IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Giovana Osorio Segovia
10/06/2025

Outline

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

Executive Summary

- Data integration with SpaceX API and web scraping.
- Data cleansing and EDA with visualizations.
- Advanced SQL queries.
- Interactive maps with Folium.
- Interactive dashboards (Dash + Plotly).
- Predictive modeling (classification) with tuning and evaluation.

Introduction

- Develop a complete pipeline from data acquisition to modeling with SpaceX data.
- The main objective of this project is to demonstrate data science skills using Python and modern tools.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Flow: SpaceX REST API $\longrightarrow$ pandas.
- API call capture and JSON structure.
- Flowchart (API $\longrightarrow$ DataFrame).
- Link: notebooks/spacex_api_calls.ipynb.

Data Collection – SpaceX API

- GitHub URL: 1.0 jupyter-labs-spacex-data-collection-api.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

[9] ✓ 0.0s

We should see that the request was successfull with the 200 status response code

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

[32] ✓ 1.4s

```
response.status_code
```

[33] ✓ 0.0s

```
200
```

Data Collection - Scraping

- Technique: BeautifulSoup + requests.
- Flowchart: web → structured data.
- GitHub URL: 1.1 jupyter-labs-webscraping.ipynb

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

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

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

```
# Use soup.title attribute
print(soup.title)
```

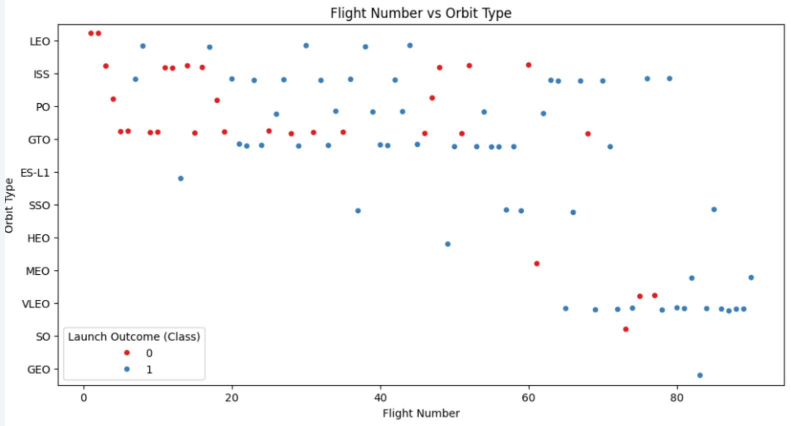
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

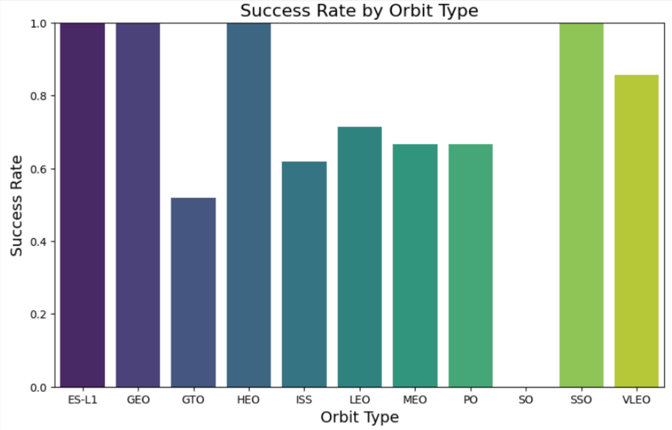
Data Wrangling

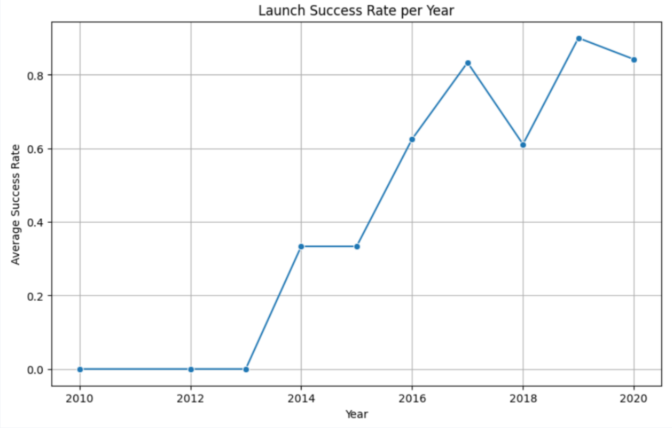
- Cleanup: duplicate removal, null handling, column renaming.
- Changed class to launch_success.
- GitHub URL: 1.2 labs-jupyter-spacex-Data wrangling.ipynb

EDA with Data Visualization

- With scatter point chart, we see if there is any relationship between FlightNumber and Orbit type.
- With bar chart, we visually check if there are any relationship between success rate and orbit type.
- And with line chart, to get the average launch success trend.
- GitHub URL: 2.2 edadataviz.ipynb







