IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nishith Sahoo

08-May-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
  - Machine Learning Prediction
- 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 space X for a rocket launch. This objective of the project is to build a model to predict if the first stage will land successfully.

Problems you want to find answers

- What factors determine if a rocket will land successfully
- The interaction amongst various features that determine the success rate of a successful landing
- What conditions does SpaceX have meet to maximize the probability of success landing rate

Section 1

Methodology

Methodology

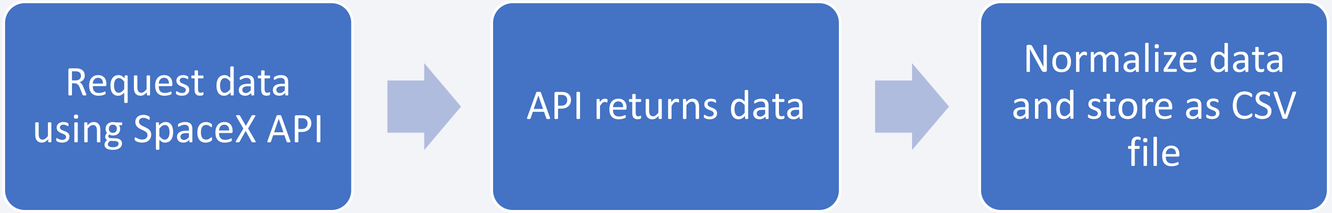
Executive Summary

- Data collection methodology:
  - Data has been collected using SpaceX REST API, &
  - Webscrapping from Wikipedia
- Perform data wrangling
  - One hot encoding was applied to categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was collected by two methods

1. SpaceX API methodology:

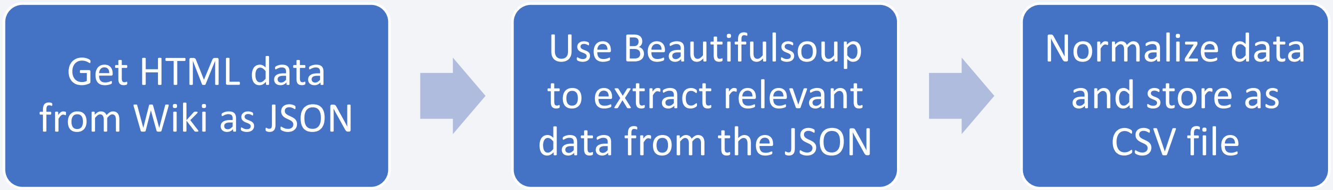


Request data using SpaceX API

API returns data

Normalize data and store as CSV file

2. Webscrapping Methodology:



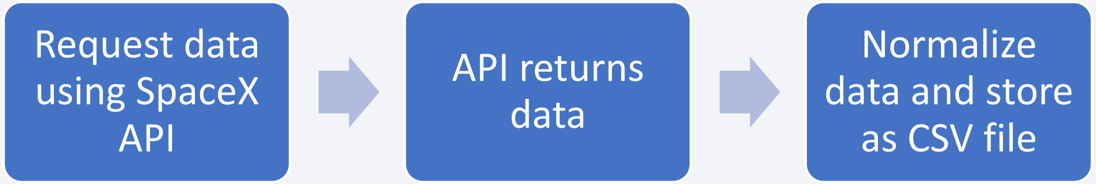
Get HTML data from Wiki as JSON

Use Beautifulsoup to extract relevant data from the JSON

Normalize data and store as CSV file

Data Collection – SpaceX API

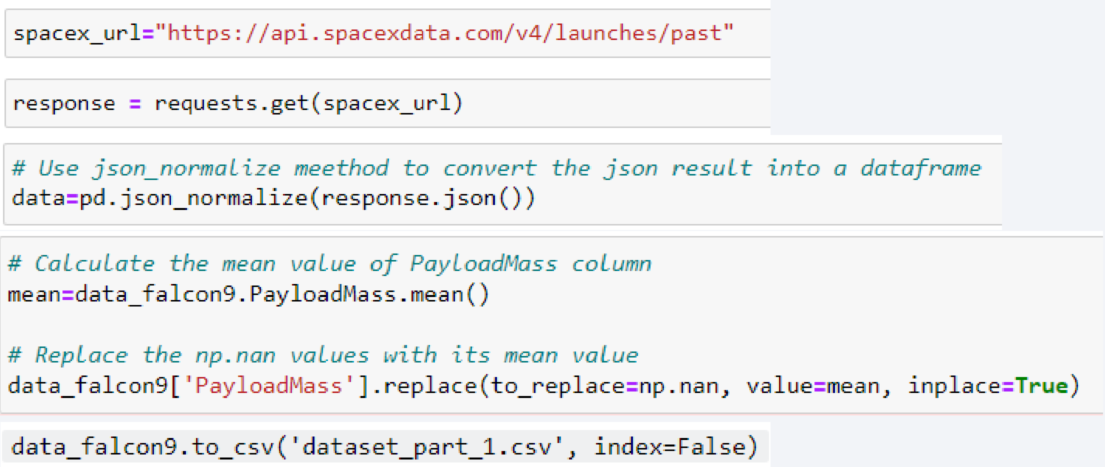
- Get request to the SpaceX API was used to collect data, which was then cleaned, normalized, missing values replaced and exported to CSV.
- GitHub URL of the completed SpaceX API calls: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week1A_%20Data%20Collection%20using%20API.ipynb



Request data using SpaceX API

API returns data

Normalize data and store as CSV file



```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

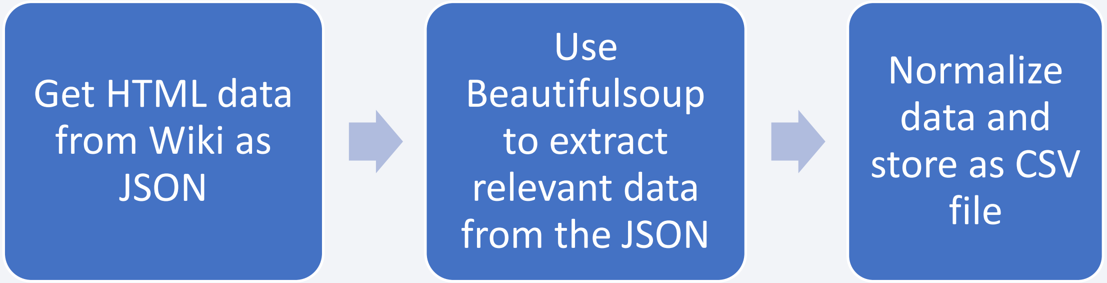
```
# Calculate the mean value of PayloadMass column
mean=data_falcon9.PayloadMass.mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(to_replace=np.nan, value=mean, inplace=True)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Webscrapping was used to extract information from Wikipedia. Tables were parsed using beautifulsoup. The data was normalized and exported to CSV.

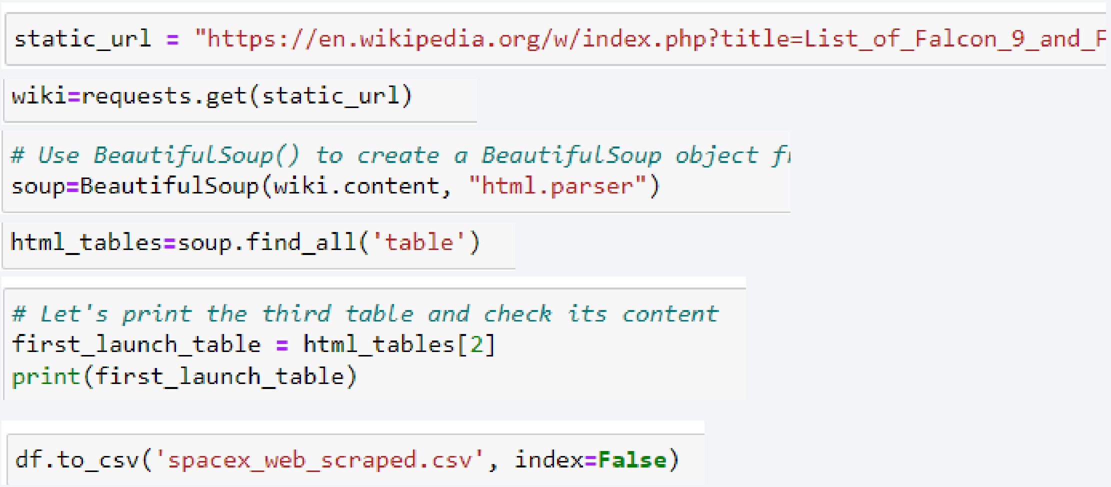
- GitHub URL of the completed web scraping notebook: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week1B_%20Data%20Collection%20using%20Webscrapping.ipynb



