



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Reihaneh
Hajisafarali>
<26 September 2024>



Outline

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

Executive Summary

Methodologies:

Data Acquisition: Data was sourced from SpaceX's internal repositories and publicly accessible online platforms.

Exploratory Data Analysis (EDA): In-depth EDA was conducted to extract meaningful insights from the collected dataset.

Predictive Modeling: Multiple machine learning algorithms were developed and evaluated to forecast launch outcomes.

The CCAFS LC-40 launch facility consistently demonstrated the highest success rate, especially for missions with payloads in the 2000-5000 kg range targeting Sun-synchronous orbits. The developed predictive model achieved a commendable accuracy of 83.33% when evaluated on unseen test data.

(Cape Canaveral Space Force Station Launch Complex 40)

Introduction

- **Project Goal:** To develop a machine learning model capable of predicting the success of SpaceX Falcon 9 first stage landings.
- **Business Problem:** The high cost of space launches, primarily due to the need for a new rocket for each mission, has hindered the growth of the commercial space industry. SpaceX has significantly reduced launch costs by successfully reusing the first stage of their Falcon 9 rockets. However, the success of these landings is not guaranteed
- **Potential Benefits:** A predictive model could enable competitors to:
 - Offer more competitive pricing for launch services.
 - Make informed decisions about launch strategies and risk management.
 - Improve their overall business performance.

Section 1

Methodology

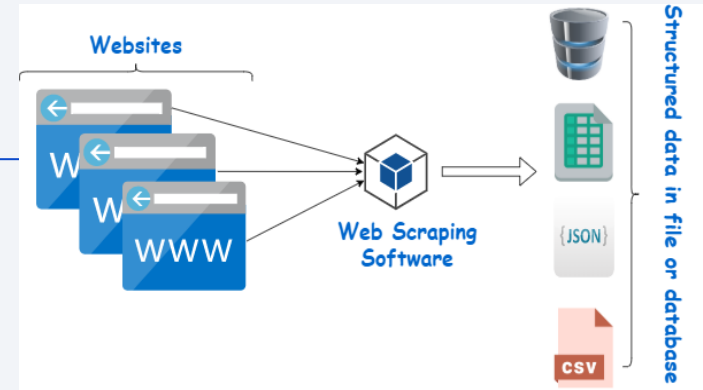
Methodology

Executive Summary

- Data collection methodology:
 - From SpaceX API and web scraping
- Perform data wrangling
 - Handling missing values
 - Class labeling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - SVM
 - Logistic regression