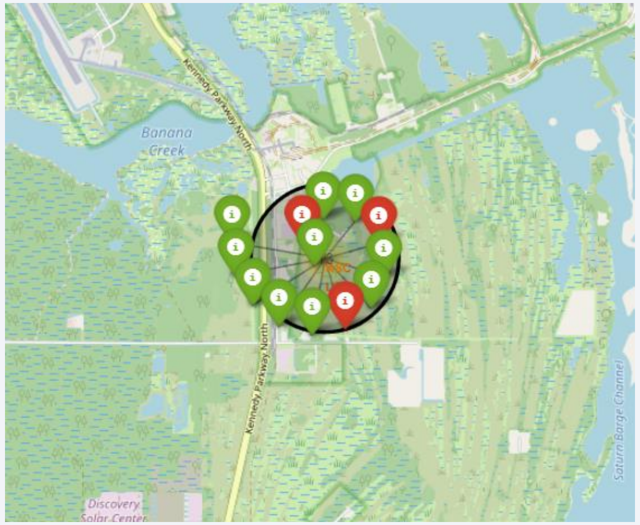
EDA with SQL

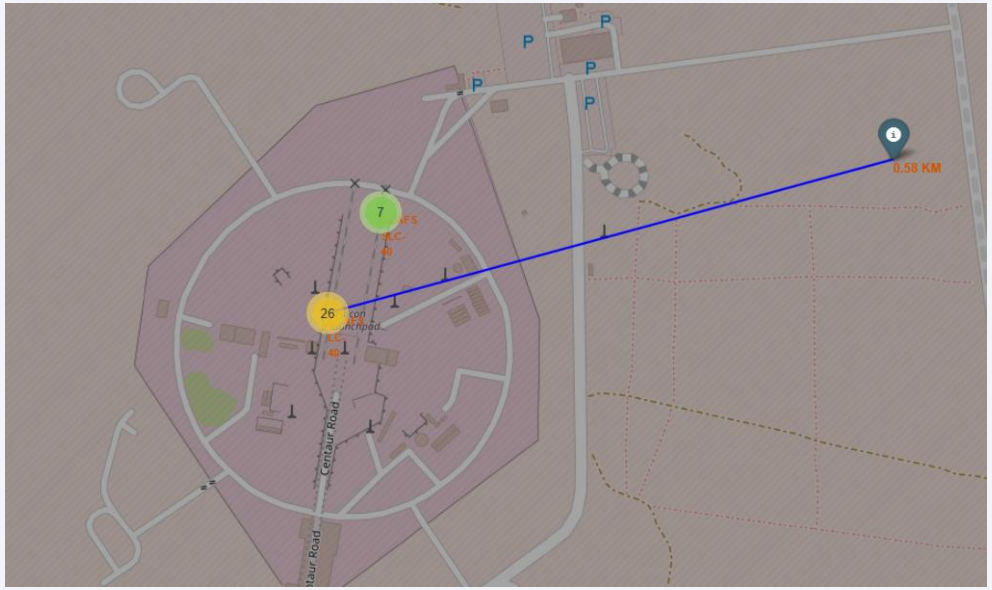
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%';
%sql SELECT AVG("Payload_Mass__kg_") AS Avg_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success%ground pad%';

GitHub URL: 2.1 jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

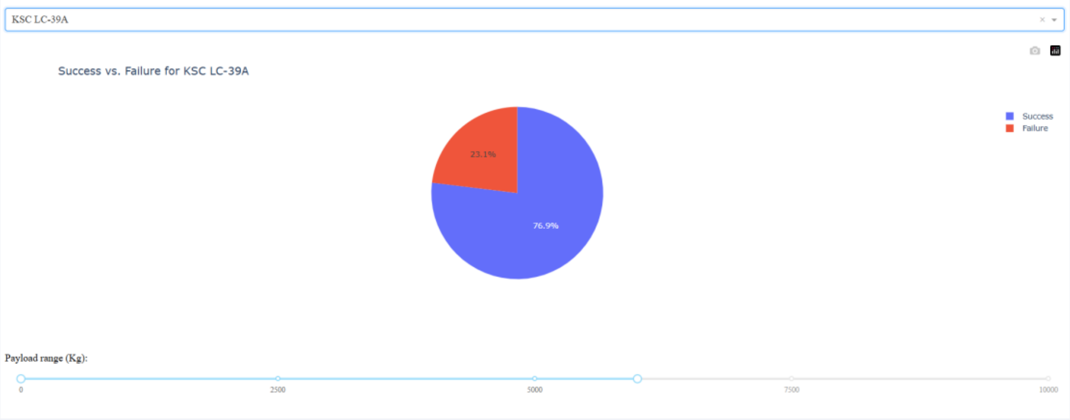
- I created a marker to indicate whether the launch was successful or a failure, also added a PolyLine from the launch site to the shore.
- GitHub URL: 3.1 lab_jupyter_launch_site_location.ipynb

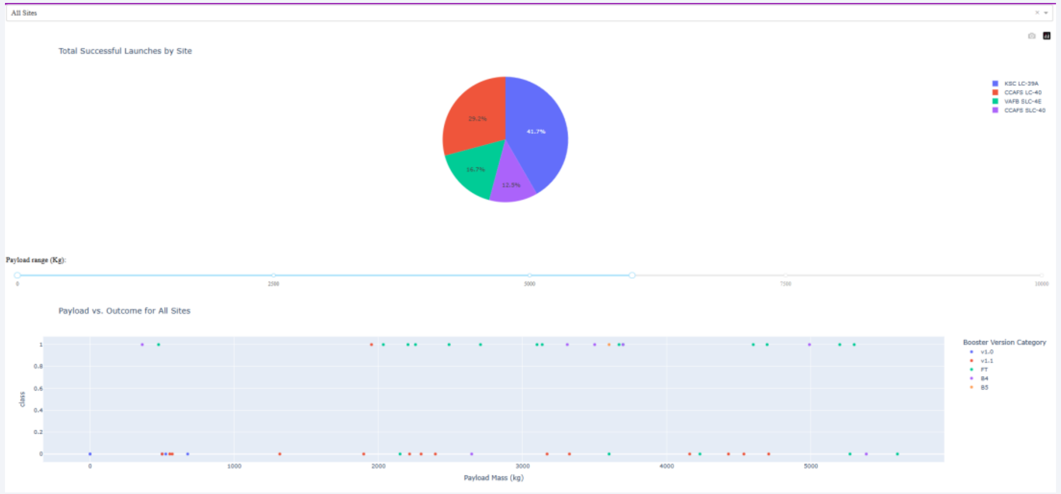


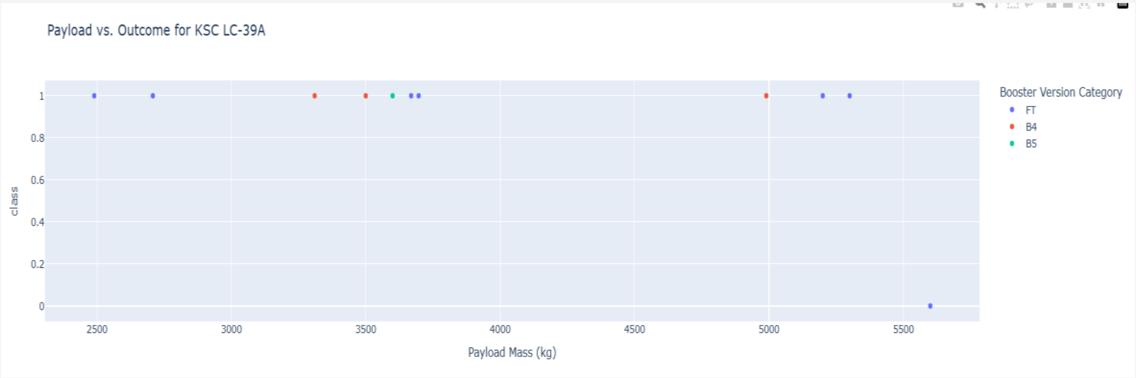


Build a Dashboard with Plotly Dash

- Dropdown de sitios.
- Pie charts (total éxitos, éxitos vs fallos).
- Slider y scatter payload/outcome.
- GitHub URL: 3.2 spacex-dash-app.py







