IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Danial Jafari
16-April-2024

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection with SpaceX API
  - Data Collection with Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA with Data Visualization
  - Build an Interactive Map with Folium
  - Build a Dashboard with Plotly Dash
  - Predictive Analysis
- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

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 you want to find answers
  - predict if the Falcon 9 first stage will land successfully.
  - What features are important for this prediction.
  - The relationship of these features with each other.
  - Chekh model has the best prediction.

Section 1

Methodology

Methodology

Executive Summary

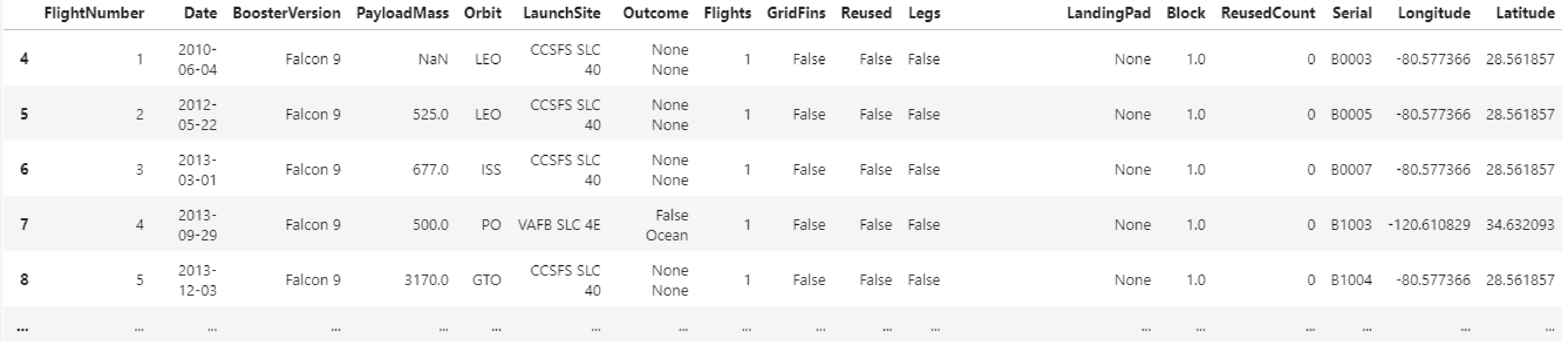
- Data collection methodology:
    - Data was collected from SpaceX API and Wikipedia
- Perform data wrangling
    - Perform exploratory Data Analysis and determine Training Labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

Data Collection

- 1. Web Scraping
  - Extract data from SpaceX API and Wikipedia using web scraping tools.
- 2. Data Integration and Cleaning
  - Combine the data from the various sources.
- 3. Data Storage in pandas data frame

Data Collection – SpaceX API

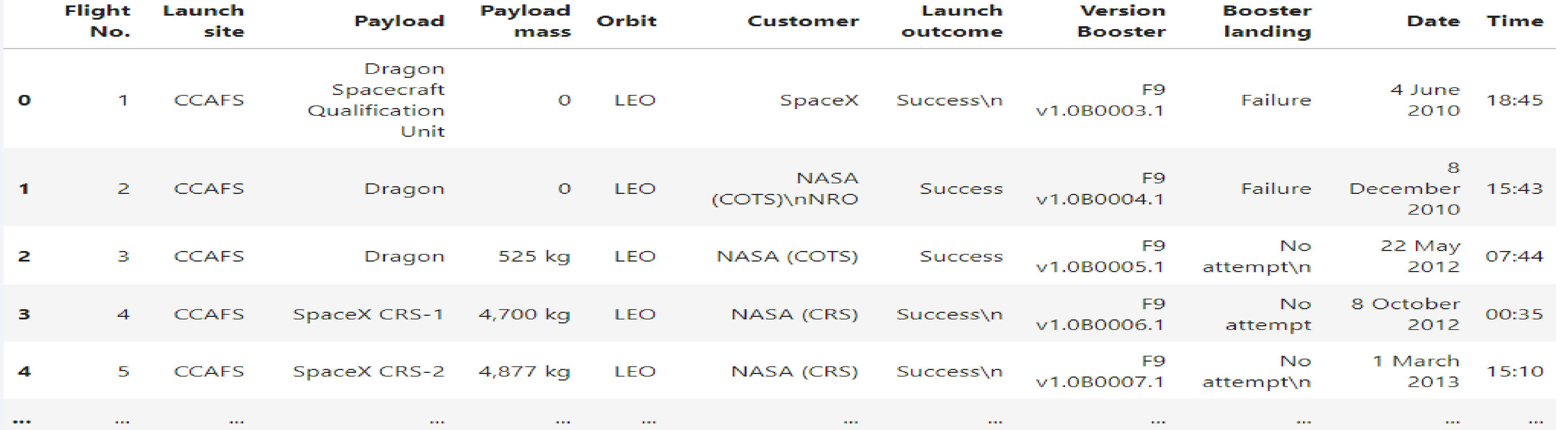
- 1. Identify the relevant API endpoints, obtain the necessary authentication credentials, and use Python's requests library to retrieve data from the SpaceX API.
- 2. Extract the JSON response data into Pandas DataFrames, clean and preprocess the data to handle any issues such as missing values or formatting problems.
- 3. Analyze the cleaned data, generate insights and visualizations, and store the processed data for future reference and further analysis.
- GitHub URL: https://github.com/djafari700/-applied-data-science-capstone/blob/main/1-jupyter-labs-spacex-data-collection-api.ipynb



	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Data Collection - Scraping

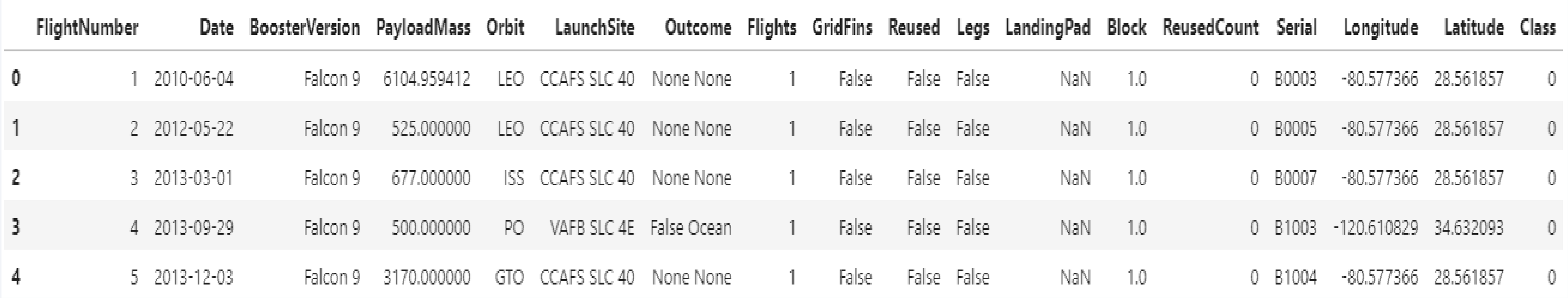
- Employ web scraping libraries like BeautifulSoup to extract relevant data from Wikipedia pages, parse the HTML content, and store the information in structured Pandas DataFrames for further analysis alongside the SpaceX API data.
- GitHub URL:https://github.com/djafari700/-applied-data-science-capstone/blob/main/2-jupyter-labs-webscraping.ipynb



	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...

Data Wrangling

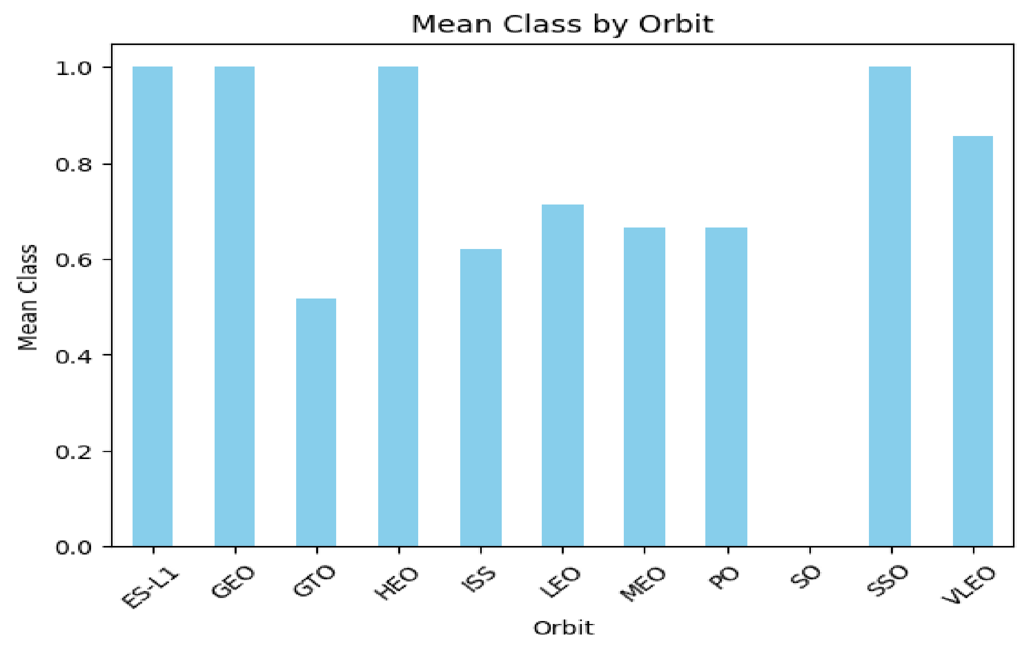
- Ingest the data from the SpaceX API and Wikipedia web scraping, clean and preprocess the datasets (handle missing values, data types, and formatting), combine the sources into a unified Pandas DataFrame, and perform exploratory data analysis to derive insights and visualizations.
- Conducted exploratory analysis to determine training labels.
- Calculated launch site and orbit metrics across the data.
- Engineered a landing outcome label and exported the processed results.
- GitHub URL:https://github.com/djafari700/-applied-data-science-capstone/blob/main/3-labs-jupyter-spacex-Data%20wrangling.ipynb

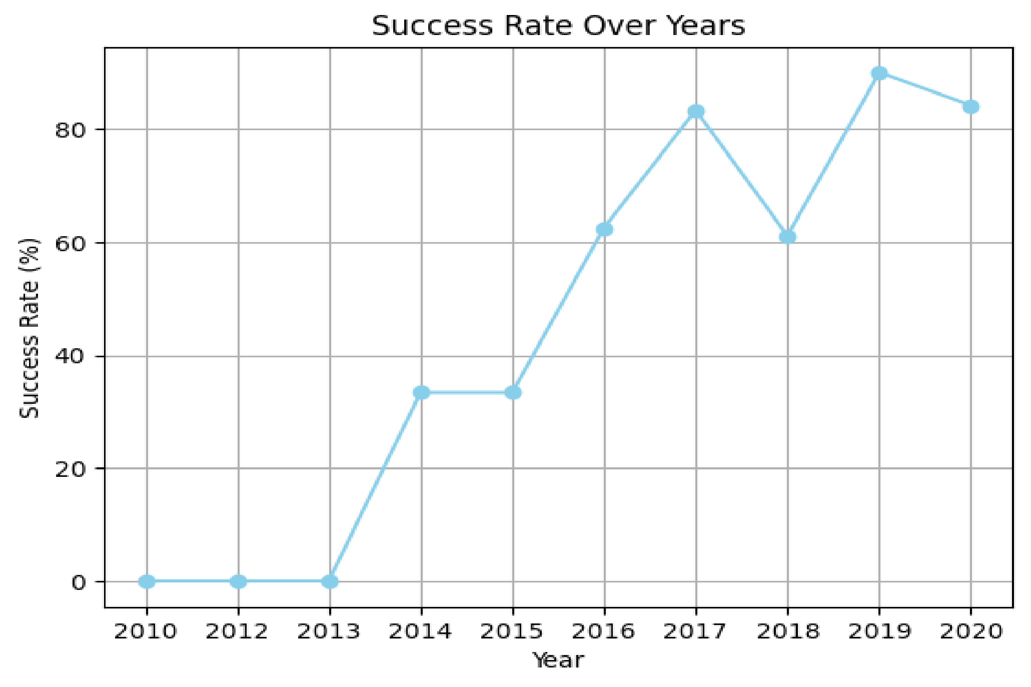


	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

EDA with Data Visualization

- scatter plots to visualize relationships between variables and bar charts to compare categorical data, such as launch site metrics and orbital patterns, providing valuable insights into the SpaceX launch data.