Get HTML data from Wiki as JSON

Use Beautifulsoup to extract relevant data from the JSON

Normalize data and store as CSV file



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_F
```

```
wiki=requests.get(static_url)
```

```
# Use BeautifulSoup() to create a BeautifulSoup object fi
soup=BeautifulSoup(wiki.content, "html.parser")
```

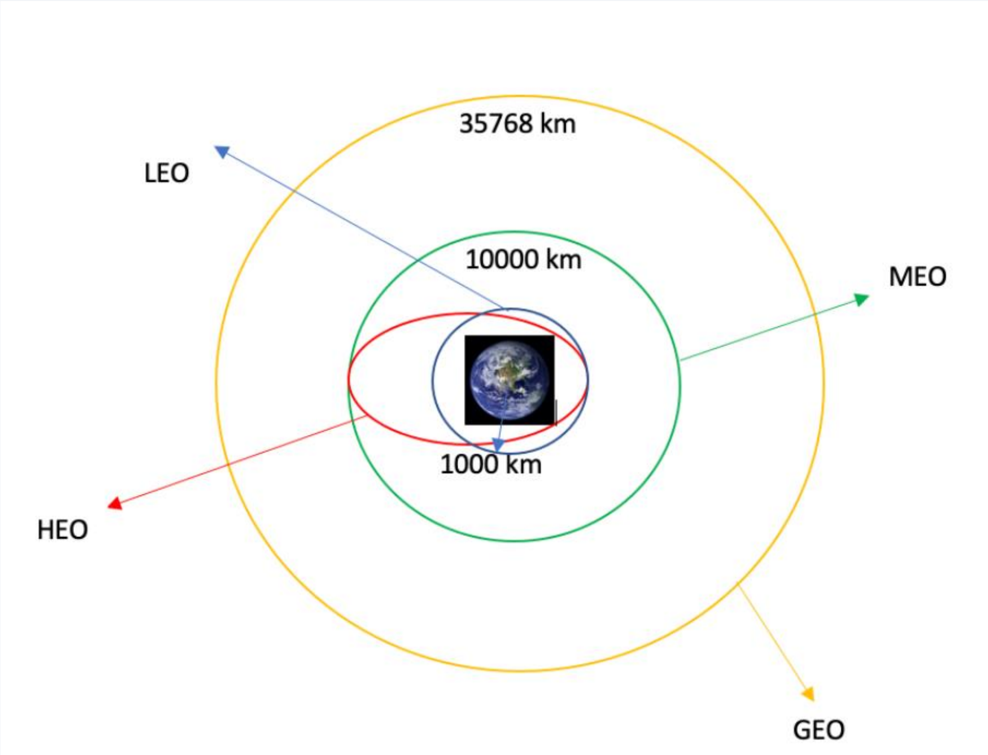
```
html_tables=soup.find_all('table')
```

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

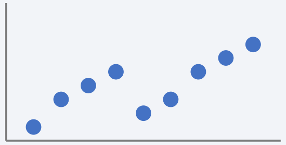
- Process Followed:
  - EDA was performed on the dataset
  - Number of launches at each site was determined
  - Number of launches to each orbit was determined
  - Mission outcome was determined based on target orbit
  - Percent of success was determined
  - Data was exported to a CSV
- GitHub URL of your completed data wrangling related notebooks: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week1C_%20Data%20Wrangling.ipynb



Some of the target orbits are shown here

EDA with Data Visualization

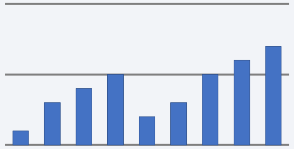
- Scatter Graphs drawn to visualize:



  - Payload Mass vs Flight Number (visualized by Class)
  - Launch site vs Flight Number (visualized by Class)
  - Payload Mass vs Launch Site (visualized by Class)
  - Orbit vs Flight Number (visualized by Class)
  - Payload Mass vs Orbit (visualized by Class)

GitHub URL: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week2B_%20EDA%20with%20Data%20Visualization.ipynb

- Bar Graph was used to visualize:



  - Success rate vs Orbit

- Line Plot was used to visualize:

  - Success rate vs Launch Year

EDA with SQL

- SQL queries were performed to gather information about the data:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'KSC’
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
  - Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the records which will display the month names, successful landing_outcomes in ground pad, booster versions, launch_site for the months in year 2017
  - Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

- Add the GitHub URL of your completed EDA with SQL notebook:
  https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week2A_%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities. Answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.
- Add the GitHub URL of your completed interactive map with Folium map: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week3A_%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Add the GitHub URL of your completed Plotly Dash lab:
  https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week3B_%20Dashboard%20Using%20Plotly.ipynb

Predictive Analysis (Classification)

- BUILDING MODEL
  - Load dataset into NumPy and Pandas
  - Transform Data
  - Split data into training and testing data sets
  - Set parameters and algorithms to GridSearchCV for different ML models
  - Fit training data into the GridSearchCV objects and train models
- EVALUATING MODEL
  - Check accuracy for each model
  - Get tuned hyperparameters for each type of algorithms
  - Plot Confusion Matrix
- FINDING THE BEST PERFORMING CLASSIFICATION MODEL
  - The model with the best testing accuracy score is the best performing model

- GitHub URL: https://github.com/nishithsahoo/IBM-Data-Science-Capstone-SpaceX/blob/c26033e576d8d0c352b63d29e6c11bc7374e4355/Week4_%20Machine%20Learning%20Prediction.ipynb

Results

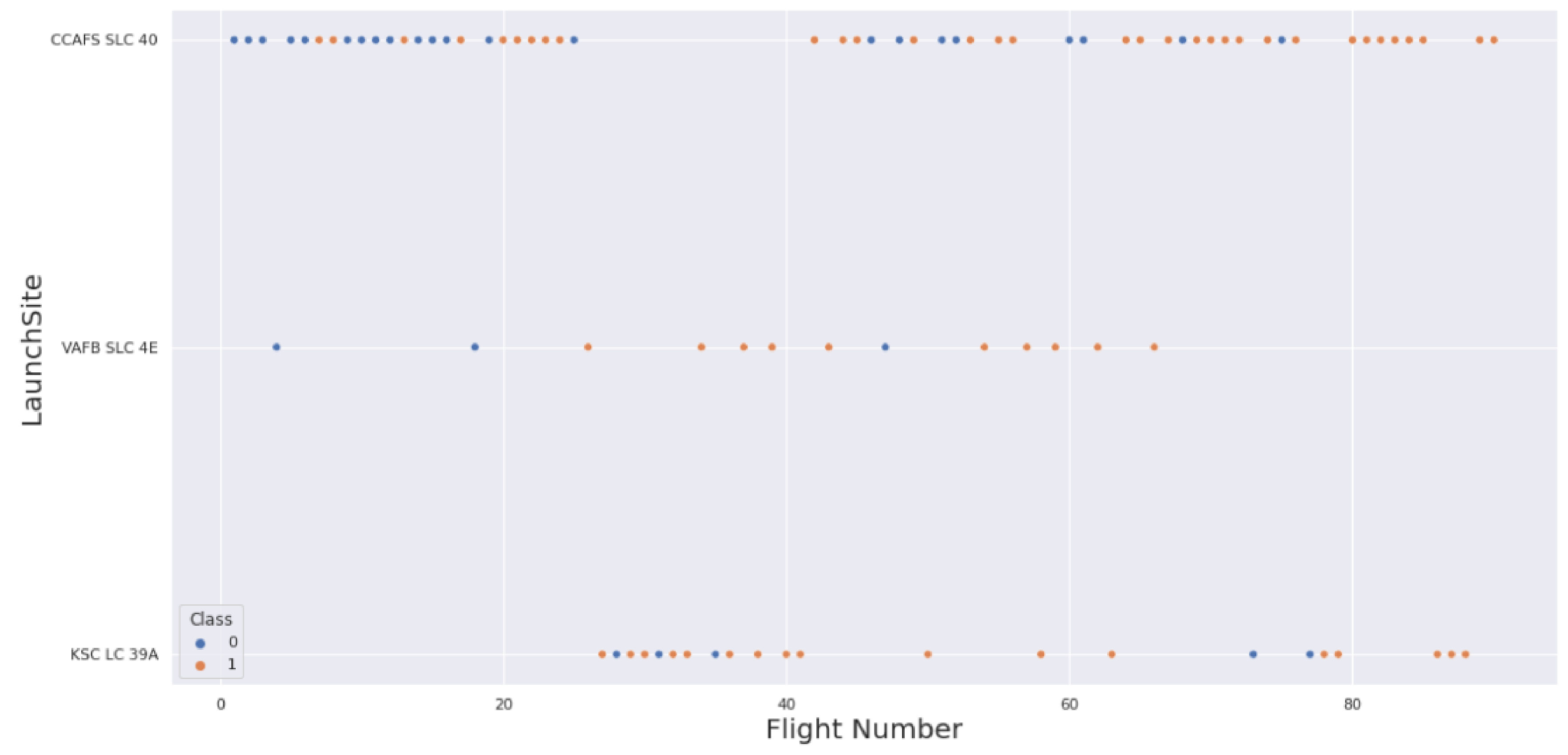
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

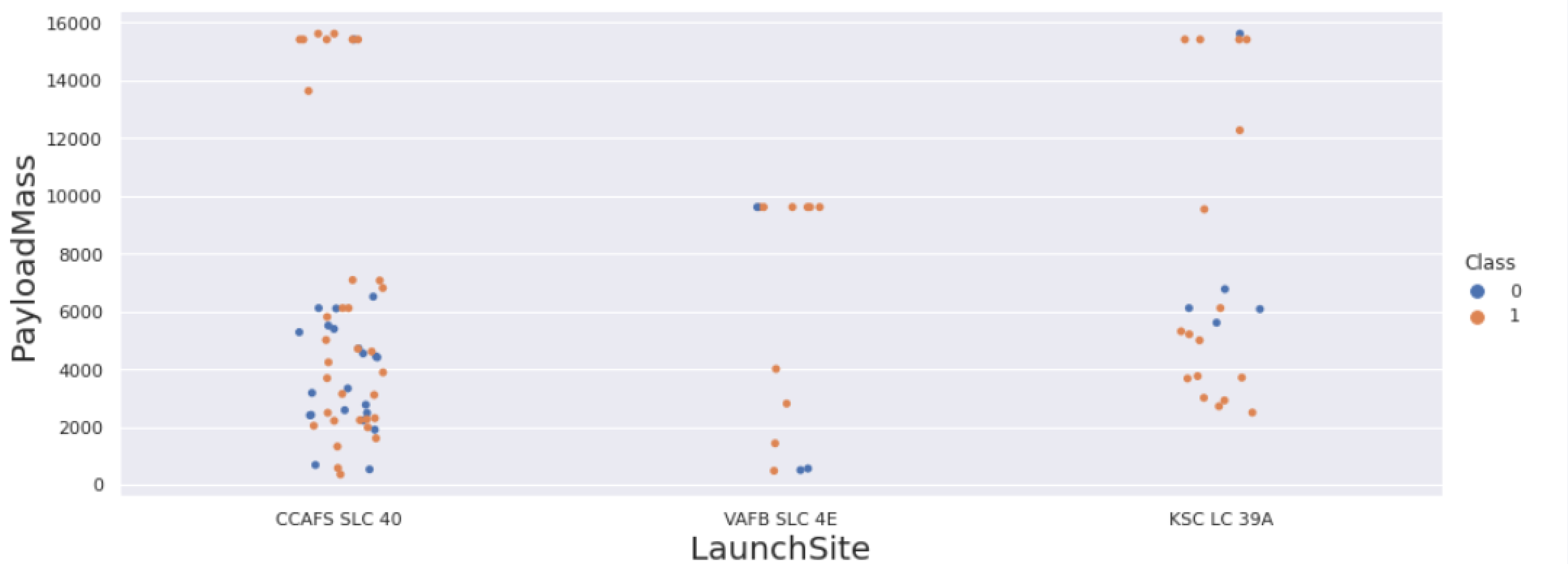
Flight Number vs. Launch Site

- It is observed that different launch sites have different success rates. CCAFS LC-40, has a lower success rate compared to KSC LC-39A and VAFB SLC 4E.



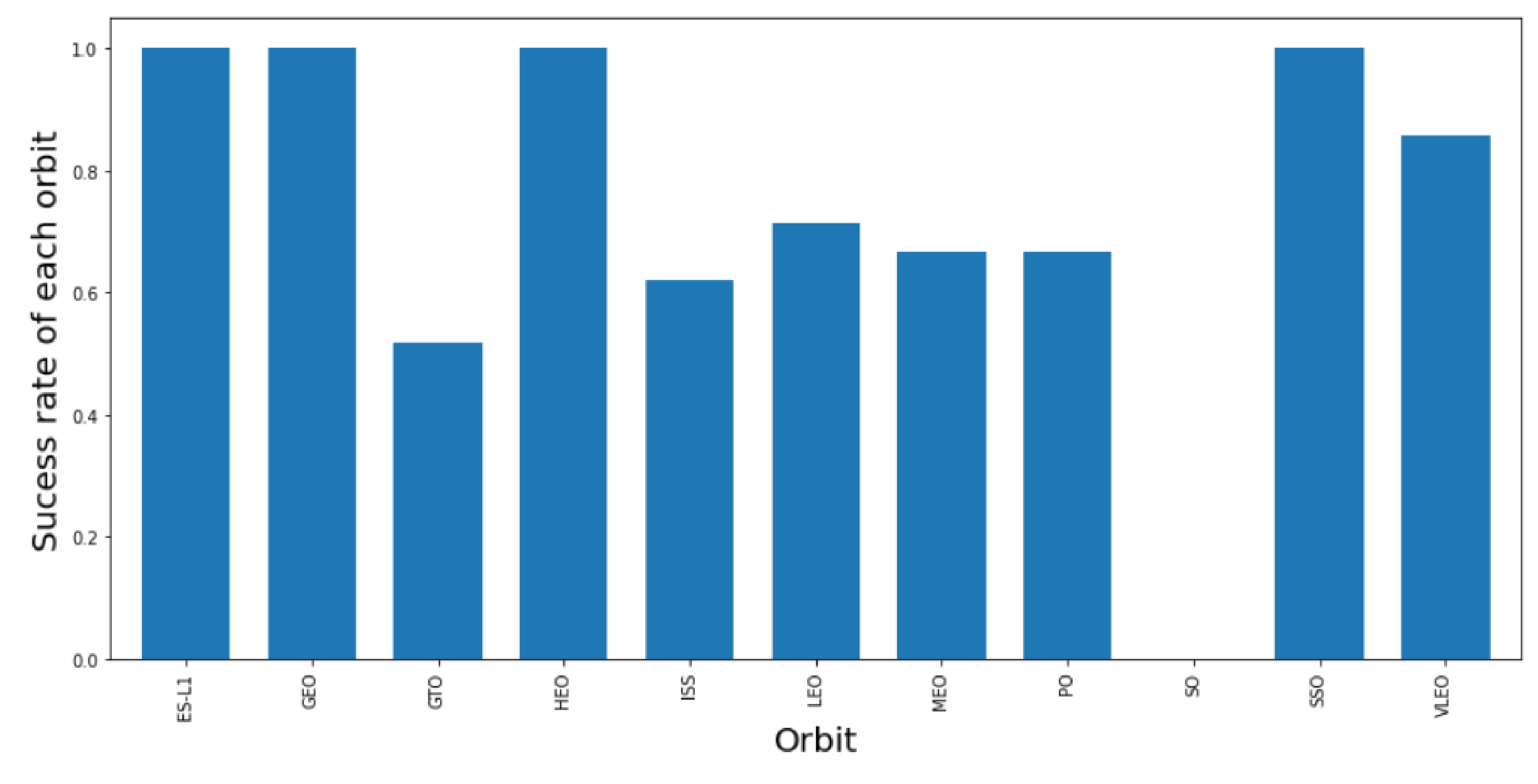
Payload vs. Launch Site

- It is observed that Launch site VAFB SLC 4E does not have any rocket launches with payload greater than 10000kg. CCAFS SLC 40 has no failures for bigger payloads (>12000kg)



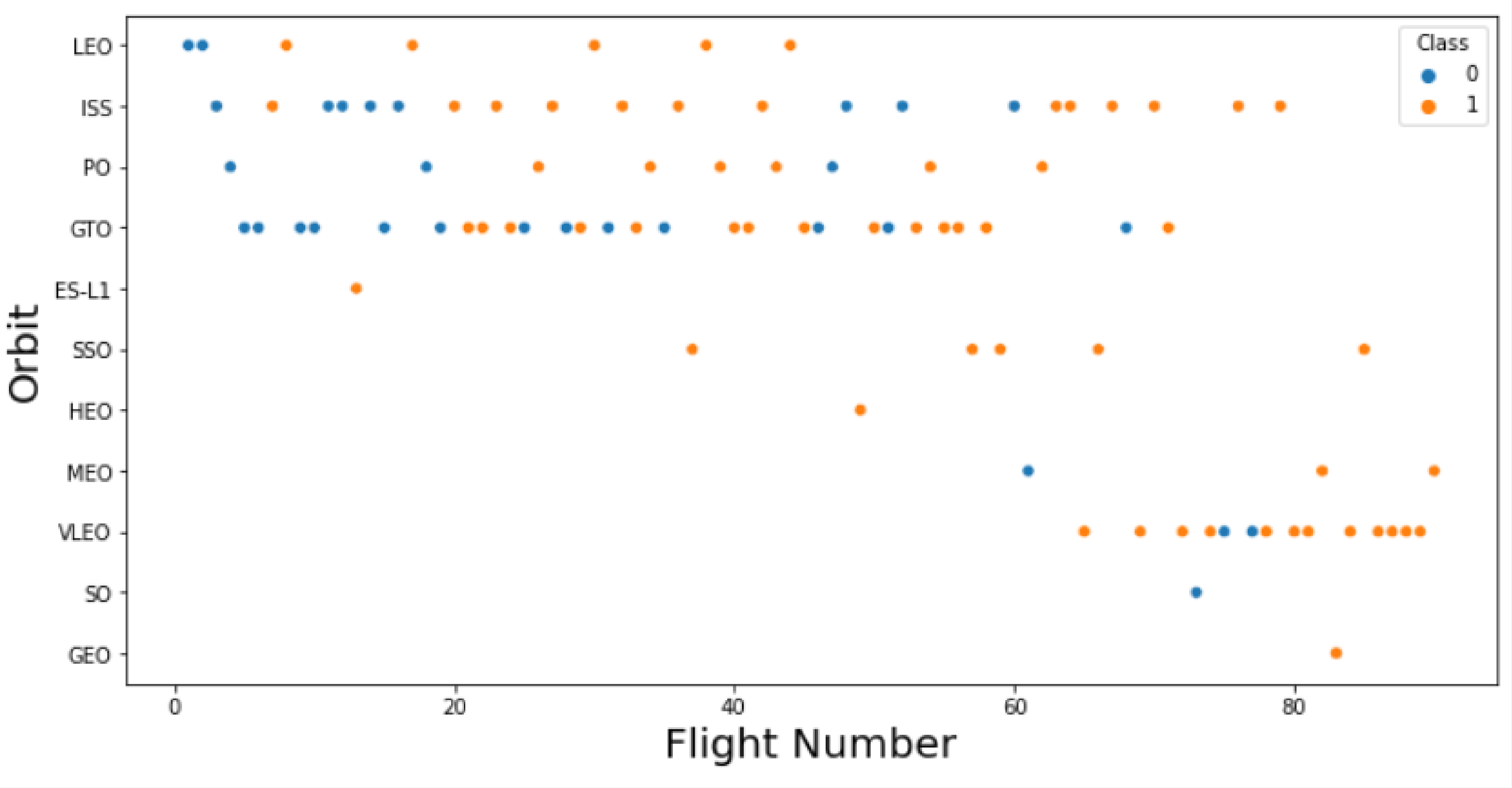
Success Rate vs. Orbit Type

- It is observed that ES-L1, GEO, HEO and SSO orbits have the highest success rates while GTO has lowest success rate.



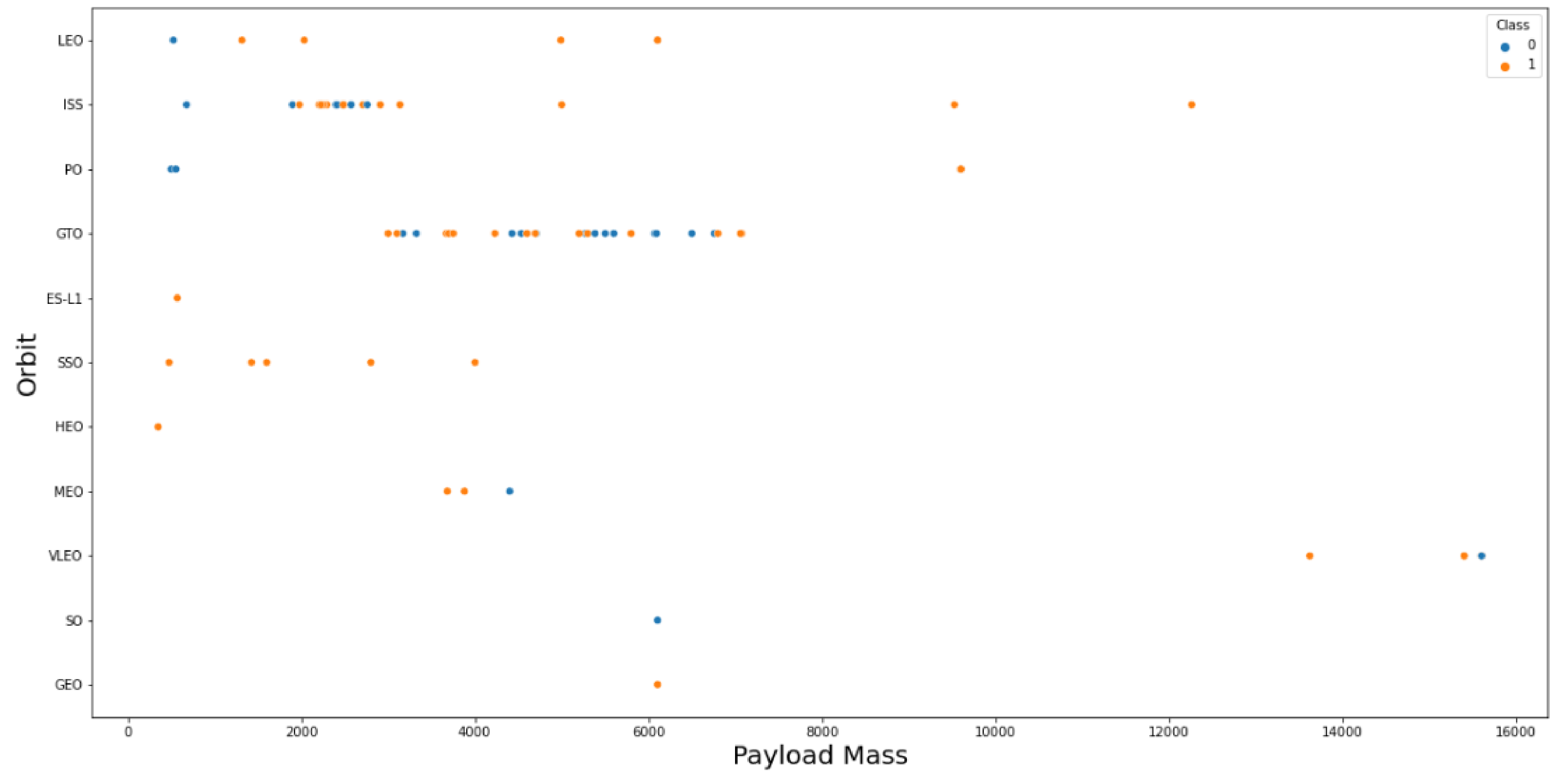
Flight Number vs. Orbit Type

- It is seen that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit. In SSO, HEO, ES-L1, and GEO orbit there are no failures.



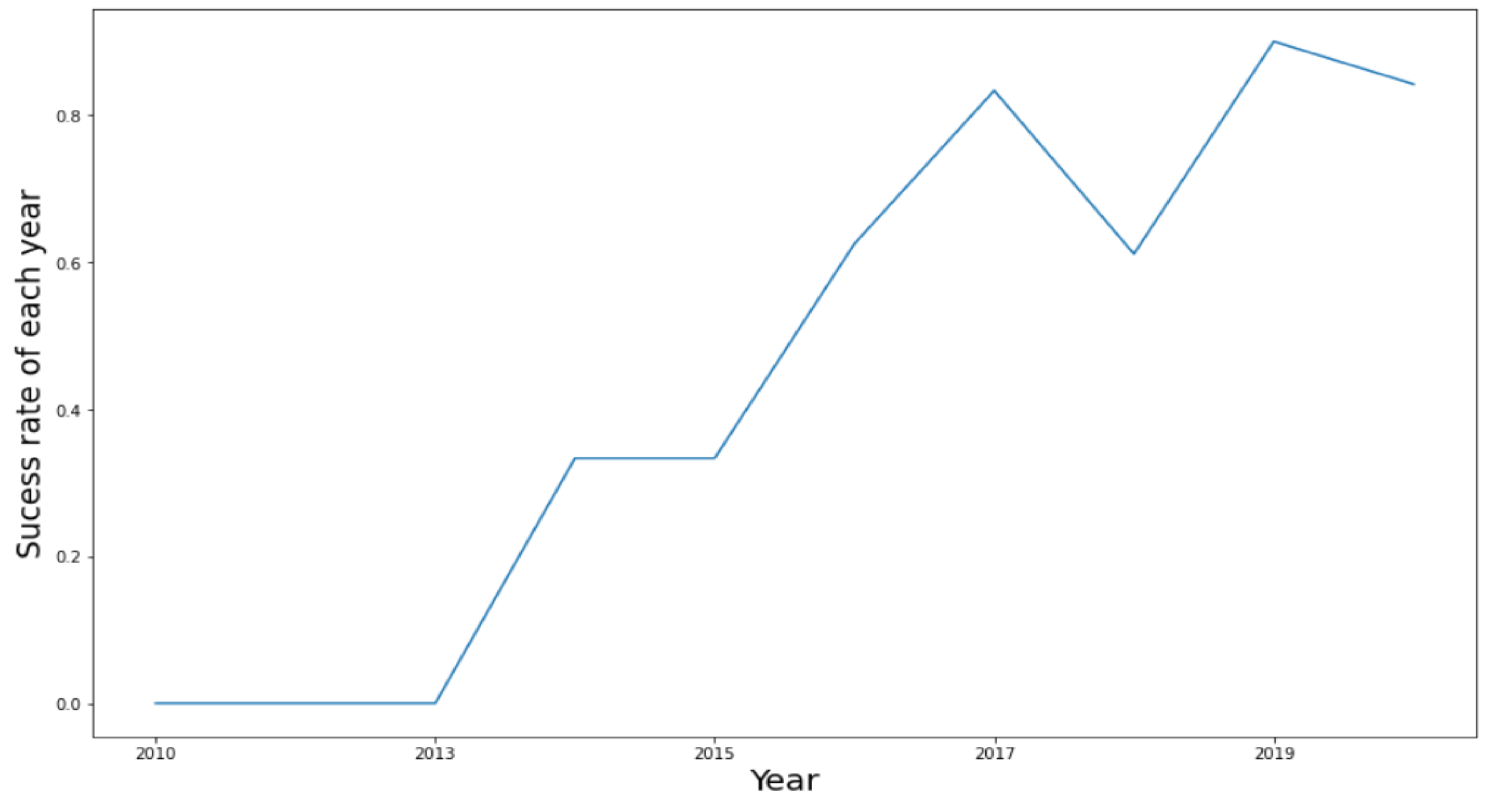
Payload vs. Orbit Type

- Heavy payloads have a positive impact on PO, LEO and ISS orbits in terms of chances of success.

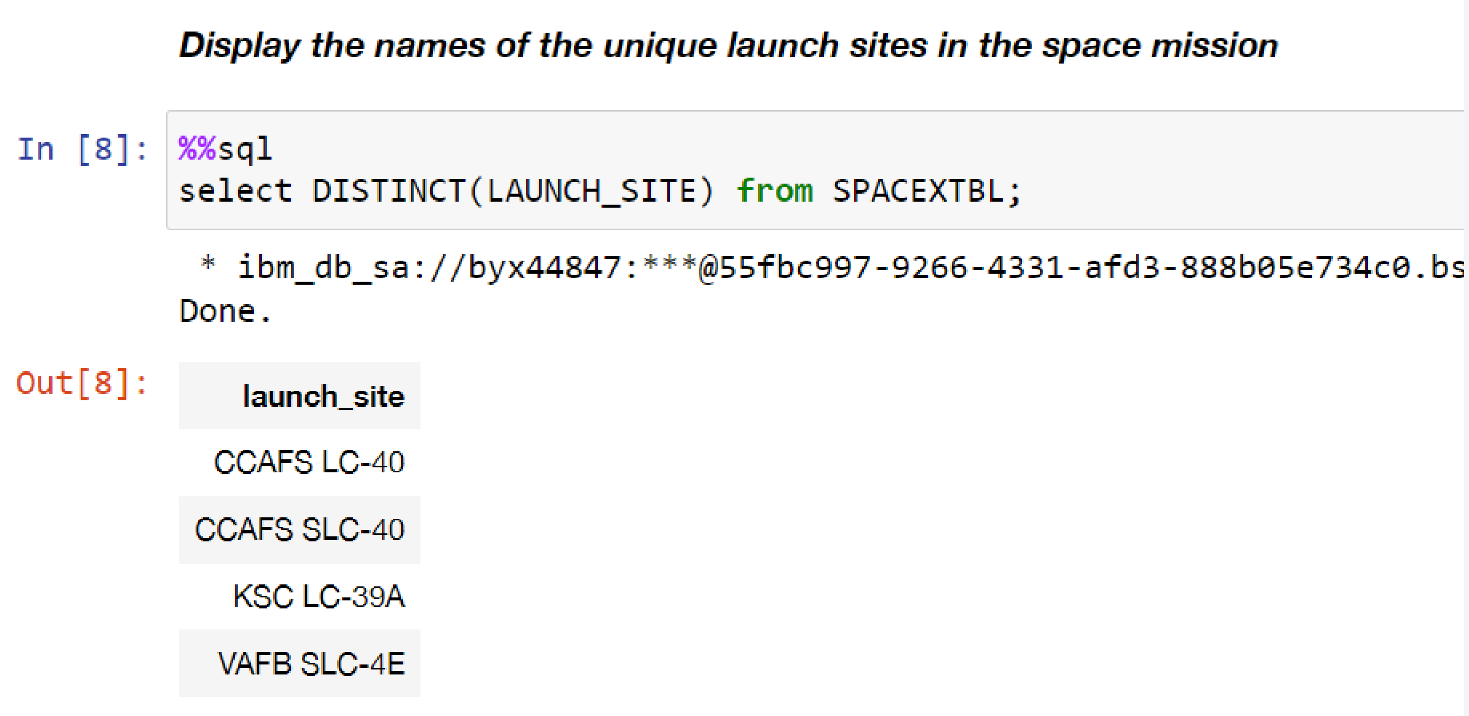


Launch Success Yearly Trend

- The success rate remained constant from 2010 to 2013 and increases from 2013 till 2020 only except 2018.



All Launch Site Names



Display the names of the unique launch sites in the space mission