Predictive Analysis (Classification)

- Modelos: Regresión logística, Random Forest, más.
- Etapas: selección de features, división, tuning.
- Resultados: tabla comparativa de precisión/RF, matriz de confusión del mejor.
- GitHub URL: 4.0 SpaceX_Machine Learning Prediction_Part_5.ipynb

Find the method performs best:

```
# Diccionario con los resultados
results = {
    'Logistic Regression': test_accuracy,
    'SVM': test_accuracy_svm,
    'Decision Tree': test_accuracy_tree,
    'KNN': test_accuracy_knn
}

# Mostrar todos los resultados
for model, acc in results.items():
    print(f"{model}: {acc:.4f}")

# Encontrar el mejor modelo
best_model = max(results, key=results.get)
best_accuracy = results[best_model]

print(f"\n✅ The best performing model is: {best_model} with accuracy = {best_accuracy:.4f}")
```

```
Logistic Regression: 0.8333
SVM: 0.8333
Decision Tree: 0.6667
KNN: 0.8333

✅ The best performing model is: Logistic Regression with accuracy = 0.8333
```

Results

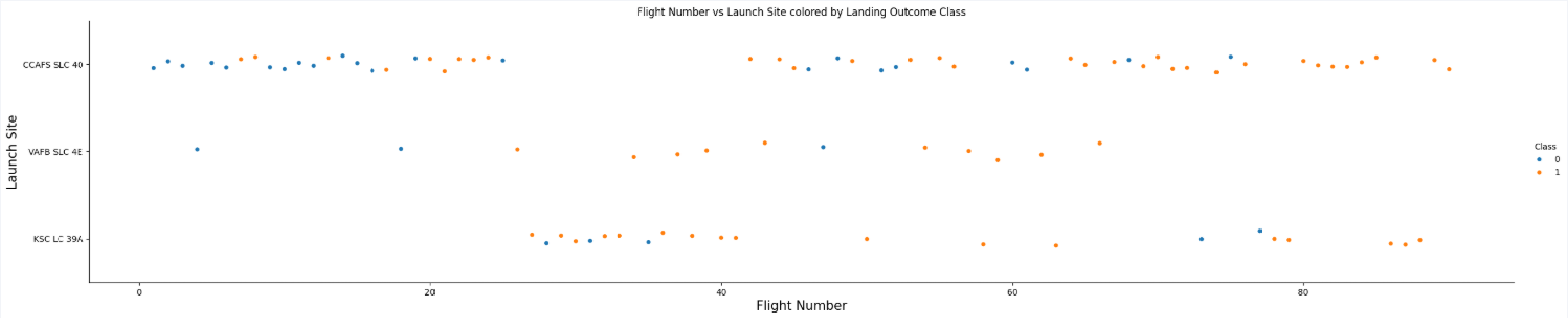
- Aprendizajes: integración de API/web, EDA, visualización, interacción, modelling.
- Retos y cómo se superaron.
- Siguientes pasos: despliegue en servidor, modelos avanzados, colaboraciones.