- GitHub: https://github.com/djafari700/-applied-data-science-capstone/blob/main/4-edadataviz.ipynb

EDA with SQL

- - Identified unique launch sites
- - Filtered records by launch site prefix
- - Calculated total payload mass for NASA CRS missions
- - Determined average payload mass for F9 v1.1 boosters
- - Found date of first successful landing
- - Queried boosters with successful drone landings in 4-6K kg range
- - Counted successful vs. failure mission outcomes
- - Identified boosters with maximum payload mass
- - Retrieved failed drone landing records in 2015
- - Ranked landing outcomes between 2010-2017
- GitHub URL: https://github.com/djafari700/-applied-data-science-capstone/blob/main/5-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers for launch site locations
- Circles to indicate 100-mile coverage areas
- Lines connecting launch sites to landing sites
- Popups with launch site details
- These map objects were added to provide a comprehensive geospatial visualization of the SpaceX launch sites and their landing outcomes, enabling spatial analysis and identification of patterns.
- GitHub URL: https://github.com/djafari700/-applied-data-science-capstone/blob/main/6-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- - Bar, pie, scatter, and line charts to visualize launch success rates, site locations, and historical trends
- - Dropdown menu and hover tooltips for interactive exploration of launch site details
- - Filtering to focus on successful or failed launches
- These plots and interactions enable comprehensive analysis of the SpaceX launch data, supporting data-driven decision making and insights.
- GitHub URL:https://github.com/djafari700/-applied-data-science-capstone/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

- 1. Data Preprocessing: Cleaned and prepared the data by handling missing values and encoding categorical variables.
- 2. Model Selection: Chose potential models such as logistic regression, decision trees, and random forests for evaluation.
- 3. Model Training and Evaluation: Trained each model on the training data and evaluated their performance using accuracy and precision.
- 4. Model Improvement: Fine-tuned the selected model by adjusting hyperparameters and conducting feature selection.
- 5. Cross-Validation: Utilized k-fold cross-validation to ensure the model's generalization performance.
- 6. Selection of Best Model: Selected the best performing model based on evaluation metrics and cross-validation results.
- URL:https://github.com/djafari700/-applied-data-science-capstone/blob/main/8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- 1. **Exploratory Data Analysis Results**: Analyzed the dataset to understand its characteristics, including feature distributions, relationships, and patterns.

- 2. **Interactive Analytics Demo in Screenshots**: Created an interactive demo allowing users to explore data insights, and captured screenshots to showcase the interactive features and key findings.

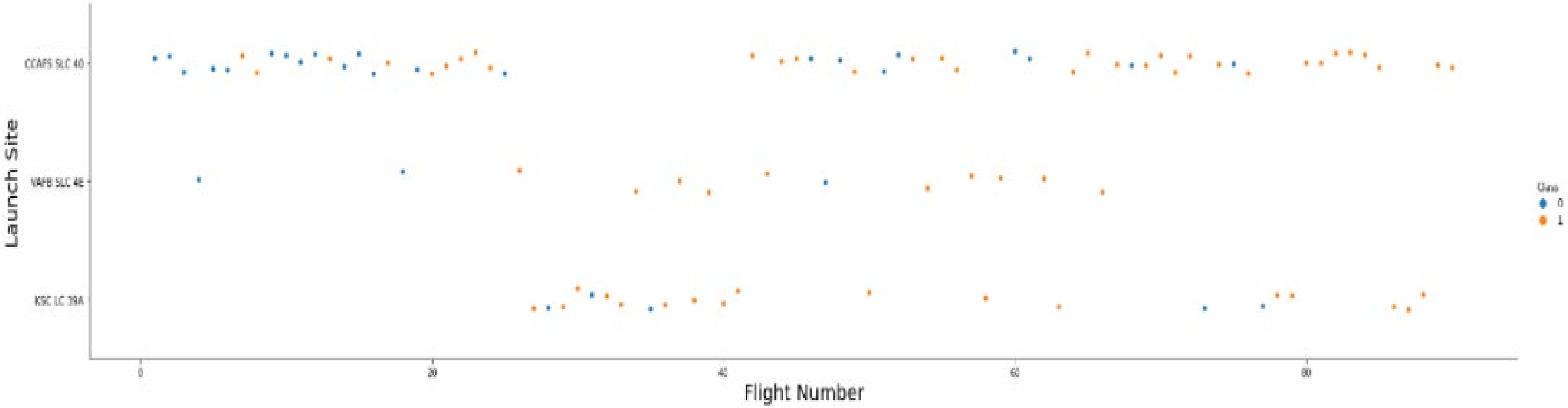
- 3. **Predictive Analysis Results**: Utilized predictive modeling to generate and evaluate predictions, using metrics such as accuracy, precision, recall, and F1 score to assess model performance.

Section 2

Insights drawn from EDA

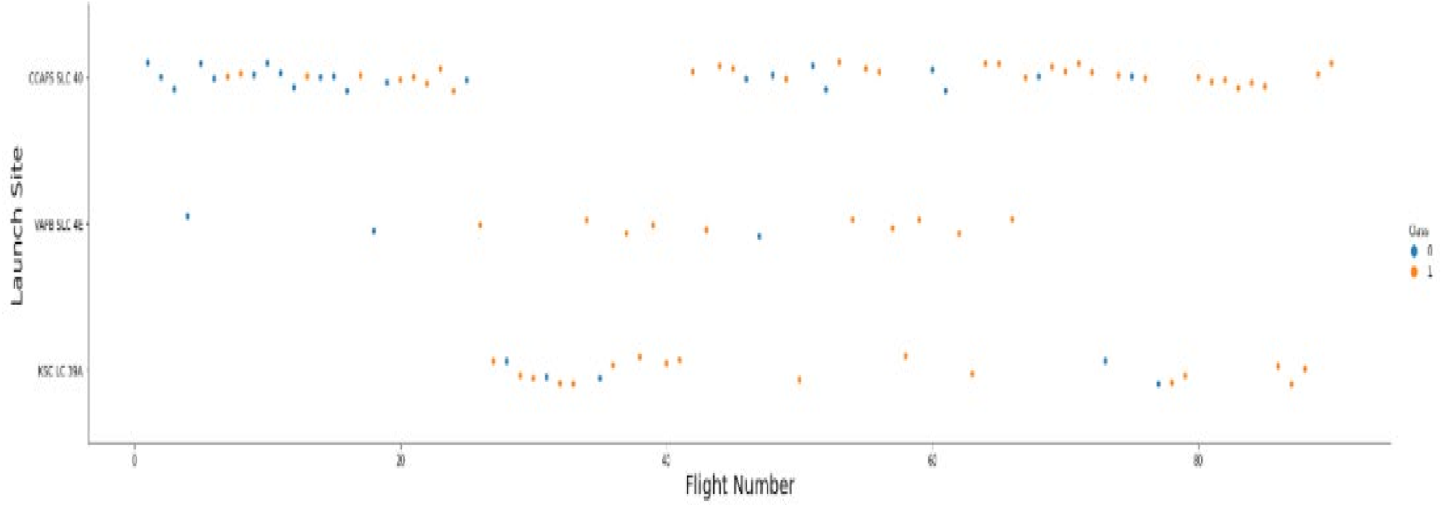
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



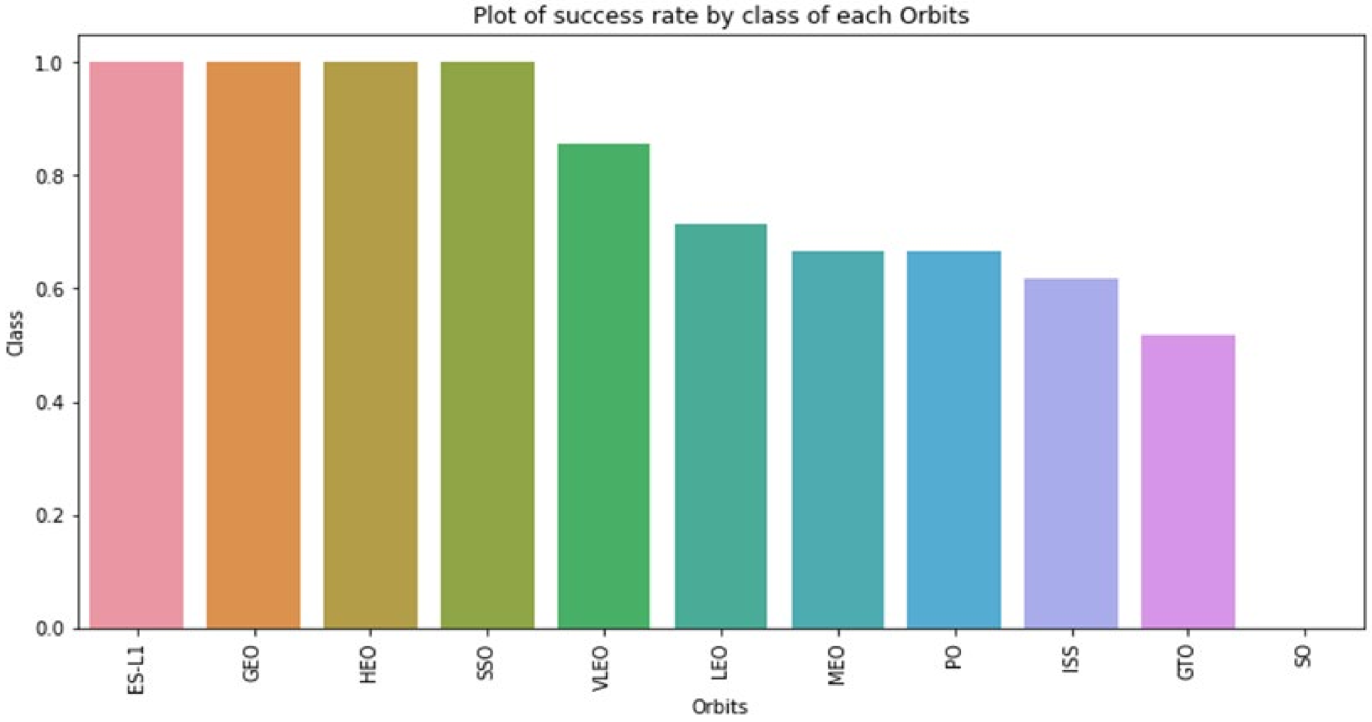
Payload vs. Launch Site

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



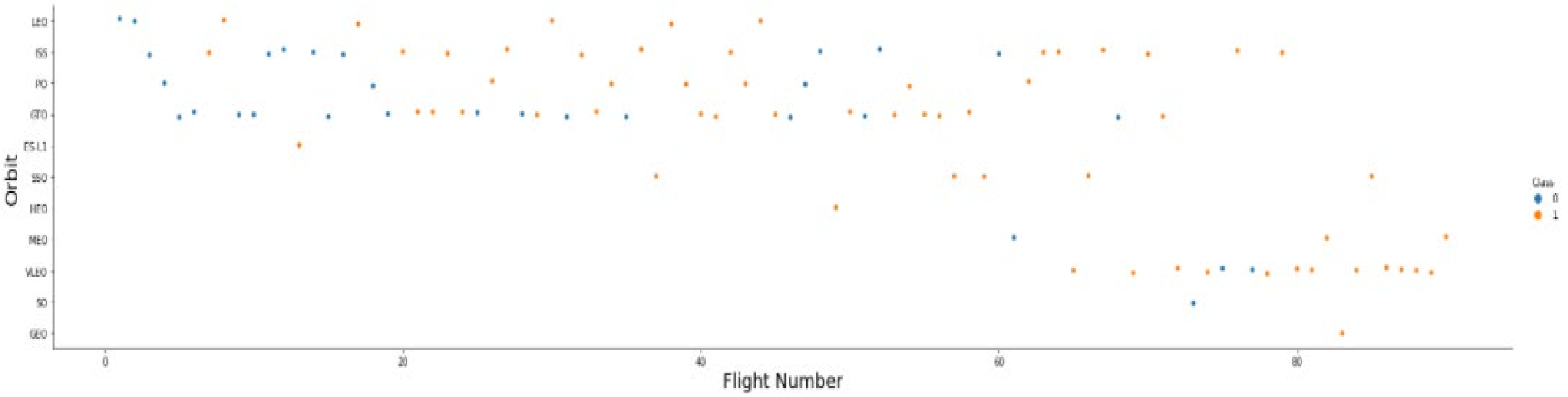
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



