling
    - Empty data is replaced with column average
    - Translate outcome to either “successful” or “failed” class
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Split the data to training and testing data. Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

- Data is collected from SpaceX API and also by webscraping

Data Collection – SpaceX API

- REST call is used to get data from SpaceX API
- Data receive is converted to right format to be further cleaned
- Github link


Get response from SpaceX API

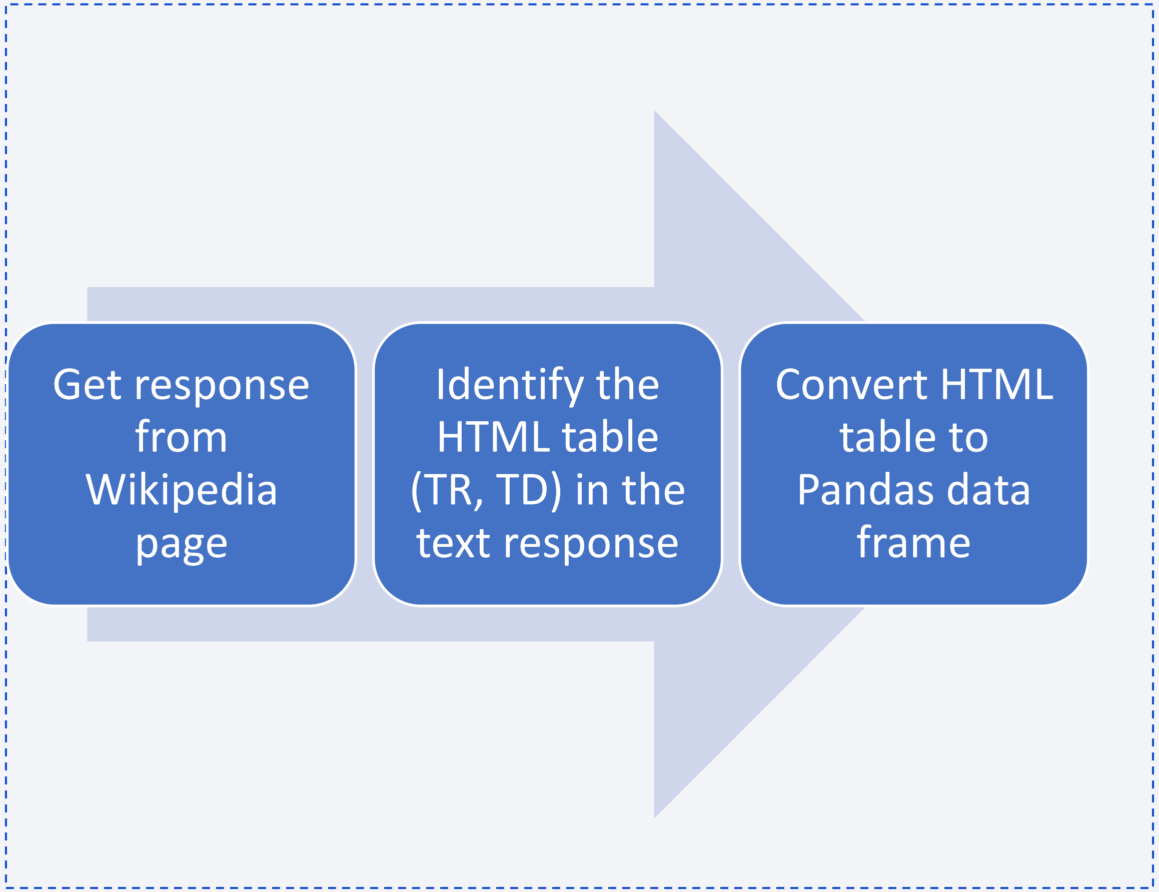
Convert to Pandas data frame

Filter to Falcon9 data only

Replace missing value with mean

Data Collection - Scraping

- Scraping is done from Wikipedia page titled “List of Falcon 9 and Falcon Heavy launches”
- In the website there is HTML table that we parse using BeautifulSoup
- Github link



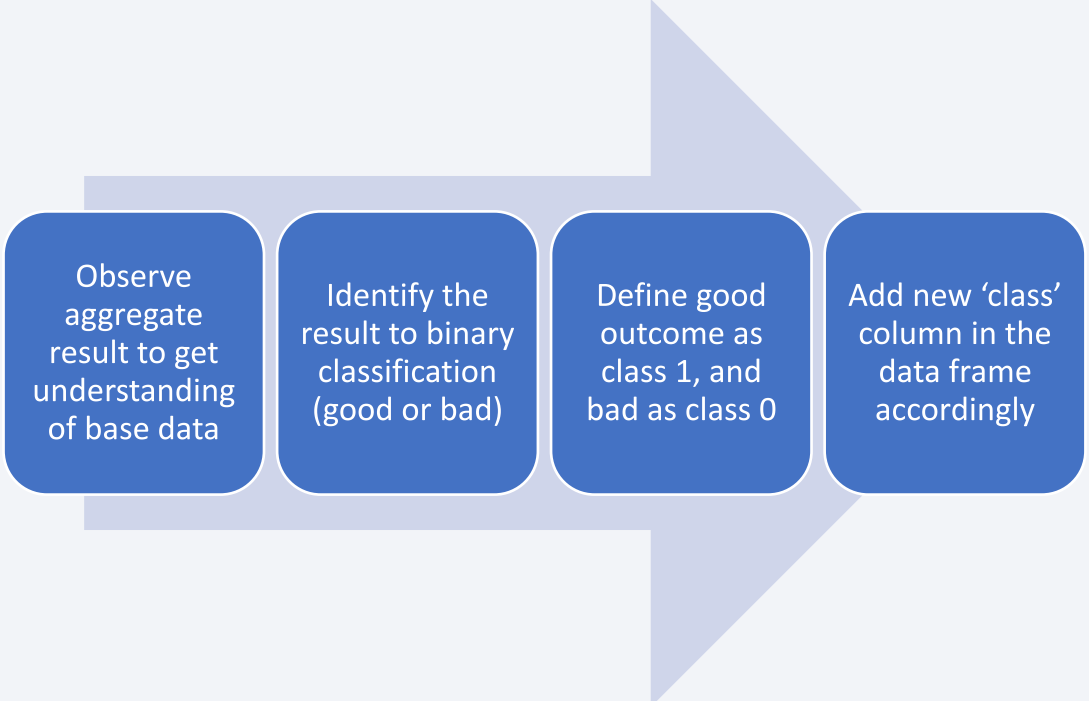
Get response from Wikipedia page

Identify the HTML table (TR, TD) in the text response

Convert HTML table to Pandas data frame

Data Wrangling

- We observed
  - # of launches for each site
  - # of each occurrence for each orbit
  - Outcome of each landing
  - All above by using value_counts() method
- We classified the “good” (Class: 1) and “bad” (Class: 0) outcomes to prepare for modeling later on.
- This become a new column in the dataframe
- Github link

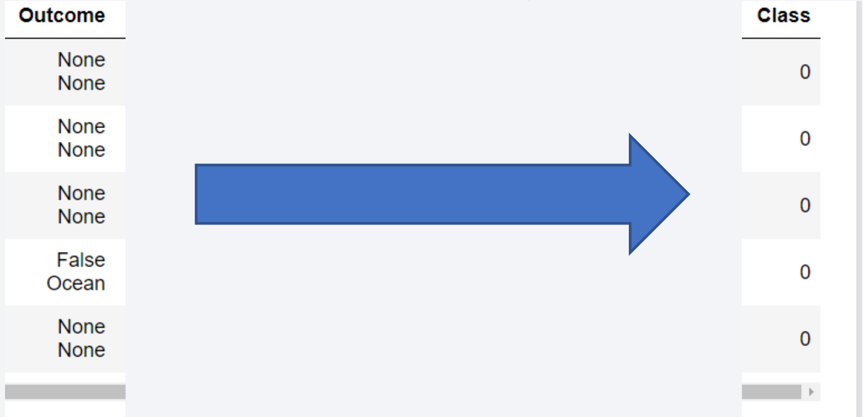


Observe aggregate result to get understanding of base data

Identify the result to binary classification (good or bad)

Define good outcome as class 1, and bad as class 0

Add new ‘class’ column in the data frame accordingly



Outcome
None None
None None
None None
False Ocean
None None

Class
0
0
0
0
0

EDA with Data Visualization

- Chart plotted
  - Scatter plot, to visualize correlation between 2 attributes + the outcomes. And to segregate if any range of values has relation to the outcomes
    - Flight # VS Payload mass, Flight # VS Launch site, Launch site VS Payload mass
    - Flight # VS Orbit, Payload mass VS orbit
  - Bar chart, to summarize which orbit has highest success rate.
  - Line chart, to see the trend of success rate given the time
- Github link

EDA with SQL

- SQL queries are used to explore:
  - Names of unique launch site
  - Record where launch site begins with 'CCA'
  - total payload mass carried by boosters launched by NASA (CRS)
  - average payload mass carried by booster version F9 v1.1
  - date when the first successful landing outcome in ground pad was achieved
  - names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - total number of successful and failure mission outcomes
  - names of the booster_versions which have carried the maximum payload mass
  - failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order
- Github link

- All are using typical SQL skeleton structure

SELECT

FROM

WHERE

GROUP BY

ORDER BY

In some cases need sub-query

Build an Interactive Map with Folium

- First, we create a folium map object
- Then we put a circle object and marker object to make a mark of the launch site and show a circle around it
- Then we have a marker cluster to have multiple marker of successful/failed landing. This is useful when we have many marker in same geolocation
- Github link

Build a Dashboard with Plotly Dash

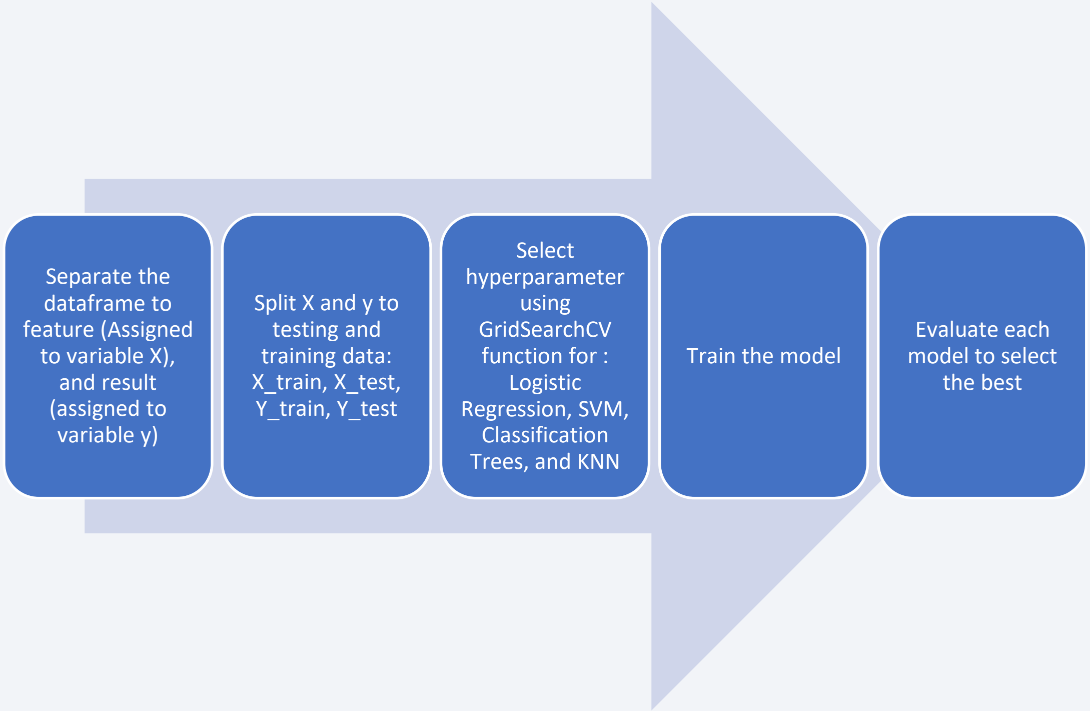
- There are 2 charts
  - Pie chart
    - Successful landing for each launch area
    - Success rate
- There are 2 filters:
  - Dropdown list of launch area
  - Slider of payload mass