Data Collection

- SpaceX API and web scraping: Web scraping



**Web
scraping**

SpaceX
API

rocket

payload
s

Launch
pad

cores

Data
_ utc

Flight_
number

Request JSON files

Pandas Data Frame

CSV file

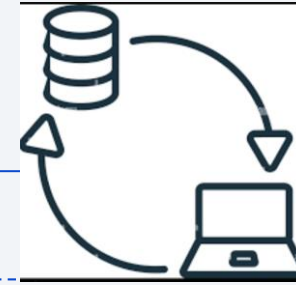


Falcon 9 HTML table from
Wikipedia

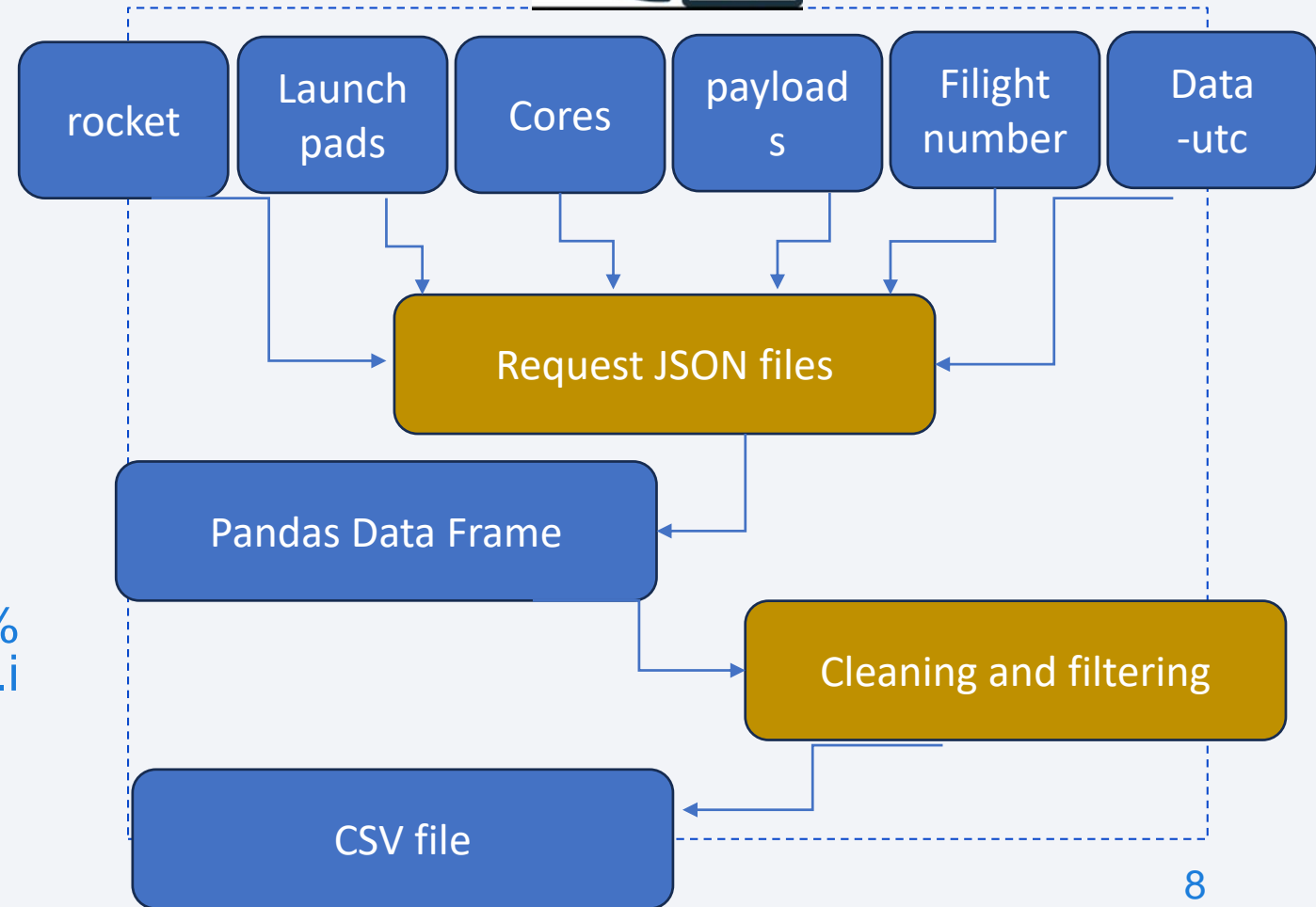
Pandas Data Frame

CSV file

Data Collection – SpaceX API



- Use the requests library to extract and concatenate past launch data.
- Replace any missing payload mass value with the column average.
- Filter the data by the relevant features for analysis.
- Add the GitHub URL of the completed SpaceX API calls notebook (<https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/Data%20Collection%20API%20Lab.reyhan.ipynb>), as an external reference and peer-review purpose



Data Collection - Scraping

- I applied web scraping to webscraping Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/web_scraping_reyhan.py

```
[ ] 1 # use requests.get() method with the provided static_url  
    2 response = requests.get(static_url)  
    3 # assign the response to a object  
    4 response.status_code
```

⇨ 200

```
[ ] 1 response.text
```

⇨ '<!DOCTYPE html>\n<html class="client-nojs vector-feature-language-in-header-enabled vector-
ge-tools-pinned-disabled vector-feature-toc-pinned-clientpref=1 vector-feature-main-menu-pin
ector-feature-custom-font-size-clientpref=1 vector-feature-appearance-pinned-clientpref=1 ve
\n<head>\n<meta charset="UTF-8">\n<title>List of Falcon 9 and Falcon Heavy launches - Wikiped
ector-feature-language-in-main-page-header-disabled vector-feature-sticky-header-disabled ve
nu-pinned-disabled vector-feature-lim...'

Create a BeautifulSoup object from the HTML response

```
[ ] 1 # Use BeautifulSoup() to create a BeautifulSoup object from a response text content  
    2 soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the BeautifulSoup object was created properly

▶ 1 soup.text

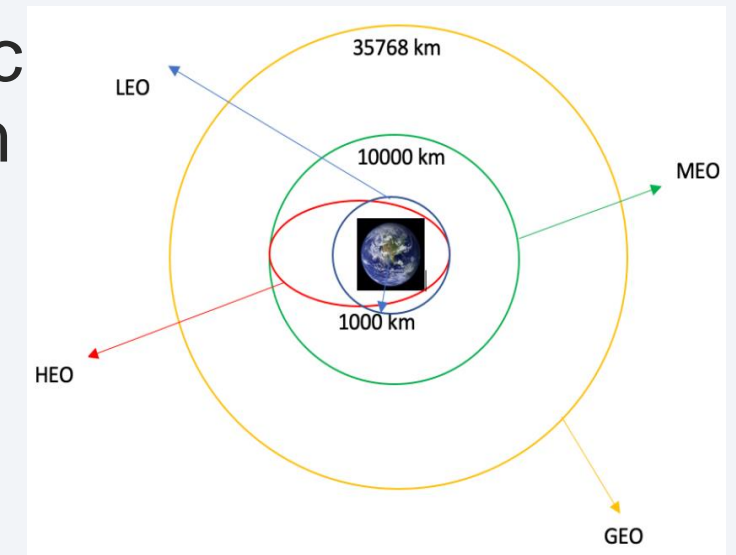
⇨ '\n\n\nList of Falcon 9 and Falcon Heavy launches – Wikipedia\n\n\n\n\nMain pageContentsCurrent eventsRa
ortalRecent changesUpload file\n\n\n\n\nSearch\n\n\n\nCreate account\nLog in\nPersonal tools\nCreate ac
\n\n\n\n\nContents\nmove to sidebar\hide\n\n\n(\nket configurations\n\n\n\n1...'