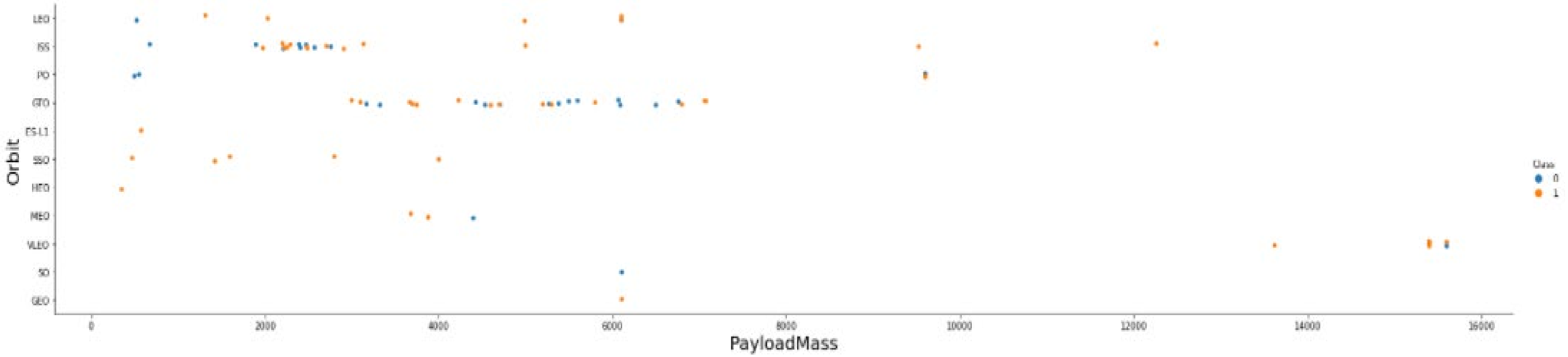
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



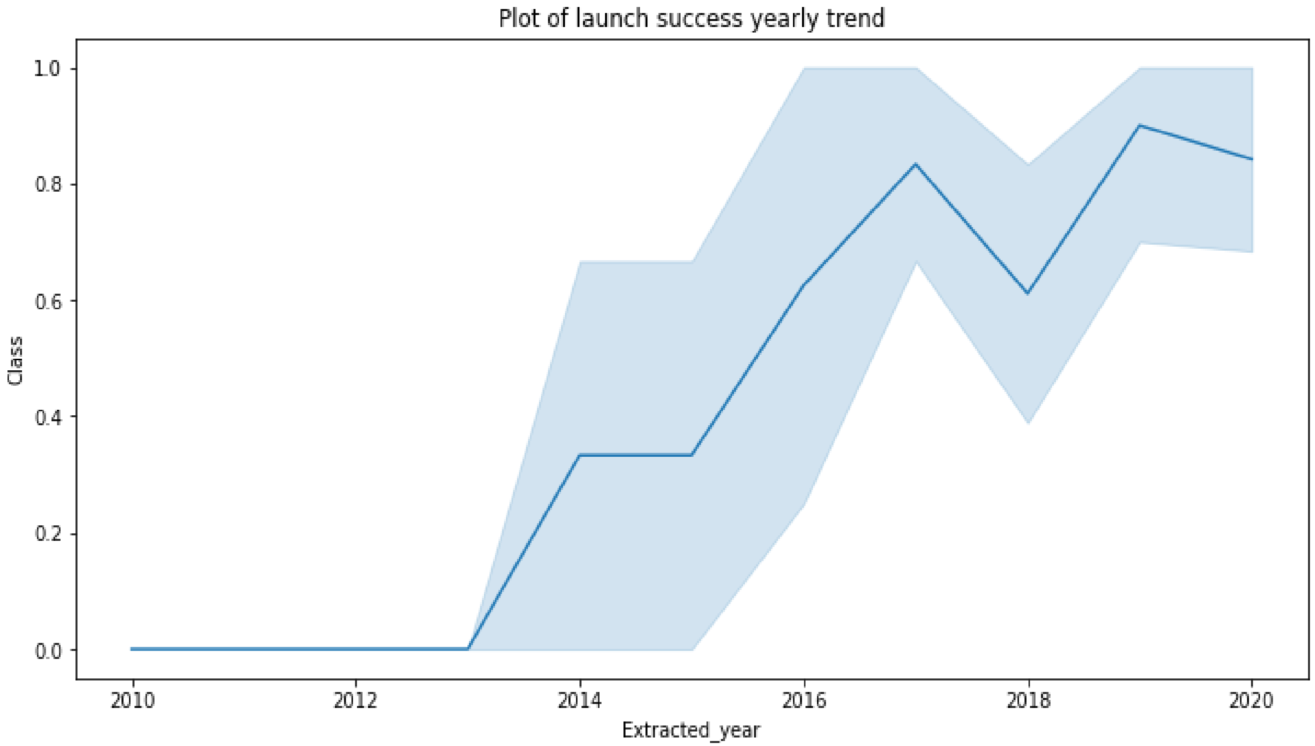
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



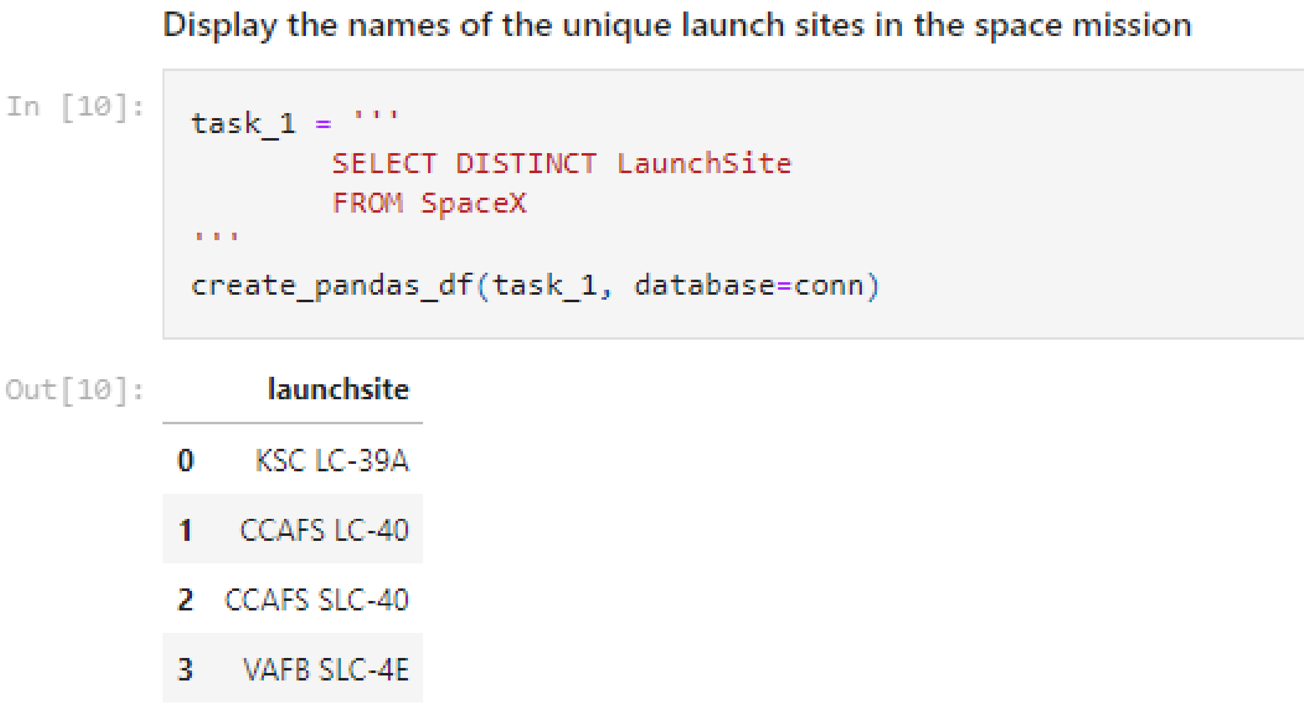
Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.



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

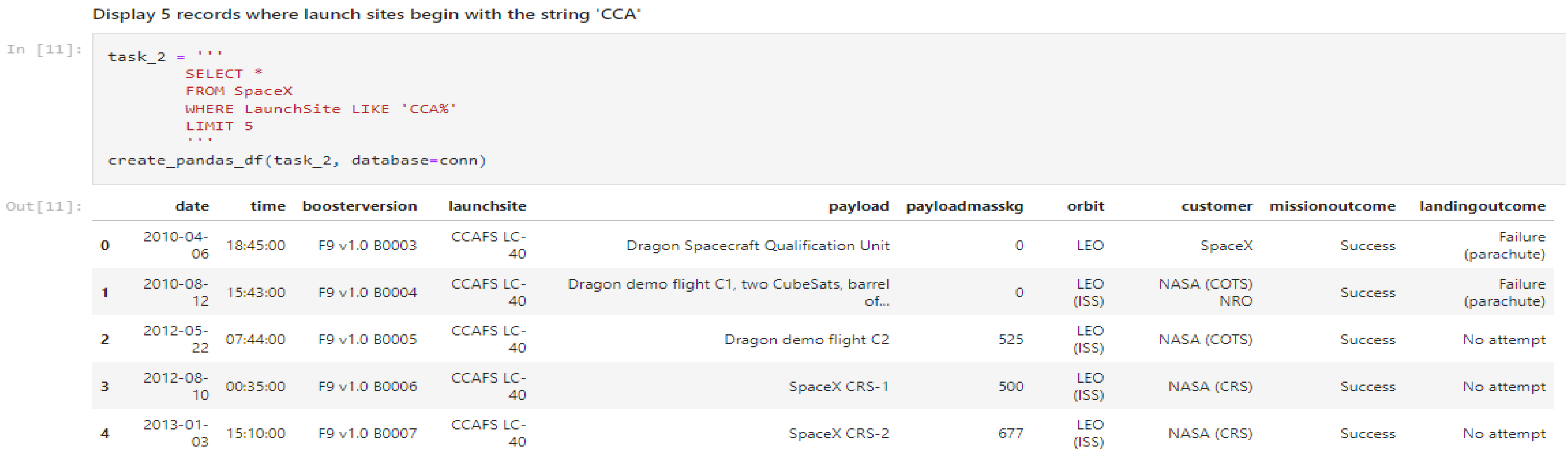
In [10]:

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'



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

```
In [11]:  task_2 = '''
                  SELECT *
                  FROM SpaceX
                  WHERE LaunchSite LIKE 'CCA%'
                  LIMIT 5
                  '''
          create_pandas_df(task_2, database=conn)
```

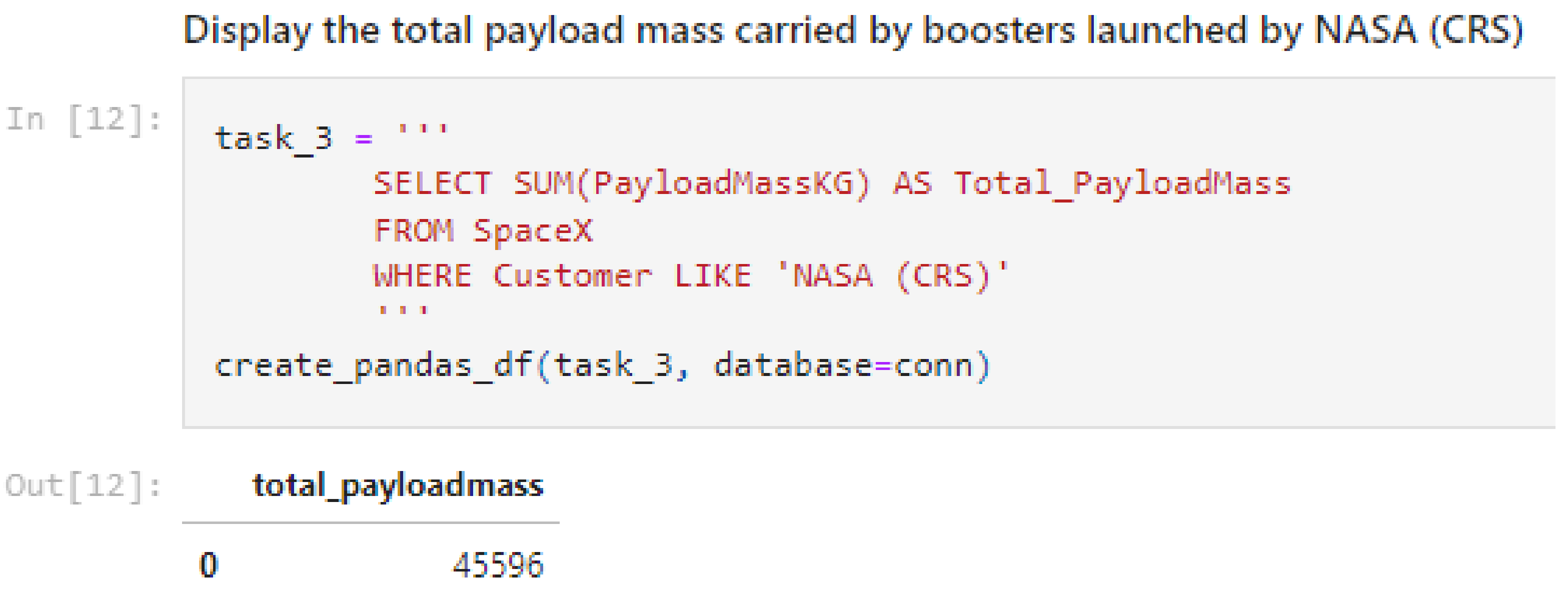
Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below



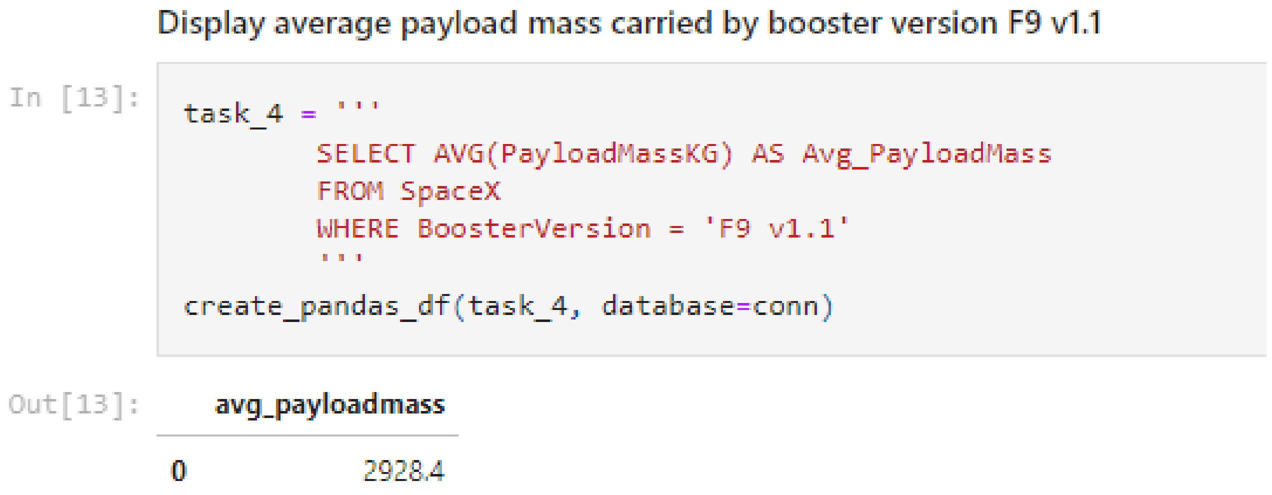
Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [12]: task_3 = '''
                 SELECT SUM(PayloadMassKG) AS Total_PayloadMass
                 FROM SpaceX
                 WHERE Customer LIKE 'NASA (CRS)'
                 '''
         create_pandas_df(task_3, database=conn)
```

```
Out[12]:    total_payloadmass
         0              45596
```

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4



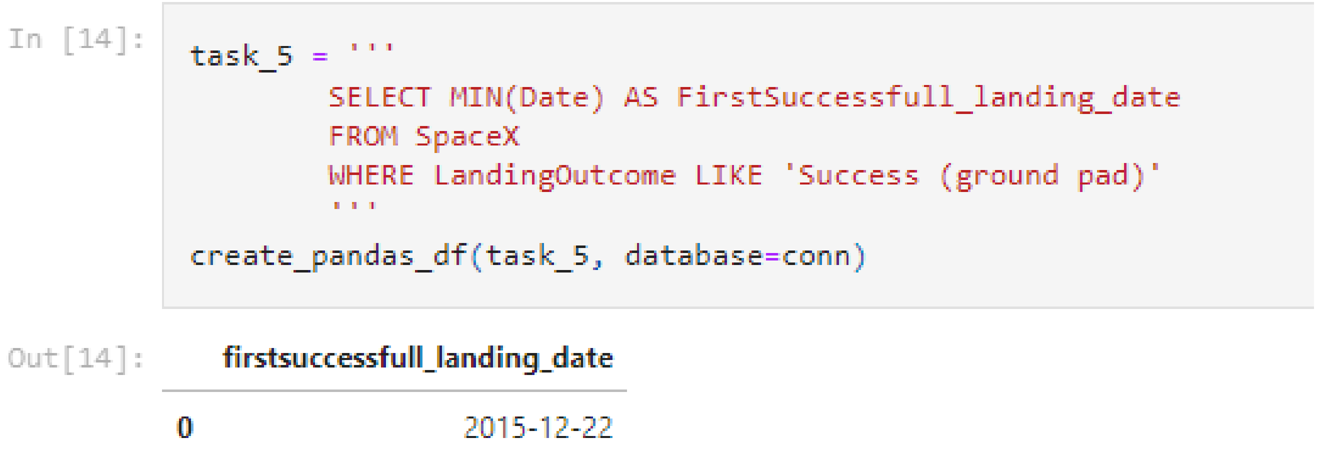
Display average payload mass carried by booster version F9 v1.1

```
In [13]:  task_4 = '''
                  SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
                  FROM SpaceX
                  WHERE BoosterVersion = 'F9 v1.1'
                  '''
          create_pandas_df(task_4, database=conn)
```

```
Out[13]:      avg_payloadmass
          0            2928.4
```

First Successful Ground Landing Date

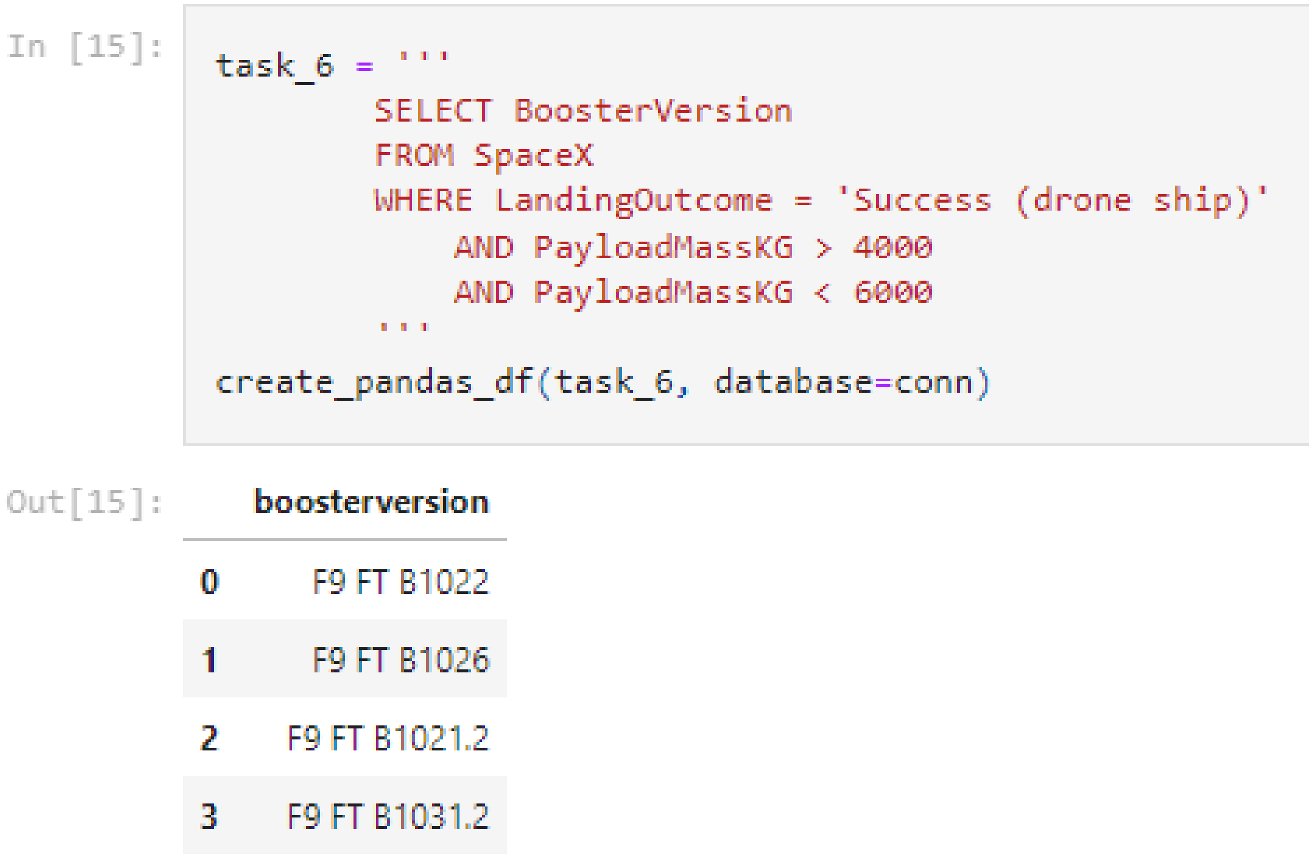
- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015