- Github link

Predictive Analysis (Classification)

- Split the data to training and testing data then check best Hyperparameter for : Logistic Regression, SVM, Classification Trees, and KNN

- GitHub link



Separate the dataframe to feature (Assigned to variable X), and result (assigned to variable y)

Split X and y to testing and training data: X_train, X_test, Y_train, Y_test

Select hyperparameter using GridSearchCV function for : Logistic Regression, SVM, Classification Trees, and KNN

Train the model

Evaluate each model to select the best

Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

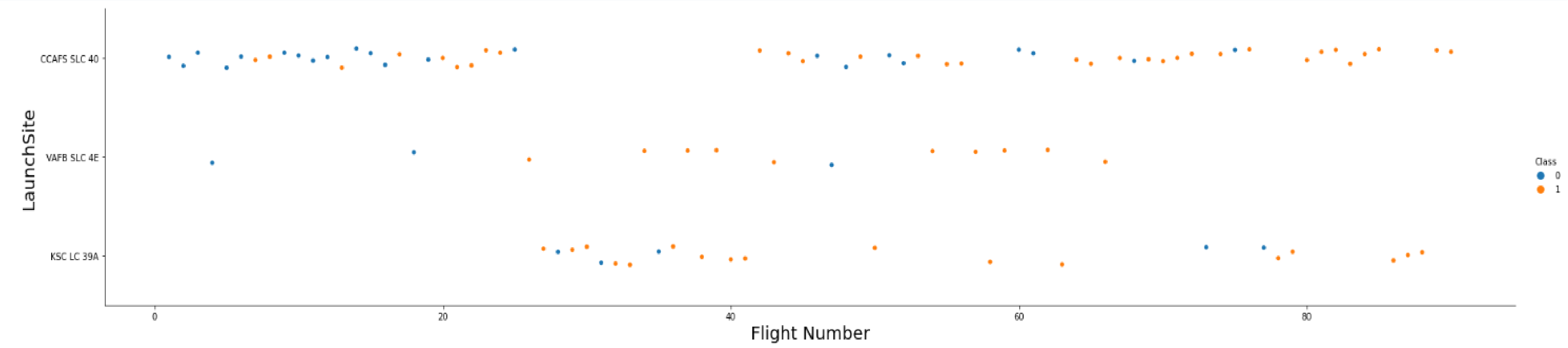
- Different success rate for each launch site. With VAFB SLC 4E having the highest rate of 77%

Success Rate

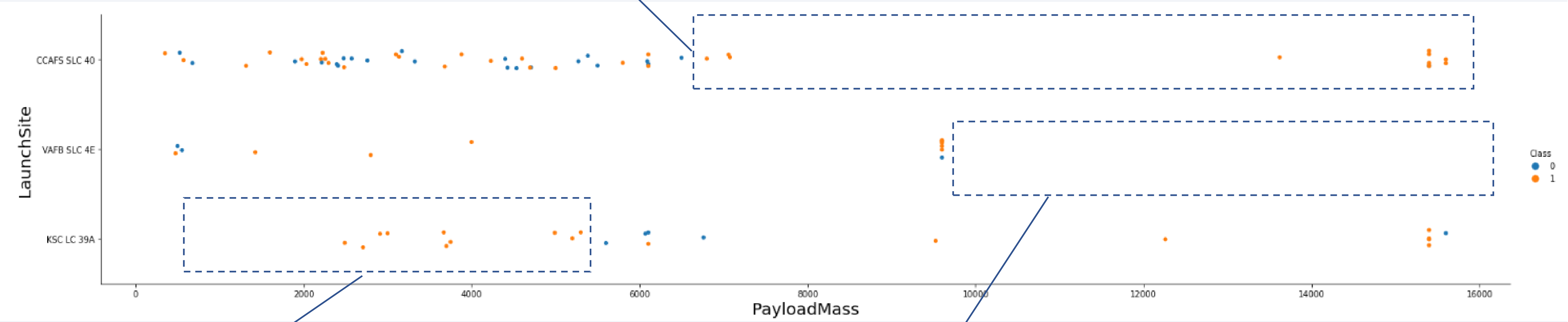
60%

77%

72%



Payload vs. Launch Site

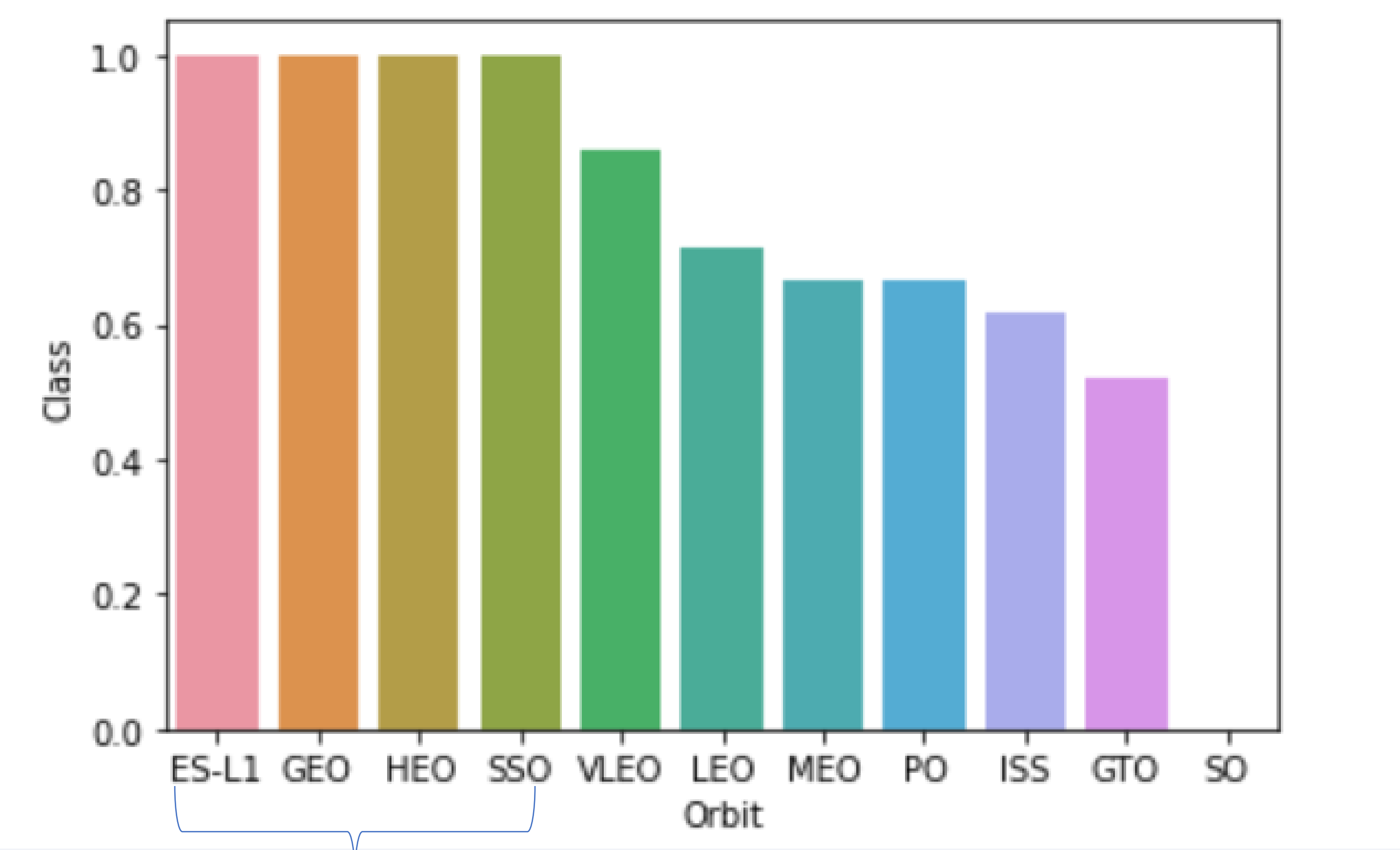


> 7000 payload resulting in high success rate

< 5500 payload resulting in high success rate

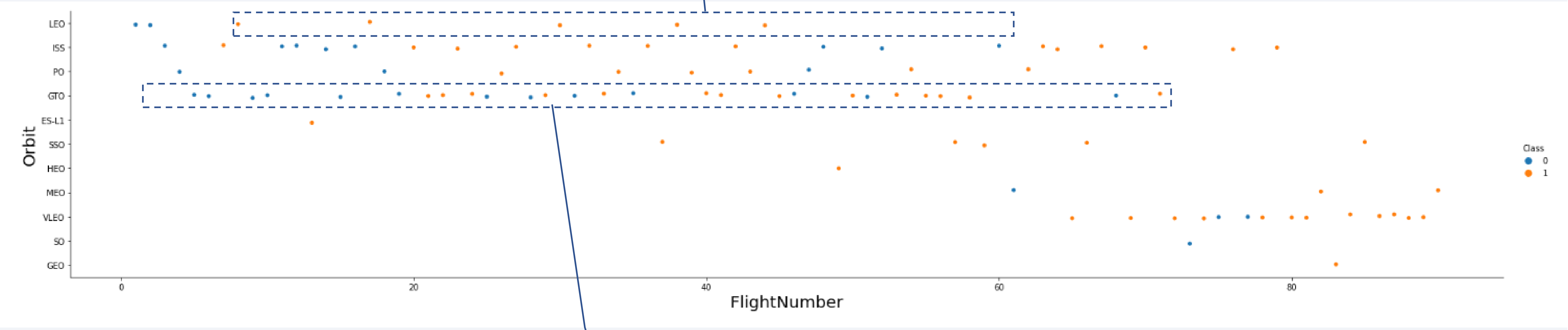
No rocket launch for payload mass > 10000 KG