```
[ ] 1 # Use soup.title attribute  
    2 soup.title
```

⇨ <title>List of Falcon 9 and Falcon Heavy launches – Wikipedia</title>

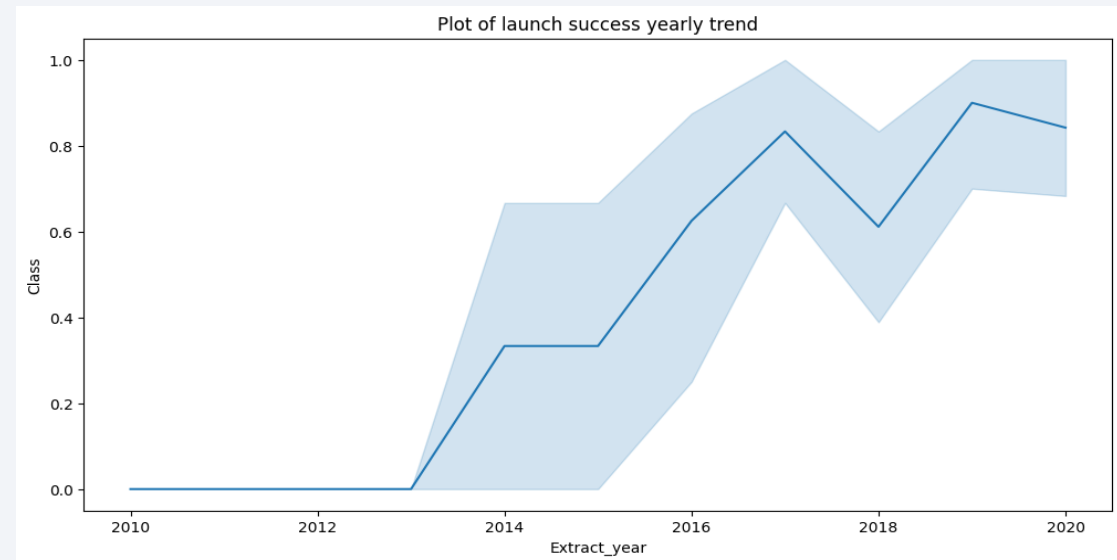
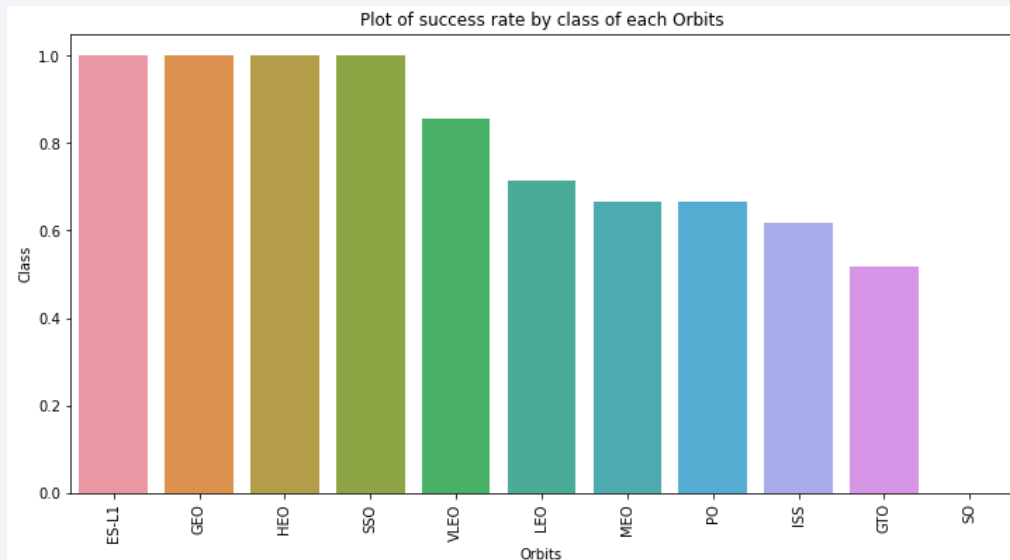
Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbit
- We created landing outcome label from outcome column and exported the results to CSV.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/data_wrangling.reihan.ipynb



EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- The link to the notebook is https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/eda_with_visualization_reyhan.py



EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
 - The names of unique launch sites in the space mission.
 - The total payload mass carried by boosters launched by NASA (CRS)
 - The average payload mass carried by booster version F9 v1.1
 - The total number of successful and failure mission outcomes
 - The failed landing outcomes in drone ship, their booster version and launch site names.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/exploratory_analysis_using_sql_reyhan.py

Build an Interactive Map with Folium

- We 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.
- We 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.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