```
In [14]: task_5 = '''
                 SELECT MIN(Date) AS FirstSuccessfull_landing_date
                 FROM SpaceX
                 WHERE LandingOutcome LIKE 'Success (ground pad)'
                 '''
         create_pandas_df(task_5, database=conn)
```

```
Out[14]:    firstsuccessfull_landing_date
         0                     2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

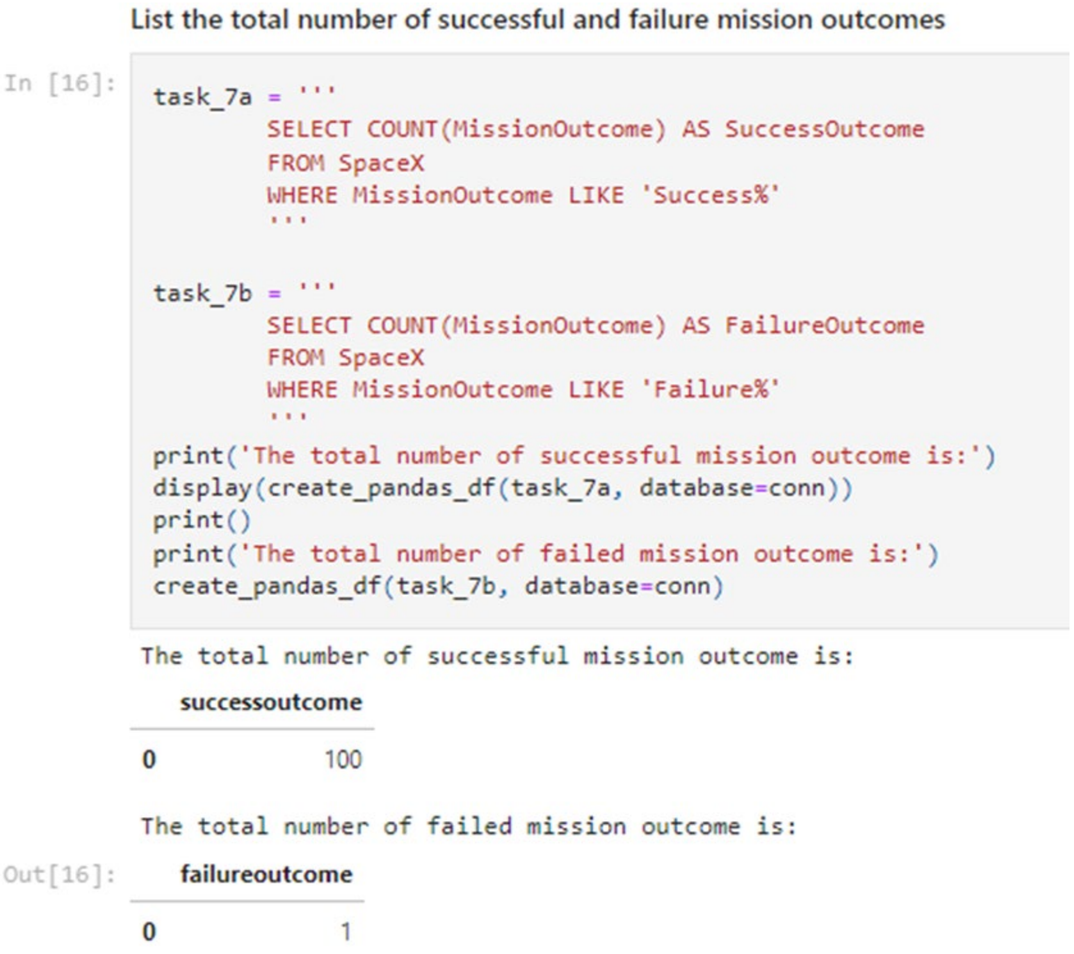


```
In [15]:  task_6 = '''
                  SELECT BoosterVersion
                  FROM SpaceX
                  WHERE LandingOutcome = 'Success (drone ship)'
                      AND PayloadMassKG > 4000
                      AND PayloadMassKG < 6000
                  '''
          create_pandas_df(task_6, database=conn)
```

```
Out[15]:     boosterversion
          0      F9 FT B1022
          1      F9 FT B1026
          2    F9 FT B1021.2
          3    F9 FT B1031.2
```

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 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:
```

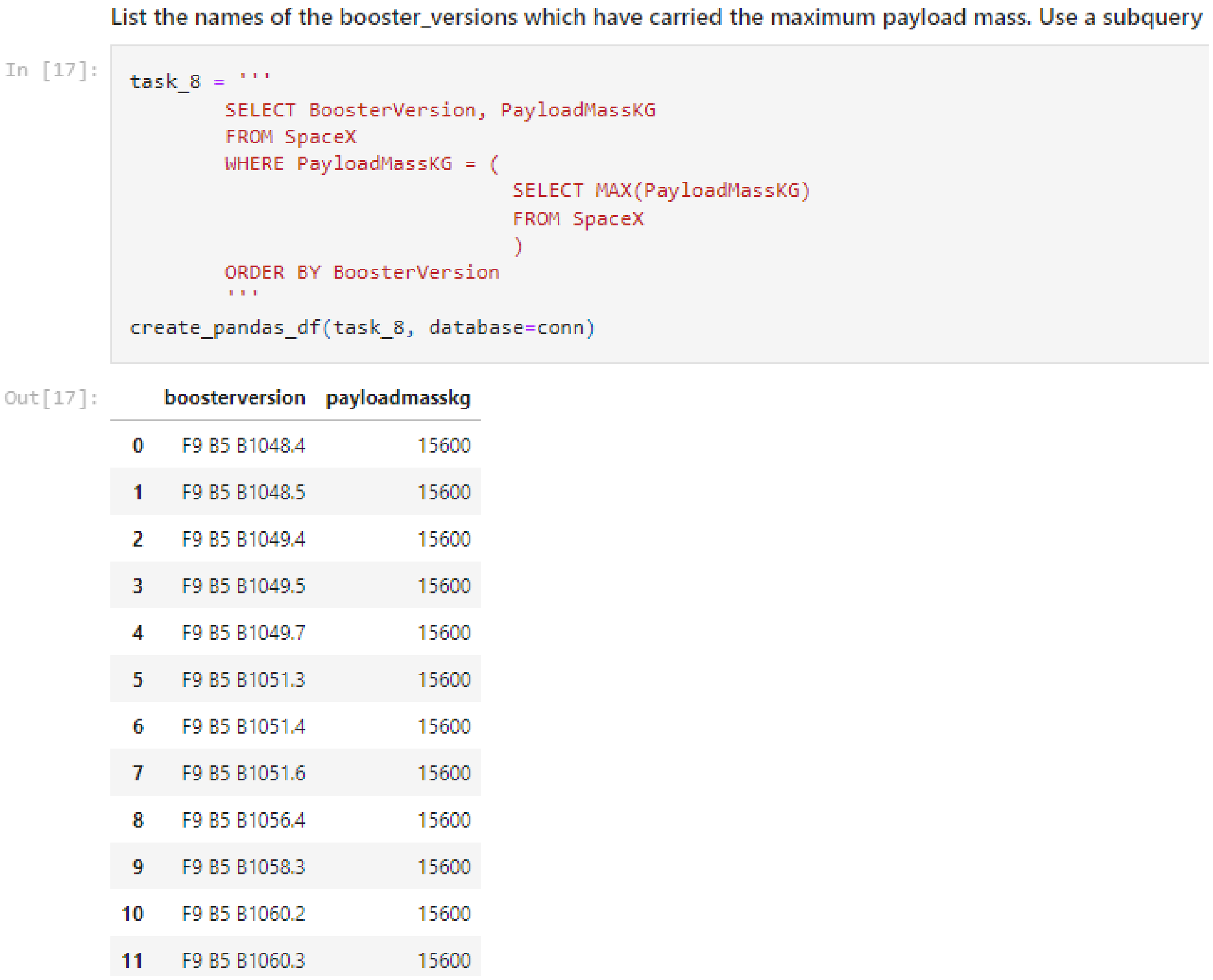
Out[16]:

	failureoutcome
0	1

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

Boosters Carried Maximum Payload

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



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

In [17]:

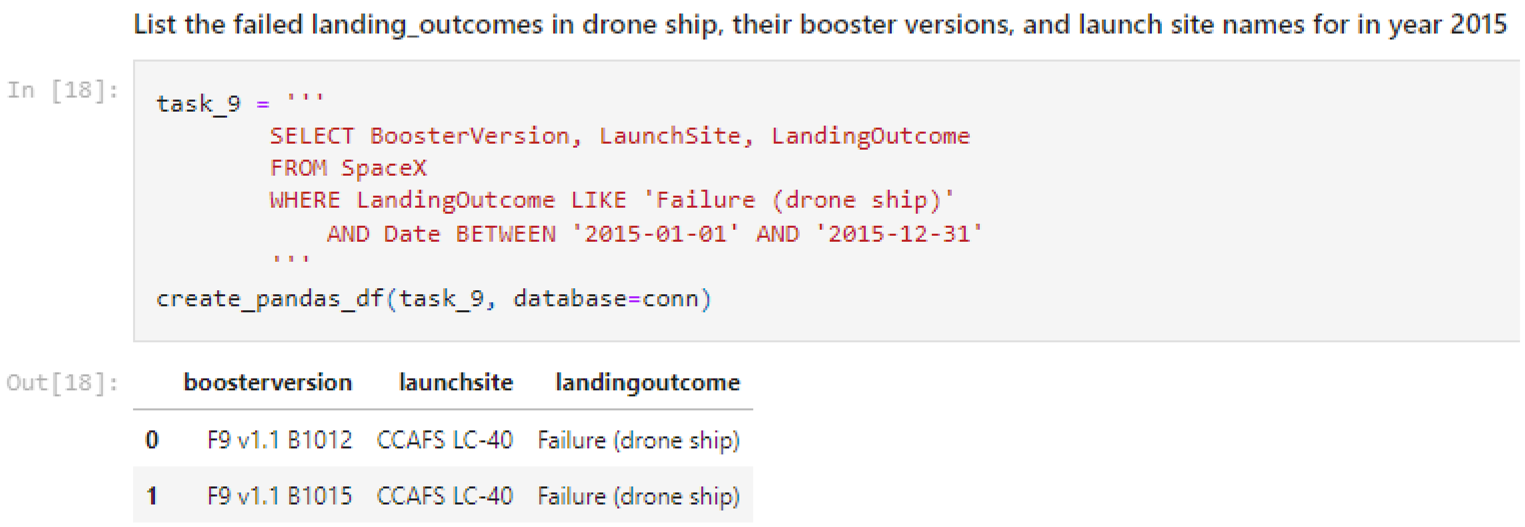
```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- We used a combinations of the **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