Success Rate vs. Orbit Type



Highest success with 100% rate

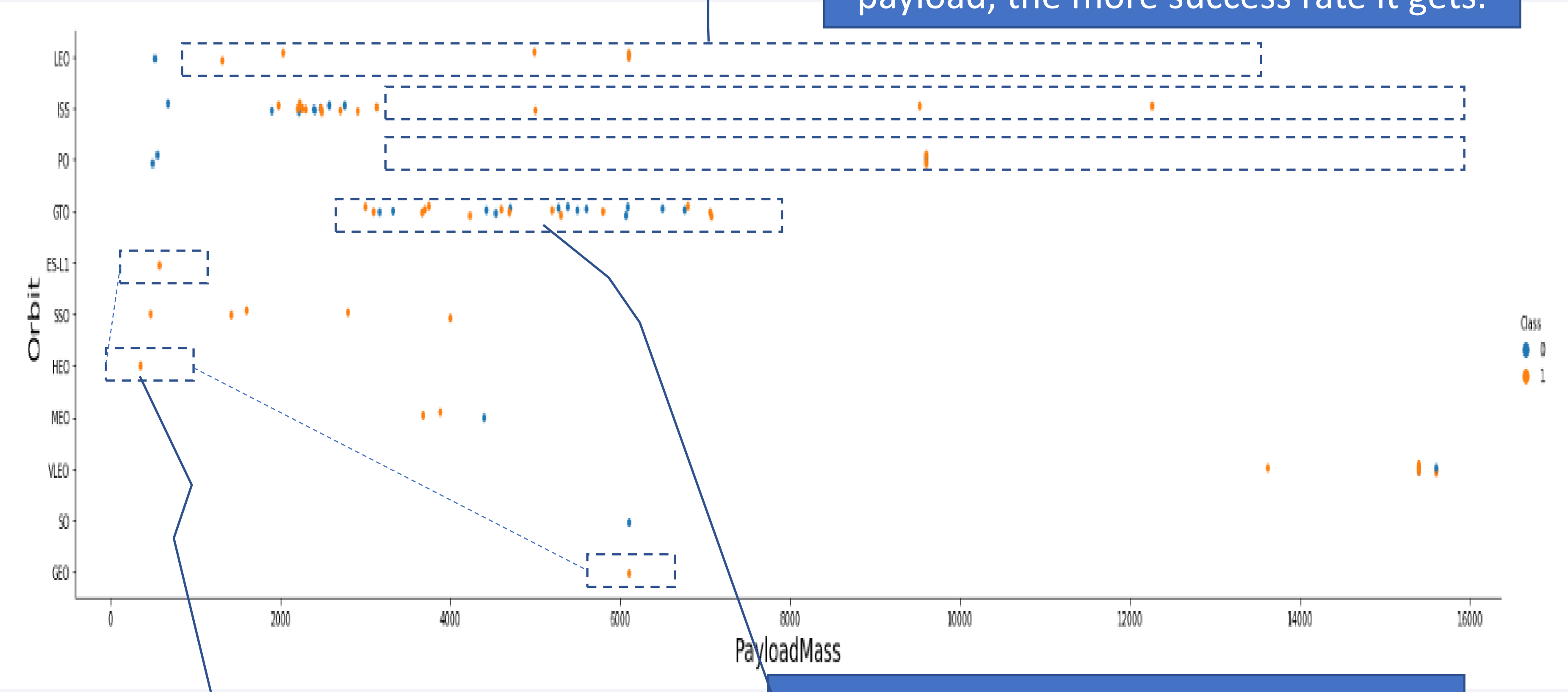
Flight Number vs. Orbit Type



More flight, more successful it gets

Could not be distinguished with flight number

Payload vs. Orbit Type

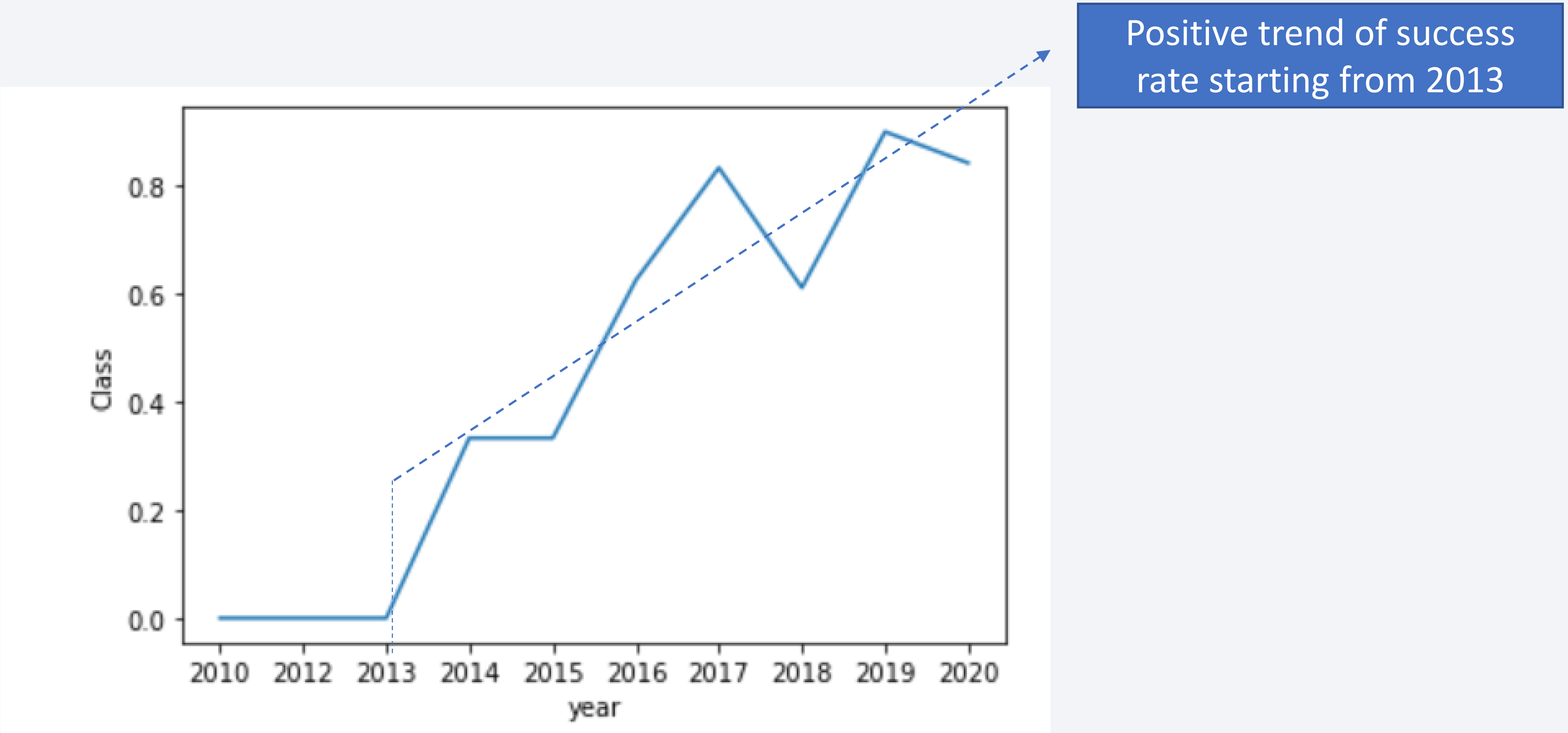


for Polar,LEO and ISS. The heavier the payload, the more success rate it gets.

GTO success rate could not be distinguished with payload mass

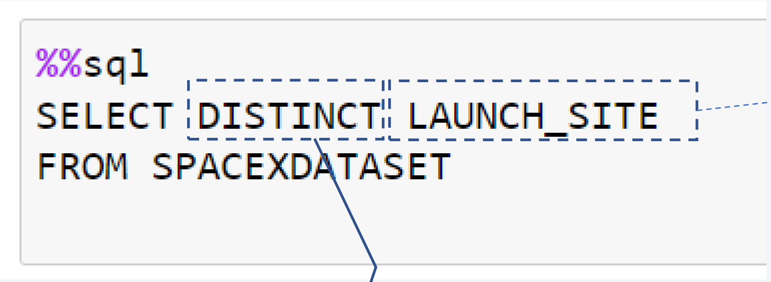
Even though ESL1, HEO, and GEO has 100% success rate, but they only have 1 data point

Launch Success Yearly Trend

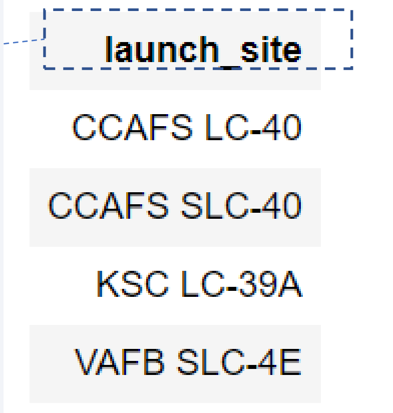


Positive trend of success rate starting from 2013

All Launch Site Names



```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXDATASET
```



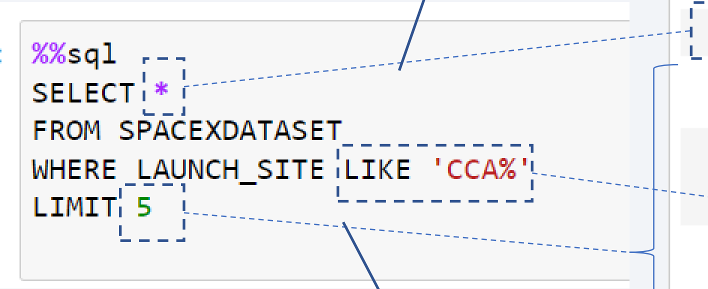
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

DISTINCT is to select only the unique result

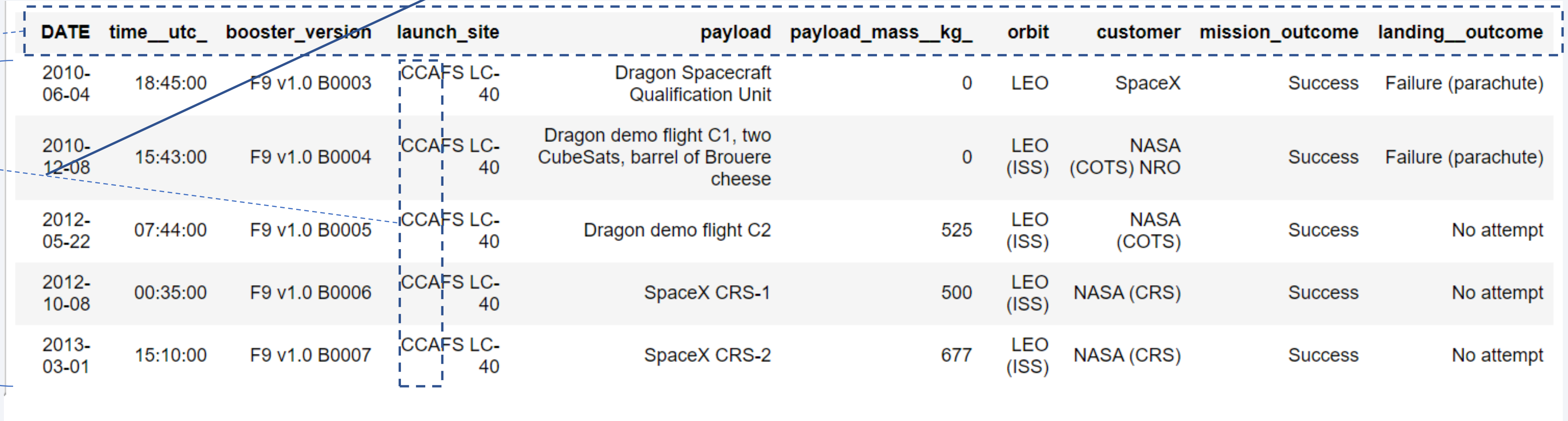
Launch Site Names Begin with 'CCA'