 - Are launch sites near railways, highways and coastlines.
 - Do launch sites keep certain distance away from cities.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/Interactive_visual_analytics_with_folium__reyhan.py

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/spacex_launch_dash_reyhan.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- https://github.com/Reihaneh-Hajisafarali/SpaceX/blob/main/spacex_machine_learning_prediction_reyhana.py

Results

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

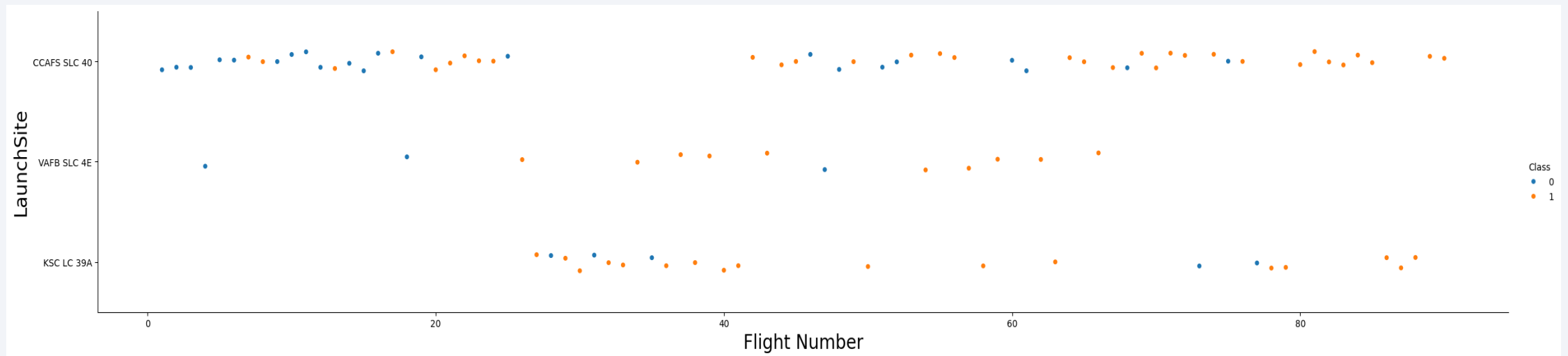
The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

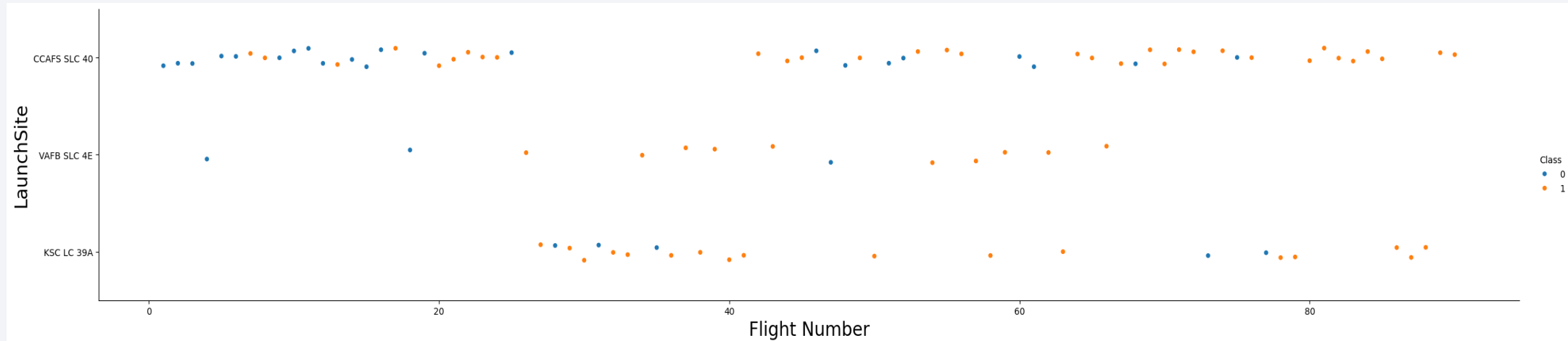
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



Payload vs. Launch Site

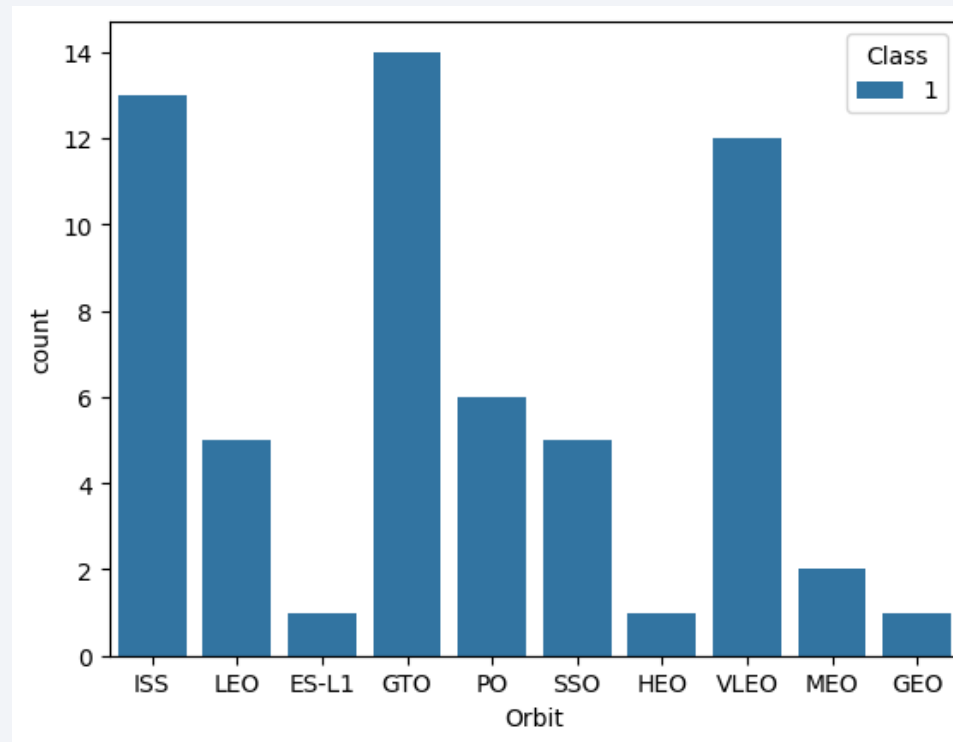


The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



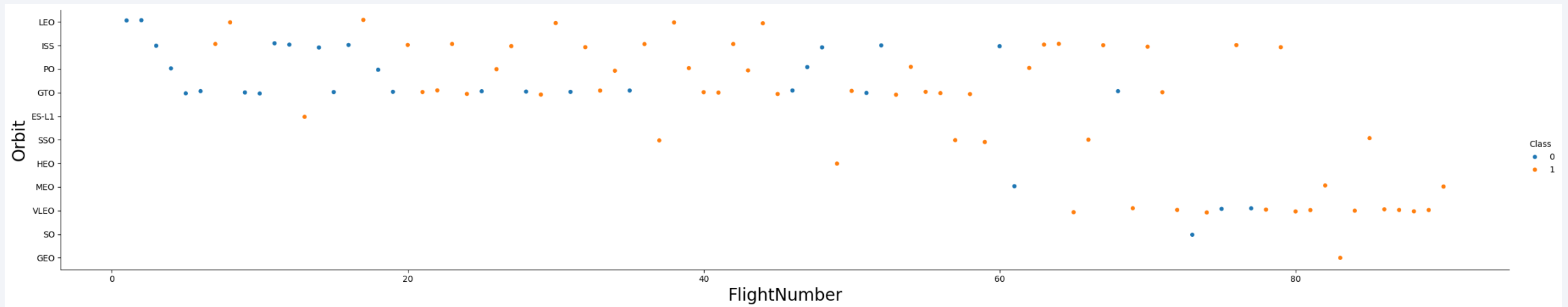
Success Rate vs. Orbit Type

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



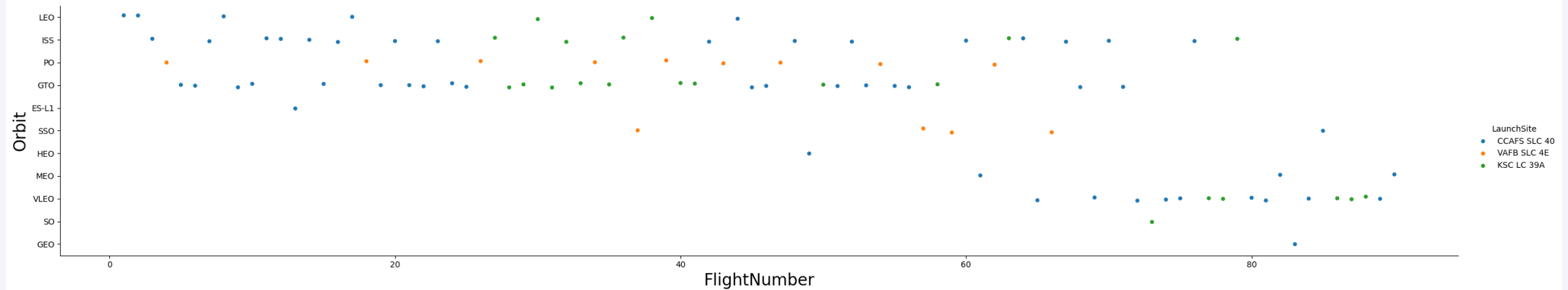
Flight Number vs. Orbit Type

- 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.



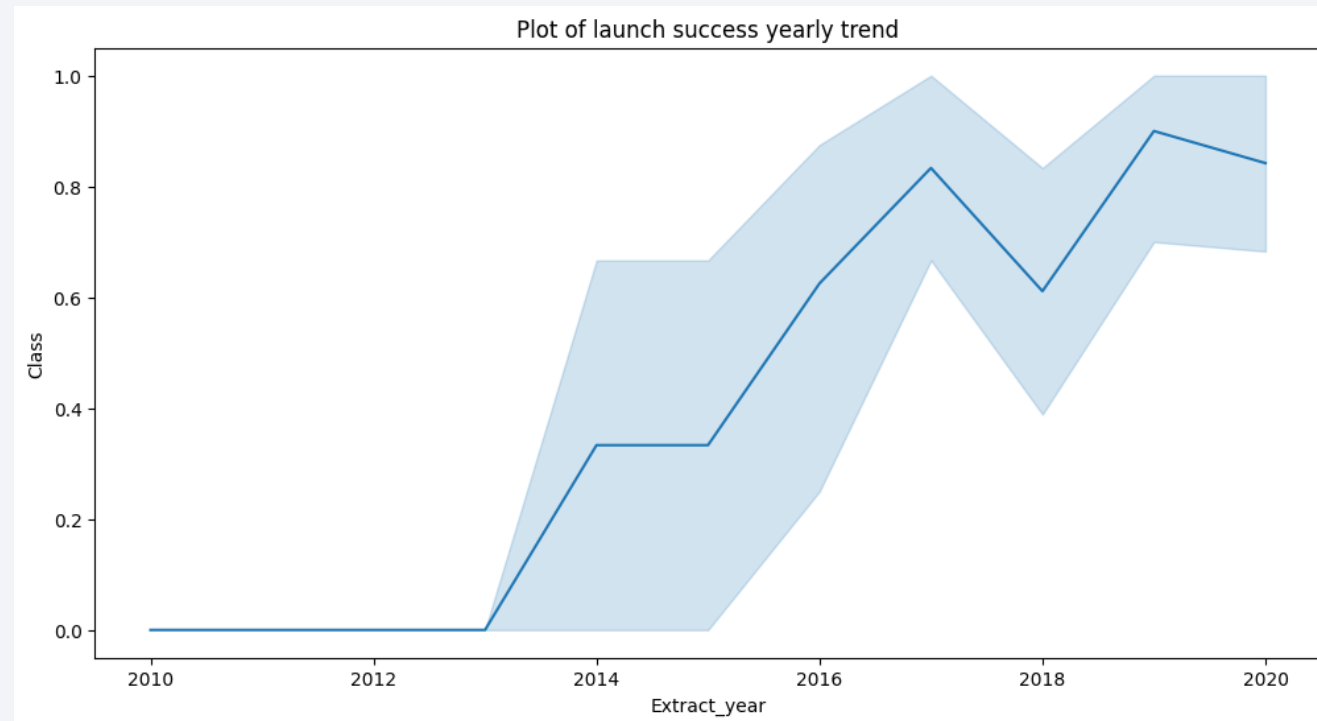
Payload vs. Orbit Type

- 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.

