IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Keyur Makwana
6th July 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Prediction through different Machine Learning Techniques
- Summary of all results
  - Exploratory Data Analysis Result
  - Interactive analytics in Screenshots
  - Predictive Analytics Result

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems to find answers
  - Will the rocket land successfully in the first stage?
  - Can the rocket be reused after the first stage?
  - Which conditions satisfy the successful launch?

Section 1

Methodology

Methodology

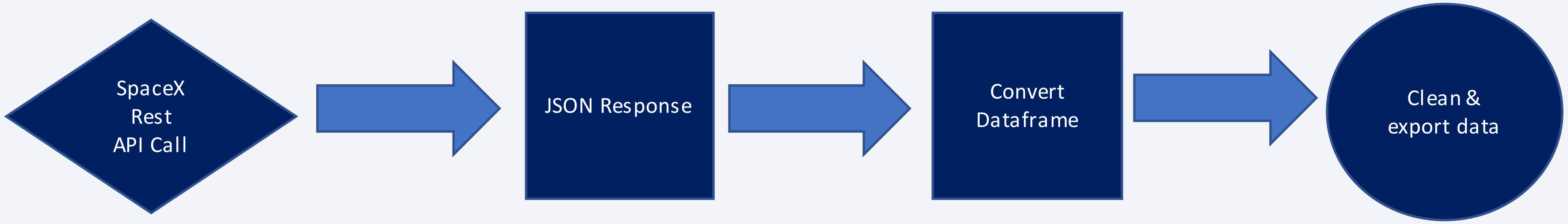
Executive Summary

- Data collection methodology:
  - Python request library is used to collect the data from SpaceX wikipedia website
- Perform data wrangling
  - Data was preprocessed & transformed into standard scaler for machine learning prediction algorithms
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Various graphs like scatter plot & bar graphs along with SQL EDA helped to show the nature of data
- Perform interactive visual analytics using Folium and Plotly Dash
  - Interactive Plotly dashboard relates launch site with success ratio using pie chart & scatter plot
- Perform predictive analysis using classification models
  - Built various algorithms & fine tuned them by finding best hyperparameter

Data Collection

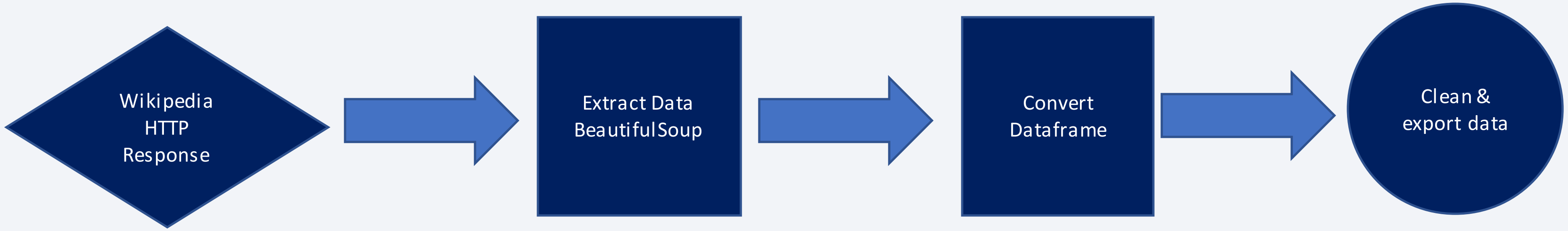
Data sets were collected using SpaceX Rest APIs

- Rockets (https://api.spacexdata.com/v4/rockets), Payloads (https://api.spacexdata.com/v4/payloads)
- Launchpad (https://api.spacexdata.com/v4/launchpads)



Falcon 9 heavy launches data collected by wikipedia site using webscrapping

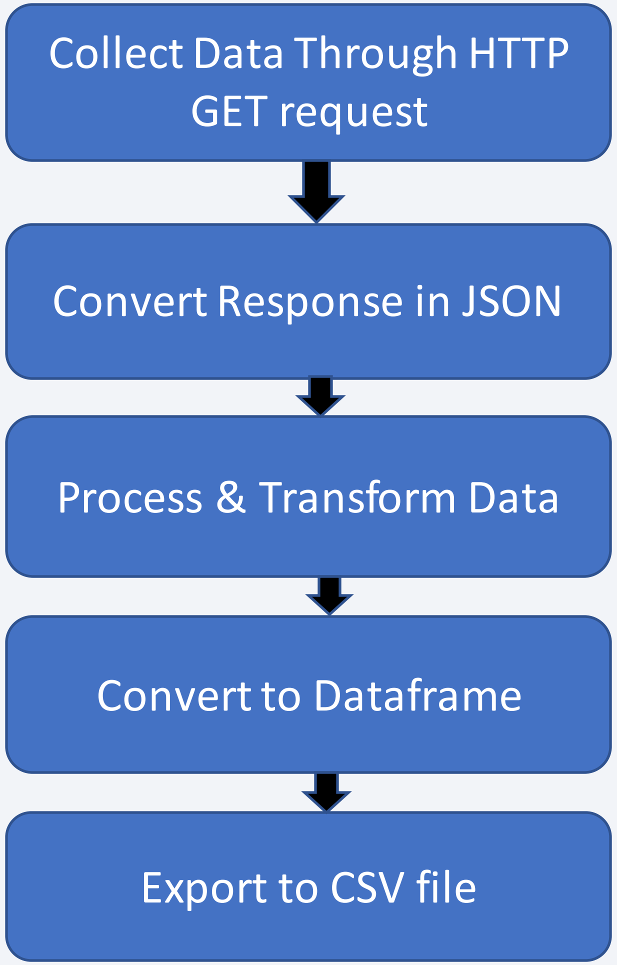
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

- We used public API of SpaceX. We have obtained data by HTTP get method. We performed data wrangling & formatting to clean & export data

- Code Link : SpaceX API



Collect Data Through HTTP GET request

Convert Response in JSON

Process & Transform Data

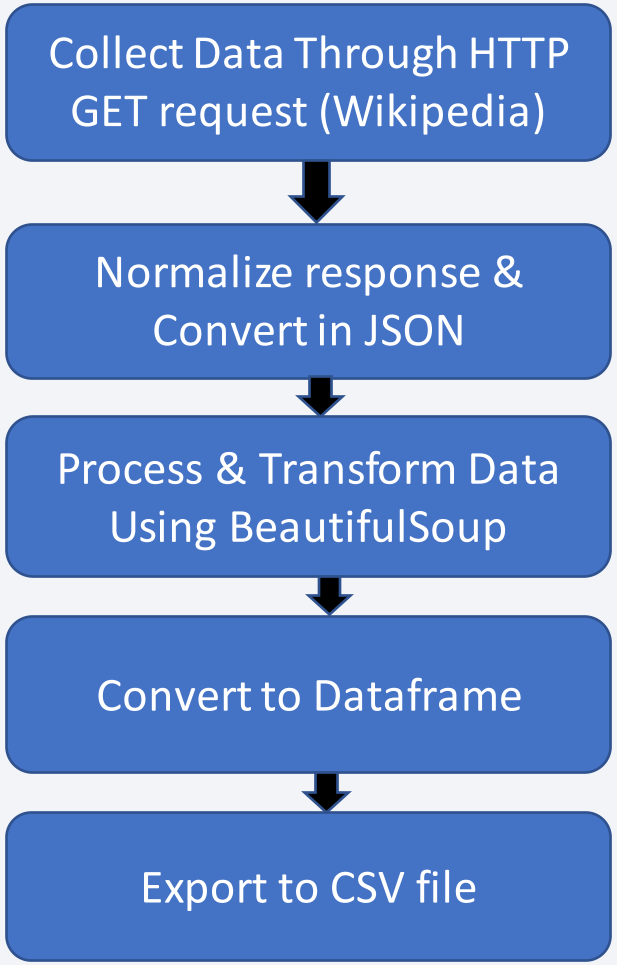
Convert to Dataframe

Export to CSV file

Data Collection - Scraping

- We used wikipedia website of SpaceX. We have obtained data, processed & transformed using BeautifulSoup library. We parsed through tables-columns to get data & converted into dataframe & export data

- Code Link : SpaceX WebScrapping



Collect Data Through HTTP GET request (Wikipedia)

Normalize response & Convert in JSON

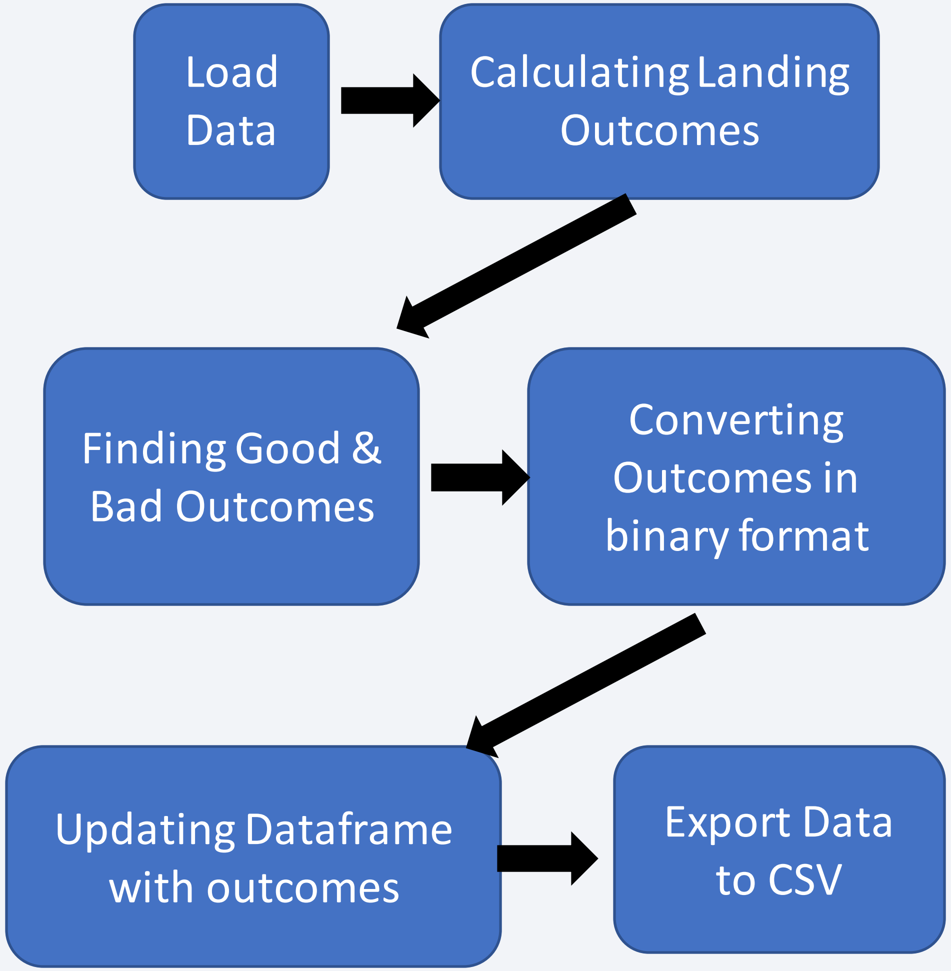
Process & Transform Data Using BeautifulSoup