Select All column

With beginning “CCA”



```
%%sql
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

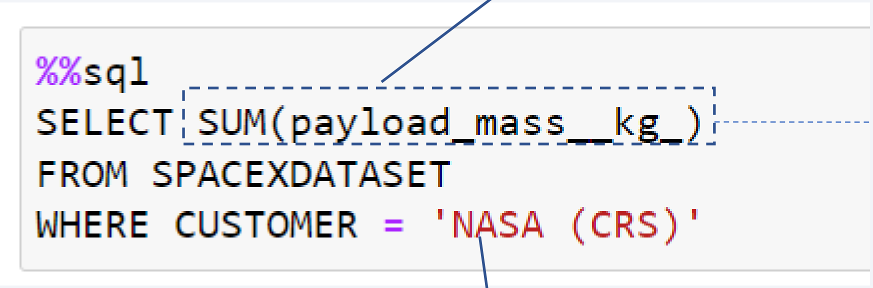


DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

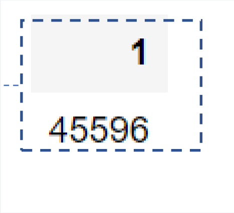
Limit to 5 row

Total Payload Mass

Selecting the TOTAL



```
%%sql
SELECT SUM(payload_mass__kg_)
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)'
```

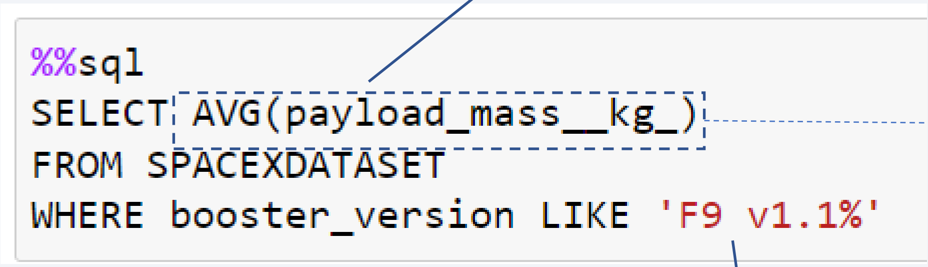


```
    1
45596
```

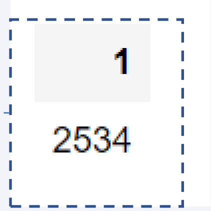
Filtering only for NASA (CRS)

Average Payload Mass by F9 v1.1

Selecting the AVERAGE



```
%%sql
SELECT AVG(payload_mass__kg_)
FROM SPACEXDATASET
WHERE booster_version LIKE 'F9 v1.1%'
```

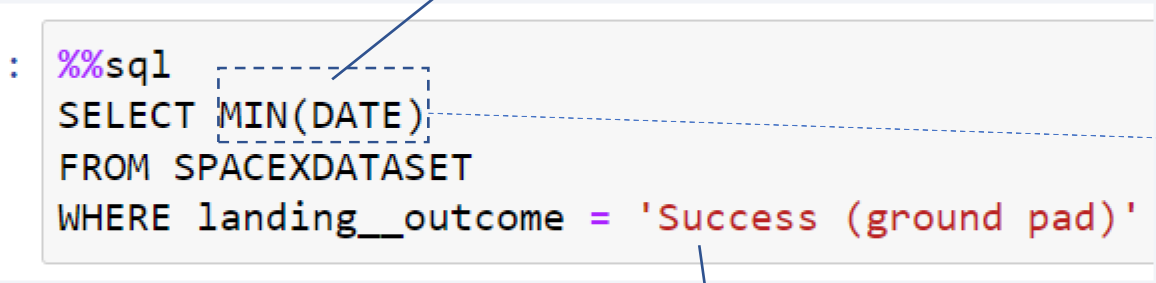


1
2534

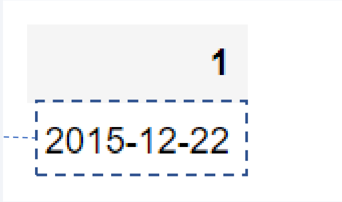
Filtering only for string beginning with F9 v1.1

First Successful Ground Landing Date

Selecting the MINIMUM



```
%%sql
SELECT MIN(DATE)
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)'
```

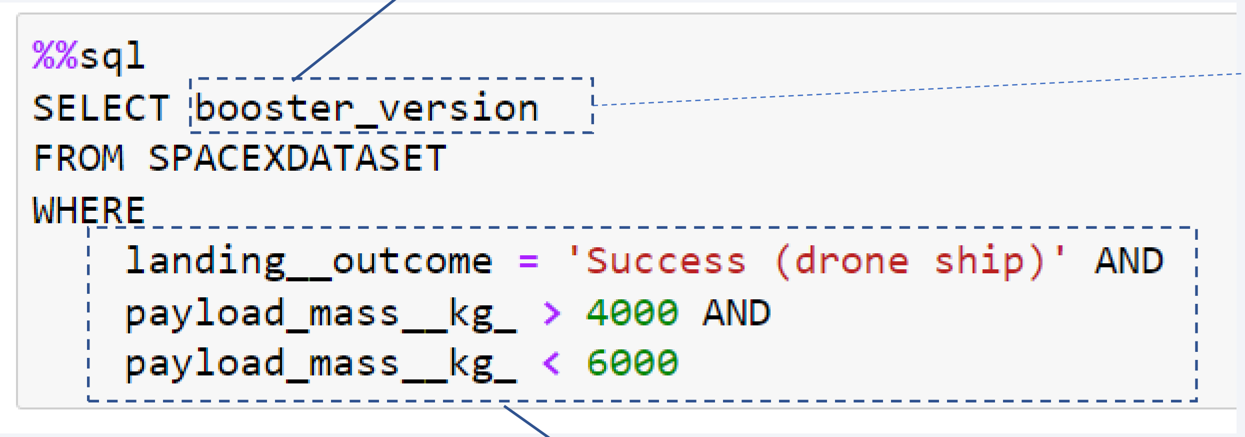


1
2015-12-22

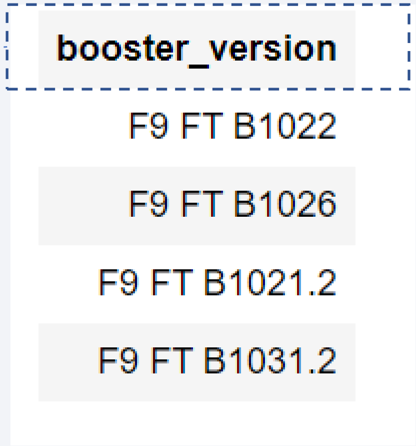
Filtering only for Successful Ground Landing

Successful Drone Ship Landing with Payload between 4000 and 6000

Selecting the booster_version column



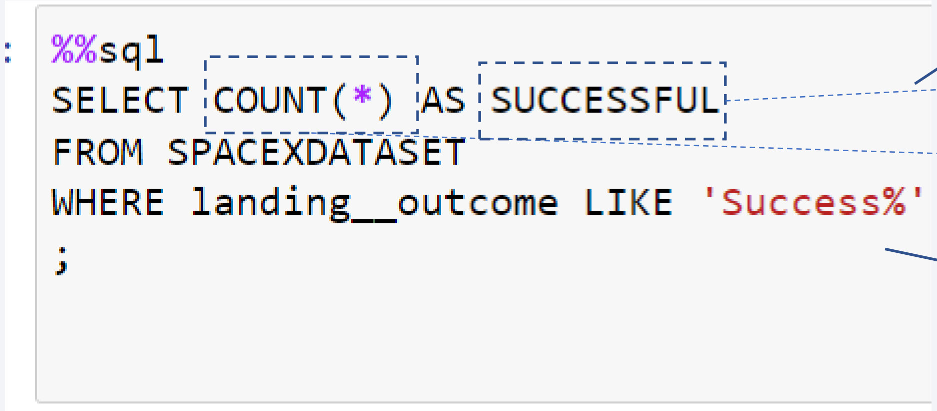
```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE
    landing__outcome = 'Success (drone ship)' AND
    payload_mass__kg_ > 4000 AND
    payload_mass__kg_ < 6000
```



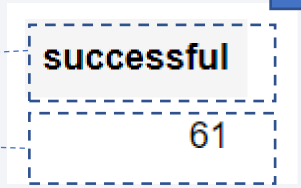
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Filtering the Successful Drone Ship Landing with Payload between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes



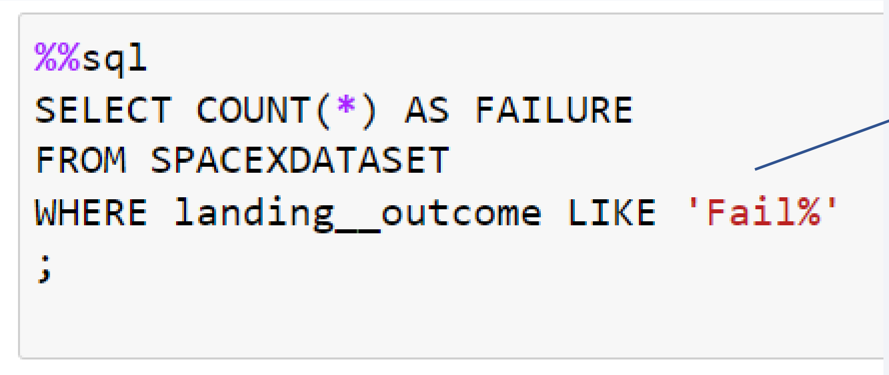
```
%%sql
SELECT COUNT(*) AS SUCCESSFUL
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Success%'
;
```



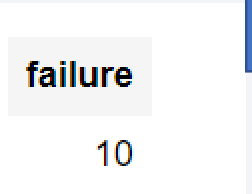
successful
61

COUNTING the event, and put it in “successful” column

Filter the success outcome



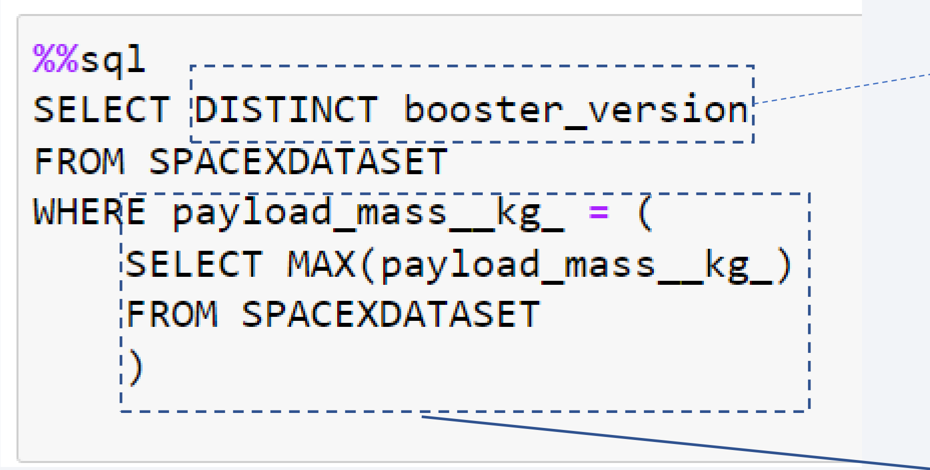
```
%%sql
SELECT COUNT(*) AS FAILURE
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Fail%'
;
```



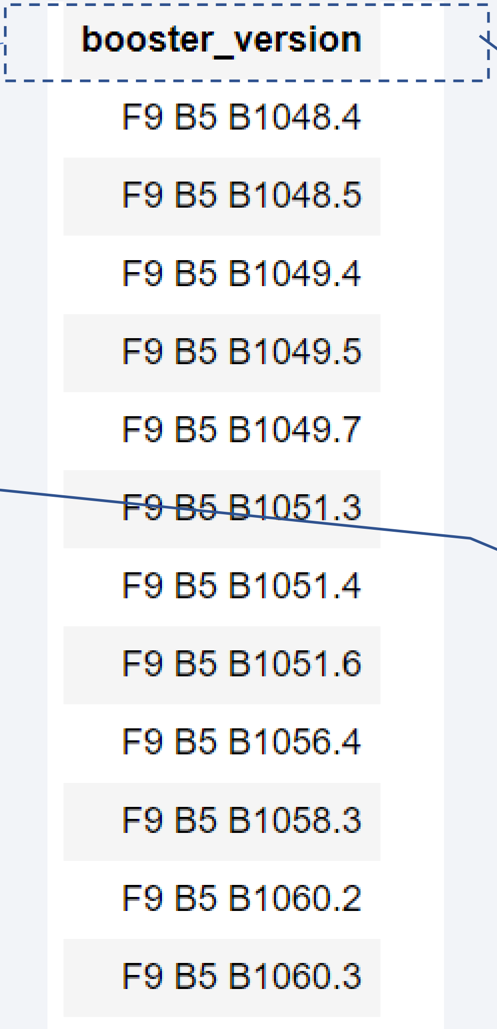
failure
10

Similar like above, however the filter is for fail

Boosters Carried Maximum Payload



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



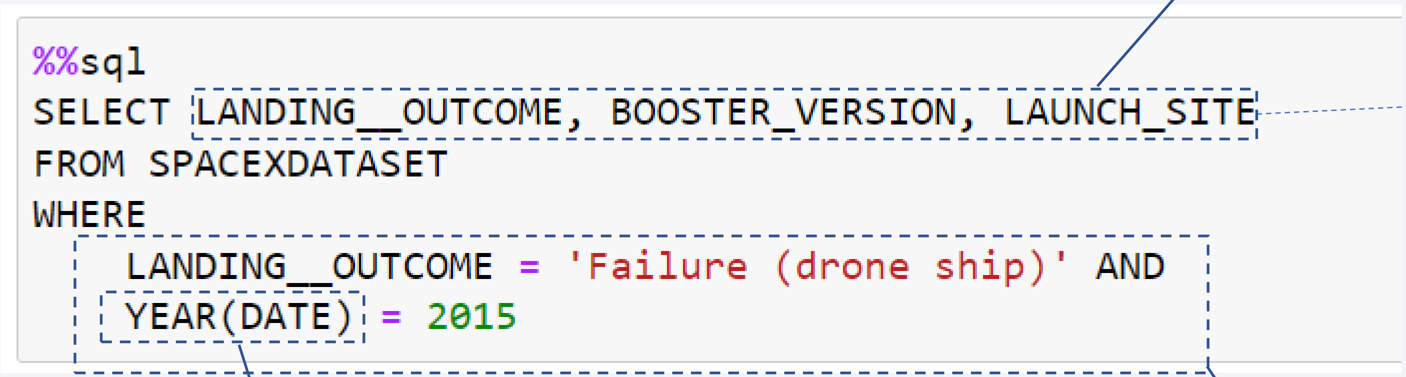
booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

Select the booster

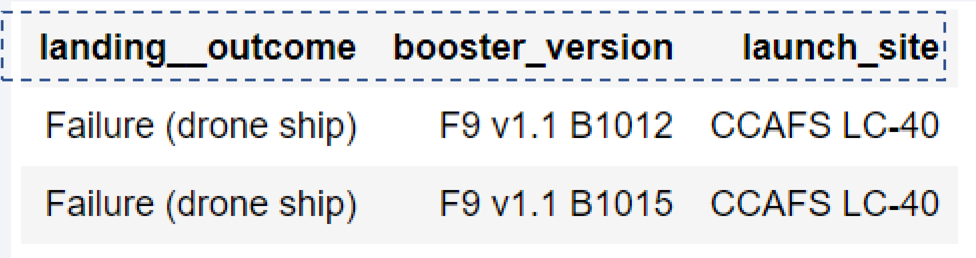
This subquery is selecting the maximum payload.
The main query is filtered with this value

2015 Launch Records

Selecting relevant columns