Section 2

Insights drawn from EDA

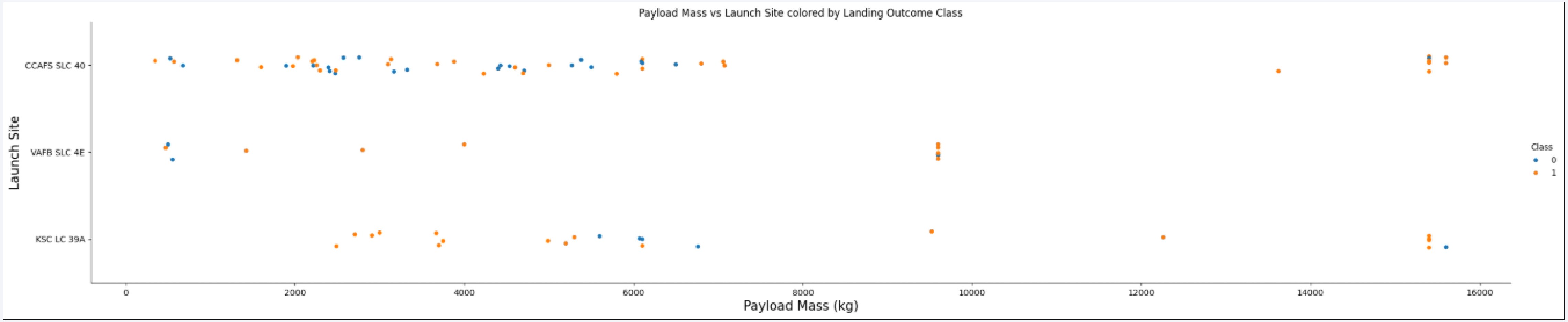
Flight Number vs. Launch Site

Depending on the location and the number of flight numbers, success increases



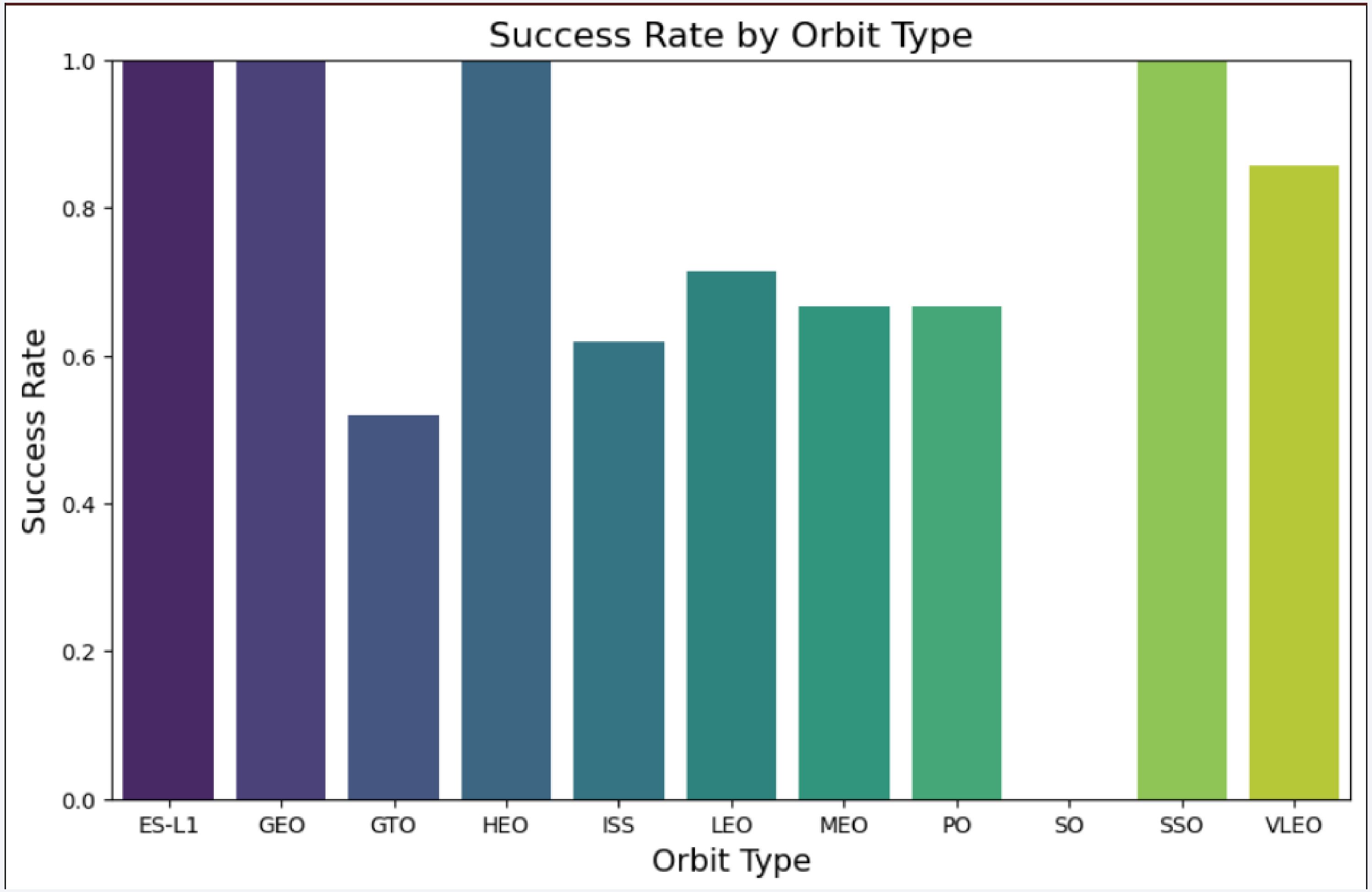
Payload vs. Launch Site

The VAFB-SLC launchsite there are no rockets launched for heavypayload mass, greater than 10000.



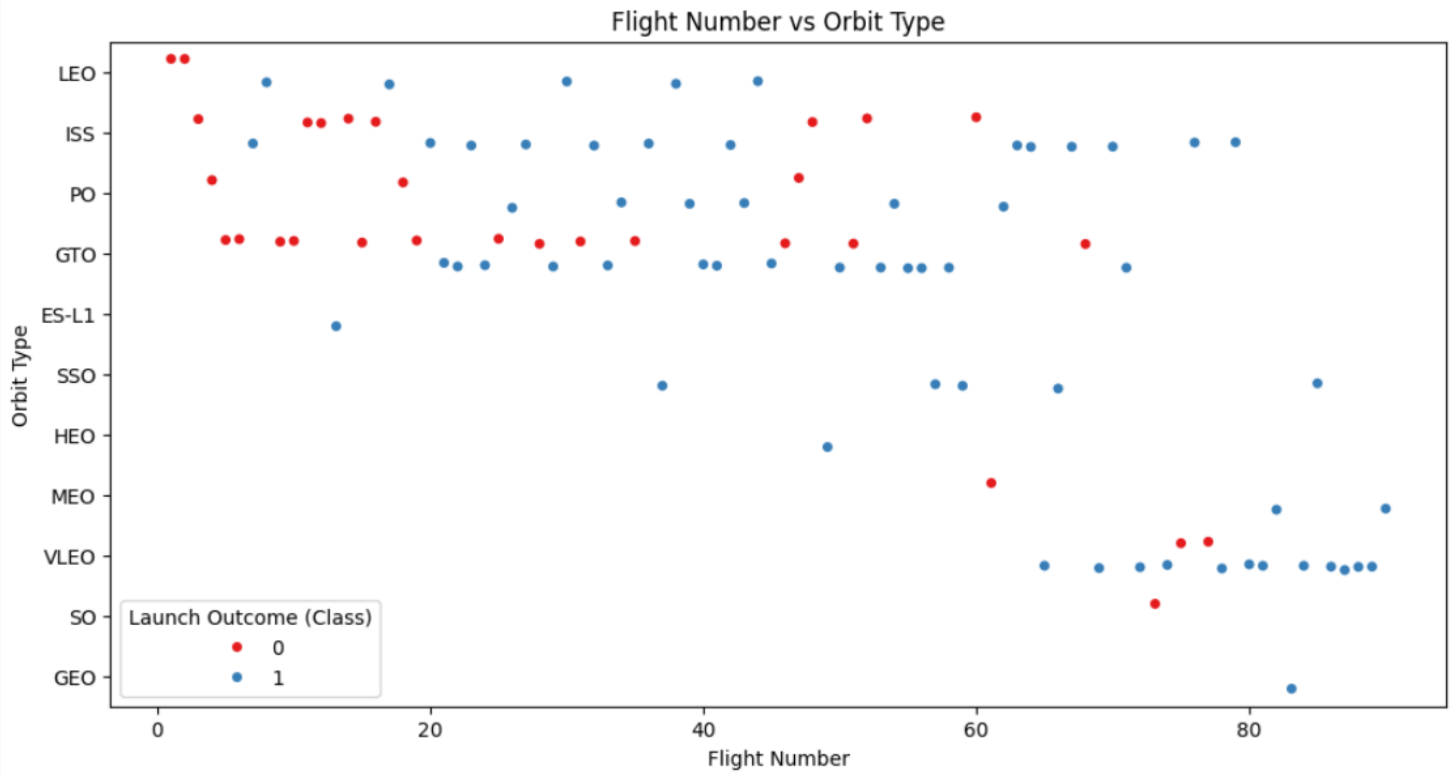
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO and VLEO have the best success rate.