```
⇒ * sqlite:///my_data1.db  
Done.  
Launch_Site  
CCAFS LC-40  
VAFB SLC-4E  
KSC LC-39A  
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

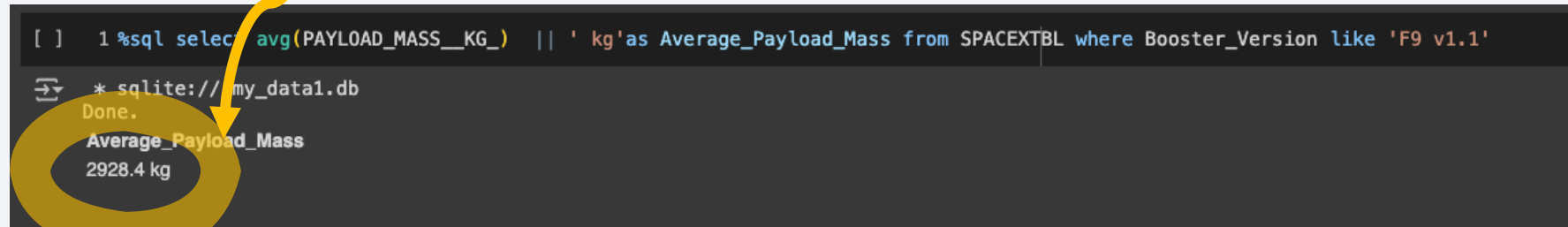
```
[ ] 1 %sql SELECT SUM(PAYLOAD_MASS_KG_) || ' kg' AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

Total_Payload_Mass
45596 kg

Average Payload Mass by F9 v1.1

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