```
In [8]: %%sql
        select DISTINCT(LAUNCH_SITE) from SPACEXTBL;
```

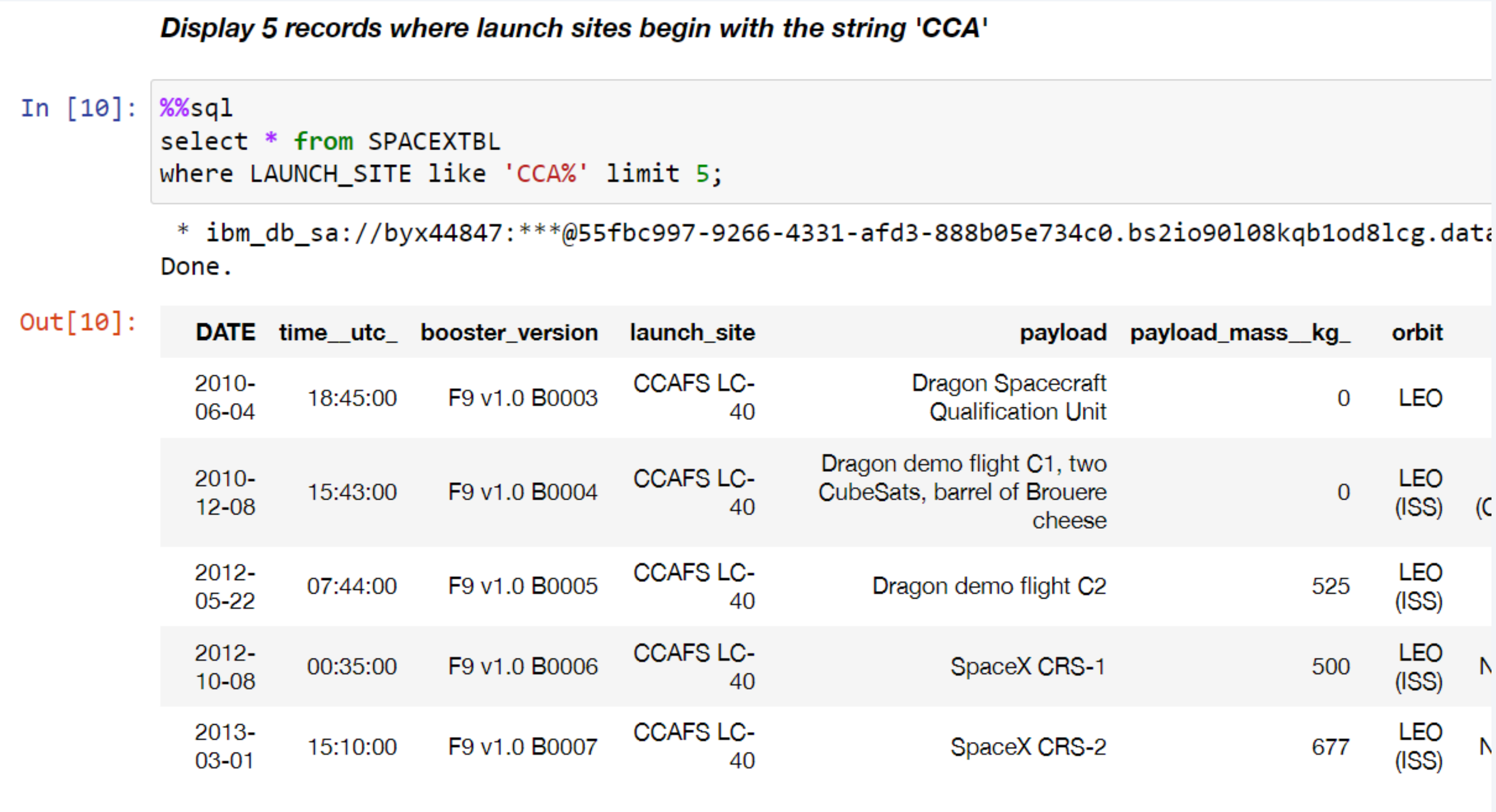
```
 * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs
Done.
```

Out[8]:

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

- DISTINCT is used in the query to list only the unique values of Launch_Site column

Launch Site Names Begin with 'CCA'



Display 5 records where launch sites begin with the string 'CCA'

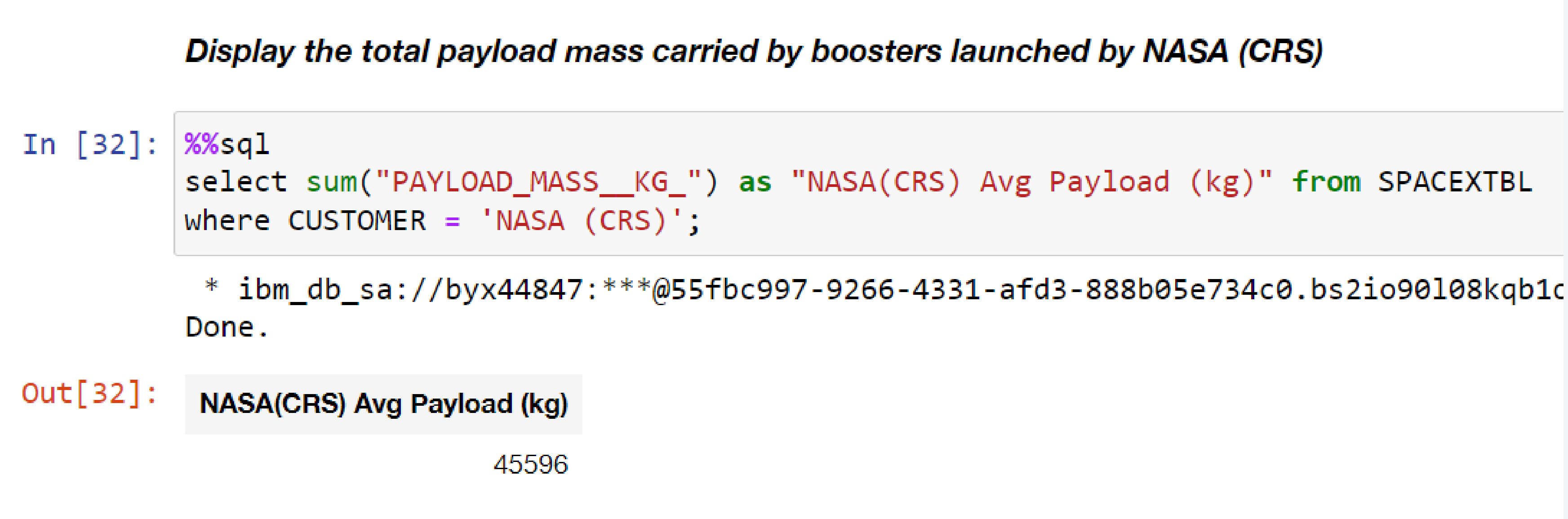
```
In [10]: %%sql
         select * from SPACEXTBL
         where LAUNCH_SITE like 'CCA%' limit 5;

          * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.data
         Done.
```

Out[10]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	
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)	(C
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	N

- We used the query above to display 5 records where launch sites begin with `CCA`. Here % is used as a wildcard. Limit operator restricts the number of entries fetched.

Total Payload Mass

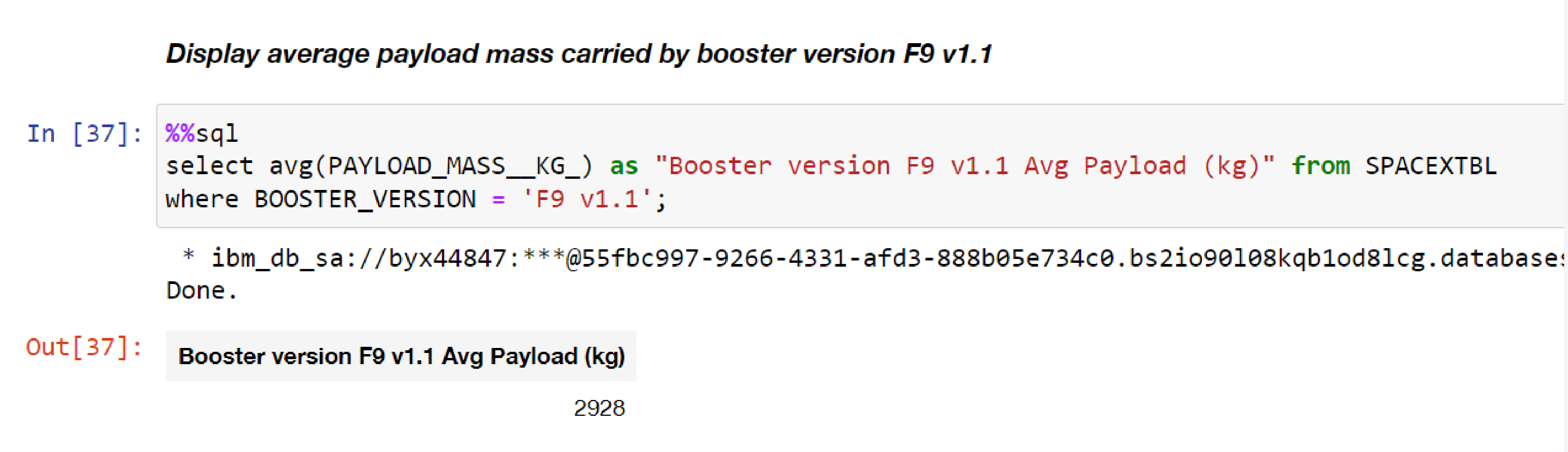


Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [32]: %%sql
         select sum("PAYLOAD_MASS__KG_") as "NASA(CRS) Avg Payload (kg)" from SPACEXTBL
         where CUSTOMER = 'NASA (CRS)';

          * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1c
         Done.

Out[32]: NASA(CRS) Avg Payload (kg)
                              45596
```

- Using the function SUM adds the column PAYLOAD_MASS_KG_; The WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS); AS clause is used to rename the column name of the output.

Average Payload Mass by F9 v1.1



Display average payload mass carried by booster version F9 v1.1

In [37]:

```
%%sql
select avg(PAYLOAD_MASS__KG_) as "Booster version F9 v1.1 Avg Payload (kg)" from SPACEXTBL
where BOOSTER_VERSION = 'F9 v1.1';
```

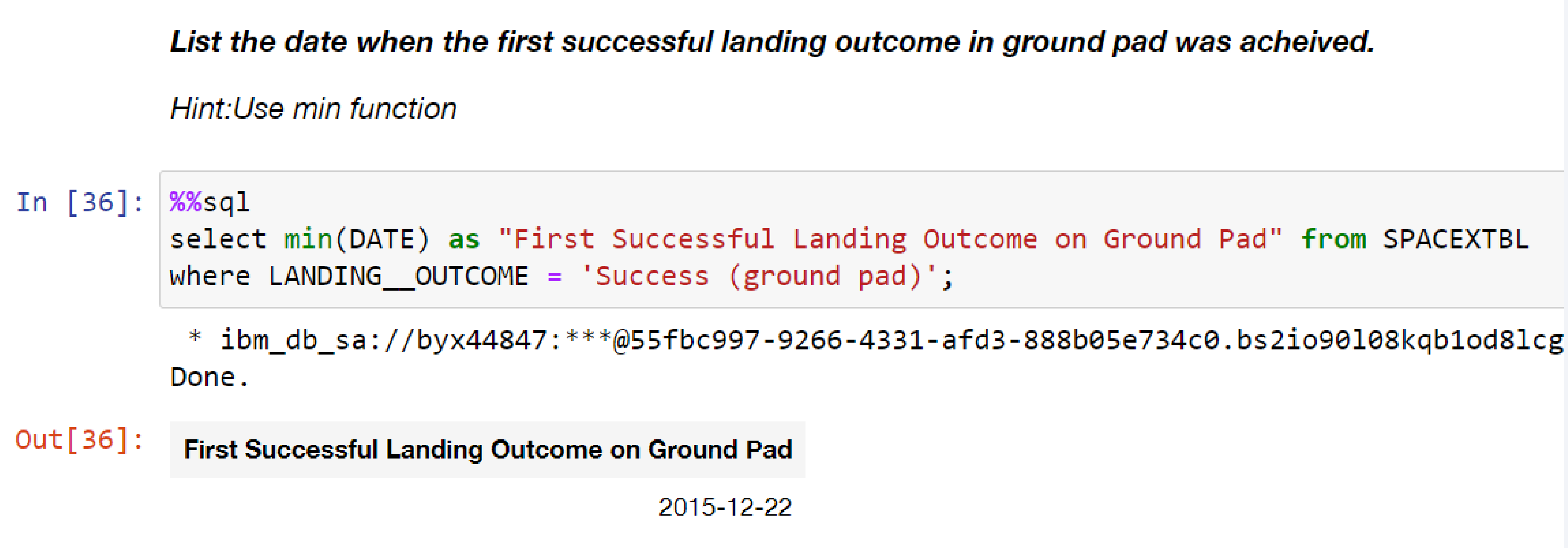
```
 * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases
Done.
```