```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXDATASET
WHERE
   LANDING__OUTCOME = 'Failure (drone ship)' AND
   YEAR(DATE) = 2015
```



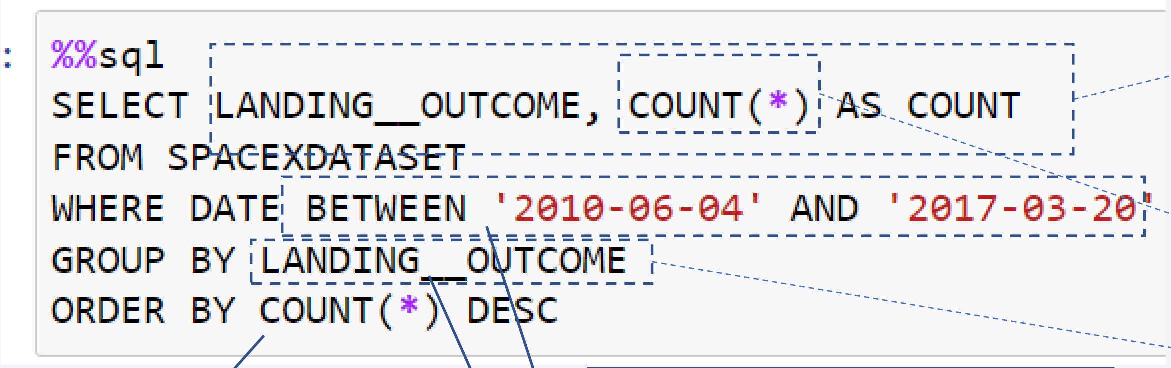
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

This is to get the YEAR out of DATE column

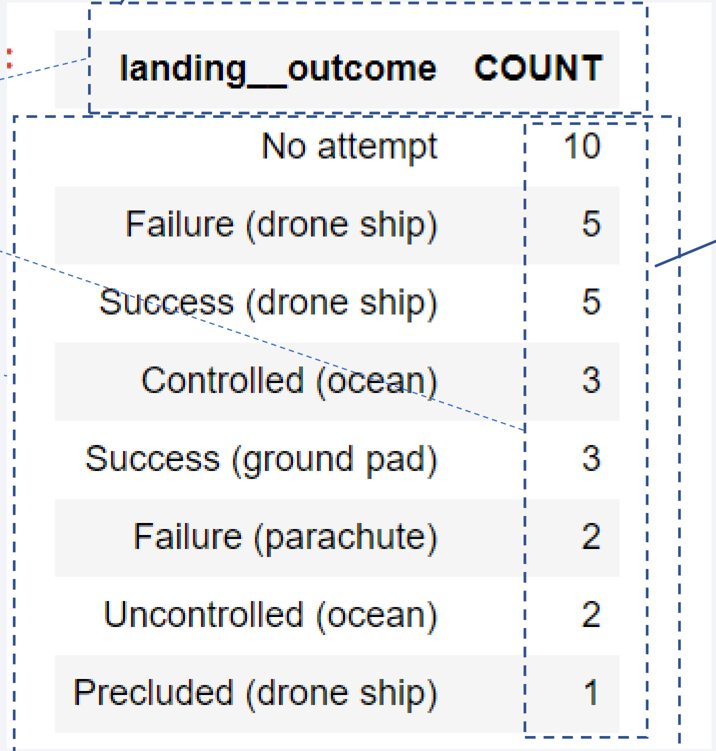
Filtering for failure (drone ship) and year 2015

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

Selecting right column