Convert to Dataframe

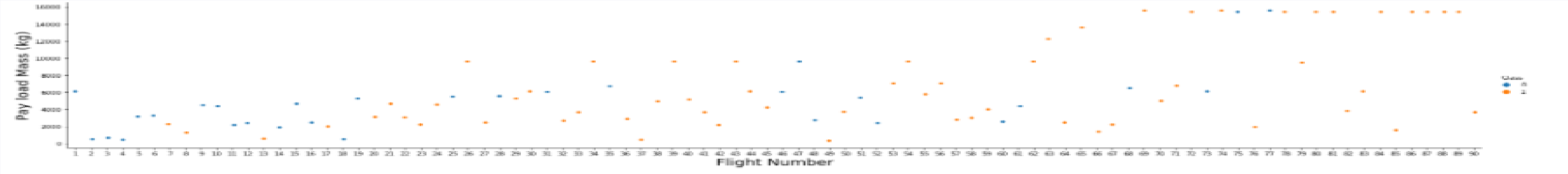
Export to CSV file

Data Wrangling

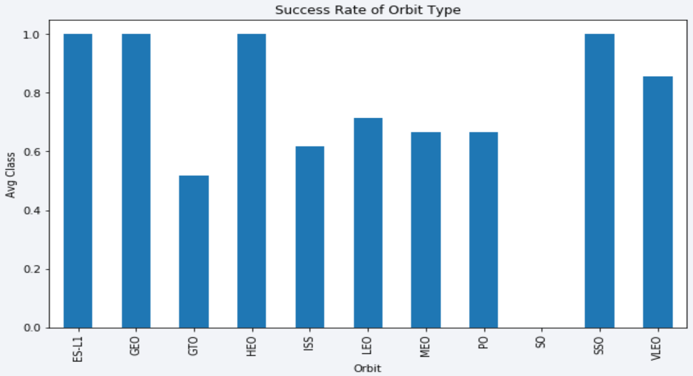
- EDA helped to find some patterns in the data and determine what would be the label for training supervised models
- In the dataset through EDA we can summarize how many launches happened from different launch sites, in which orbit they launched & the success ratio for each launch & launch sites
- Code Link : Data Wrangling


Load Data
Calculating Landing Outcomes
Finding Good & Bad Outcomes
Converting Outcomes in binary format
Updating Dataframe with outcomes
Export Data to CSV

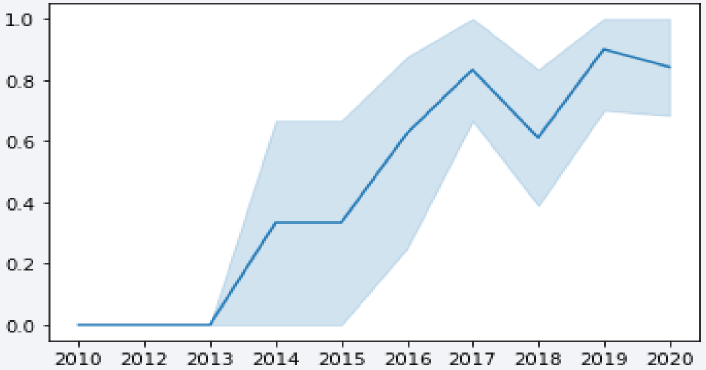
EDA with Data Visualization



Scatter plot chart to show successful launches by comparing flights & payload mass



Bar chart shows success rate per orbit type



Success rates of the launches over the years

Code Link : EDA Visualization

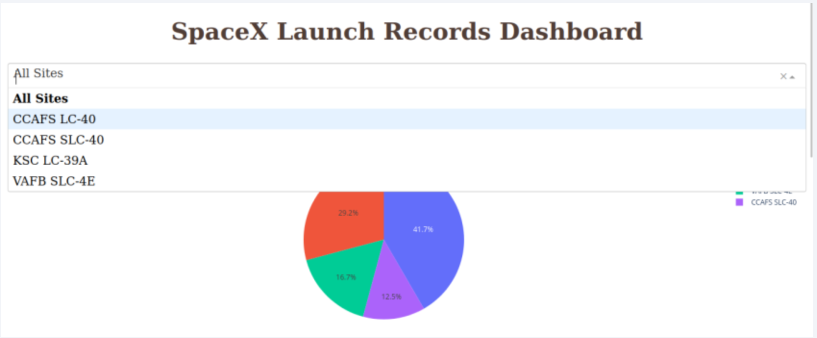
EDA with SQL

- EDA with SQL helps Data Scientists to analyze the data & give insights of overall dataset. It helps Data Scientists to see through the data & help in making decision on which data to manipulate & in what way. We derived below insights by performing EDA through SQL
    - Display the names of the unique launch sites in the space mission
    - Display of the booster versions which have carried the maximum payload mass.
    - Display of the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
    - Ranked 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
- Code Link : EDA SQL

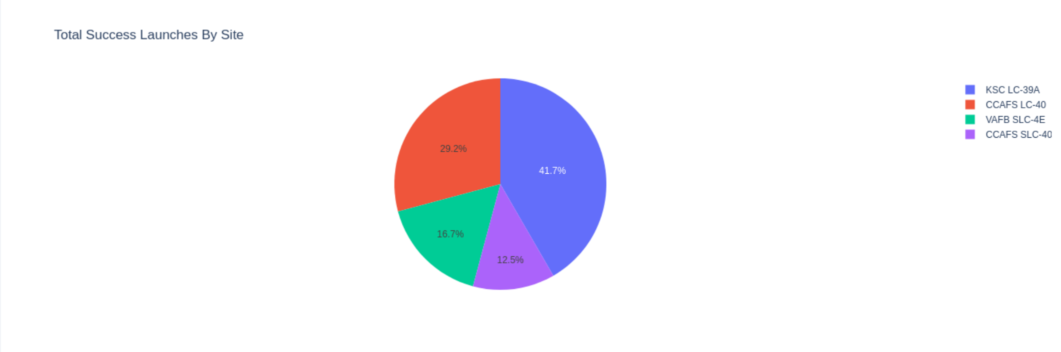
Build an Interactive Map with Folium

- Map objects & reason to add:
  - Folium.map : Creates a custom pane to hold map elements which is centered to country USA where all spaceX launches happened
  - Folium.map.marker : Create a simple stock Leaflet marker on the map, with optional popup text or Vincent visualization. It helped to mark different launch sites on the map
  - Folium.circle : Class for drawing circle overlays on a map. It helped to ma the areas of different launch sites.
  - Folium.polyline : Draw polygon overlays on a map. It helped to measure distance from launch site to nearest city, railway line, coastlines.
- Code Link : Site Locations

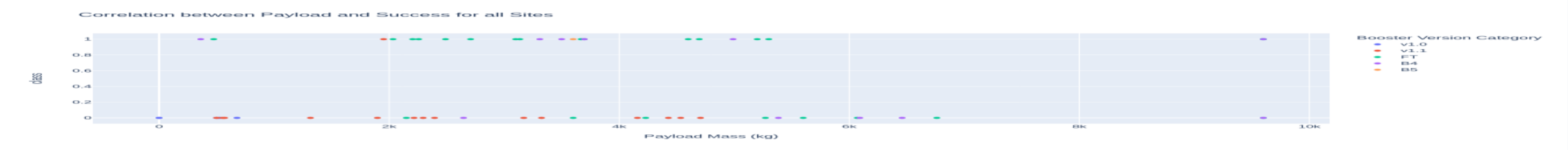
Build a Dashboard with Plotly Dash



Display all launch sites in dropdown to analyze them individually or collectively



Successful launches from all launch sites on pie chart



Scatter plot shows correlation between payload mass and success for all sites

Code Link : SpaceX Dashboard

Predictive Analysis (Classification)

- We first loaded the data from CSV file for both training & testing purpose
- We transformed the data & splitted using StandardScaler() of sklearn's preprocessing library
- We have fitted the training data to 4 different machine learning algorithms which were Logistic Regression, SVM, Decision Tree & K-nearest neighbor
- We have found best hyperparameter for each algorithm using GridsearchCV which is cross validation method
- We have evaluated model by different test score methods for each model
- Code Link : Predictive Analysis

Results

Exploratory data analysis results

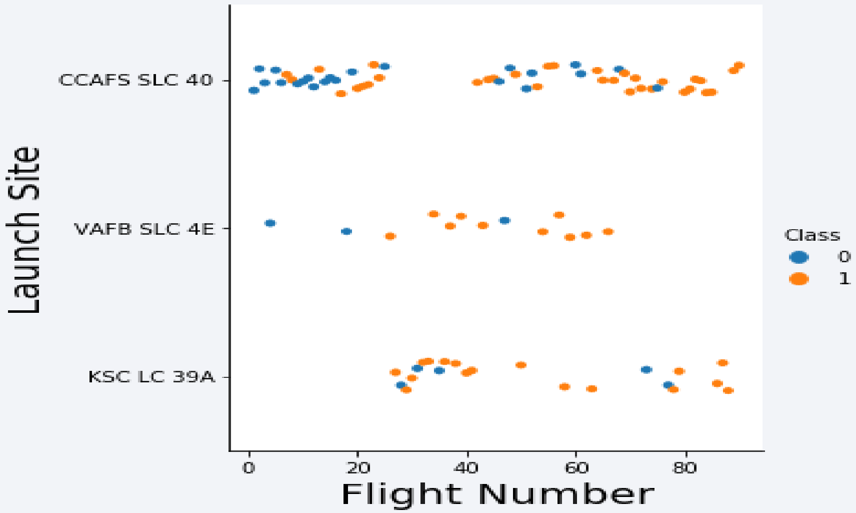
Interactive analytics demo in screenshots