```
[ ] 1 %sql select avg(PAYLOAD_MASS_KG_) || ' kg' as Average_Payload_Mass from SPACEXTBL where Booster_Version like 'F9 v1.1'
```

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

```
Done.
```

Average_Payload_Mass
2928.4 kg

First Successful Ground Landing Date

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

```
[ ] 1 %sql select min(Date) as First_Successful_Landing_Date from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

```
2  
  
* sqlite:///my_data1.db  
Done.  
First_Successful_Landing_Date  
2015-12-22
```


Successful Drone Ship Landing with Payload between 4000 and 6000

- 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

```
[ ] 1 %sql select Booster_Version,PAYLOAD_MASS_KG_ , 'Success (drone ship)' from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS_KG_ between 4000 and 6000
```



```
↳ * sqlite:///my_data1.db  
Done.  
Booster_Version PAYLOAD_MASS_KG_ 'Success (drone ship)'  
F9 FT B1022      4696             Success (drone ship)  
F9 FT B1026      4600             Success (drone ship)  
F9 FT B1021.2    5300             Success (drone ship)  
F9 FT B1031.2    5200             Success (drone ship)
```

Total Number of Successful and Failure Mission Outcomes

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

```
[ ] 1 %sql SELECT Count(Mission_Outcome) as The_Number_Of_Mission_Outcomes, Mission_Outcome from SPACEXTBL group by Mission_Outcome order by Mission_Outcome
```

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

```
Done.
```

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

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.

```
1 %sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ like (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL) ORDER BY booster_version;
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
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



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

```
Done.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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 2017-03-20.
- 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.

```
[ ] 1 %sql SELECT Landing_Outcome, COUNT(landing_outcome) AS Count, DATE FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Count DESC;
```

* sqlite:///my_data1.db
Done.

Landing_Outcome	Count	Date
No attempt	10	2012-05-22
Success (drone ship)	5	2016-04-08
Failure (drone ship)	5	2015-01-10
Success (ground pad)	3	2015-12-22
Controlled (ocean)	3	2014-04-18
Uncontrolled (ocean)	2	2013-09-29
Failure (parachute)	2	2010-06-04
Precluded (drone ship)	1	2015-06-28

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky and a view of the Earth's surface, which is covered in a dense network of city lights and clouds. The lights are concentrated in the lower right portion of the image, while the upper left portion shows a clear blue sky.