```
%%sql
SELECT LANDING__OUTCOME, COUNT(*) AS COUNT
FROM SPACEXDATASET
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY COUNT(*) DESC
```



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

COUNT the event of the outcome

Filtering the date range

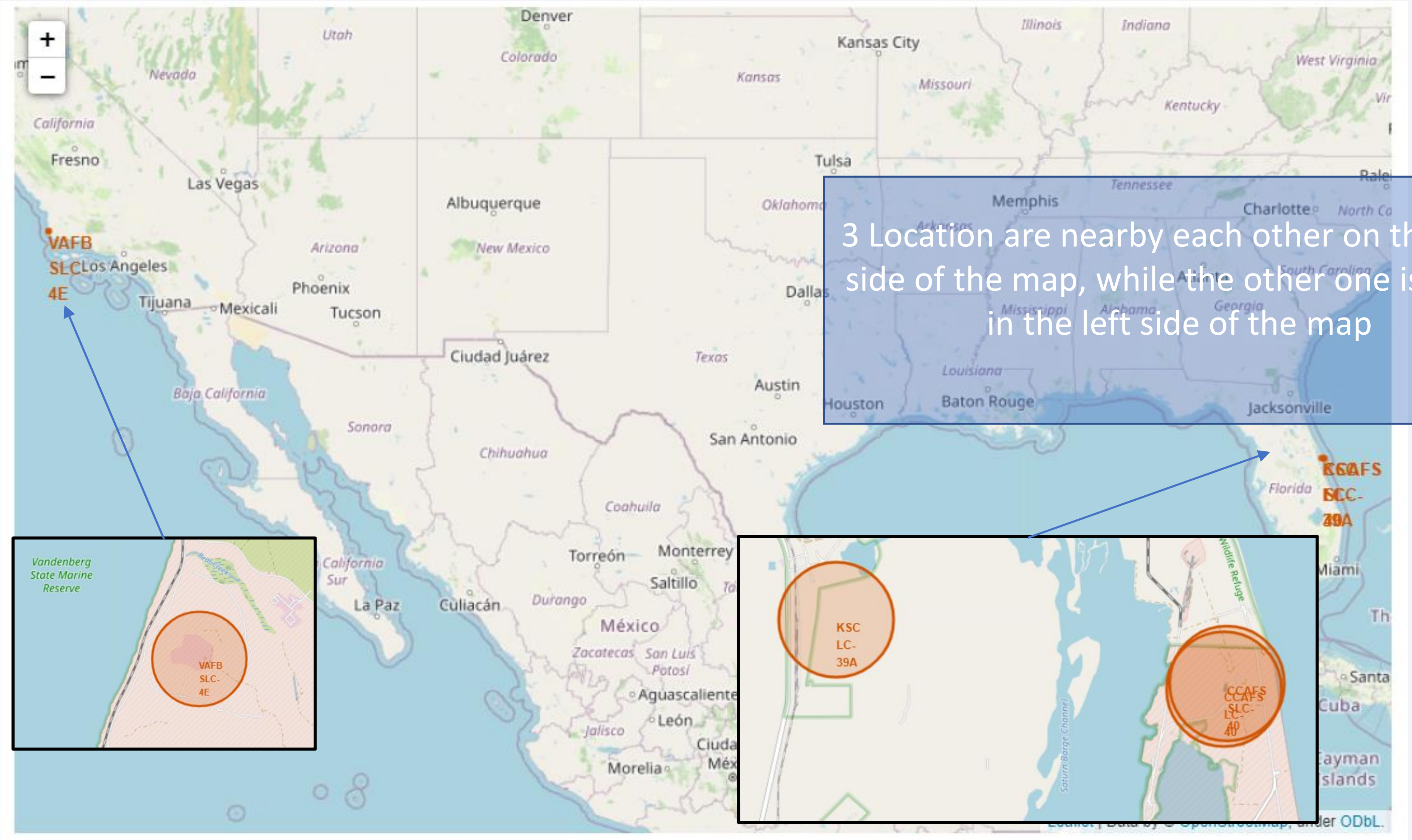
Grouping or aggregating

Sort by the COUNT in DESCENDING order

Section 3

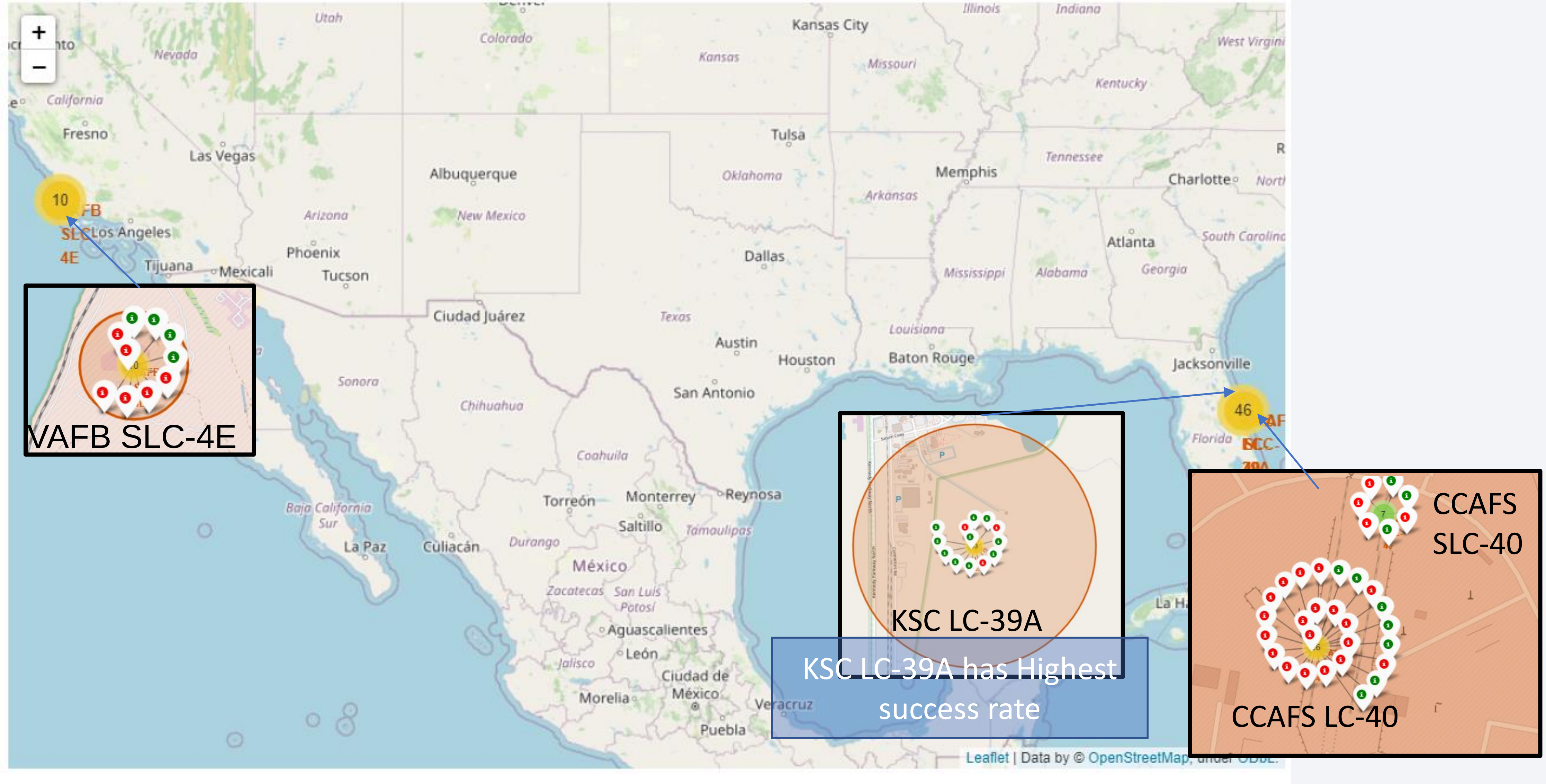
Launch Sites Proximities Analysis

All launch sites' location



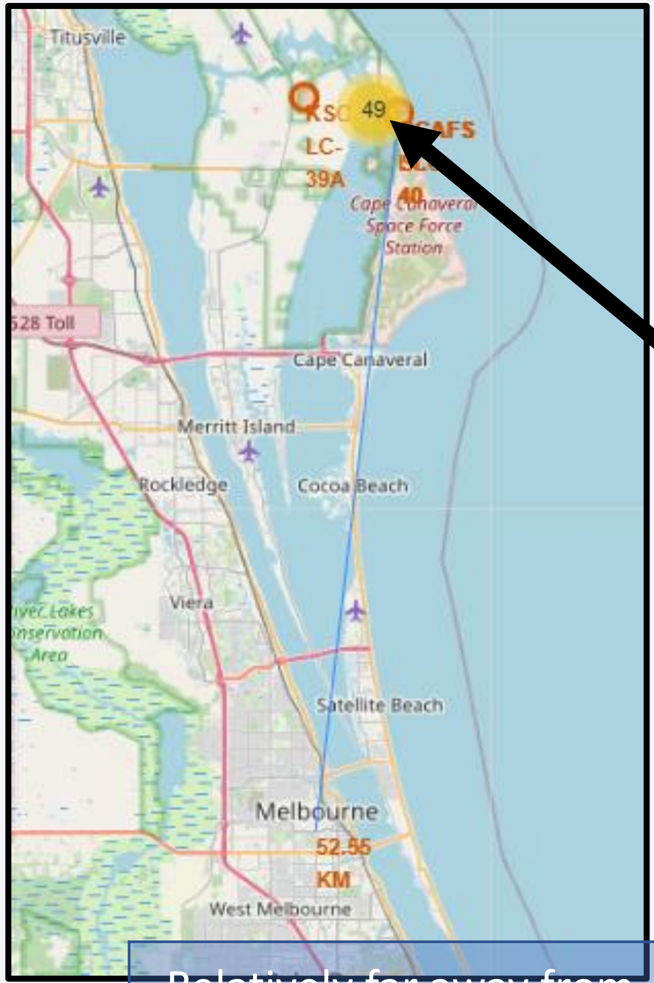
3 Location are nearby each other on the right side of the map, while the other one is alone in the left side of the map

Color-labeled launch outcomes

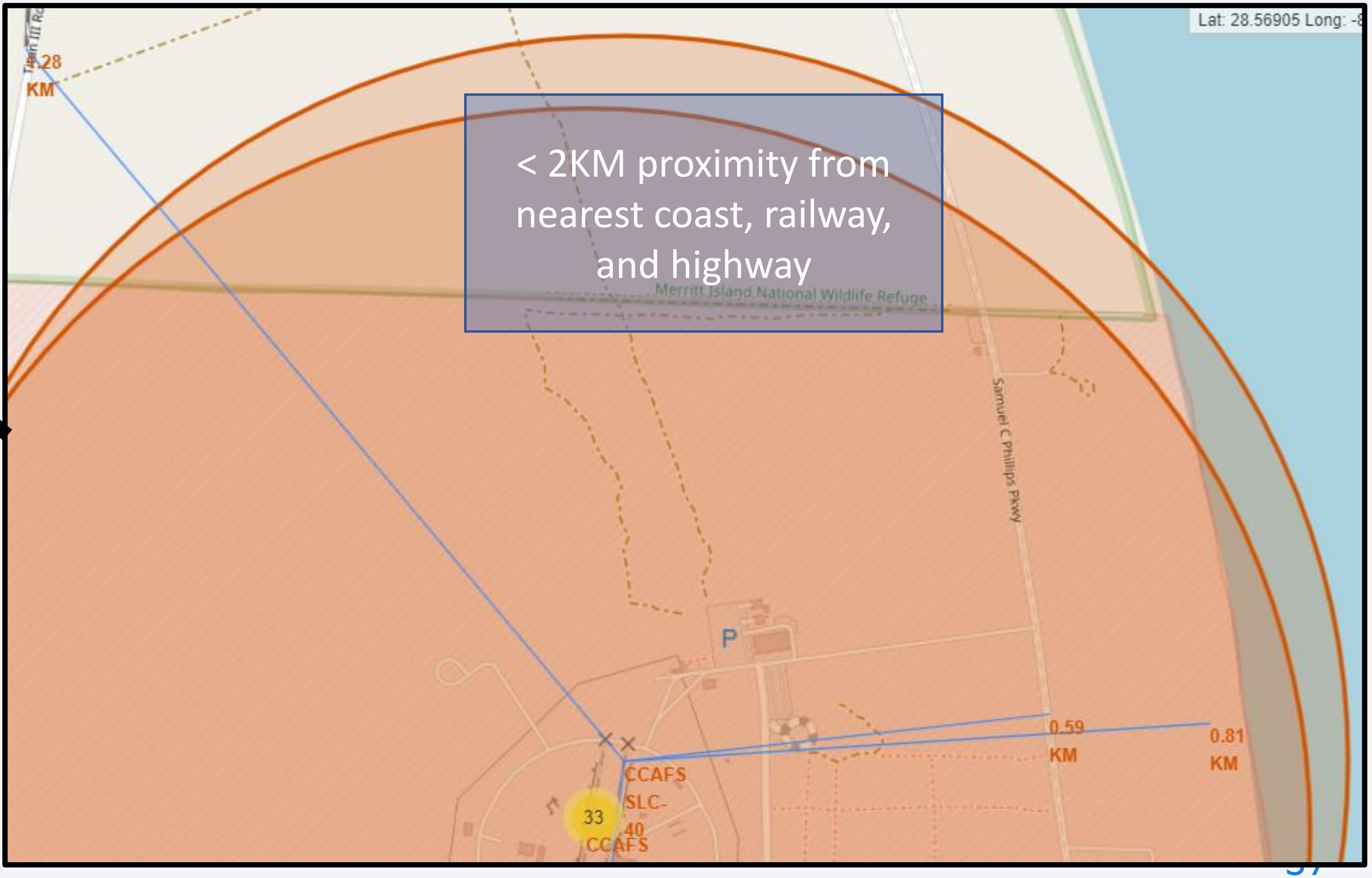


KSC LC-39A has Highest success rate

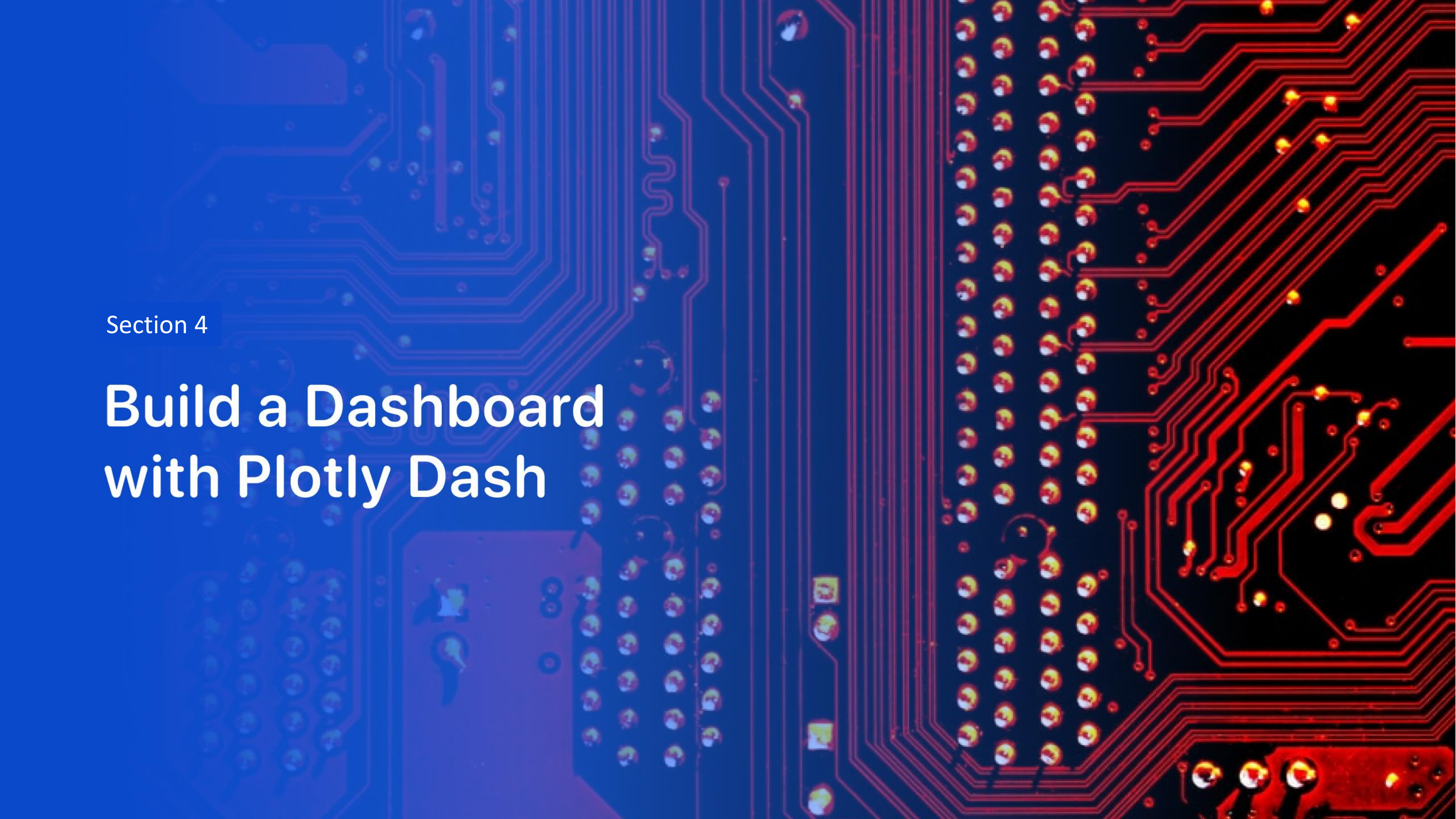
Launch site to its proximities



Relatively far away from nearest city



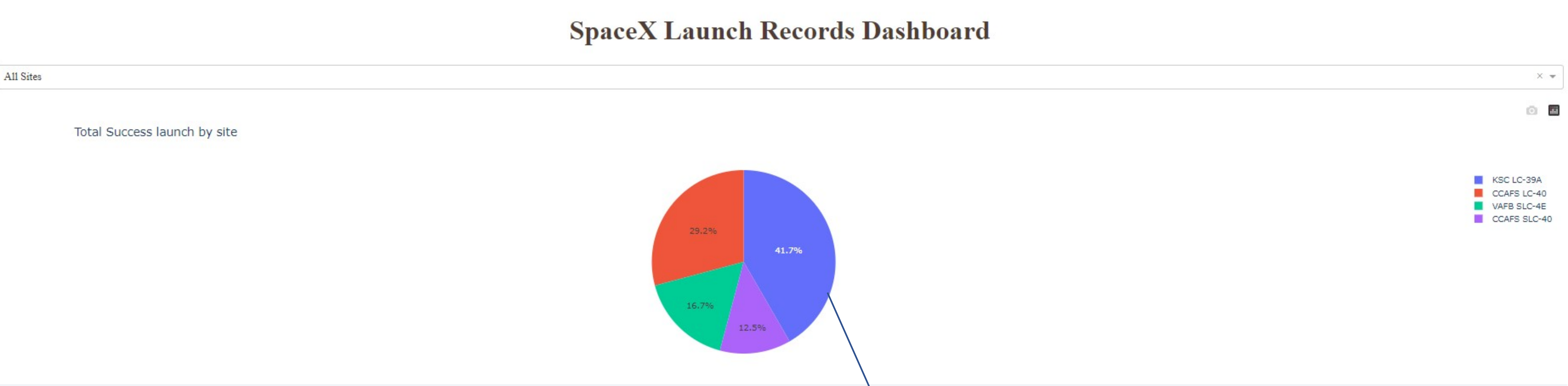