Out[37]:

Booster version F9 v1.1 Avg Payload (kg)
2928

- The function AVG determines the average of the column PAYLOAD_MASS_KG_; WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

First Successful Ground Landing Date



List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
In [36]: %%sql
         select min(DATE) as "First Successful Landing Outcome on Ground Pad" from SPACEXTBL
         where LANDING__OUTCOME = 'Success (ground pad)';
```

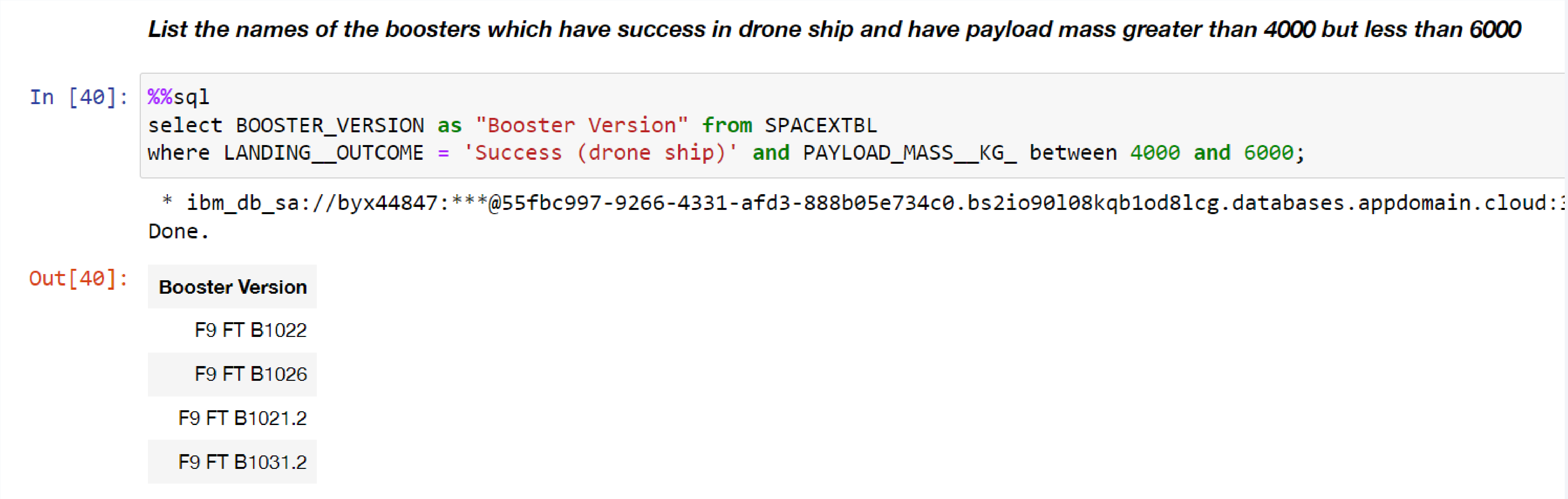
```
 * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg
Done.
```

Out[36]:

First Successful Landing Outcome on Ground Pad
2015-12-22

- The function MIN works determines the earliest date in the column Date; WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [40]:
```
%%sql
select BOOSTER_VERSION as "Booster Version" from SPACEXTBL
where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

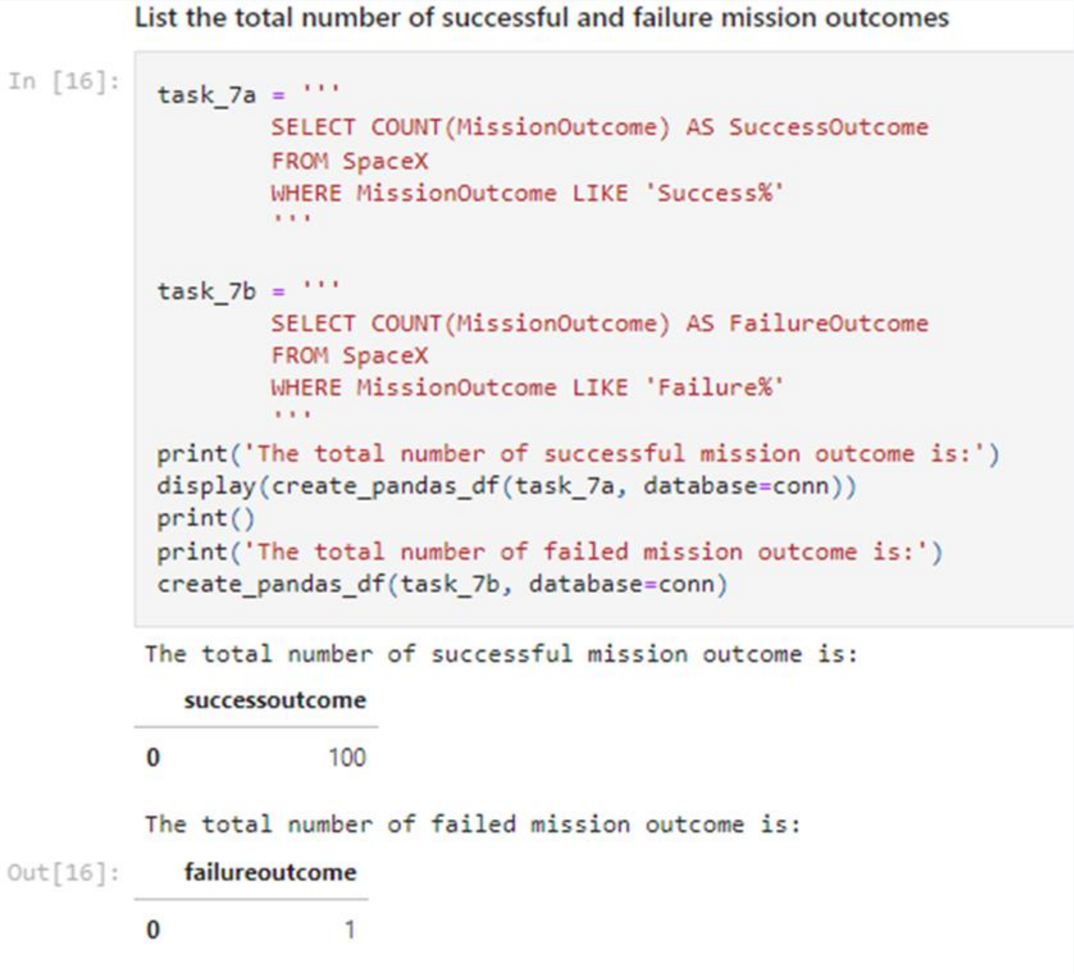
```
 * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:
Done.
```

Out[40]:

Booster Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- The WHERE clause filters the dataset to Landing_Outcome = Success (drone ship); The AND clause specifies additional filter conditions Payload_MASS_KG_ between 4000 AND 6000

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
In [16]:
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

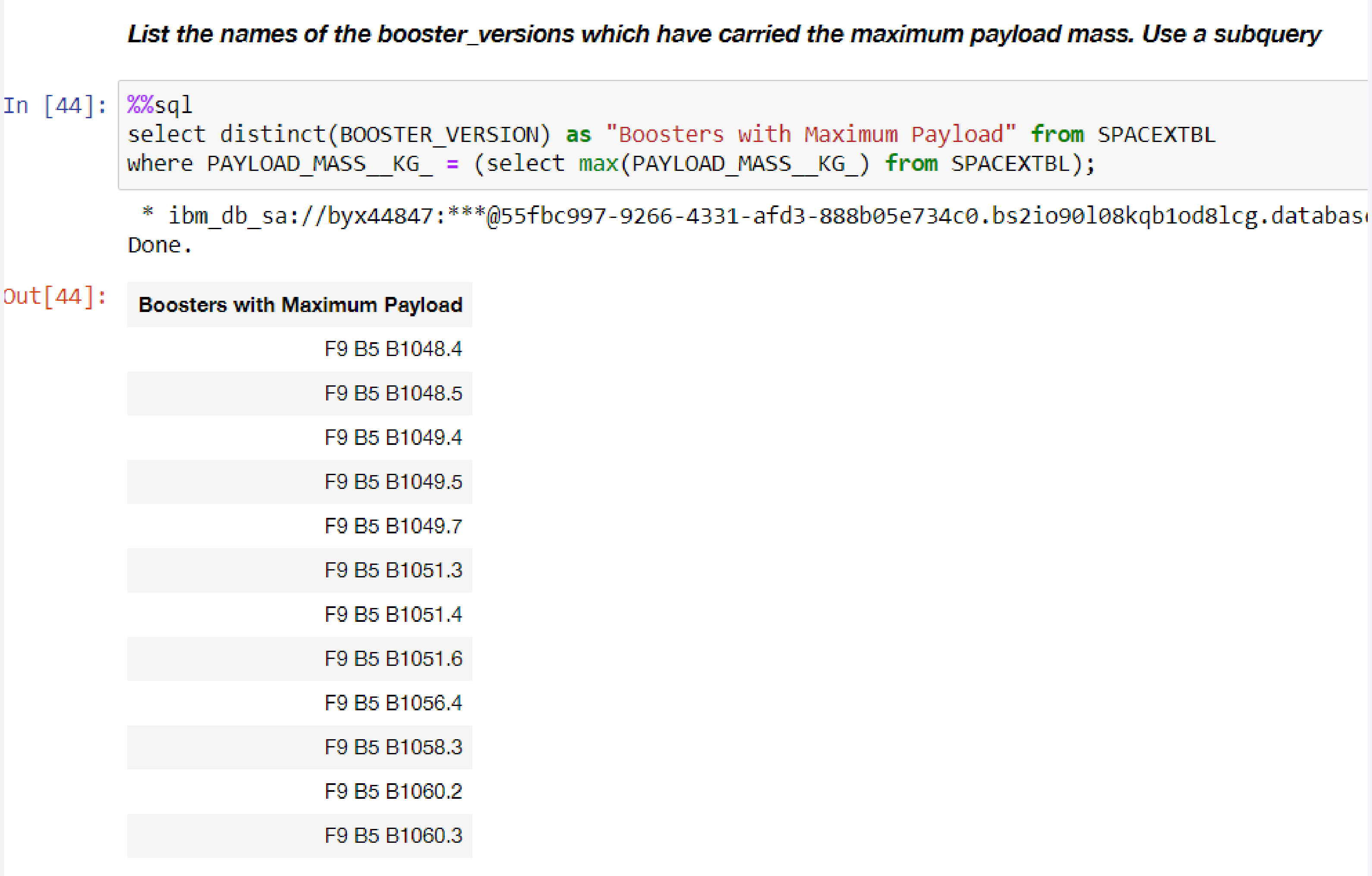
```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

- This statement determines the total success and failure mission outcomes

Boosters Carried Maximum Payload



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

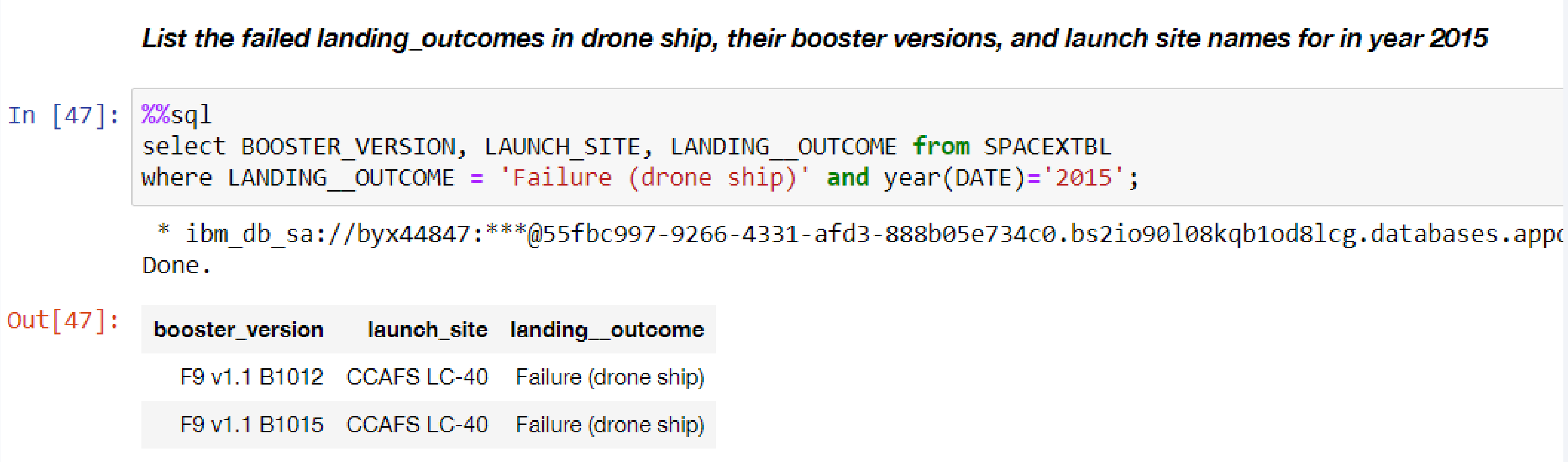
```
In [44]: %%sql
         select distinct(BOOSTER_VERSION) as "Boosters with Maximum Payload" from SPACEXTBL
         where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);

          * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databas
         Done.
```

Out[44]:

Boosters with Maximum Payload
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

- Determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 Launch Records



List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

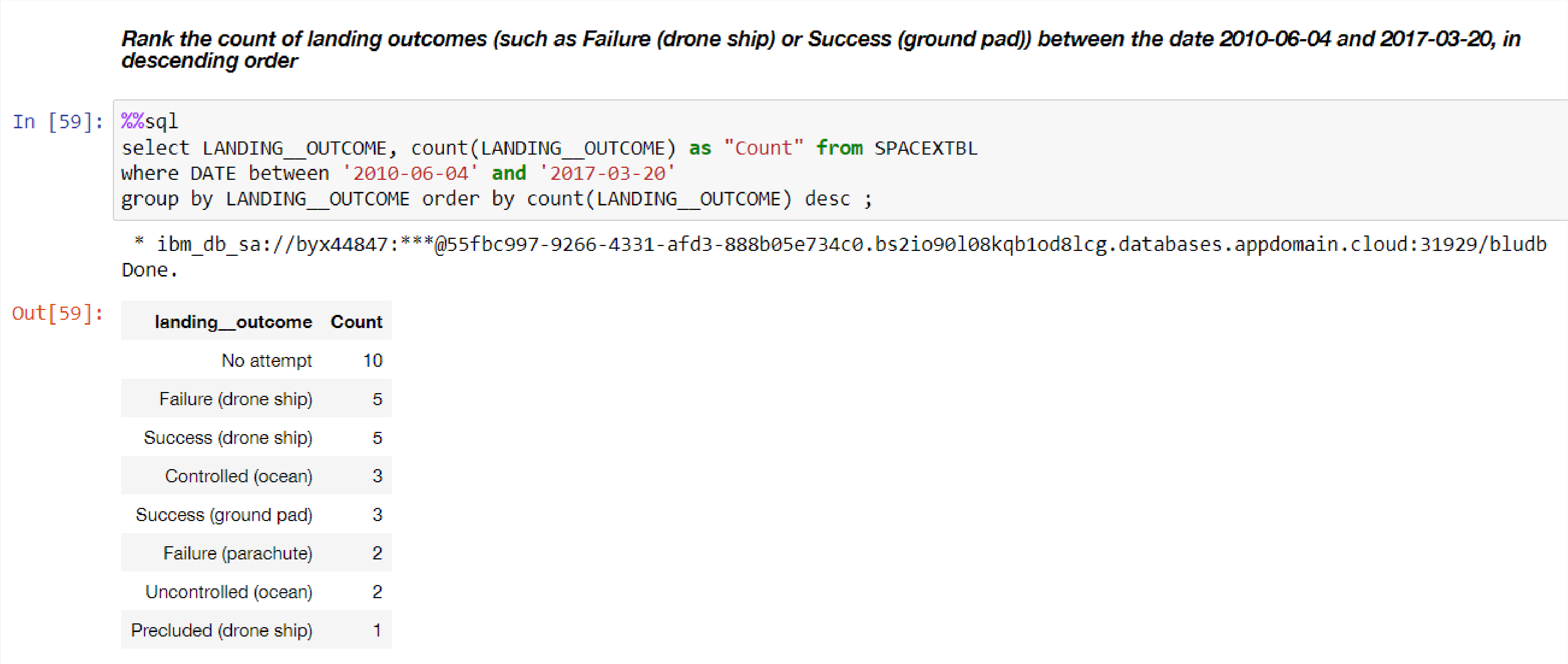
```
In [47]: %%sql
         select BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME from SPACEXTBL
         where LANDING__OUTCOME = 'Failure (drone ship)' and year(DATE)='2015';

          * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appd
         Done.
```

Out[47]:

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

- Used a combination of **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

```
In [59]: %%sql
         select LANDING__OUTCOME, count(LANDING__OUTCOME) as "Count" from SPACEXTBL
         where DATE between '2010-06-04' and '2017-03-20'
         group by LANDING__OUTCOME order by count(LANDING__OUTCOME) desc ;

 * ibm_db_sa://byx44847:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[59]:

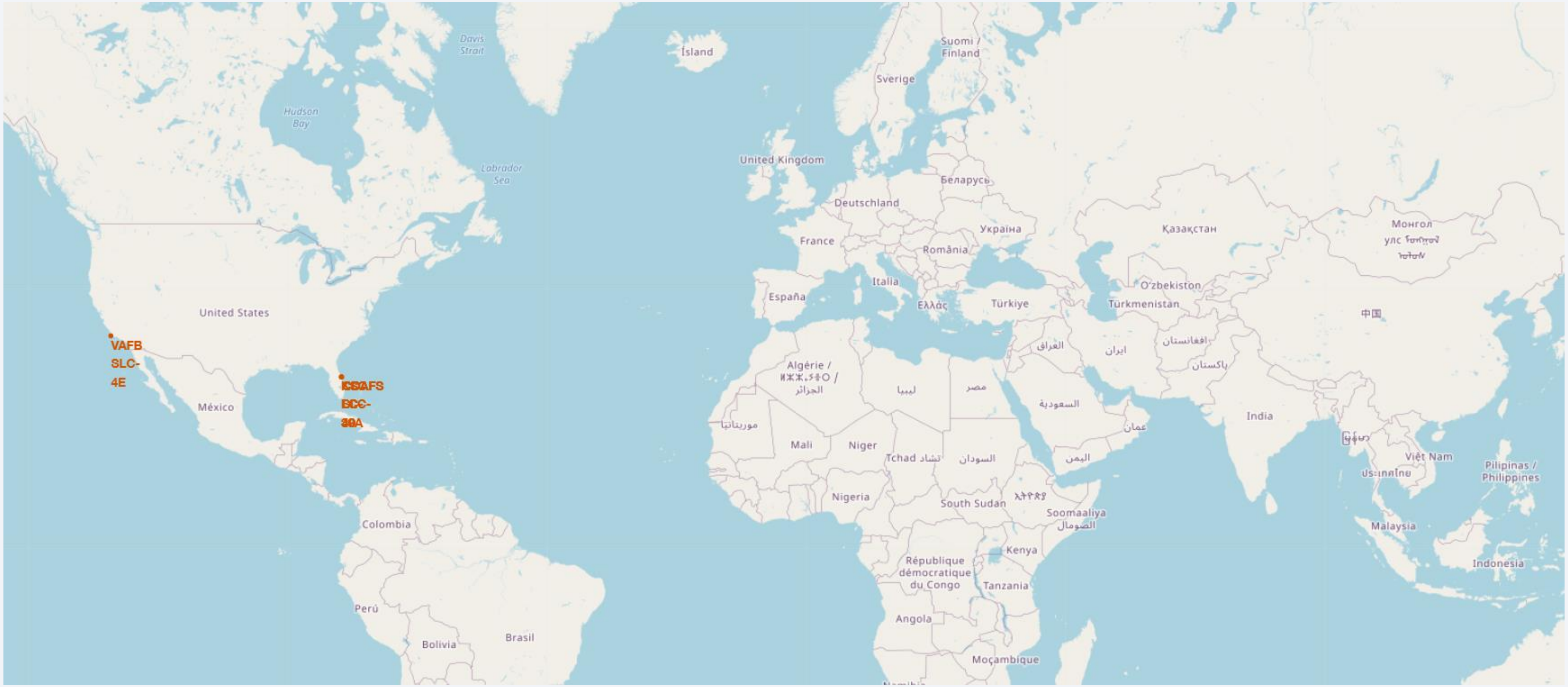
landing__outcome	Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- Selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2017-03-20. Applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY & DESC** clause to order the grouped landing outcome in descending order.

Section 3

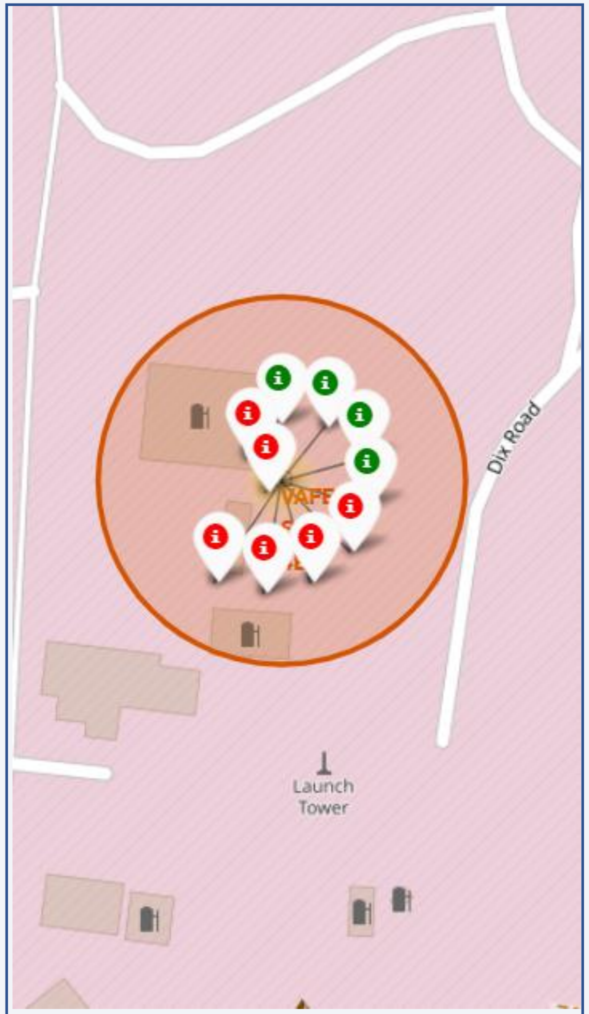
Launch Sites Proximities Analysis

All launch sites global map markers

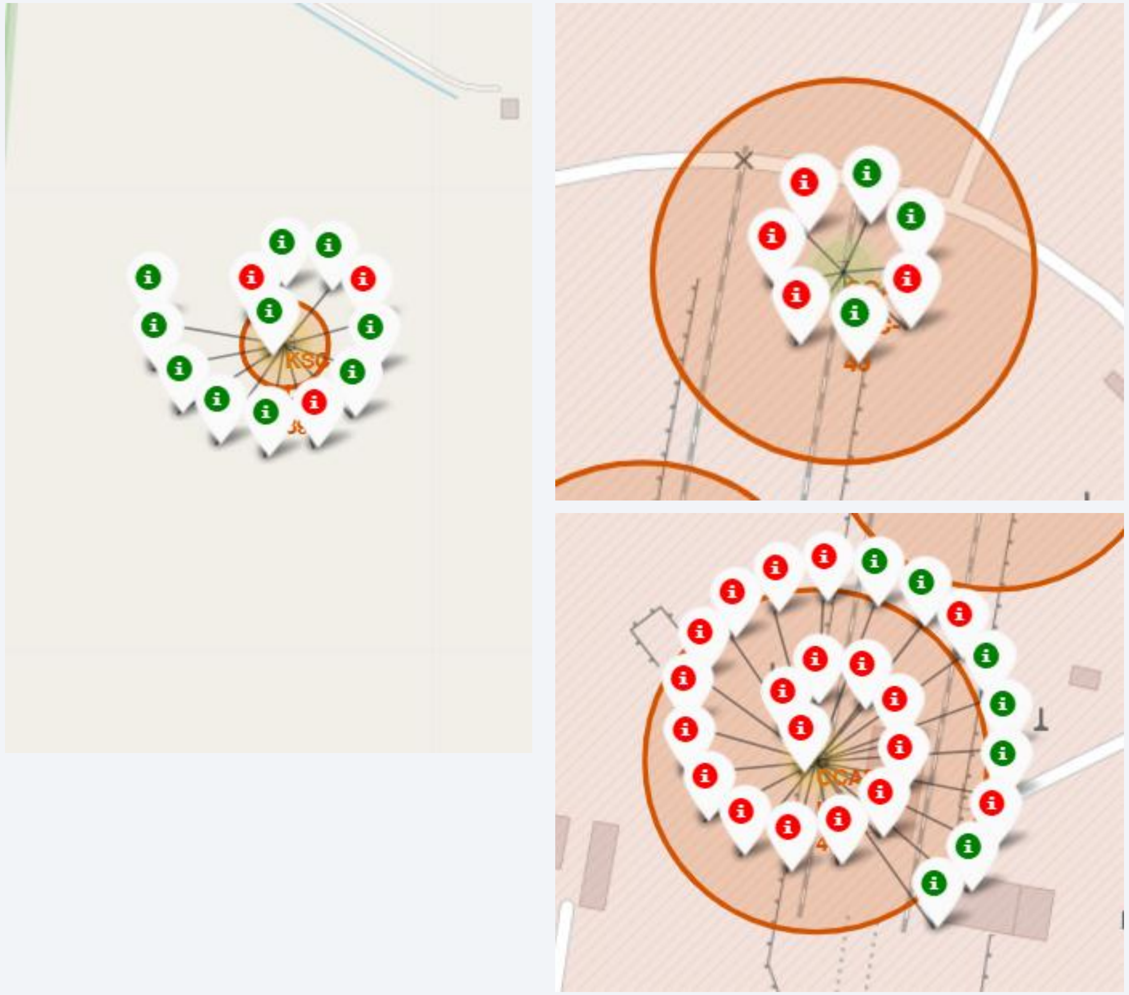


- All the SpaceX launch sites are located on the southern USA and are located in close proximity to beaches