Predictive analysis results

Section 2

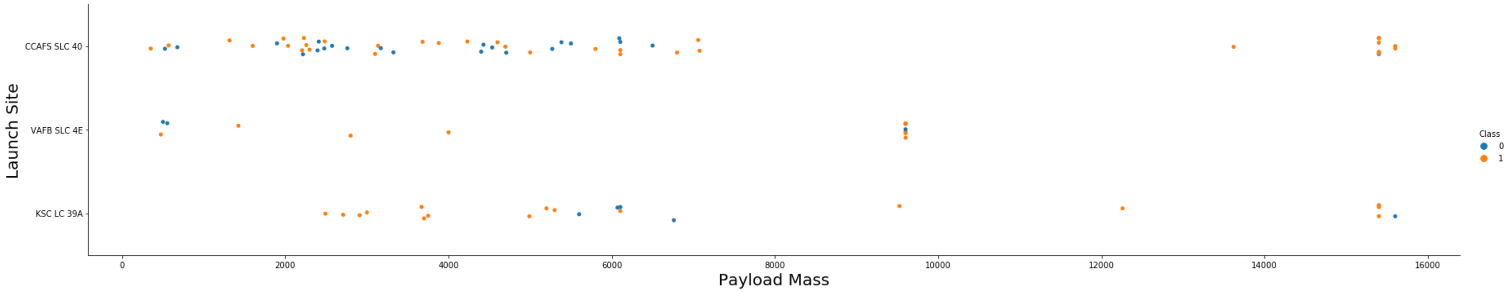
Insights drawn from EDA

Flight Number vs. Launch Site



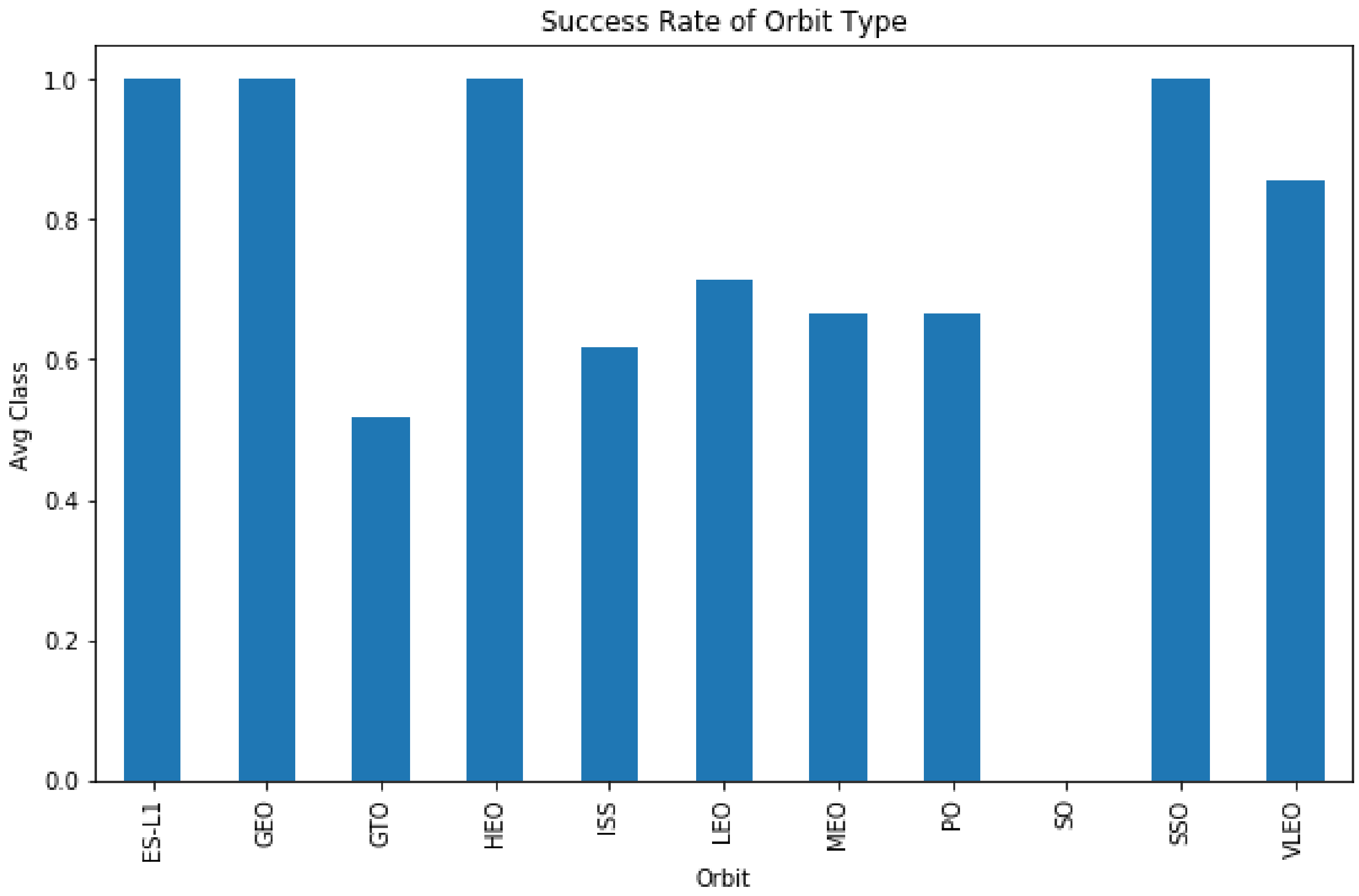
- This scatter plot shows success ratio with the increase of flight numbers at a specific launch site

Payload vs. Launch Site



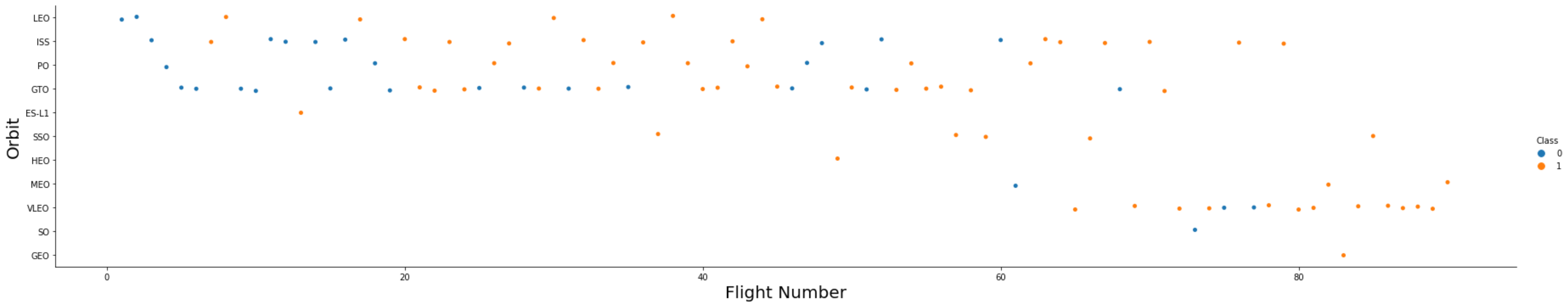
- This scatter plot shows us with increase of payload weight the success rate increases for locations KSC LC 39A and VAFB SLC 4E.
- We can also see almost 100% success rate at site CCAF5 SLC 40 for payload more than 7000.

Success Rate vs. Orbit Type



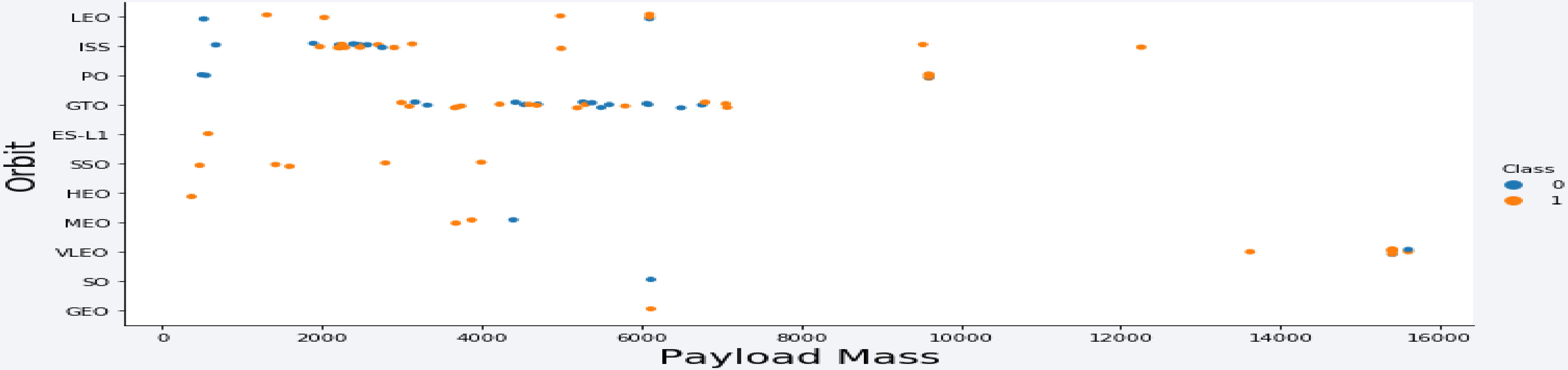
- This bar chart shows orbit types ES-L1, GEO, HEO, SSO & VLEO have higher success rate than other orbit types

Flight Number vs. Orbit Type



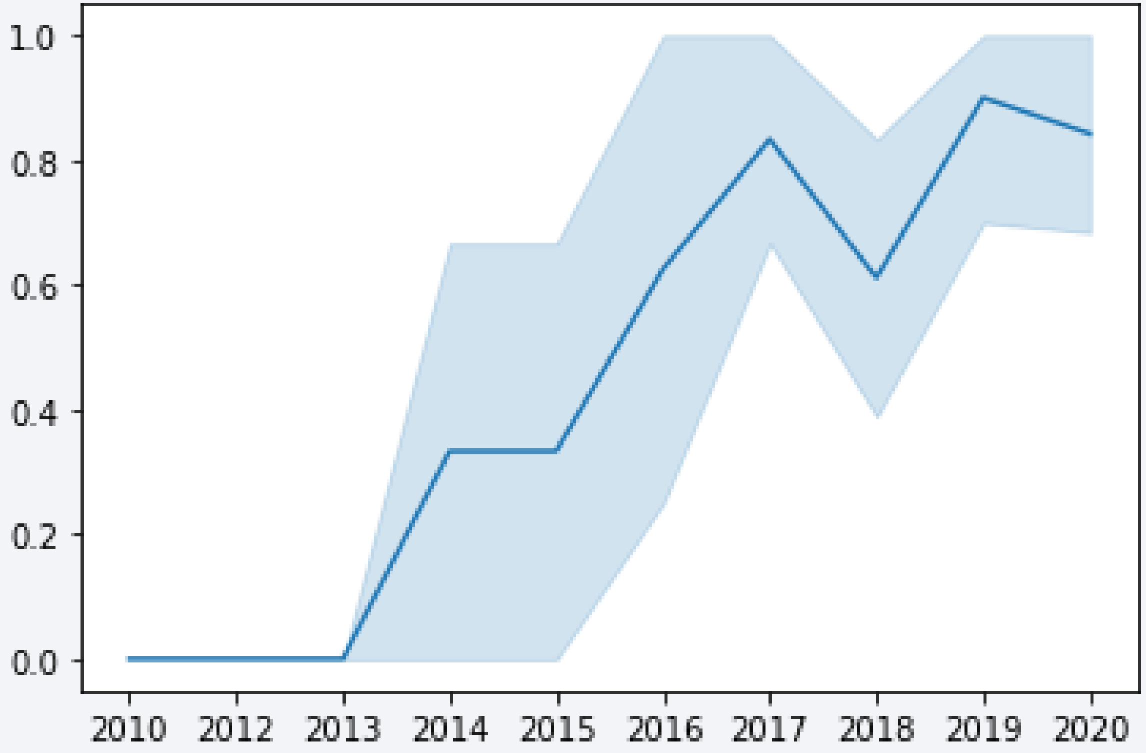
- This bar chart shows that with increase of flight number success rate is also increased for some orbit types