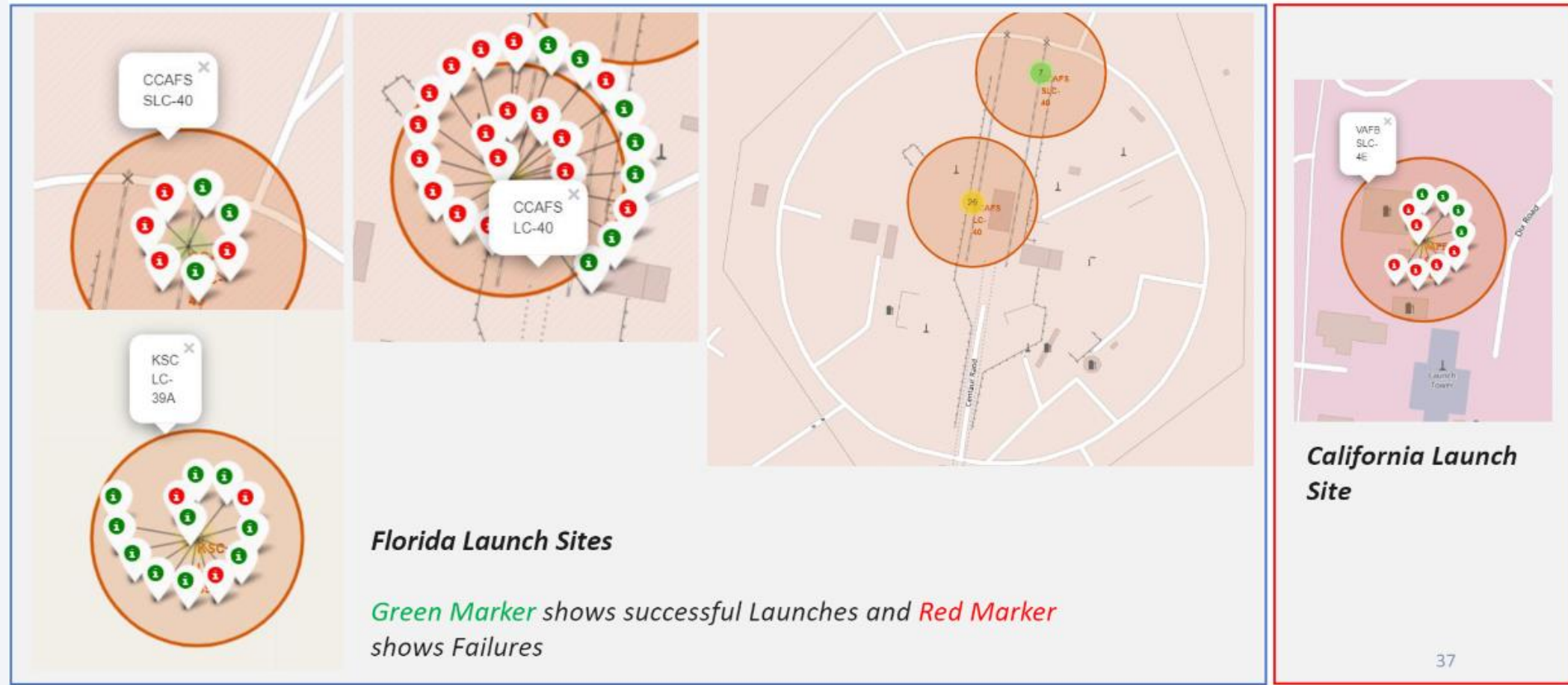
Section 3

Launch Sites Proximities Analysis

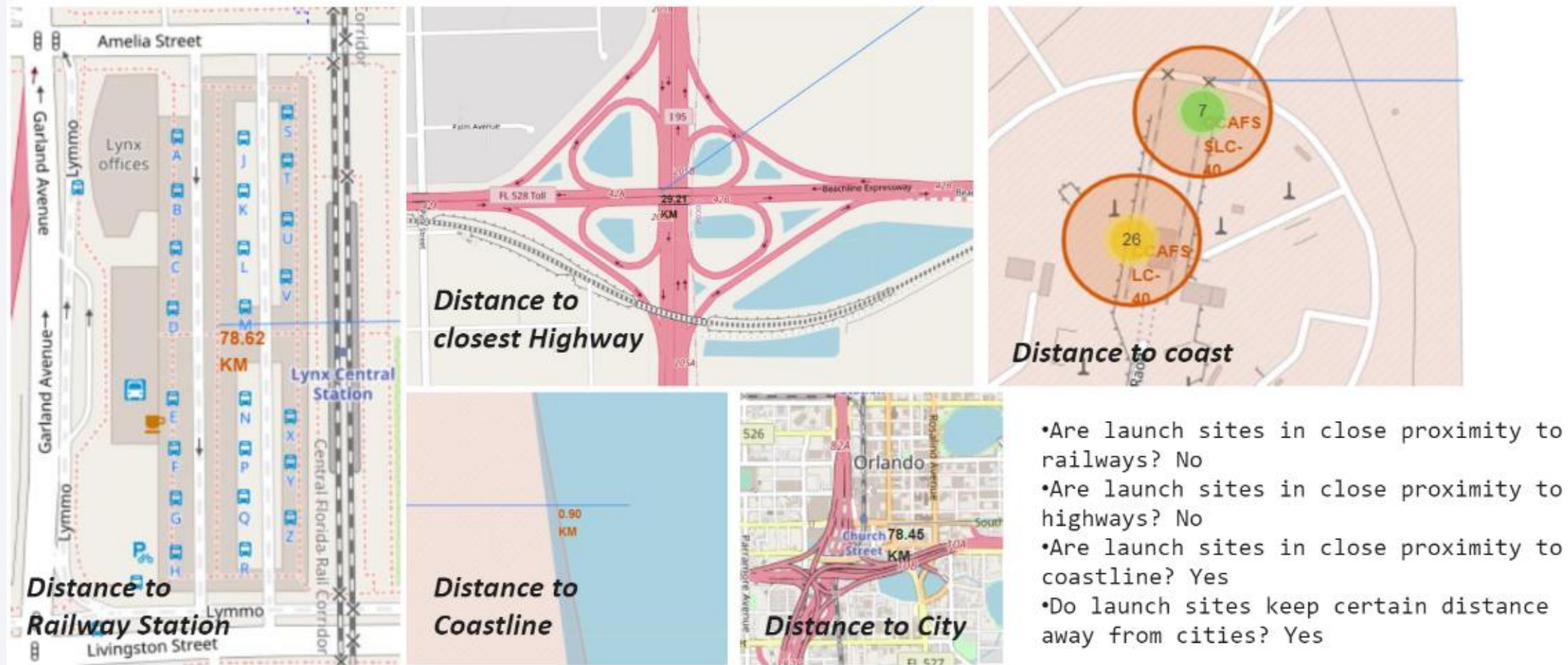
All launch sites global map markers



Markers showing launch sites with color labels



Launch Site distance to landmarks





Section 4

Build a Dashboard with Plotly Dash

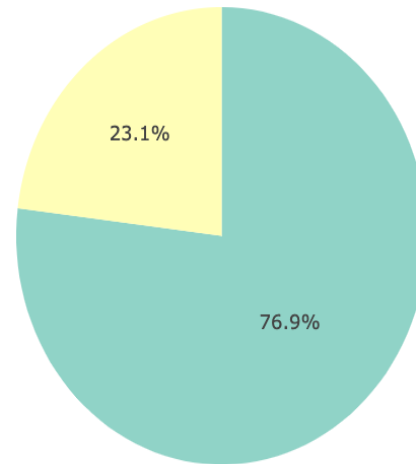
Pie chart showing the success percentage achieved by each launch site

Total Success Launch by sites



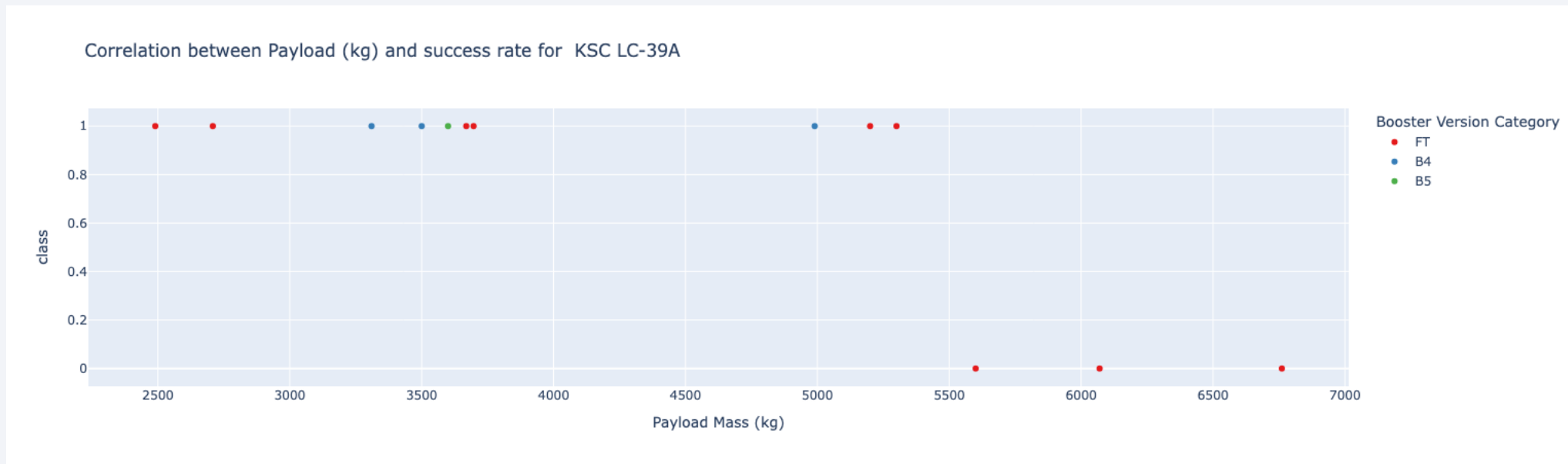
Pie chart showing the Launch site with the highest launch success ratio