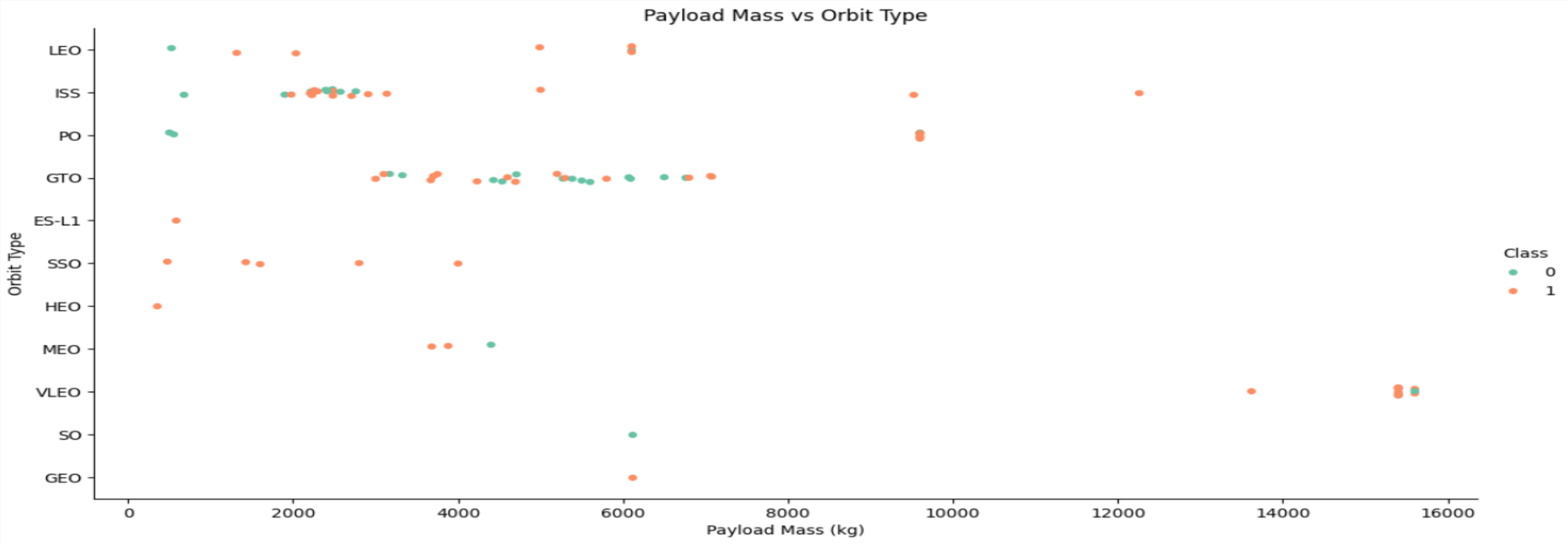
Flight Number vs. Orbit Type

- LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



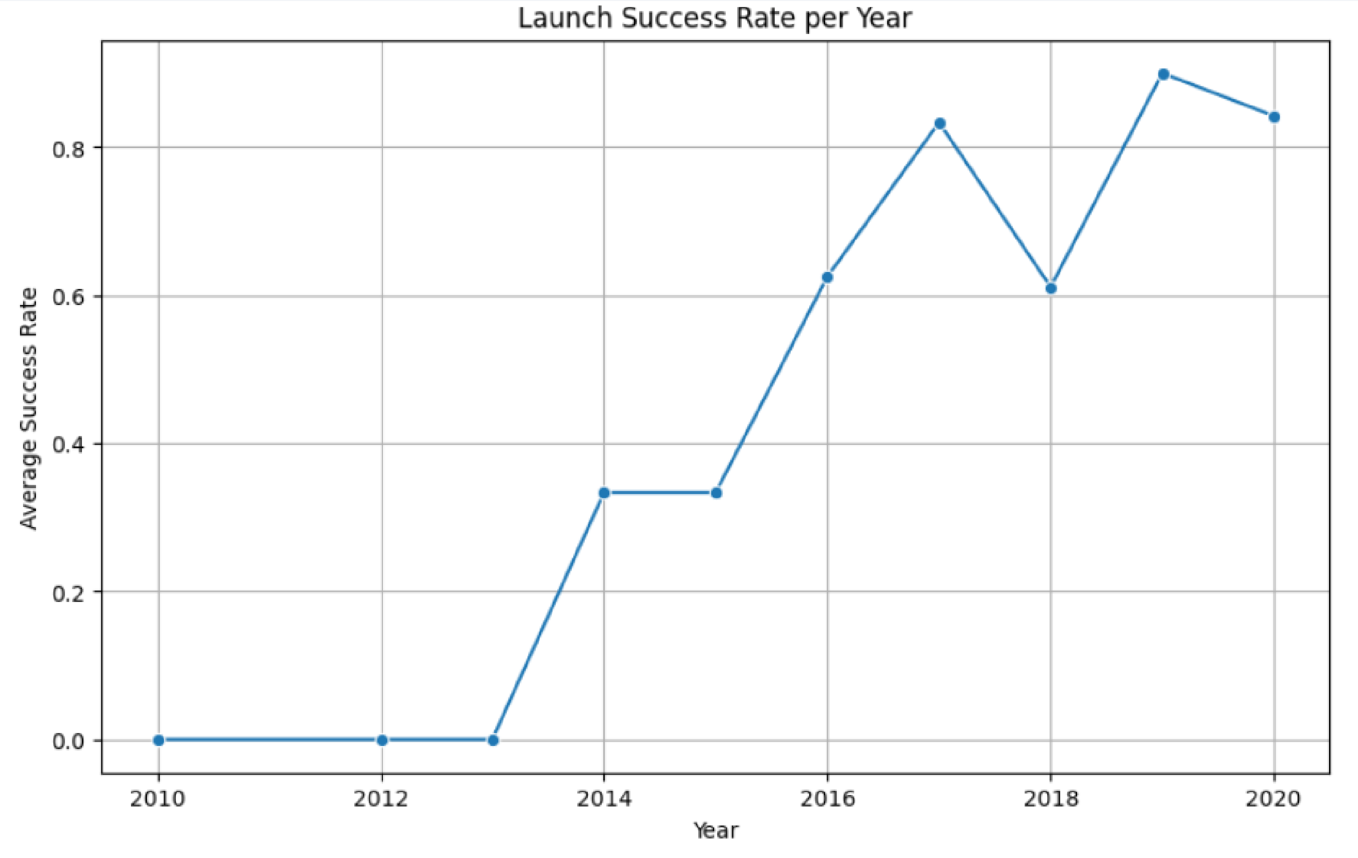
Payload vs. Orbit Type

- The successful landing or positive landing rate are more for Polar, LEO and ISS.