Payload vs. Orbit Type



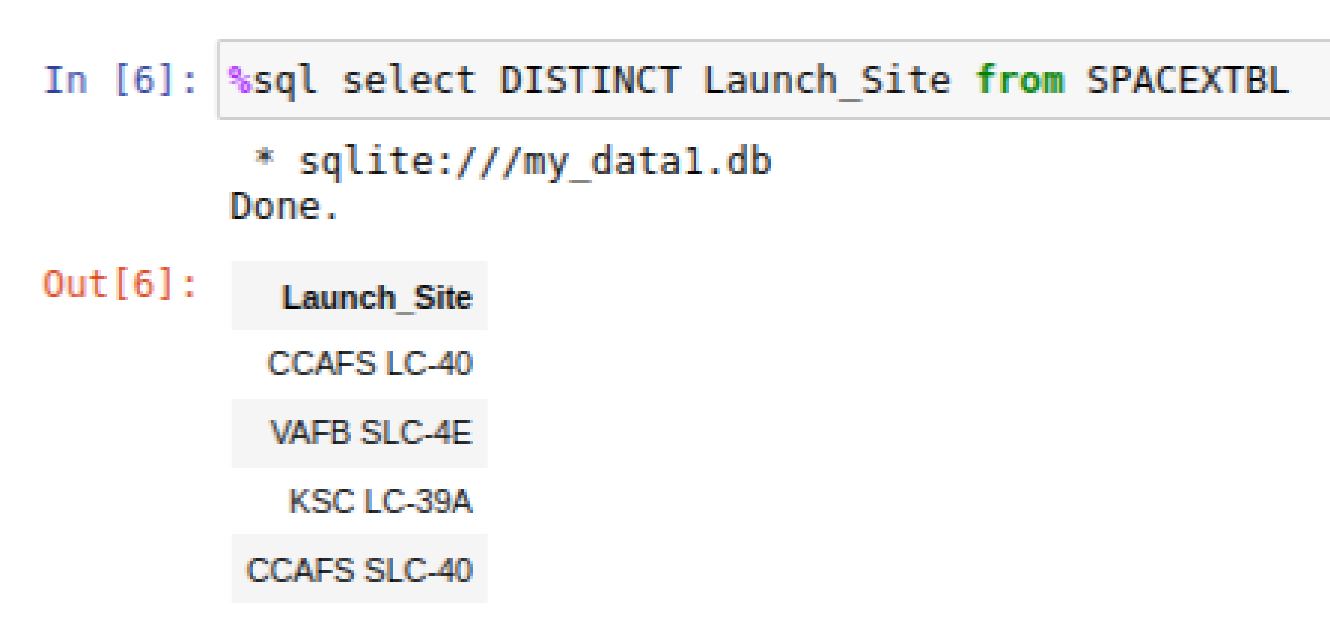
- This scatter plot suggests if the payload mass is between 100 to 4000 then SSO, HEO, ES-L1 are proven effective

Launch Success Yearly Trend



- This line chart shows the success rate of each launch has succeed over the years, specially after 2013

All Launch Site Names



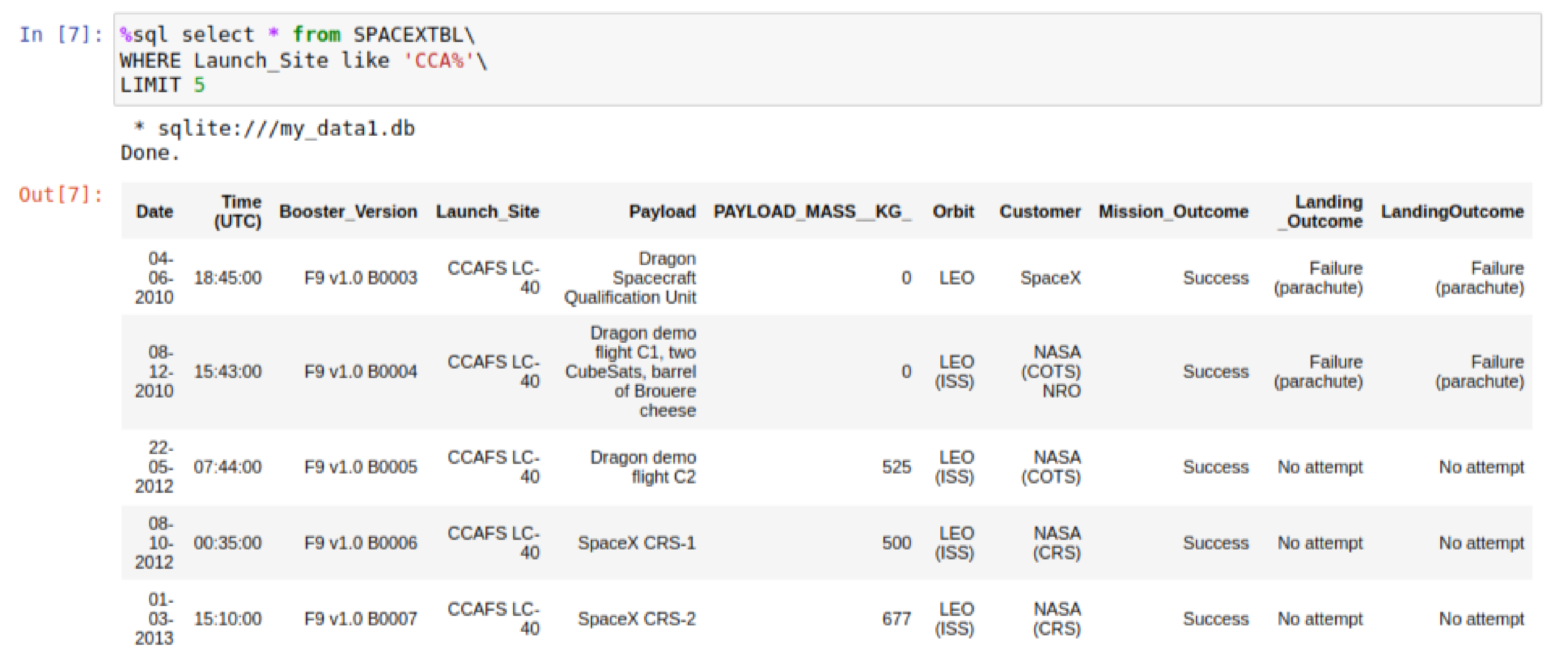
```
In [6]: %sql select DISTINCT Launch_Site from SPACEXTBL

         * sqlite:///my_data1.db
        Done.
```

Out[6]:

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

- Distinct in query gives unique result to be applied on query
- We gathered unique Launch site names using Distinct keyword on column Launch_Site

Launch Site Names Begin with 'CCA'



In [7]:

```
%sql select * from SPACEXTBL\
WHERE Launch_Site like 'CCA%'\
LIMIT 5
```

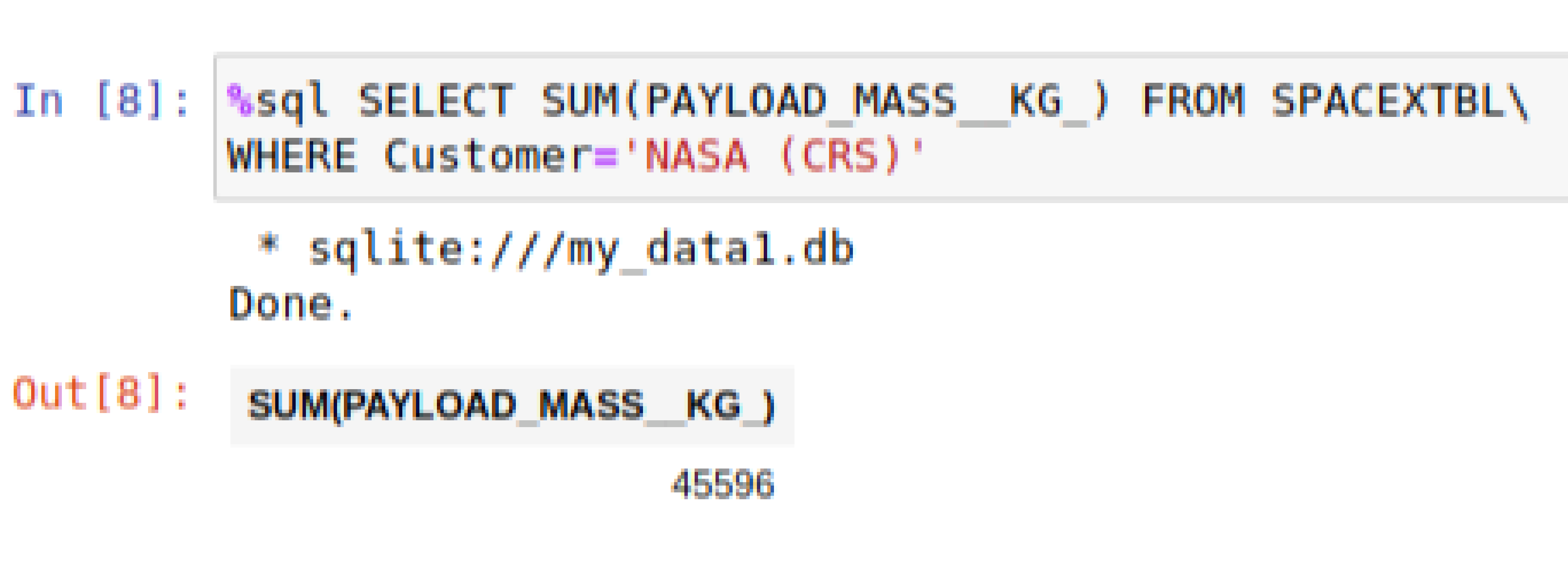
```
 * sqlite:///my_data1.db
Done.
```

Out[7]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome	LandingOutcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	Failure (parachute)
08-12-2010	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)	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	No attempt

- "Like" operator matches similar records with given letters
- Using "like" operator we have filtered records whose Launch Site starts with letters 'CCA'

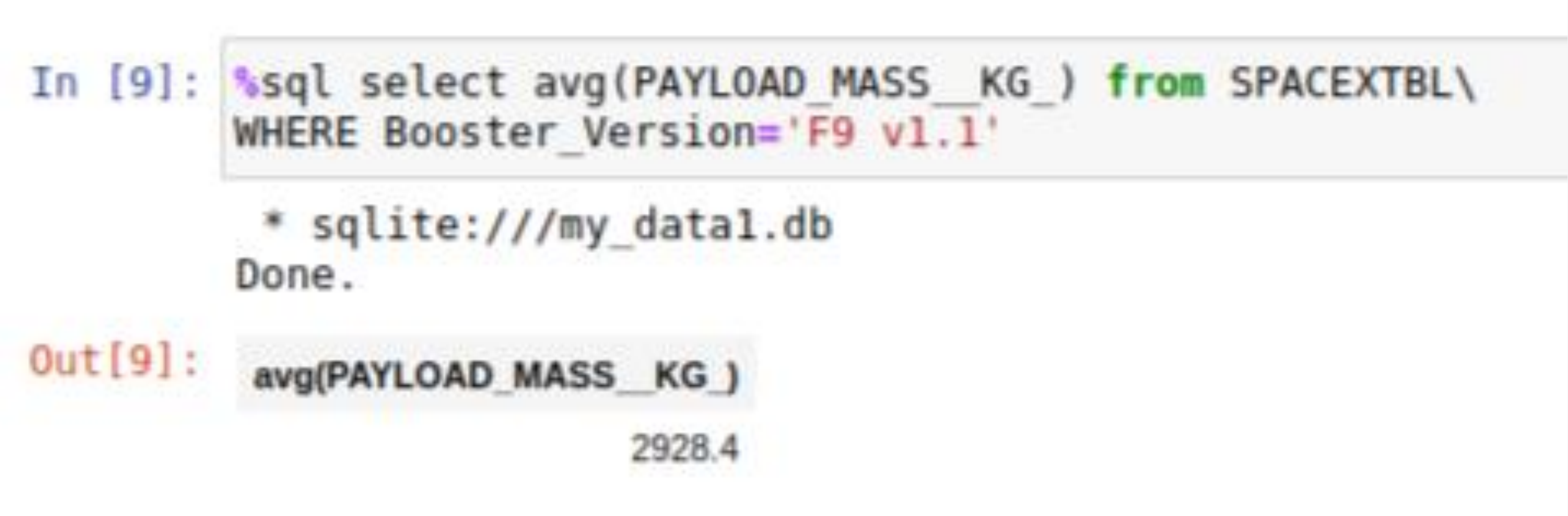
Total Payload Mass



```
In [8]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL\
        WHERE Customer='NASA (CRS)'

         * sqlite:///my_data1.db
        Done.

Out[8]:  SUM(PAYLOAD_MASS__KG_)
                          45596
```