Markers showing launch sites with color labels



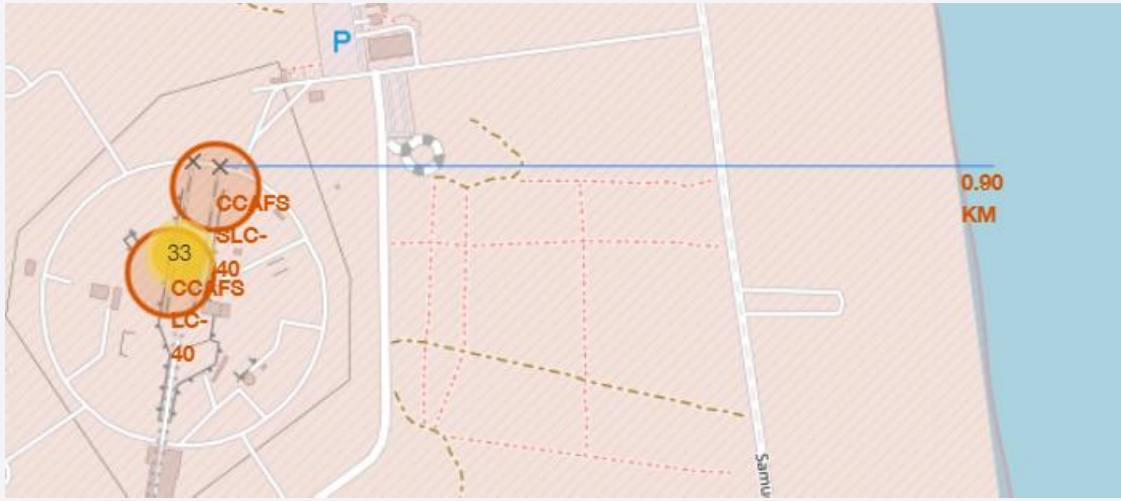
- California Launch Sites

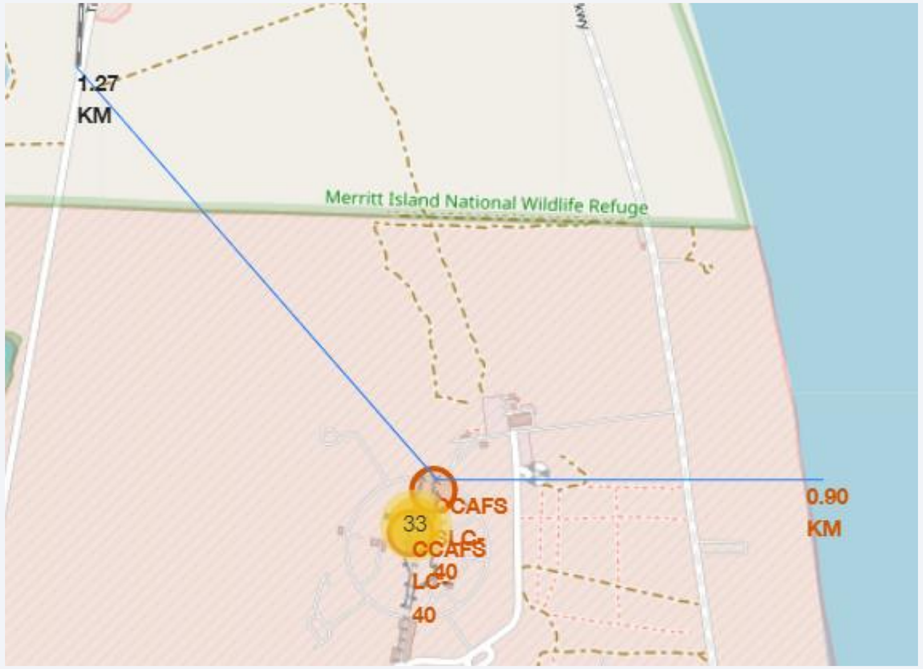


- Florida Launch Sites

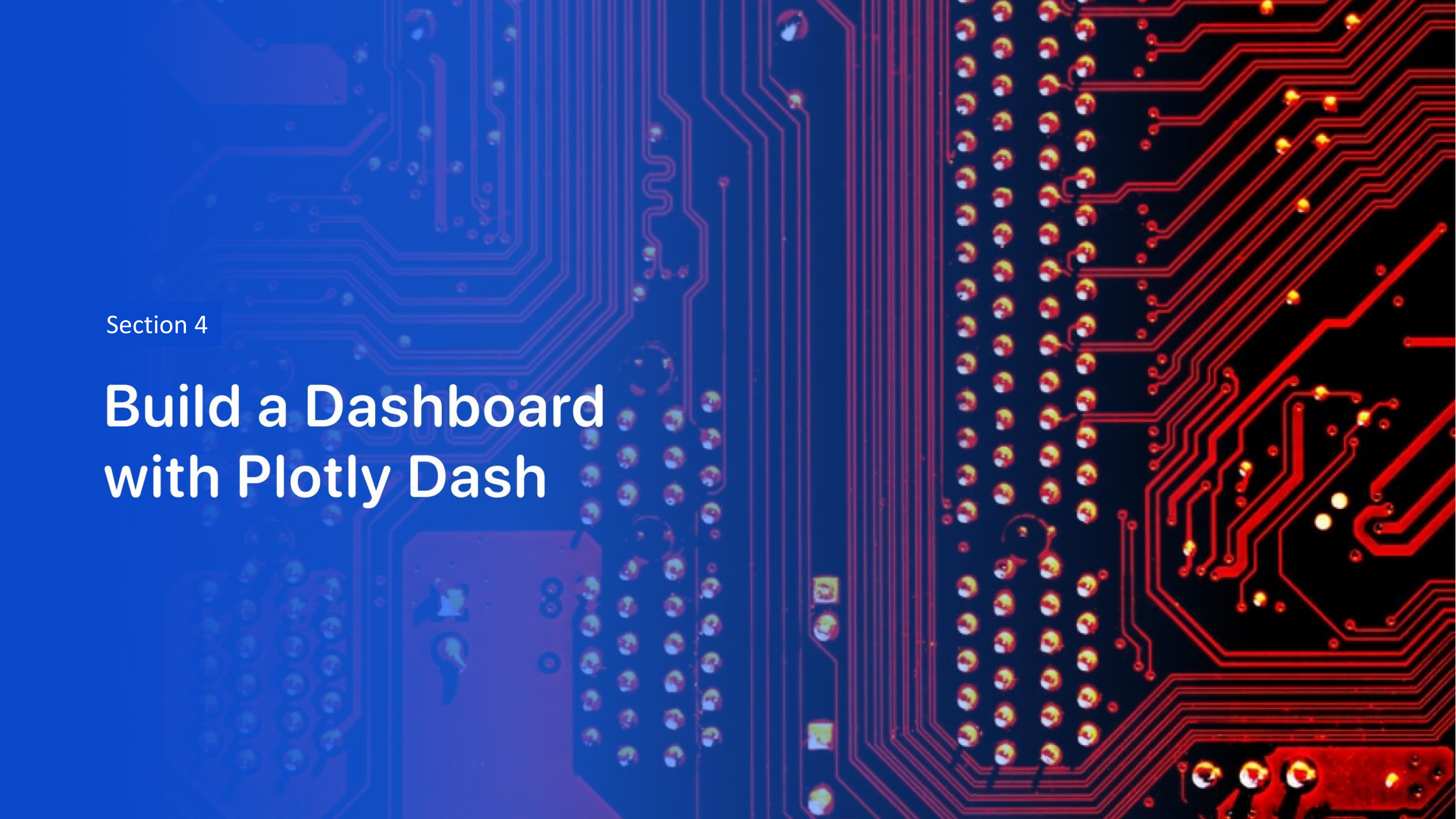
- The green indicators show successful landings and red indicators show failed landings.
- It is observed that the lunch sites follow a spiral pattern in 3 out of 4 locations.

Launch Site distance to landmarks





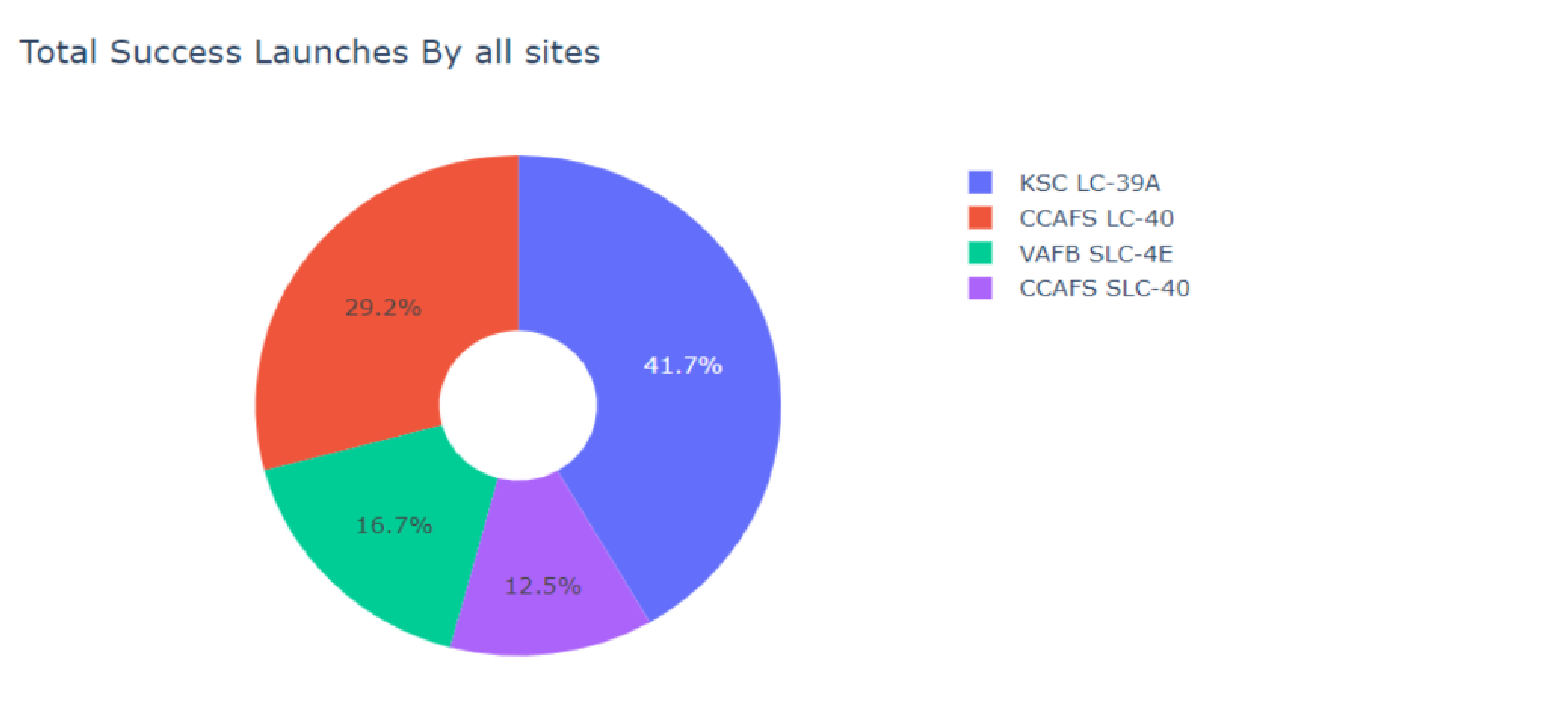
- The above images show distances to coast and nearest railway line for the florida launch site.