Total Success Launches for site KSC LC-39A



1
0

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





Section 5

Predictive Analysis (Classification)

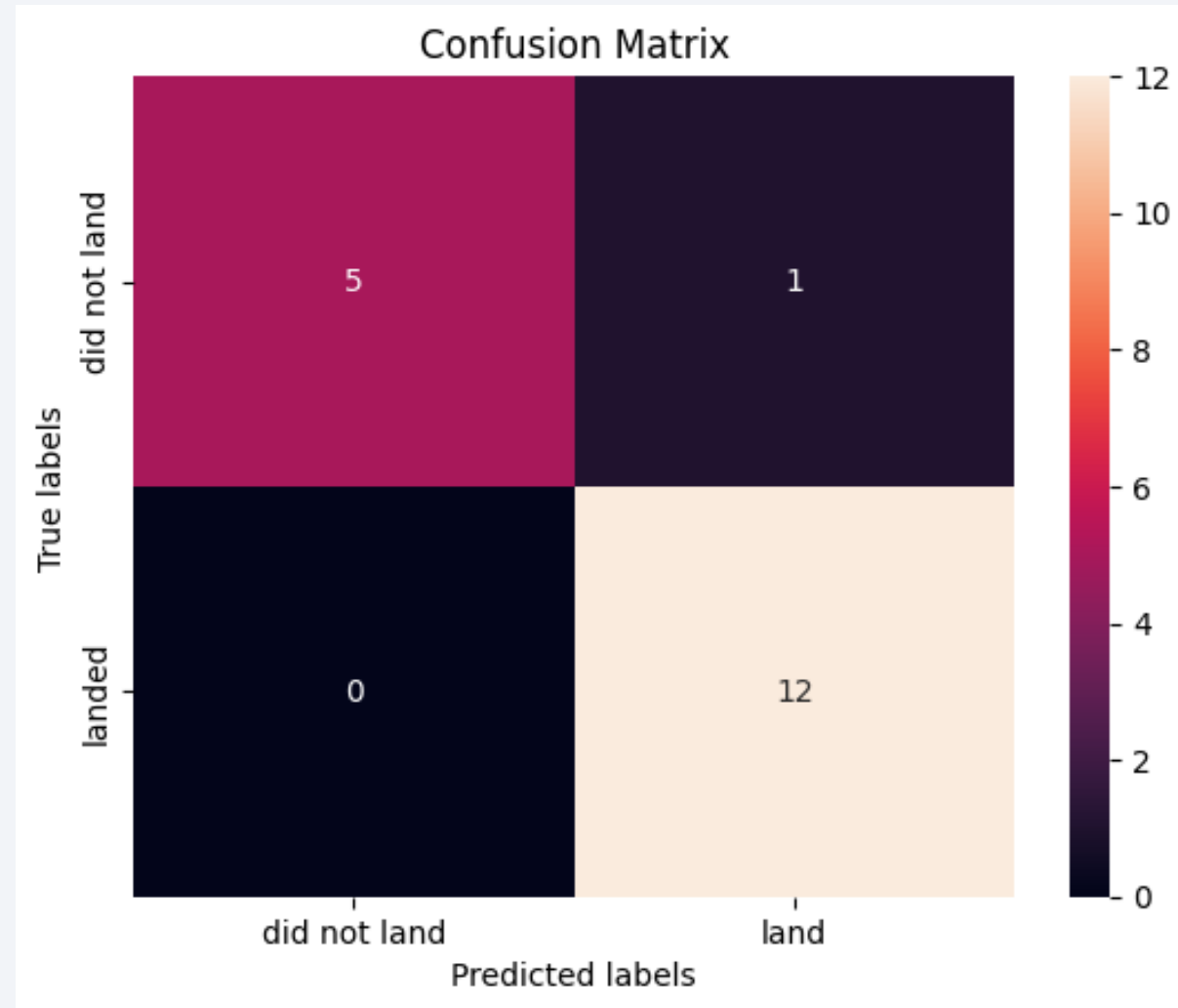
Classification Accuracy

```
1 accuracy = []
2 Method = []
3
4 Method.append('Logistic Regression')
5 Method.append('SVM')
6 Method.append('Decision Tree')
7 Method.append('KNN')
8
9 accuracy.append(Score_log)
10 accuracy.append(Score_svm)
11 accuracy.append(score_tree)
12 accuracy.append(score_knn)
13
14 print (accuracy)
15 print (Method)
```

[0.8333333333333334, 0.8333333333333334, 0.9444444444444444, 0.8333333333333334]
['Logistic Regression', 'SVM', 'Decision Tree', 'KNN']

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- [**https://github.com/maryam-asadi-coder**](https://github.com/maryam-asadi-coder)

Thank you!