- We got total payload mass in KG by Sum keyword in sql query

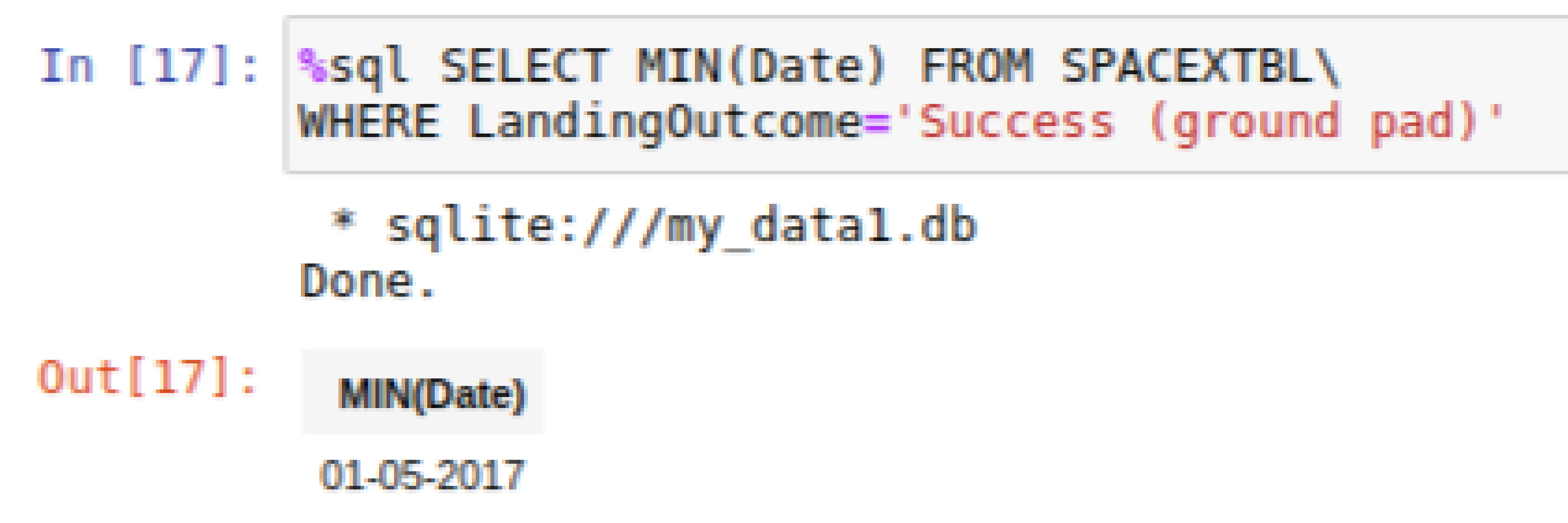
Average Payload Mass by F9 v1.1



```
In [9]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL\
        WHERE Booster_Version='F9 v1.1'

         * sqlite:///my_data1.db
        Done.

Out[9]: avg(PAYLOAD_MASS__KG_)
                        2928.4
```

- We used avg keyword to get average payload mass of Booster version F9 v.1

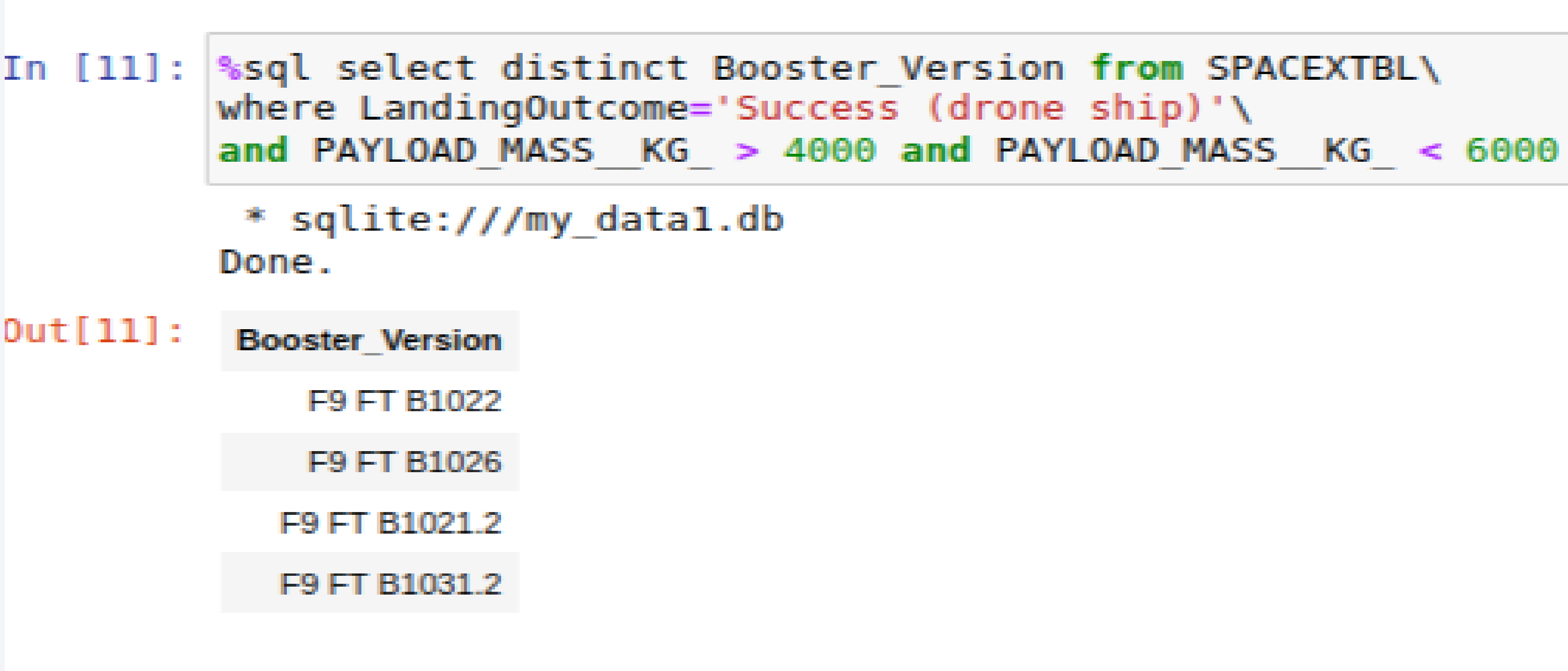
First Successful Ground Landing Date



```
In [17]: %sql SELECT MIN(Date) FROM SPACEXTBL\
         WHERE LandingOutcome='Success (ground pad)'

          * sqlite:///my_data1.db
         Done.

Out[17]:  MIN(Date)
          01-05-2017
```

- We used "min" keyword in sql query to get first successful ground landing date

Successful Drone Ship Landing with Payload between 4000 and 6000

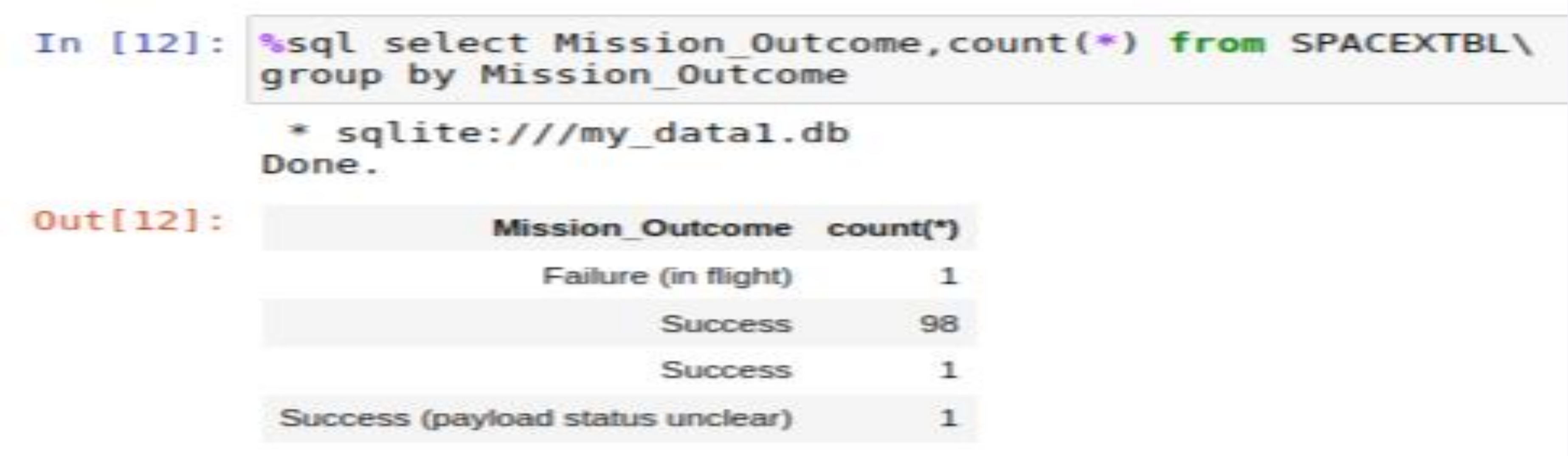


```
In [11]: %sql select distinct Booster_Version from SPACEXTBL\
         where LandingOutcome='Success (drone ship)'\
         and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

          * sqlite:///my_data1.db
         Done.

Out[11]:
```

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

- By above sql query we get unique booster versions. It is important to note that we've added multiple conditions in where clause to get the expected result

Total Number of Successful and Failure Mission Outcomes



```
In [12]: %sql select Mission_Outcome,count(*) from SPACEXTBL\
         group by Mission_Outcome
```

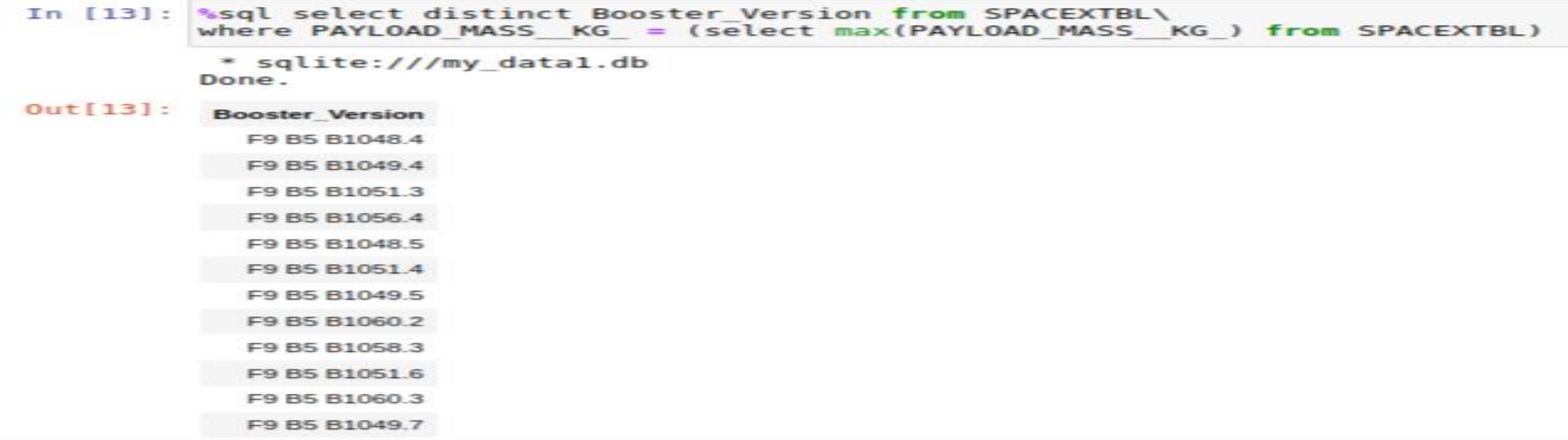
```
 * sqlite:///my_data1.db
Done.
```

Out[12]:

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- In this sql query we used count & group by together to get total no. Of successful & failure mission outcomes

Boosters Carried Maximum Payload



In [13]:
```
%sql select distinct Booster_Version from SPACEXTBL\
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

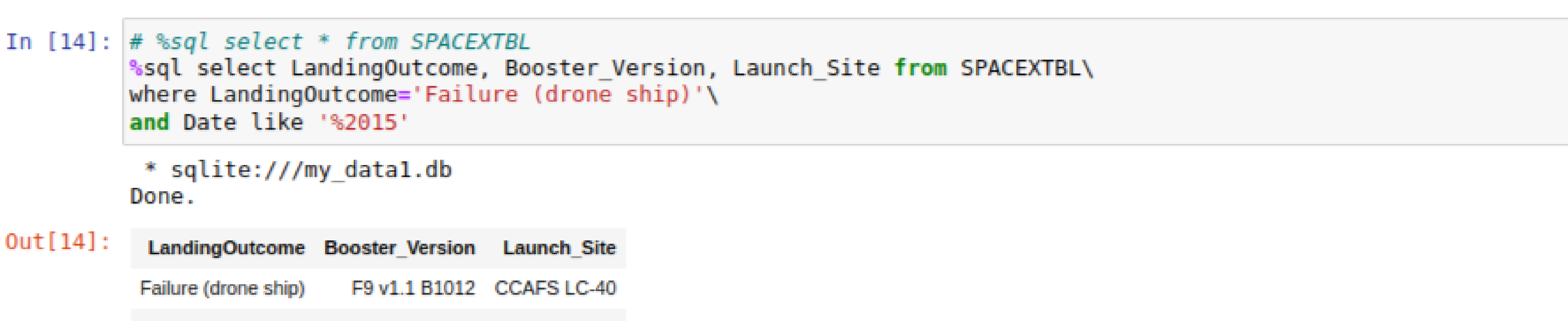
```
 * sqlite:///my_data1.db
Done.
```

Out[13]:

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

- With max subquery & distinct in the main query we found out maximum payload carried out by different booster versions

2015 Launch Records



```
In [14]: # %sql select * from SPACEXTBL
         %sql select LandingOutcome, Booster_Version, Launch_Site from SPACEXTBL\
         where LandingOutcome='Failure (drone ship)'\
         and Date like '%2015'
```

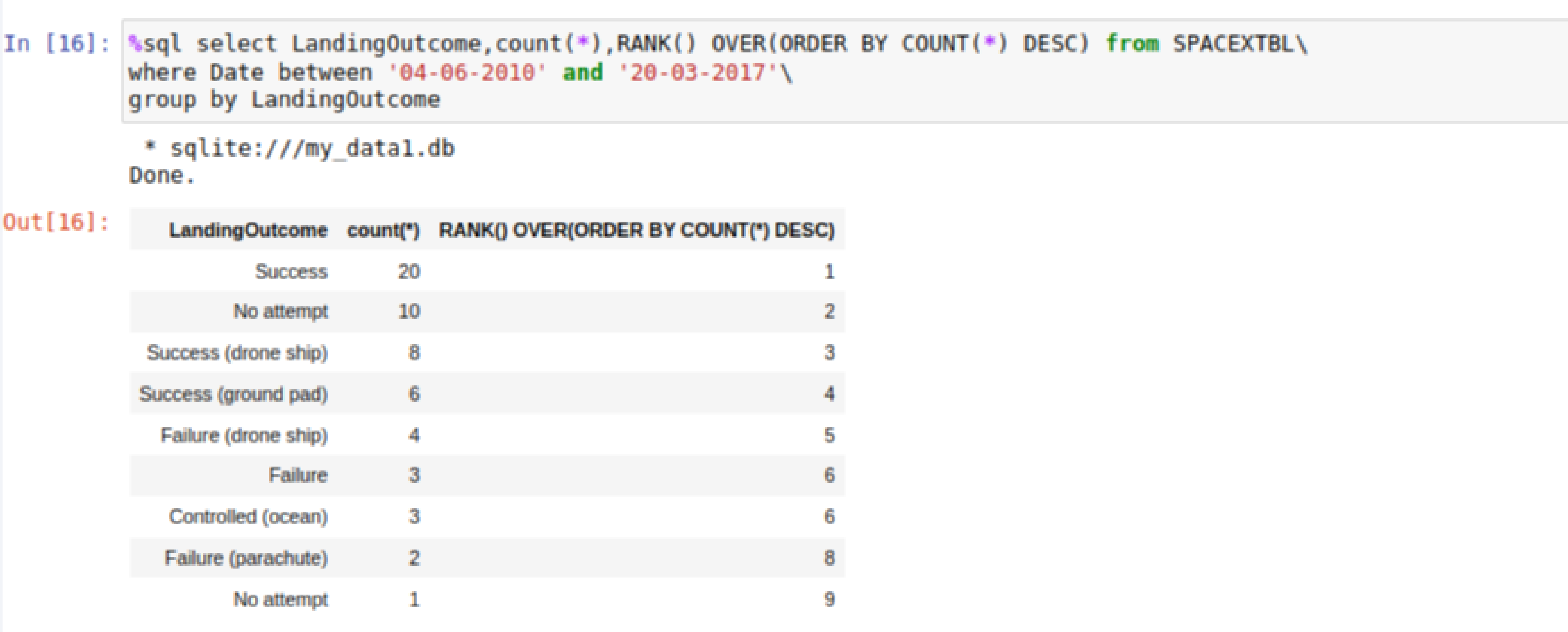
```
 * sqlite:///my_data1.db
Done.
```

Out[14]:

LandingOutcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

- We've filtered records of 2015 with like operator

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



In [16]:

```
%sql select LandingOutcome,count(*),RANK() OVER(ORDER BY COUNT(*) DESC) from SPACEXTBL\
where Date between '04-06-2010' and '20-03-2017'\
group by LandingOutcome
```