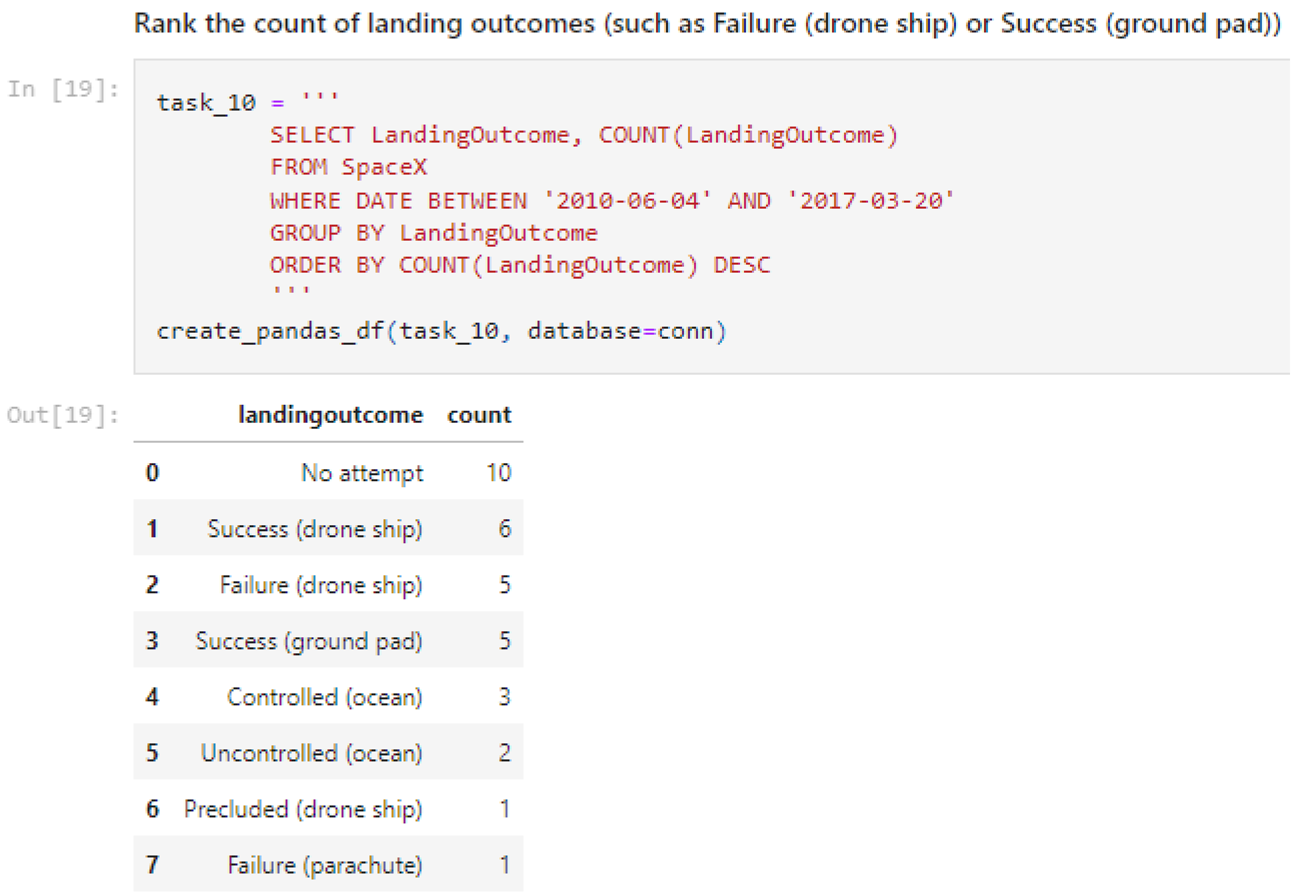
List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [18]:
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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))

```
In [19]:  task_10 = '''
                  SELECT LandingOutcome, COUNT(LandingOutcome)
                  FROM SpaceX
                  WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
                  GROUP BY LandingOutcome
                  ORDER BY COUNT(LandingOutcome) DESC
                  '''
          create_pandas_df(task_10, database=conn)
```

Out[19]:

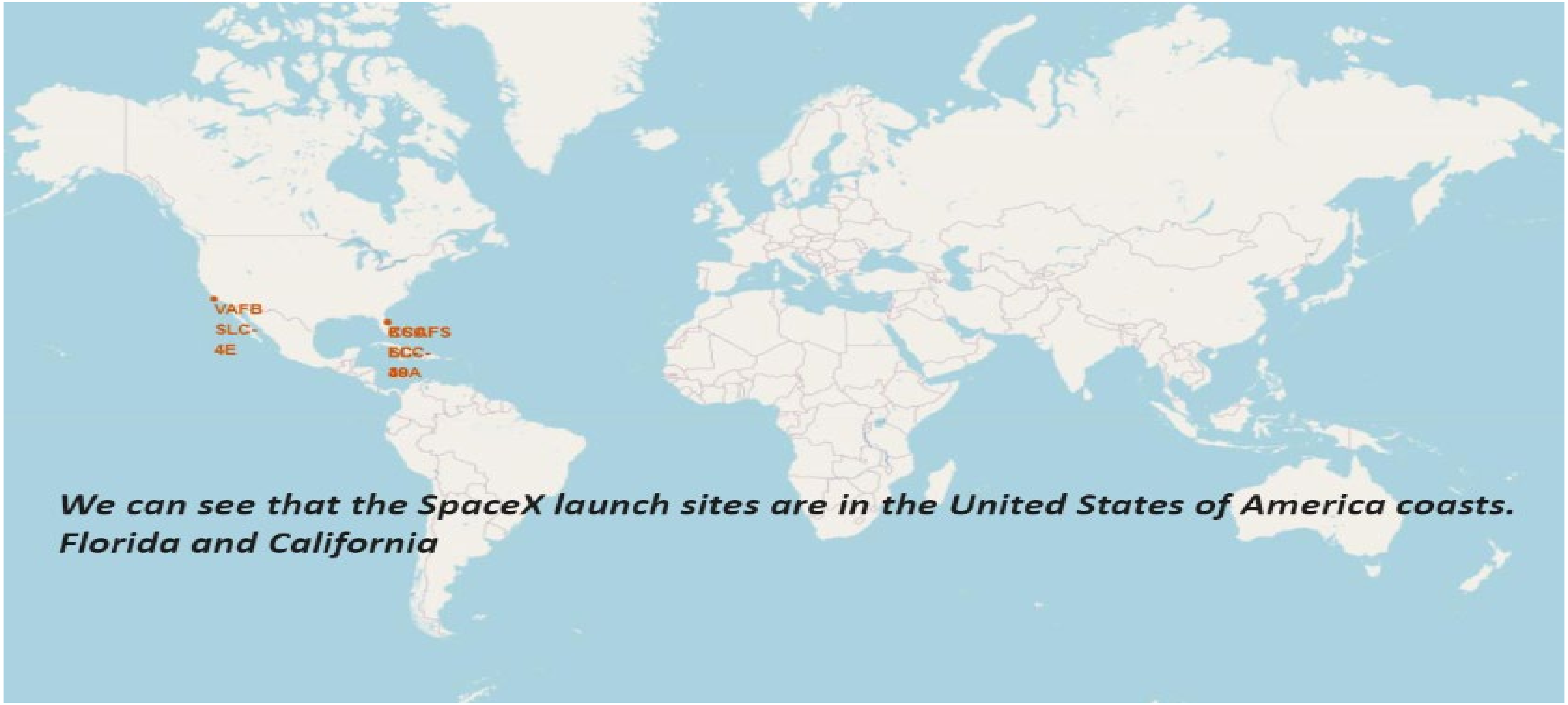
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

- We 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 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 3

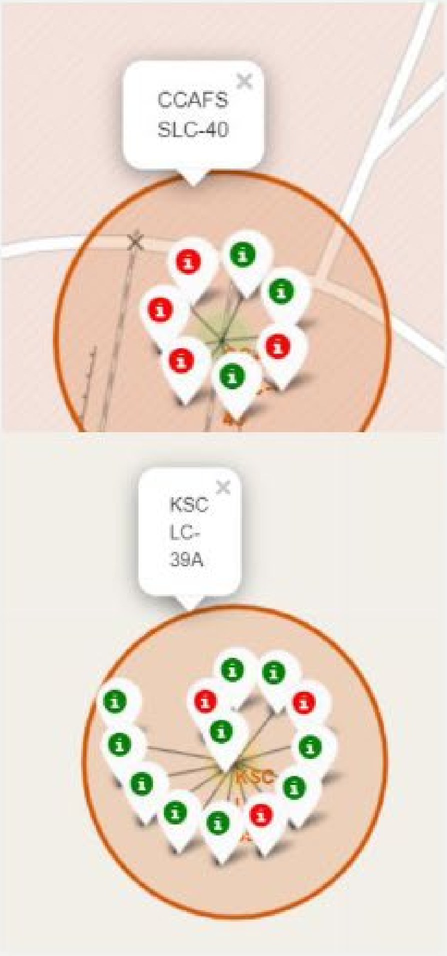
Launch Sites Proximities Analysis

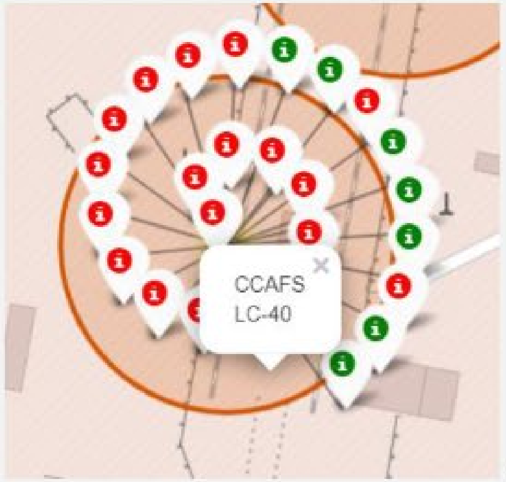
All launch sites global map markers



We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

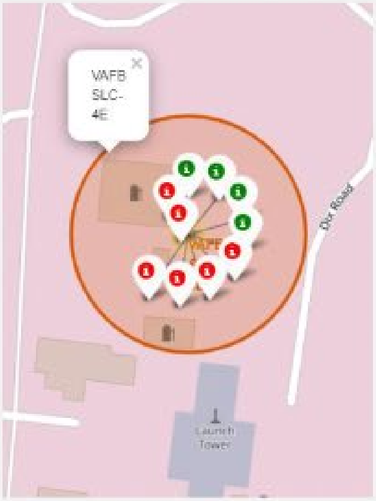




CCAFS SLC-40
CCAFS LC-40
Centaur Road

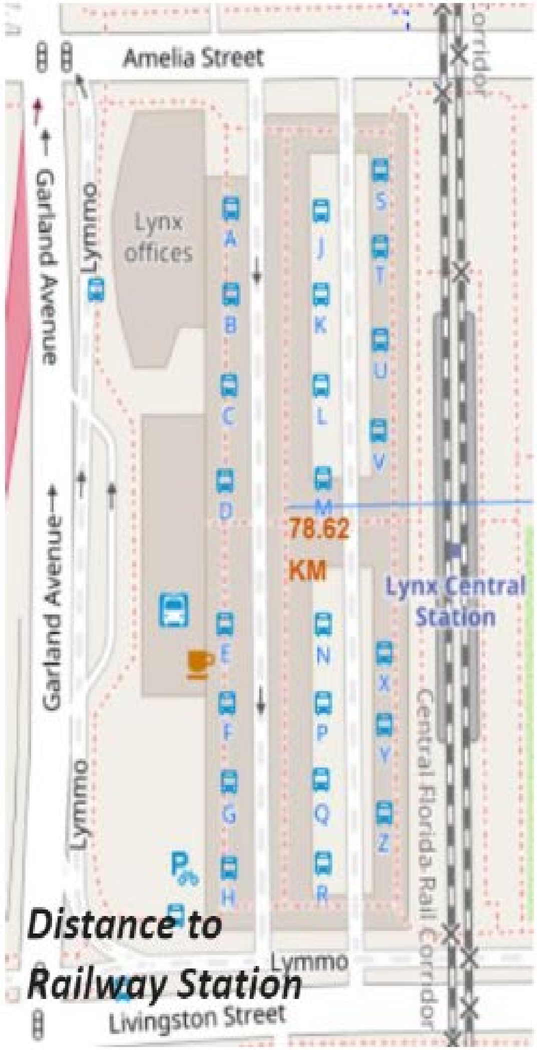
Florida Launch Sites

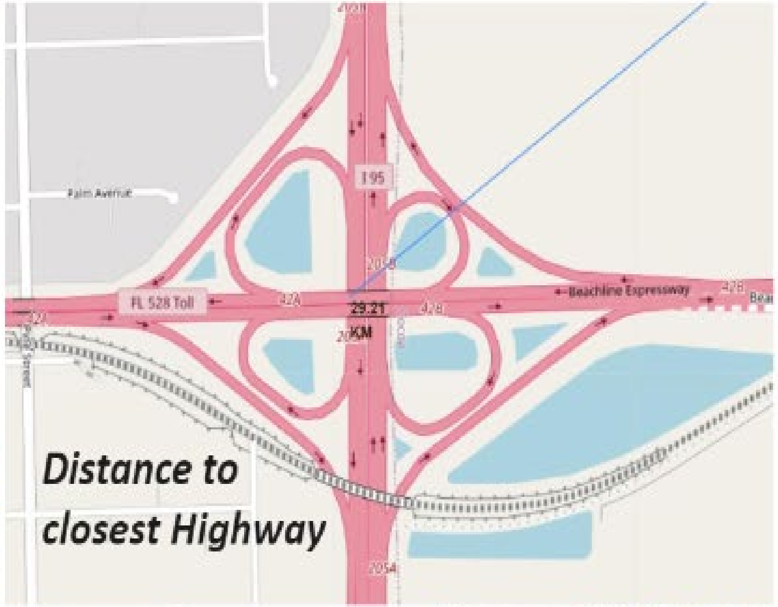
Green Marker shows successful Launches and Red Marker shows Failures