Launch Success Yearly Trend

- The sucess rate since 2013 kept increasing till 2020



All Launch Site Names

These are the names of the unique launch sites on the space misión.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

```
 * 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'

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

```
 * 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

The first 5 records where launch sites begin with the string 'CCA' are displayed.

Total Payload Mass

The total payload mass carried by NASA-launched boosters (CRS) is shown.

```
%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

Total_Payload_Mass
48213

Average Payload Mass by F9 v1.1

The average payload mass carried by the F9 v1.1 version of the booster is shown.

```
%sql SELECT AVG("Payload_Mass__kg_") AS Avg_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

Avg_Payload_Mass
2928.4

First Successful Ground Landing Date

The date on which the first successful landing on the ground platform was achieved.

```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success%ground pad%';
```

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

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listed are the names of propulsors that are successful on unmanned vessels and have a payload mass greater than 4000 but less than 6000. For this case, there are none.

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success%drone ship%' AND "Payload_Mass" > 4000 AND "Payload_Mass" < 6000;
```

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

Booster_Version

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failed mission outcomes is listed

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Total_Missions FROM SPACEXTABLE GROUP BY "Landing_Outcome";
```

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

Landing_Outcome	Total_Missions
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- Lists all booster_versions that have carried the maximum payload mass.

```
%sql SELECT booster_version, "Payload_Mass__kg_" FROM SPACEXTABLE WHERE Payload_Mass__kg_ = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);
```

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

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

2015 Launch Records

- Listed are records showing month names, results of failed landings on the unmanned spacecraft, booster versions, and the launch site for the months of 2015.

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

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

- The landing result count (as Failure (unmanned craft) or Success (ground platform) between the date of June 4, 2010, and March 20, 2017, is ranked in descending order.

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

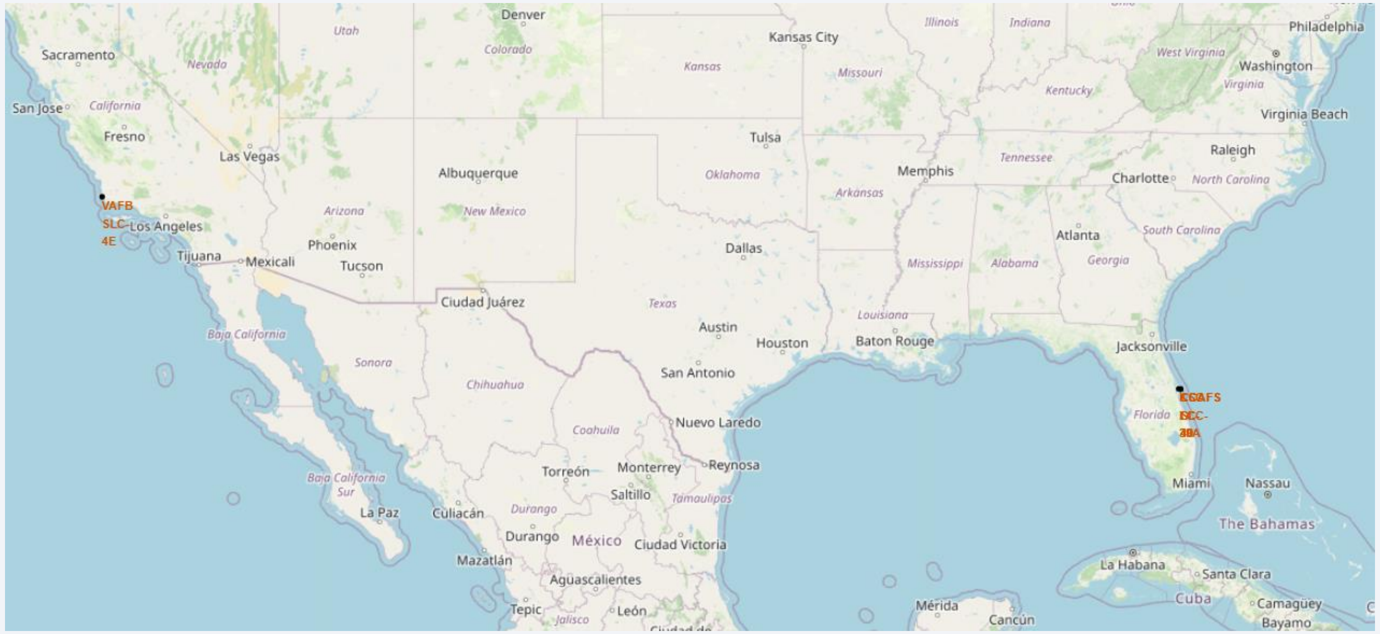
Section 3

Launch Sites Proximities Analysis

All launch sites on a map

- The location of each launch site is displayed on a map using the site's latitude and longitude coordinates.