< 2KM proximity from nearest coast, railway, and highway



Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

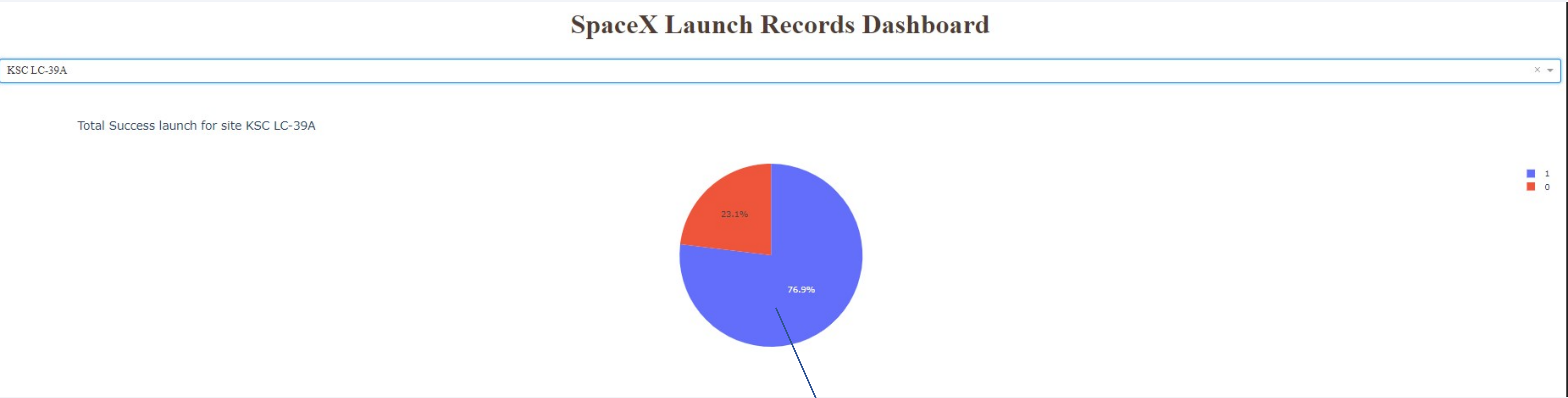


SpaceX Launch Records Dashboard

All Sites

KSC LC-39A is the highest with 42% success count vs total

Launch site with highest launch success ratio



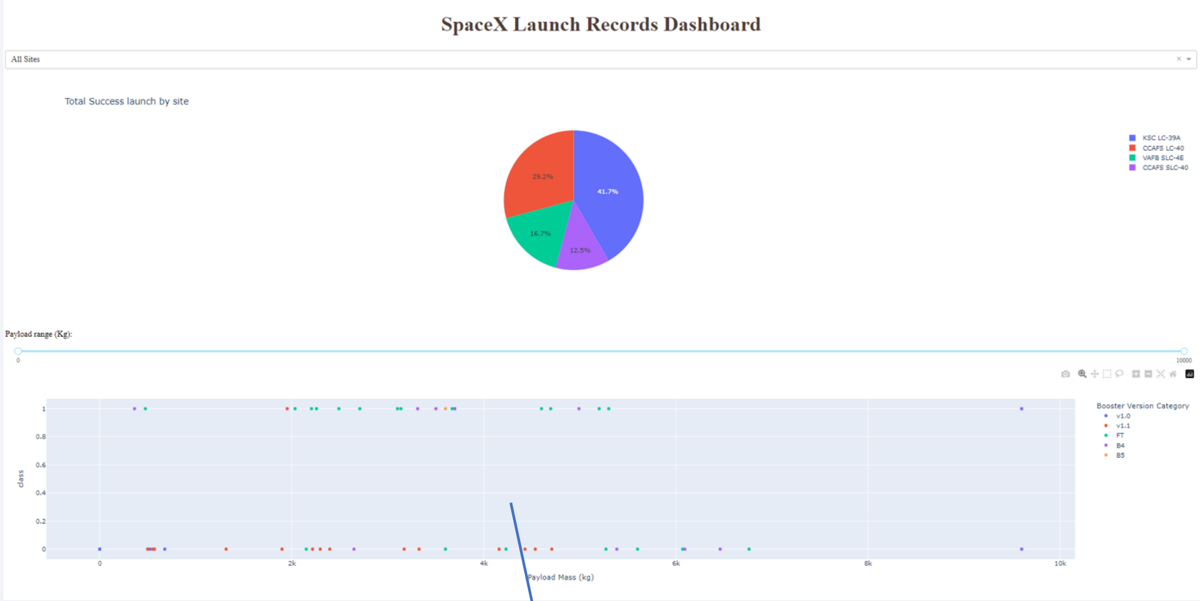
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success launch for site KSC LC-39A

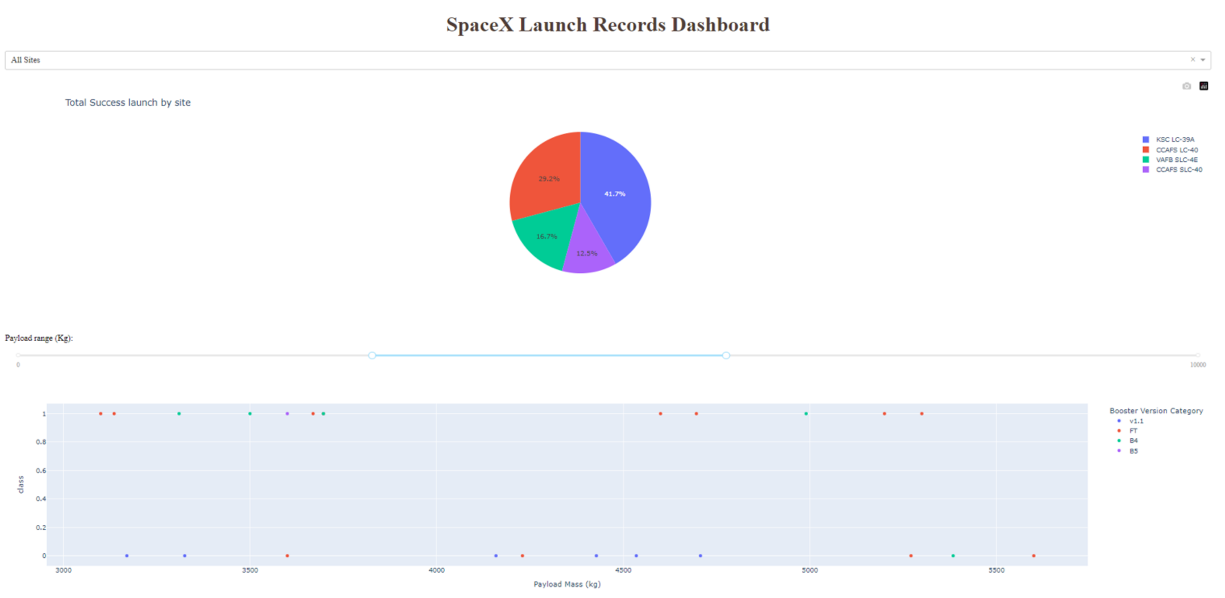
KSC LC-39A has the highest success ratio of 76.9%

<Dashboard Screenshot 3>



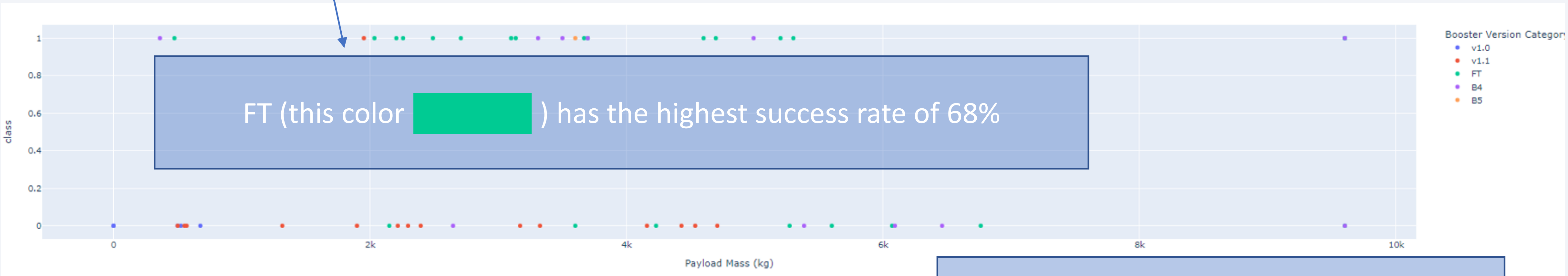
SpaceX Launch Records Dashboard

All Sites



SpaceX Launch Records Dashboard

All Sites



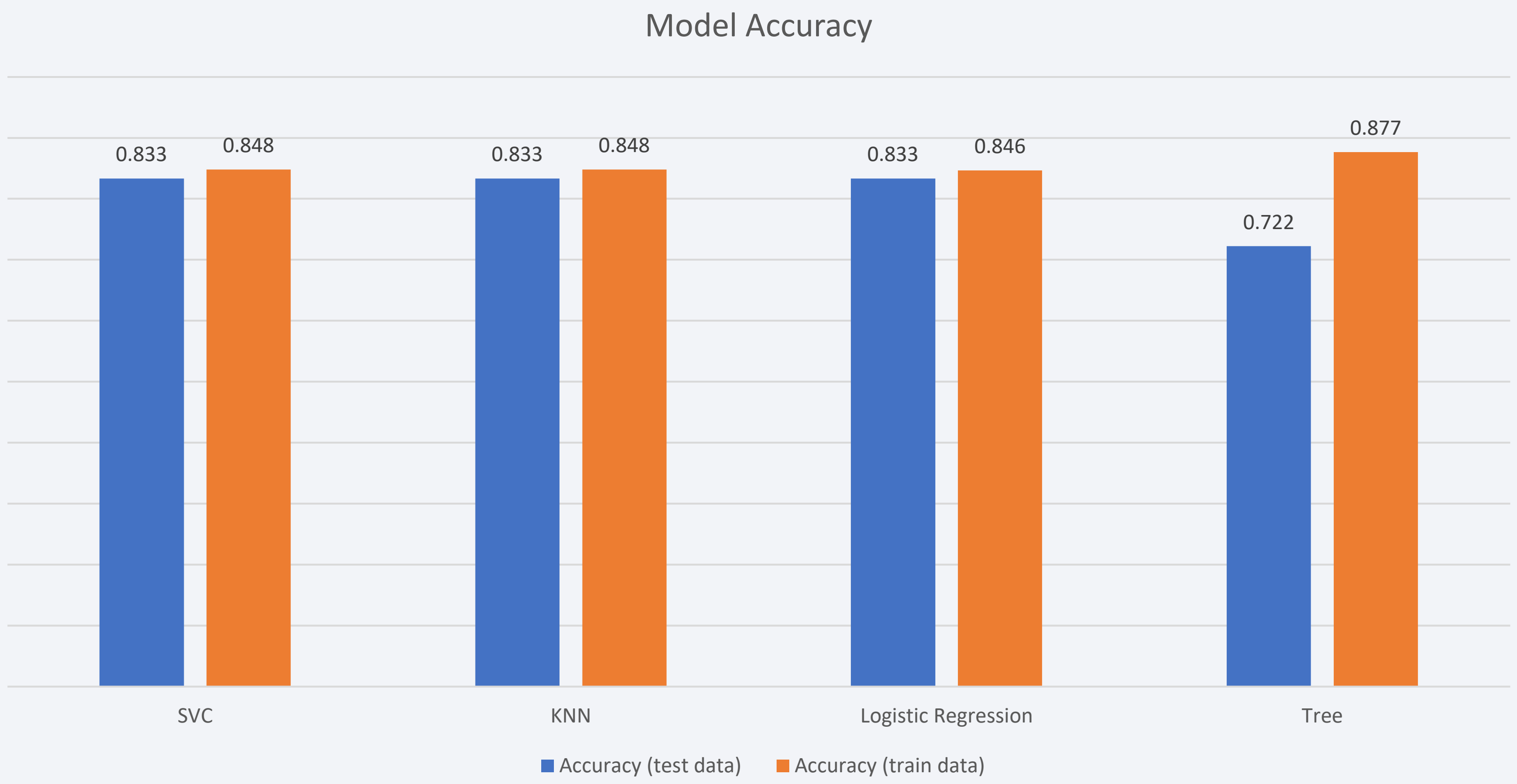
FT (this color 🟩) has the highest success rate of 68%

Payload mass > 6k has highest failure rate

Section 5

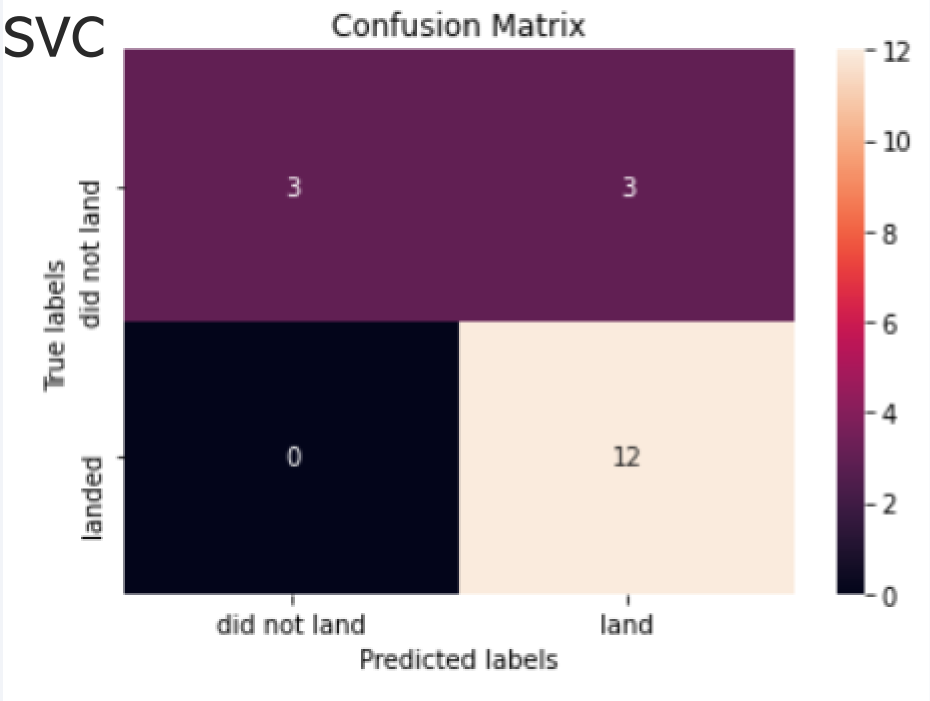