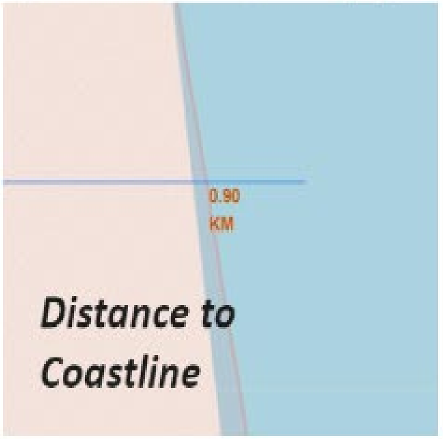
California Launch Site

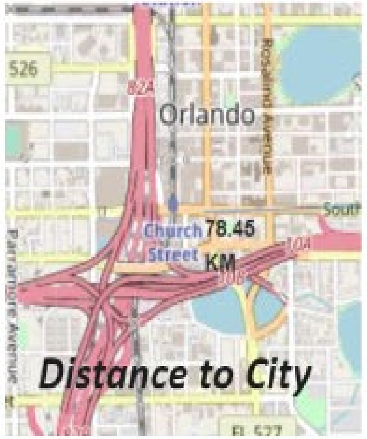
Launch Site distance to landmarks



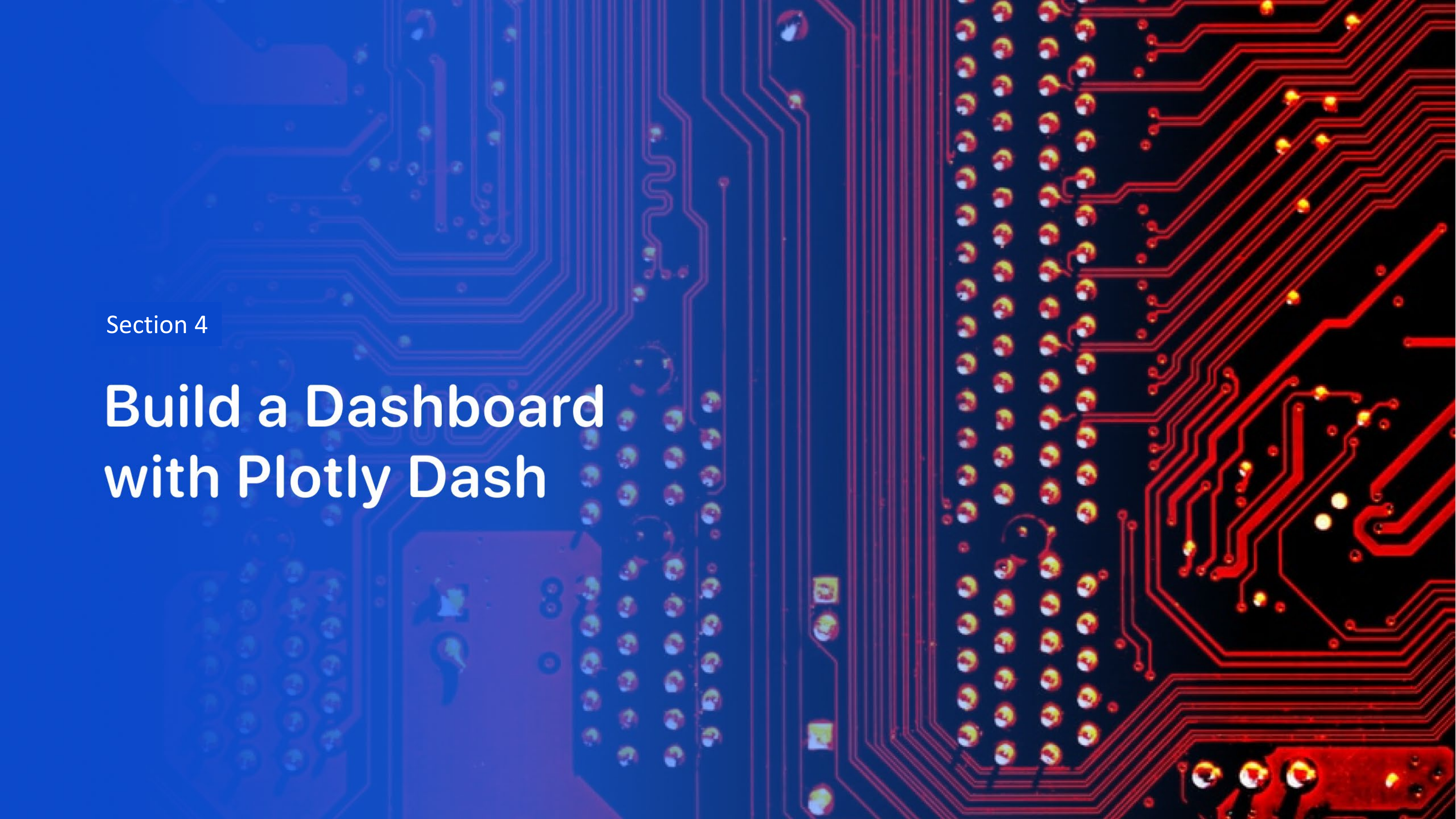


7
CCAFS
SLC-
40
26
CCAFS
LC-
40
Distance to coast





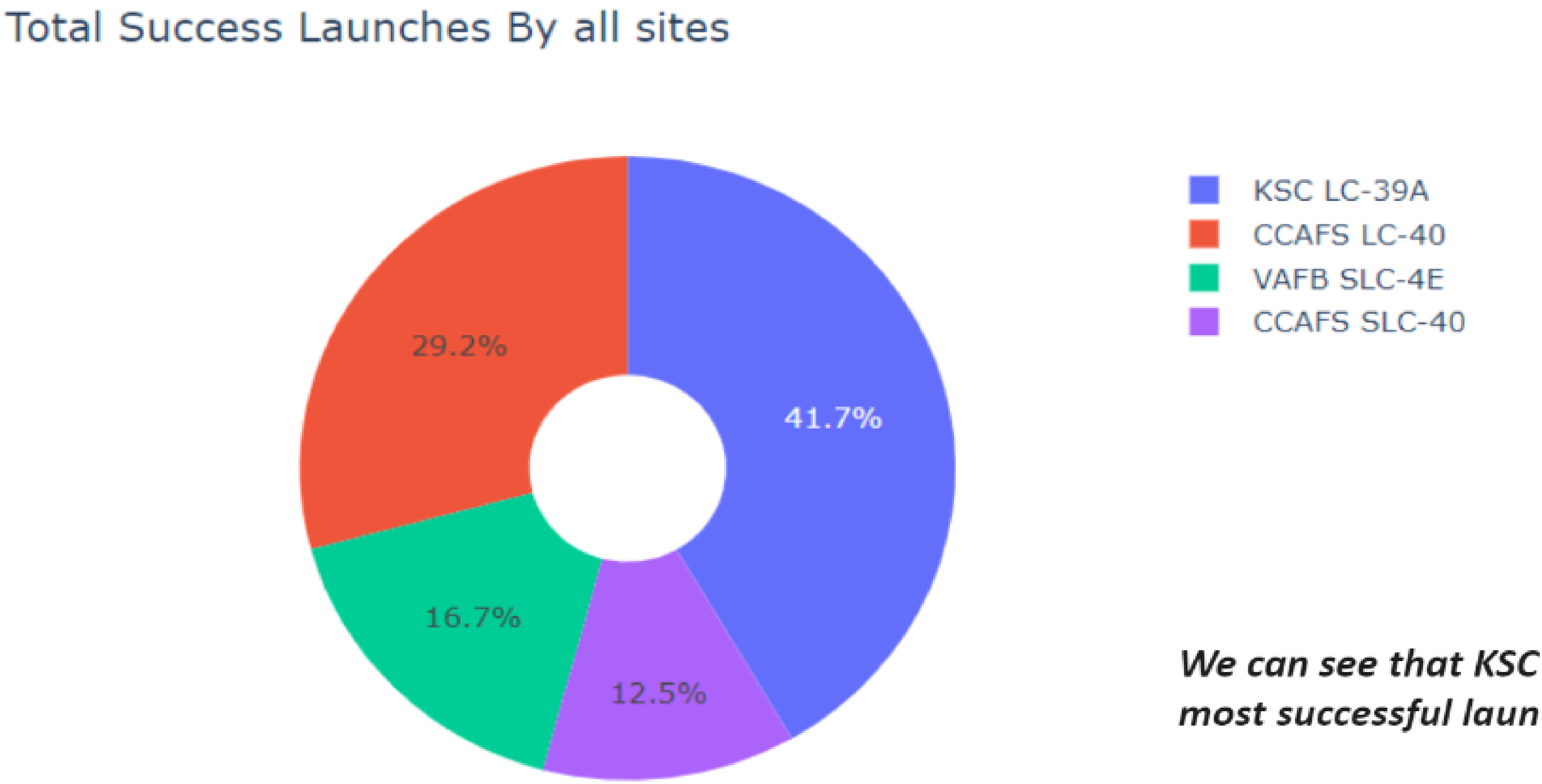
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 4

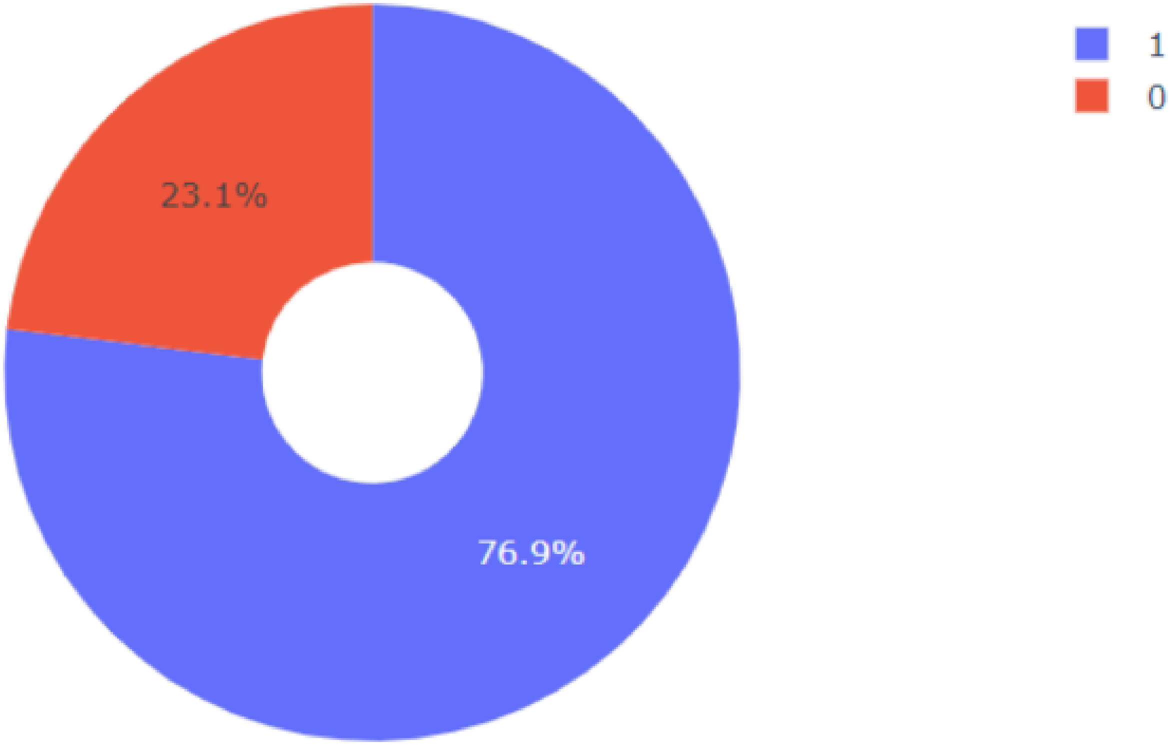
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site