```
 * sqlite:///my_data1.db
Done.
```

Out[16]:

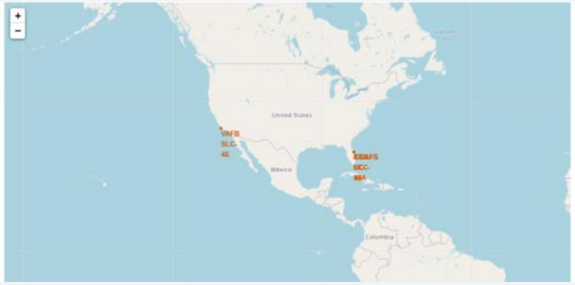
LandingOutcome	count(*)	RANK() OVER(ORDER BY COUNT(*) DESC)
Success	20	1
No attempt	10	2
Success (drone ship)	8	3
Success (ground pad)	6	4
Failure (drone ship)	4	5
Failure	3	6
Controlled (ocean)	3	6
Failure (parachute)	2	8
No attempt	1	9

- We have used rank over sql function to give rank of landing outcomes between 2010-06-04 and 2017-03-20

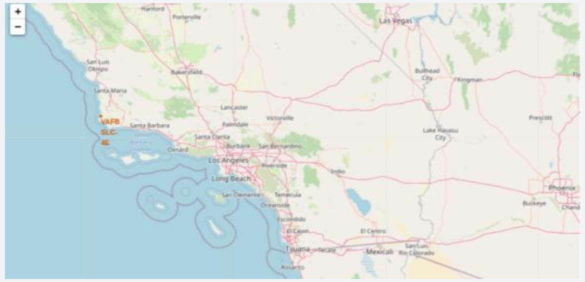
Section 3

Launch Sites Proximities Analysis

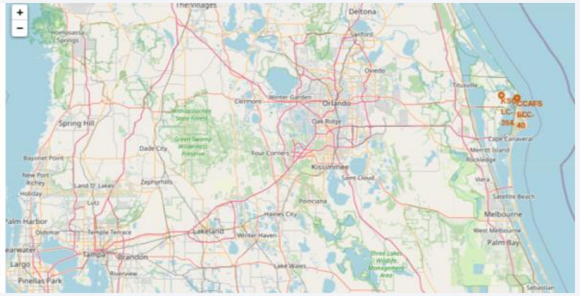
All Launch Sites Location on Map



We can see launch sites on 2 different coasts of USA



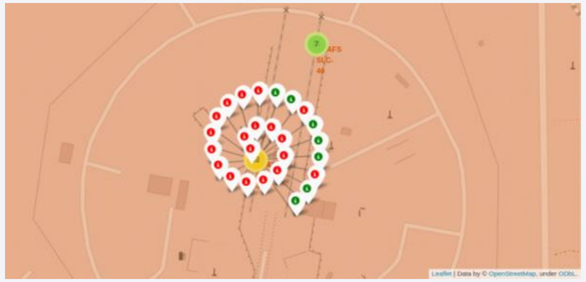
Above launch site we can see on west coast



Above launch site we can see on east coast

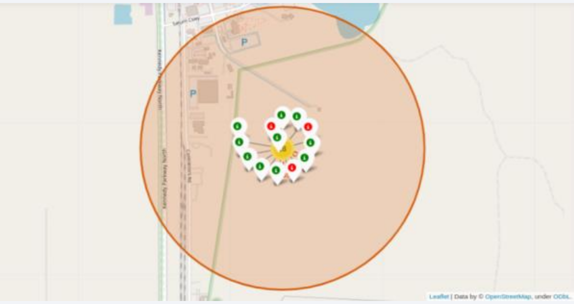
Launch outcomes with color labeled

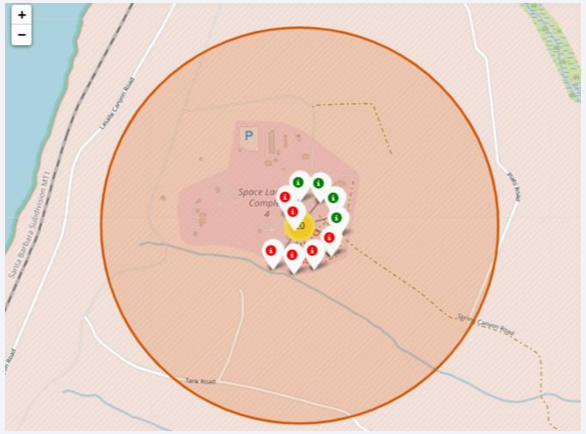
We can see launching outcomes in different coded system, green suggests the outcome is successful whereas red suggest the outcome was a failure



We can see launch site CCAFS LC-40 has more failed outcomes than successful ones

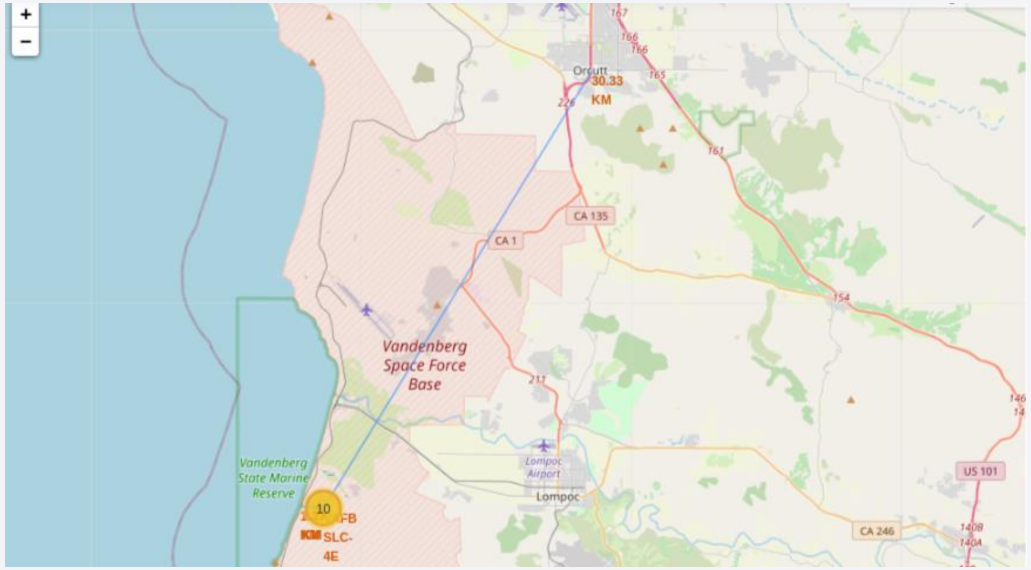
We can see highest success rate of launch than any other launch site at site KSC LC-39A



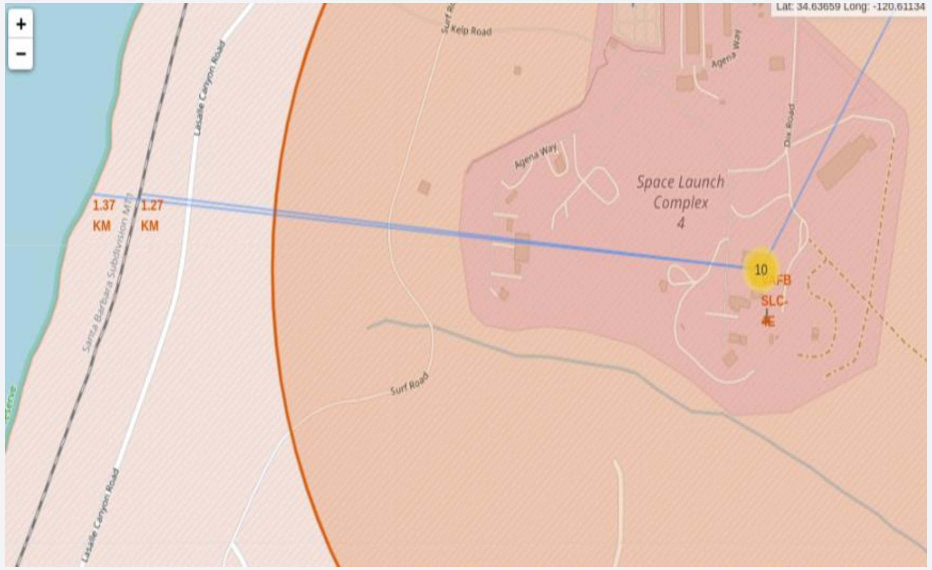


We can see the success ratio is 50% for Launch Site VAFB SLC-4E

Coastline, Railway & City distance to Launch Site

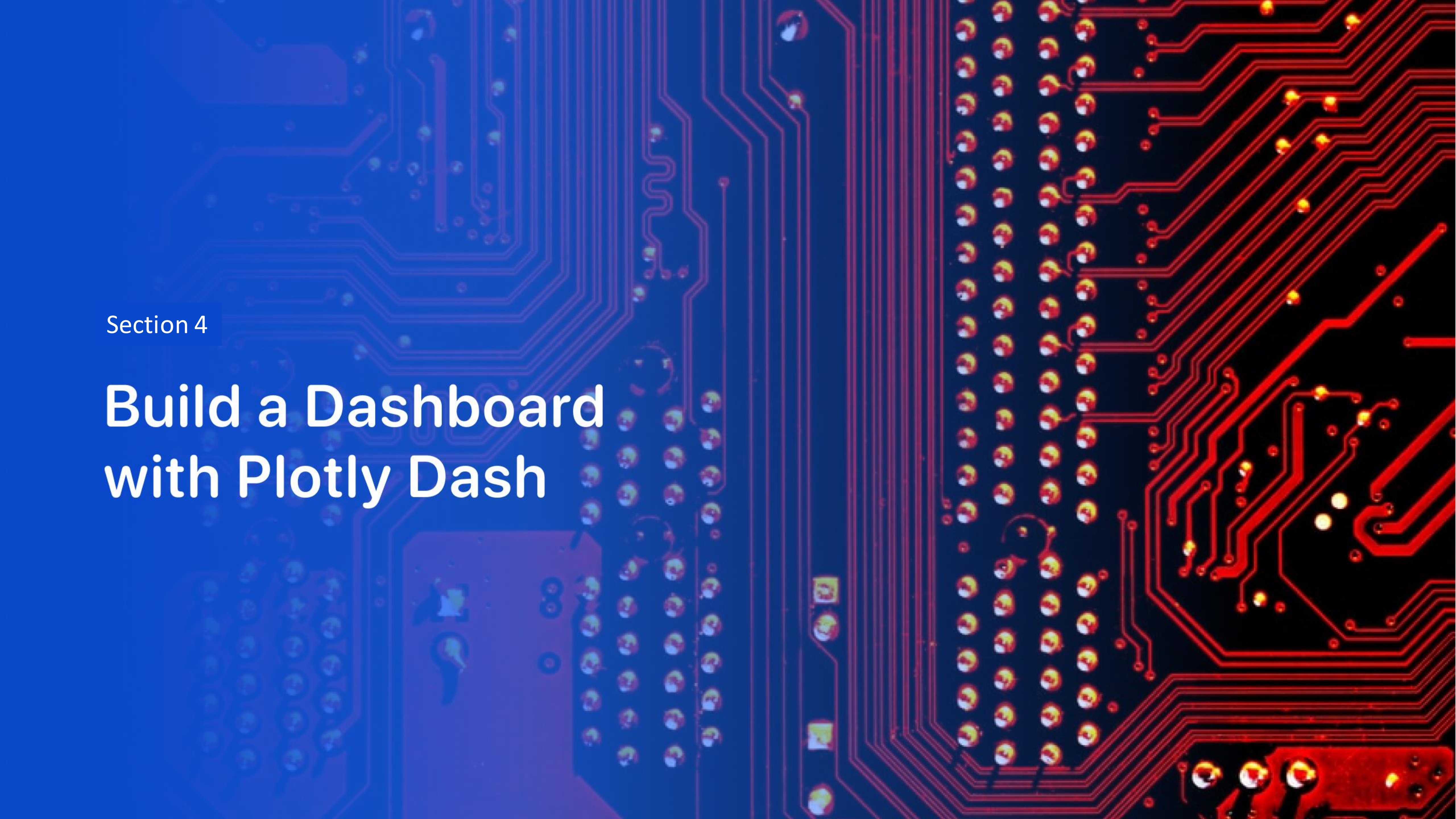


We can see the nearest city is about 33 KM from the launch site



We can see the nearest coastline & railway line from the launch site

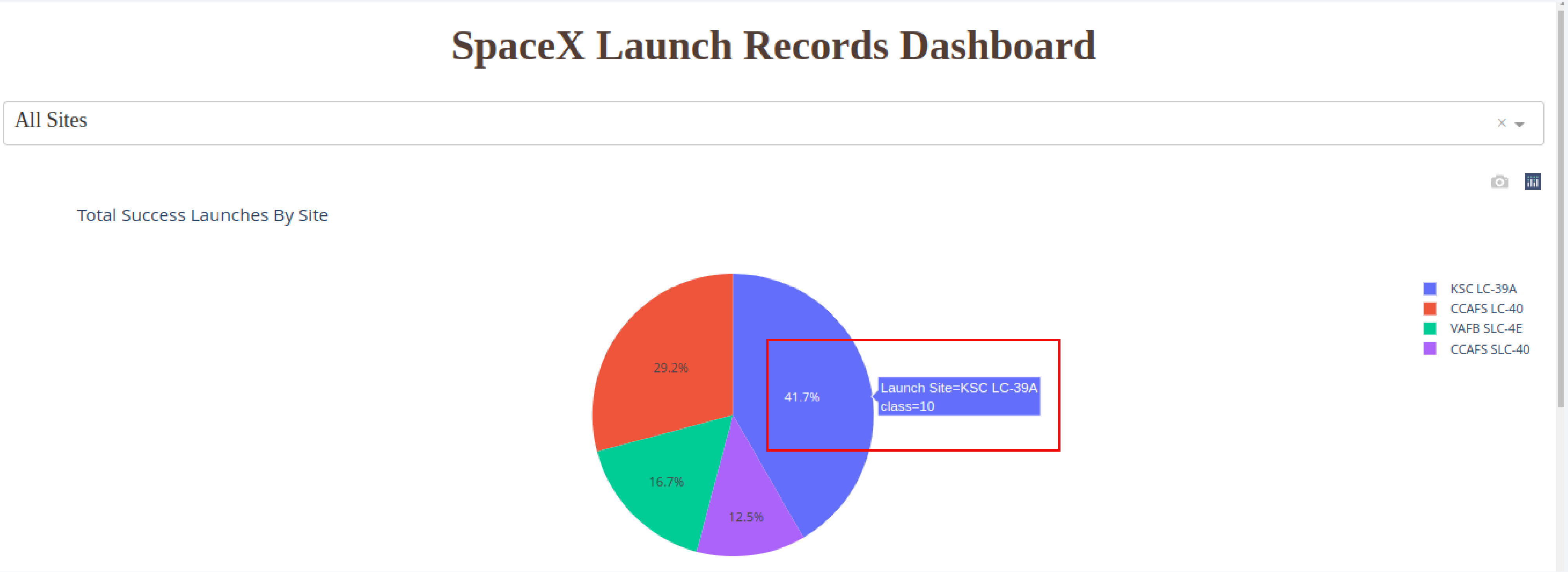
It is very important to know the nearest city, railway lines, coast lines to the launch site. Payload sometimes are heavy in weight & only means of transportation could be via maritime or railway



Section 4

Build a Dashboard with Plotly Dash

Dashboard – Launch Success Pie Chart