Section 4

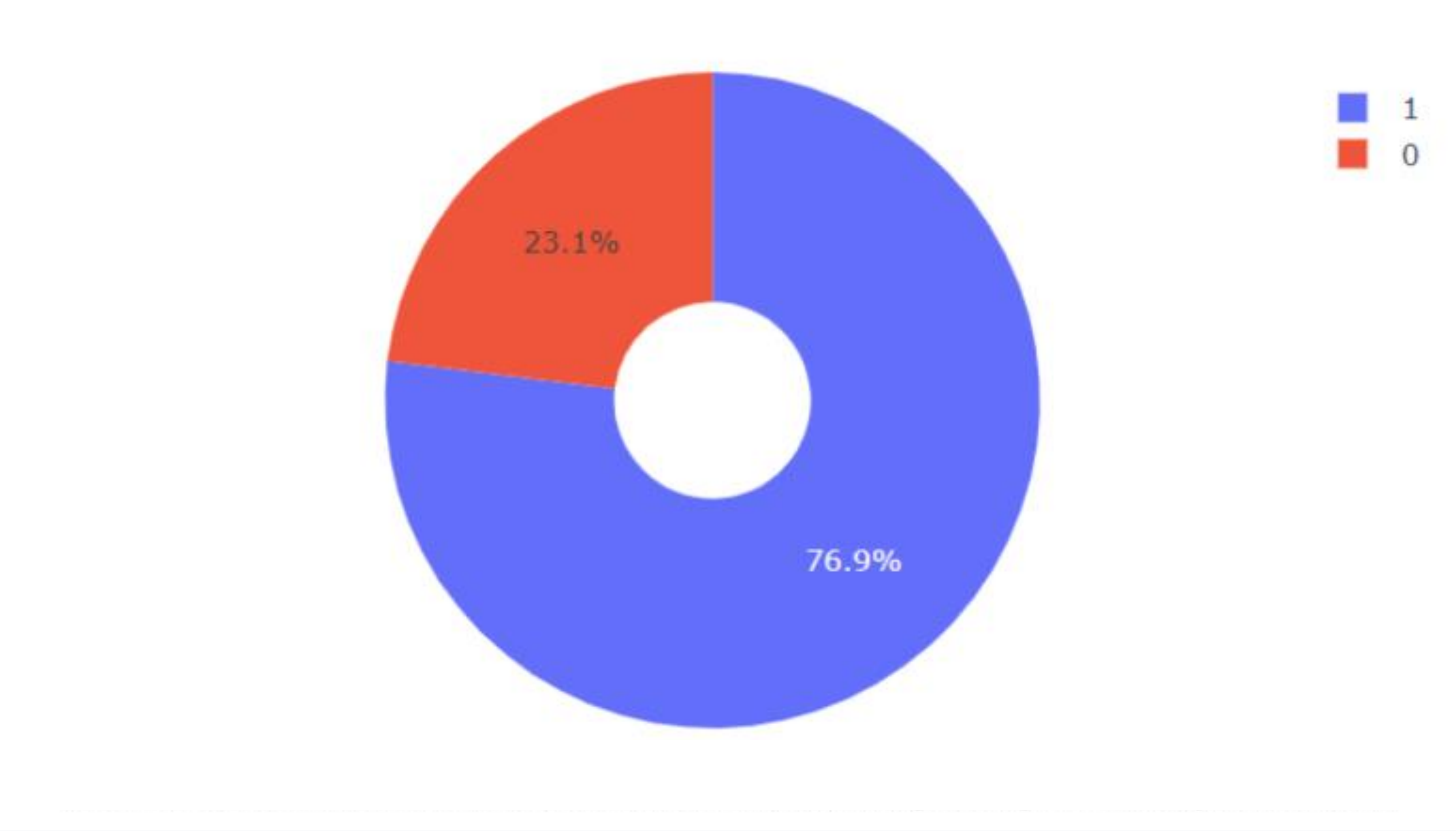
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



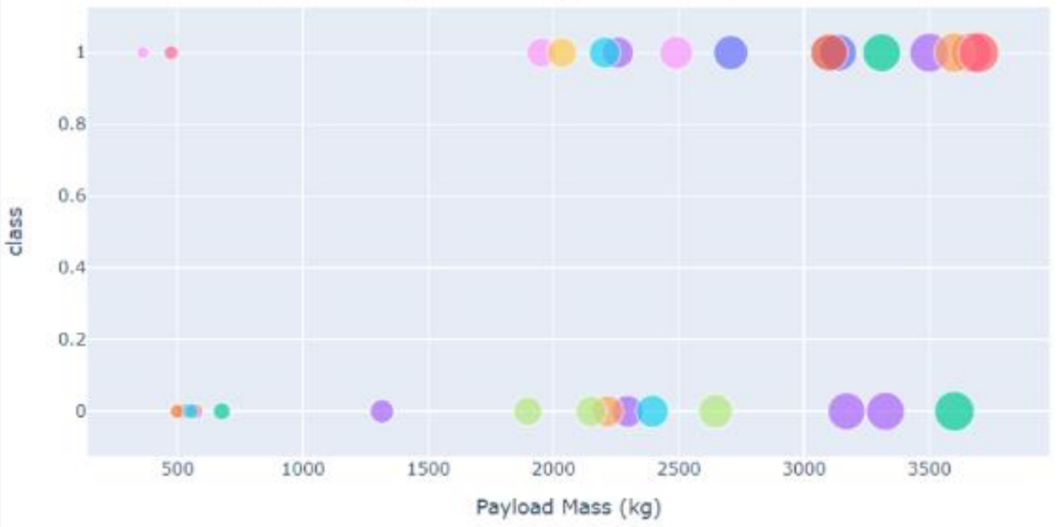
- KSC LC-39A has the most successful launches and CCAFS SLC-40 has the least successful launches

Pie chart showing the Launch site with the highest launch success ratio

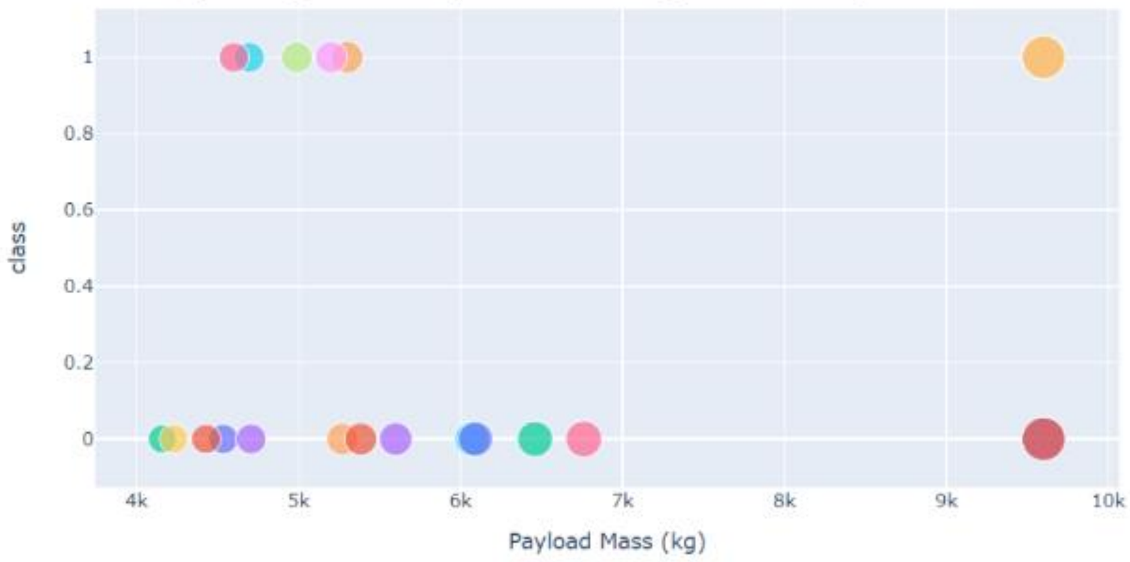


- KSC LC-39A has 76.9% success rate and a mere 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites,



Low Weighted Payload 0kg – 4000kg

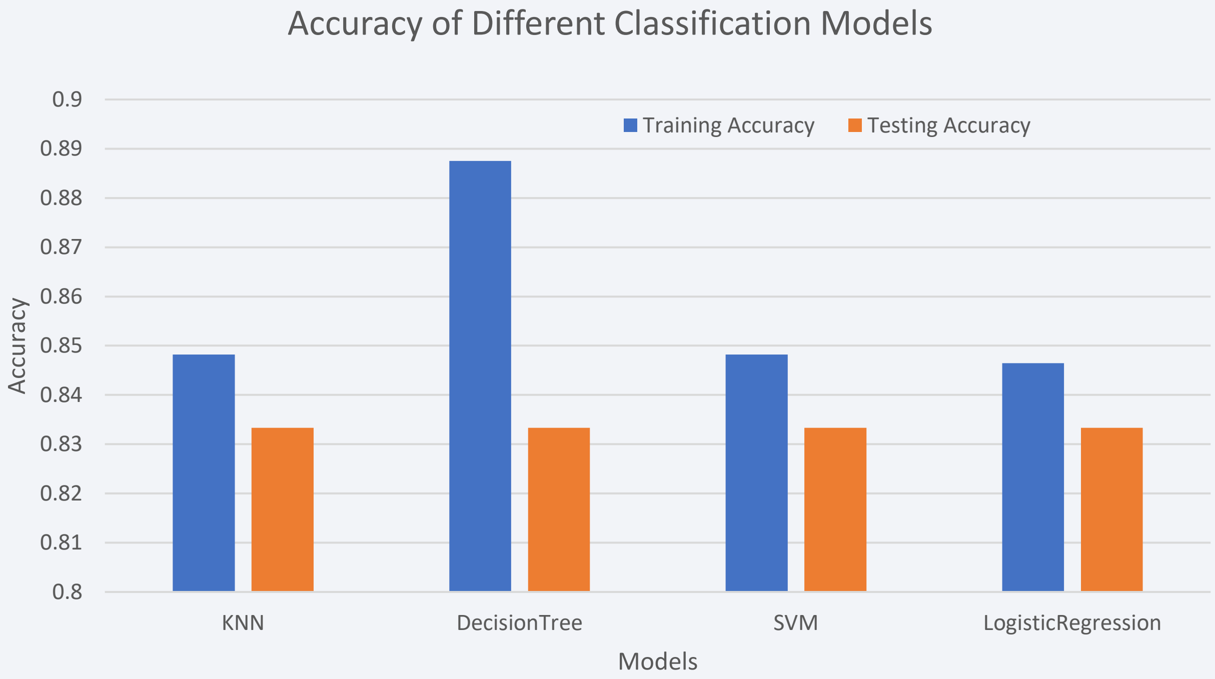


Heavy Weighted Payload 4000kg – 10000kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

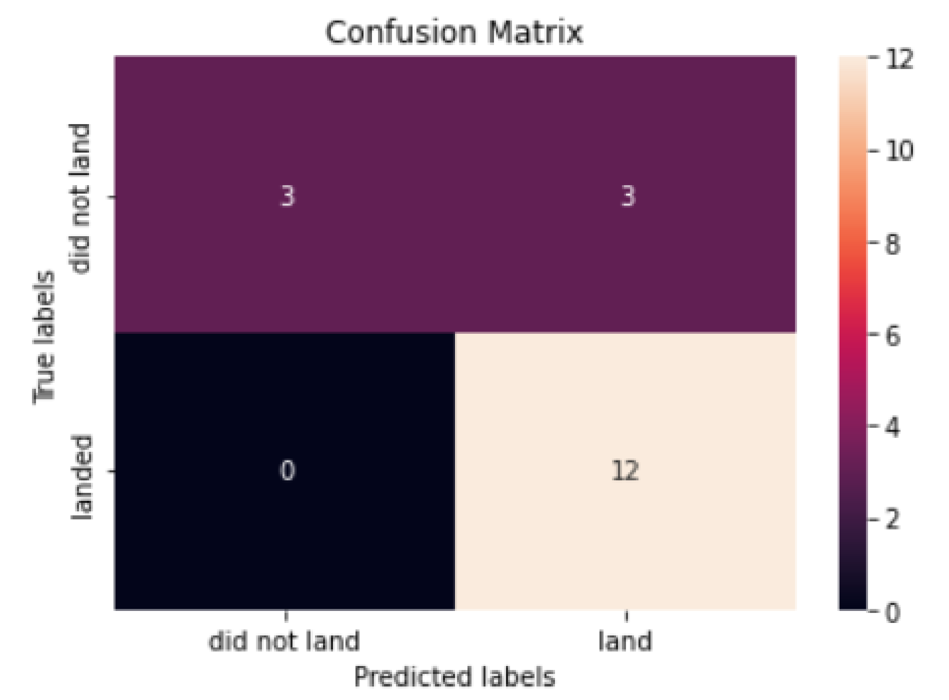


Algorithm	Training Accuracy	Testing Accuracy
KNN	0.84821	0.83333
DecisionTree	0.8875	0.83333
SVM	0.84821	0.83333
LogisticRegression	0.84643	0.83333

- Decision Tree Classifier has highest training accuracy of 88.75%.
- All models have same testing accuracy of 83.3%.

Confusion Matrix

- Confusion Matrix for all four models is same.
- The problem with the defined models is that it has 3 values in false positives.



Conclusions

- The Decision Tree Classifier Algorithm is the best for Machine Learning algorithm for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is increasing with time in years and eventually they will perfect the launches. Launch success rate started to increase in 2013 till 2020.
- KSC LC-39A had the most successful launches from all the sites.
- Orbit GEO, HEO, SSO, ES-L1 has the best Success Rates.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!