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745



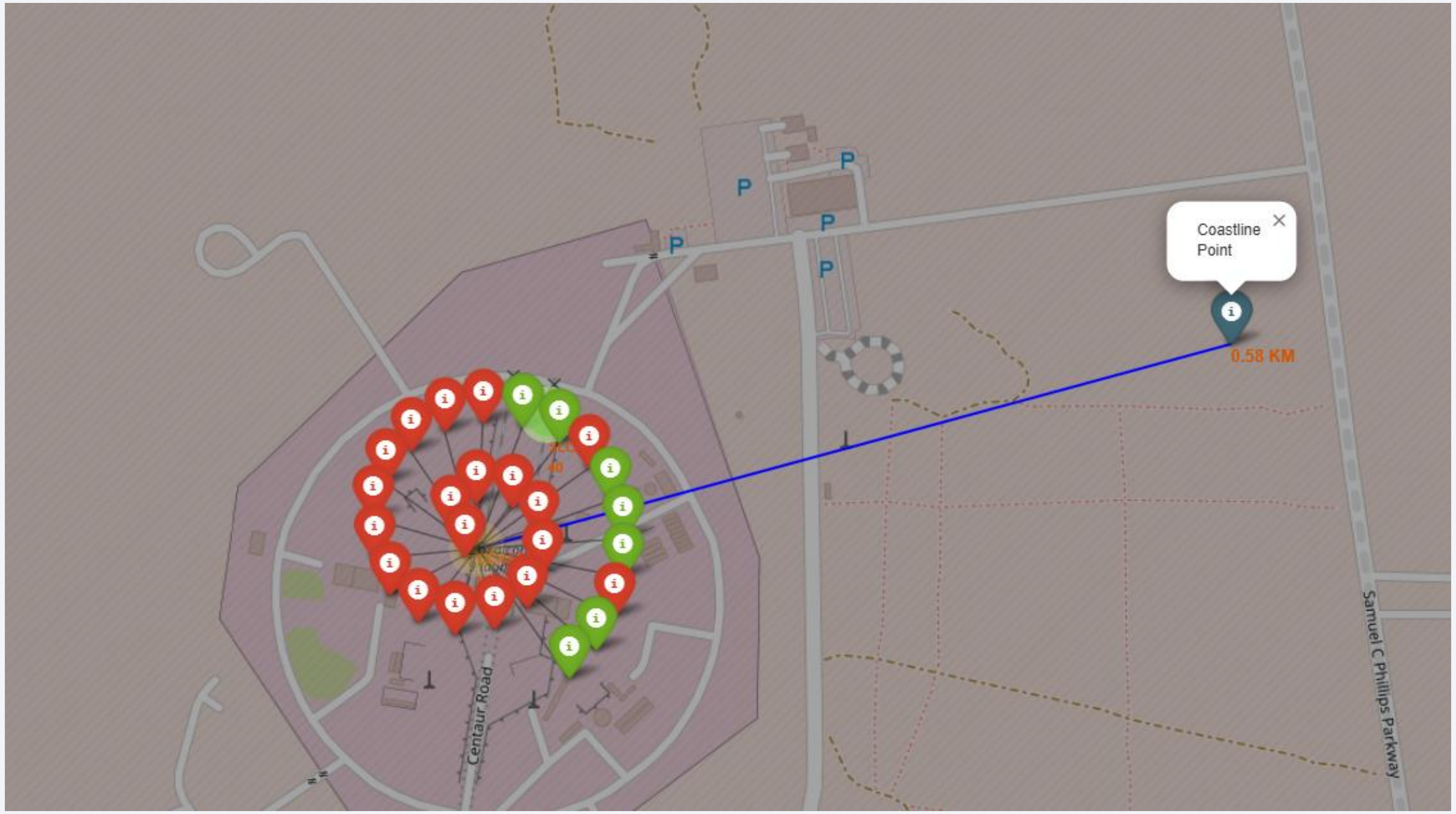
The launch results for each site are aggregated, and a list of sites with high success rates is shown. It's important t

Successful and failed launches of each site on the map

- The launch results for each site are aggregated, and a list of sites with high success rates is shown. It's important to remember that the spacex_df data frame contains detailed launch records, and the class column indicates whether the launch was successful or not.

Distances between a launch site to its proximities

- In this case, we see the distance from a launch site to a coastline.