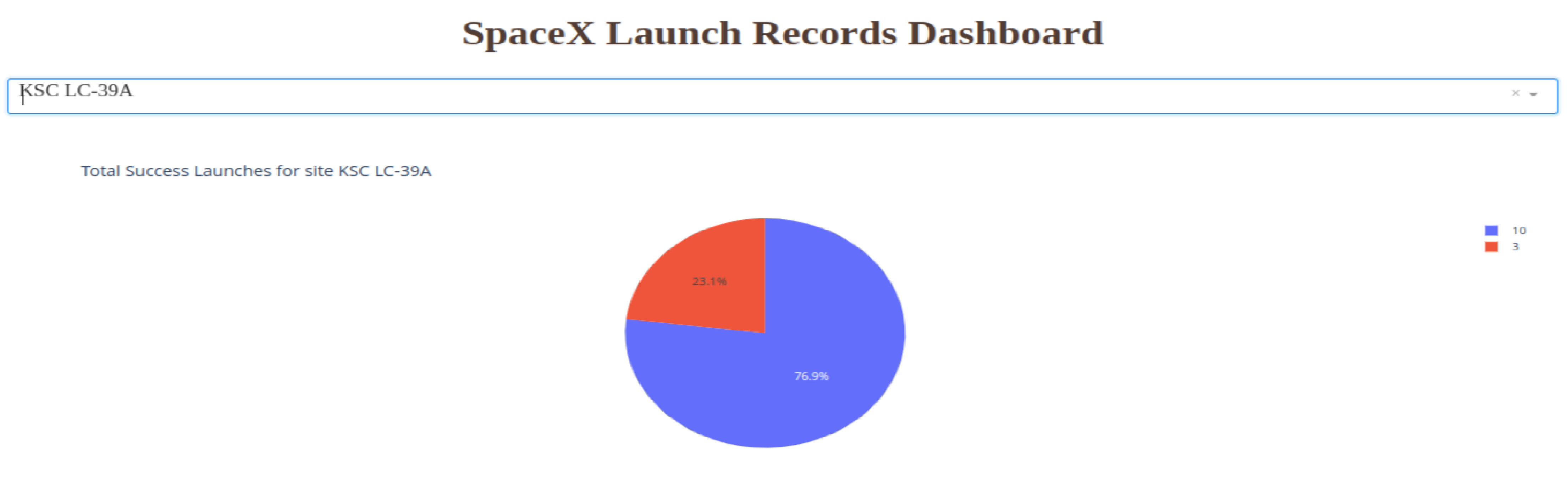


SpaceX Launch Records Dashboard

All Sites

In above pie chart we can see launch site KSC LC-39A has highest success rate

Dashboard – Best Launch Site



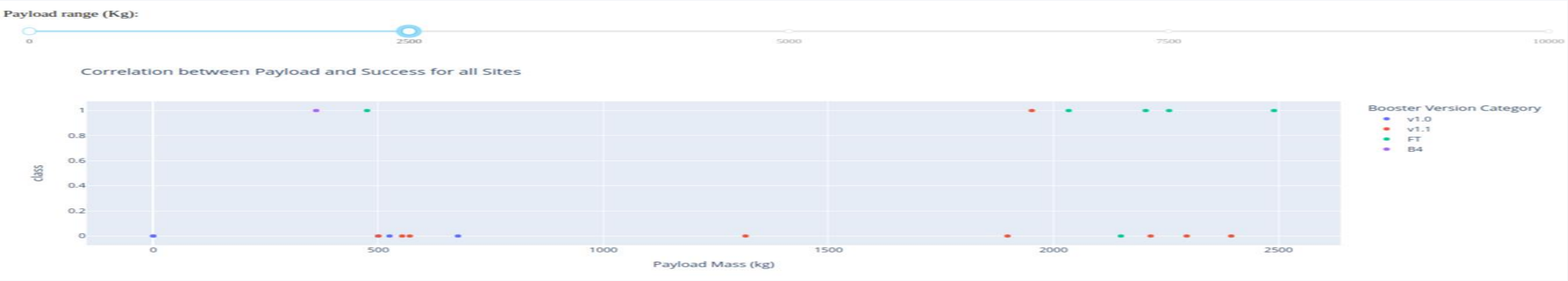
SpaceX Launch Records Dashboard

KSC LC-39A

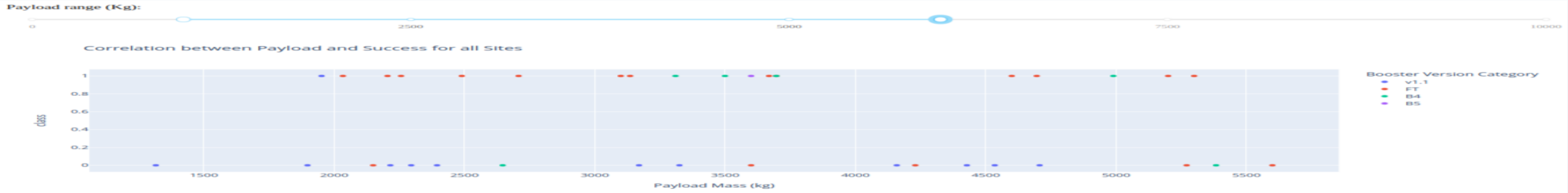
Total Success Launches for site KSC LC-39A

As we can see KSC LC-39A has the highest success rate of 76.9% among all other launch sites

Dashboard – Payload vs. Outcome Scatter Plot



We can see in above scatter plot, the outcome was failure for **booster version 1.1** when payload is **0-2500 KG** in weight

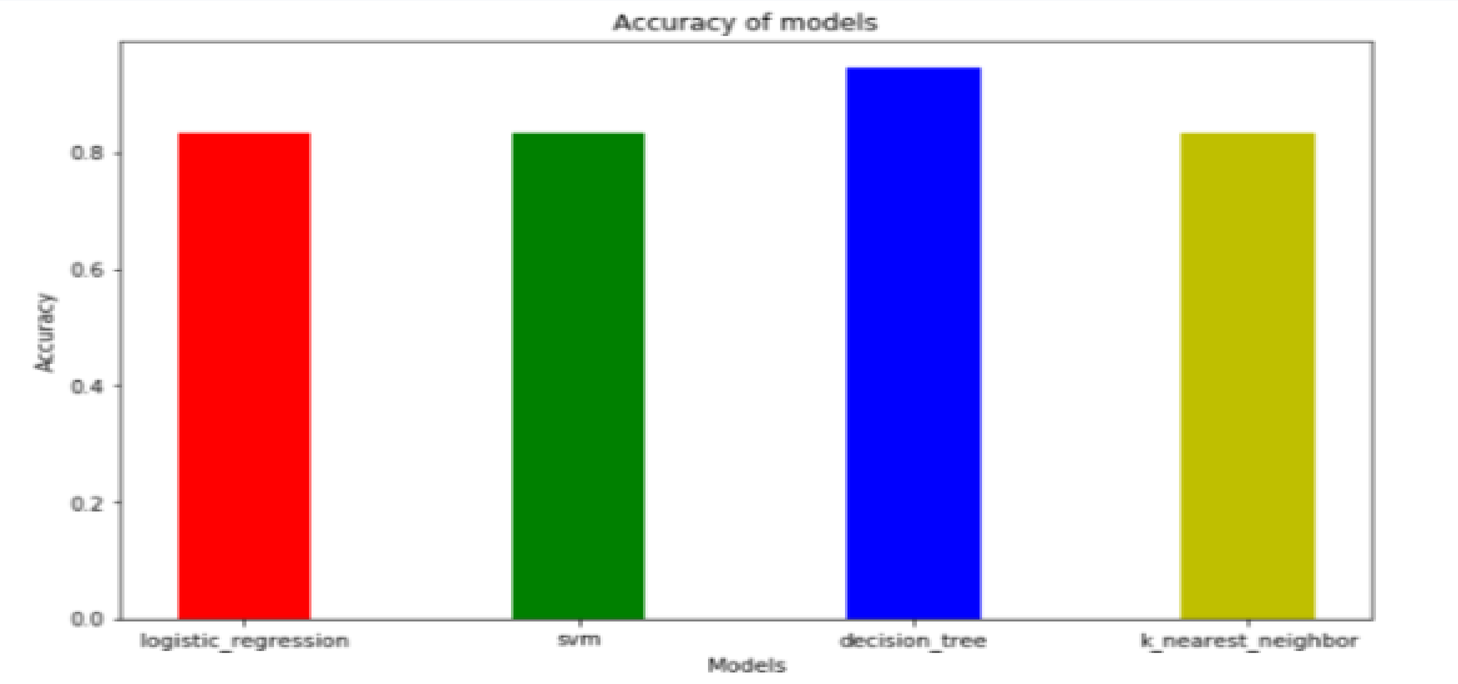


In above scatter plot we can see **booster version FT** has higher success rate when payload weight is between **2000-5500 KG**

Section 5

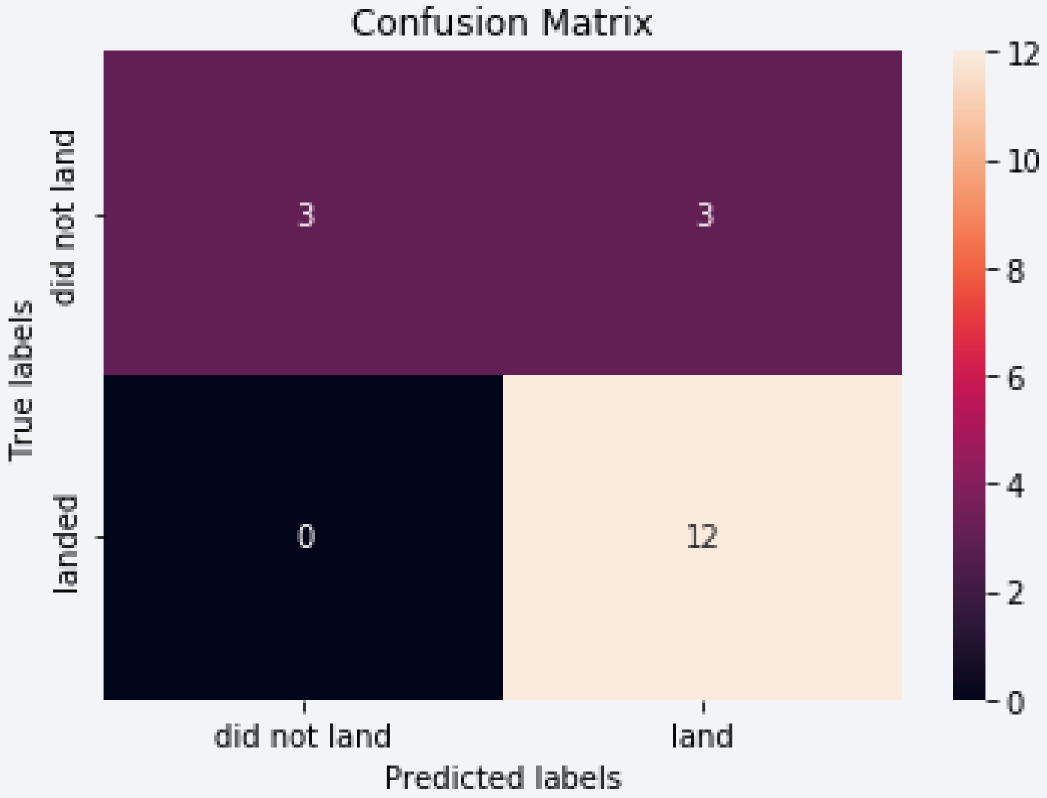
Predictive Analysis (Classification)

Classification Accuracy



We have trained & tested Logistic Regression, SVM, Decision Tree & K-Nearest Neighbor models. Among them we have found highest accuracy with Decision Tree model with 94%

Confusion Matrix



In the above Confusion Matrix of Decision Tree model we can see we predicted landing outcomes with high success rates, the failure outcomes prediction we can see is 100%

Conclusions

- By EDA we revealed hidden insights like orbit type plays important role in successful launch
- We also covered the relation between payload weight & booster versions
- We can also see that after year 2013 the success of launches has been increasing
- Out of SVM, Logistic Regression, Decision Tree & K-Nearest Neighbor models we trained & tested them & the best accuracy we got with Decision Tree
- KSC LC-39A proved to have most successful launches than any other launch sites
- Orbit type ES-L1, GEO, HEO & SSO have the best success rate

Appendix

- Github link : https://github.com/keyur077/IBM_DS_Capstone

Thank you!