Predictive Analysis (Classification)

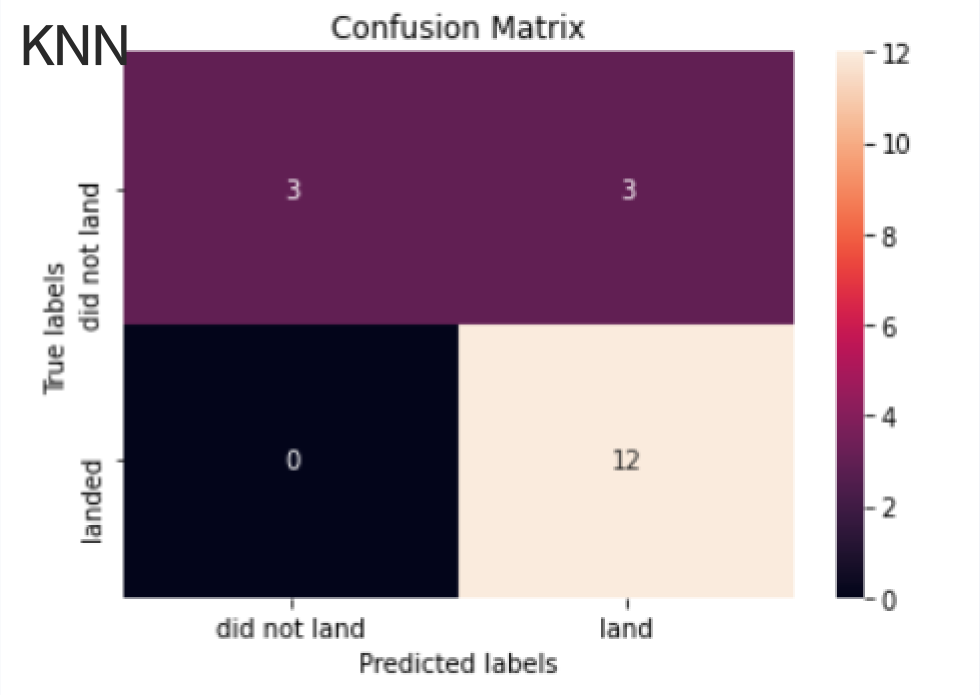
Classification Accuracy



SVC, and KNN has the highest overall accuracy

Confusion Matrix





Identical confusion matrix result for both SVC and KNN.

The model has a bit of problem with false positive (predicted land, but not correct).

When the model predict “land”, it has 80% probability that it will actually land.

Conclusions

- We can find the best predictive model by evaluating several model. KNN or SVC is recommended
- Exploratory data analysis is successfully done by both visualization and SQL
- Pandas dataframe is very useful to collect and transform table-like structure. It's powerful since it has function to convert JSON or HTML file to dataframe. It is also compatible with major visualization library.

Appendix

- Link to dataset
- Relevant python snippets can be found here

Thank you!