Total Success Launches By all sites

We can see that KSC LC-39A had the most successful launches from all the sites

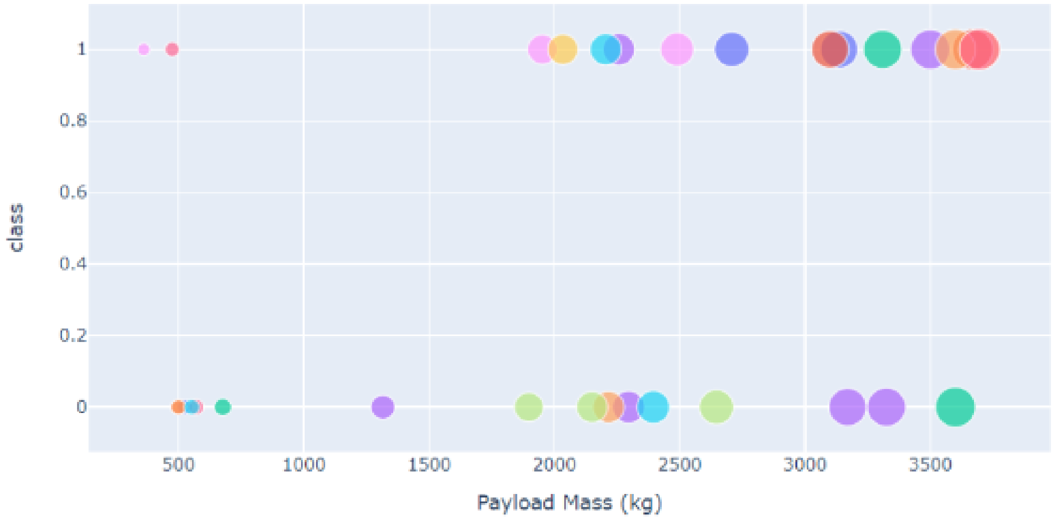
Pie chart showing the Launch site with the highest launch success ratio



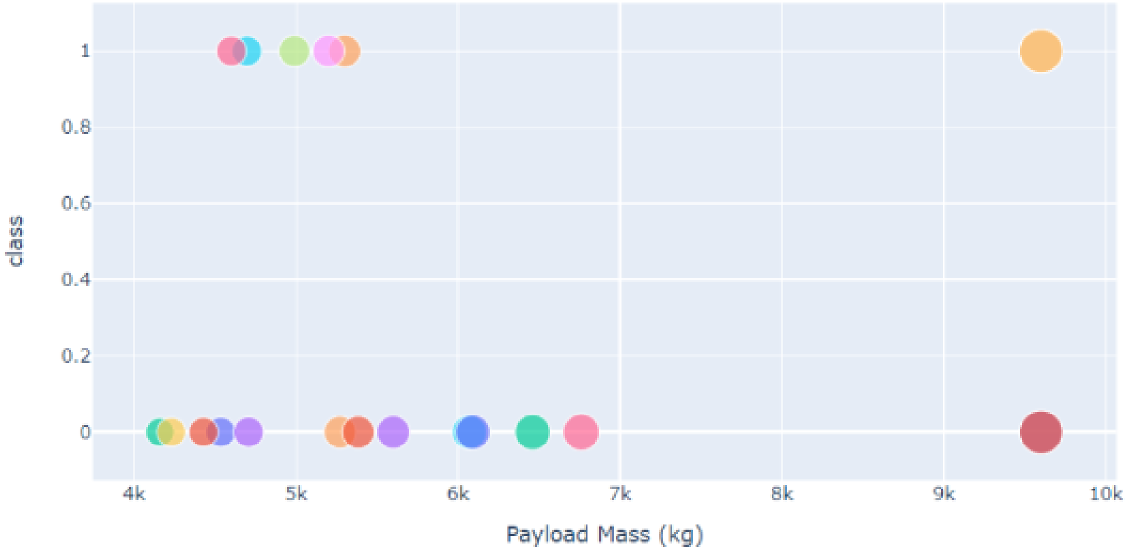
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



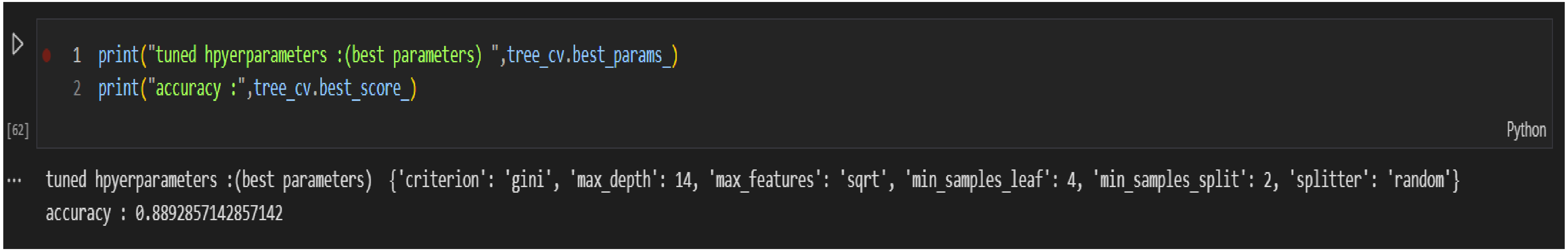
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

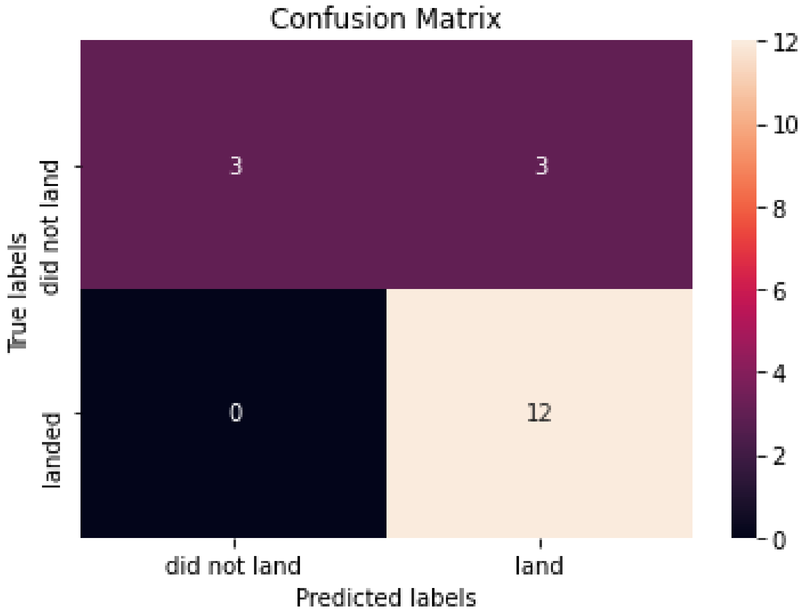


```python
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8892857142857142
```

Confusion Matrix

- The confusion matrix generated for the decision tree classifier reveals its ability to differentiate between various classes. However, a notable issue arises with false positives, where unsuccessful landings are incorrectly classified as successful ones by the model.



Conclusions

Based on our analysis, we draw the following conclusions:

1. There's a positive correlation between the number of flights at a launch site and its success rate, suggesting that higher activity contributes to better success rates.

2. The trend in launch success rates shows a steady increase from 2013 to 2020, indicating improvements in launch technology and operational efficiency during this period.

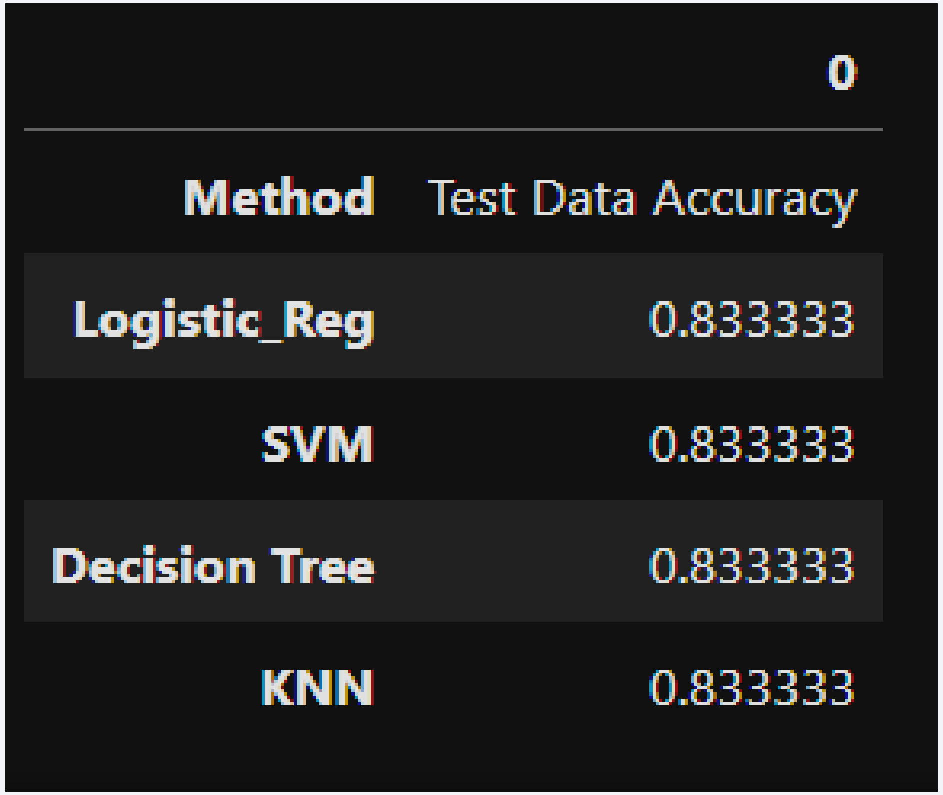
3. Orbits such as ES-L1, GEO, HEO, SSO, and VLEO exhibit the highest success rates, indicating their suitability for successful launches.

4. KSC LC-39A stands out as the launch site with the highest number of successful launches, highlighting its reliability and effectiveness in facilitating successful missions.

5. Among the machine learning algorithms evaluated, the Decision Tree classifier emerges as the most effective for this task, demonstrating superior performance in predicting launch outcomes based on the provided data.

Classification Accuracy

- Practically all these algorithms give the same result



	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Appendix

- Github like for all notebook: https://github.com/djafari700/-applied-data-science-capstone

Thank you!