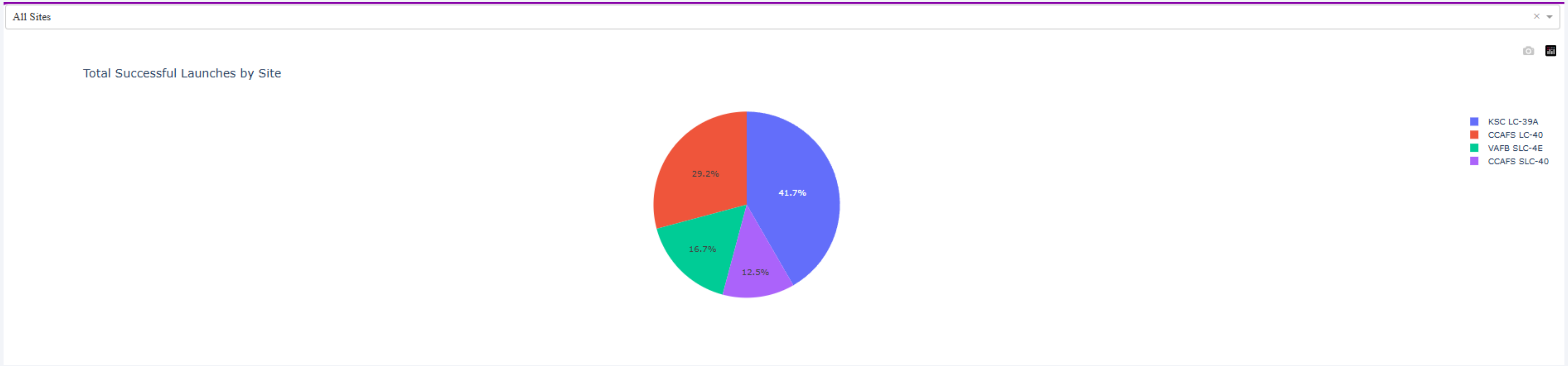


Section 4

Build a Dashboard with Plotly Dash

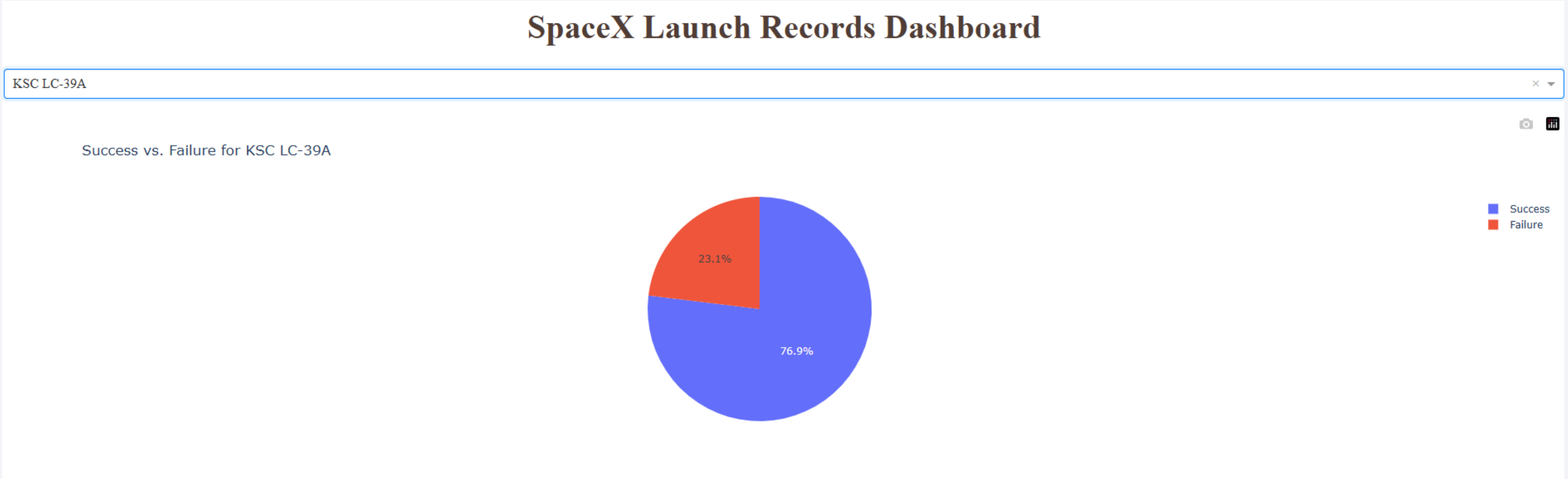
Dashboard piechart for all sites

- In this pie chart, we can see each site with their respective percentages of successful launches.

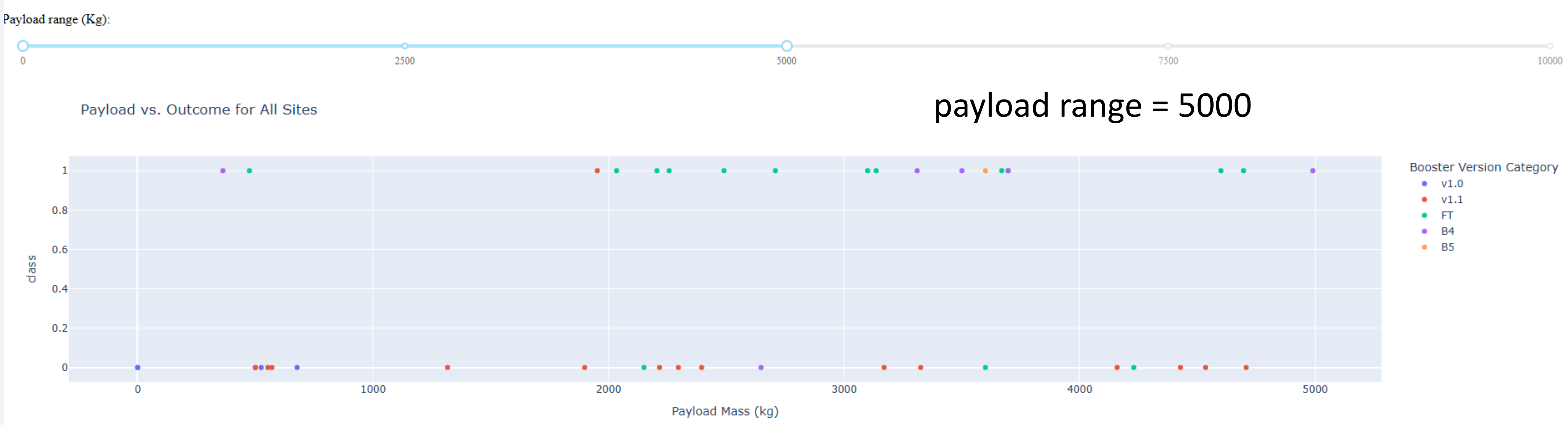


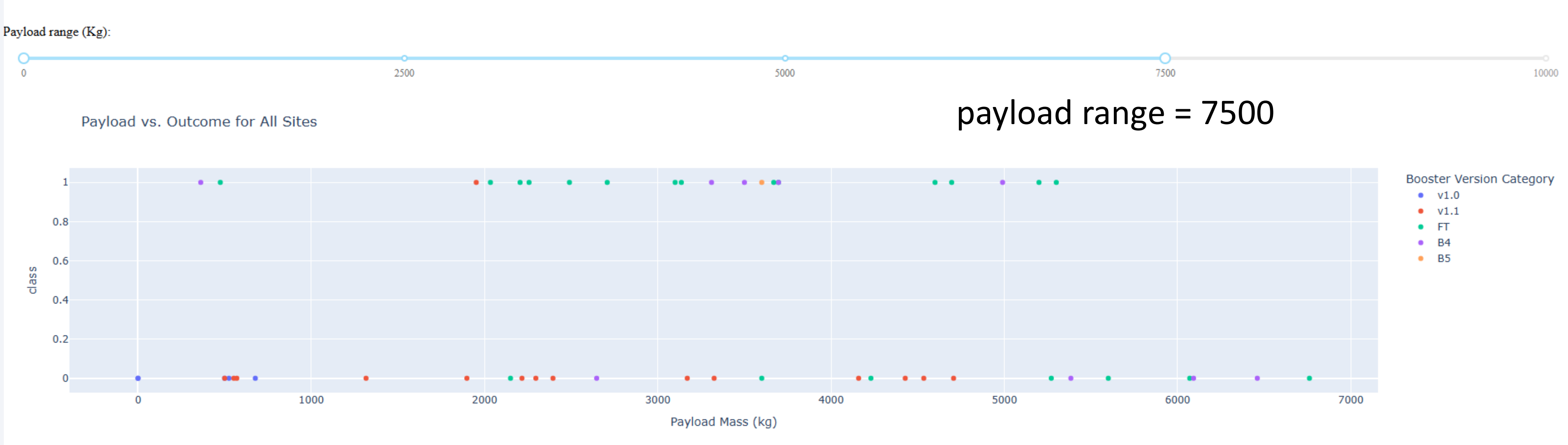
The highest launch success ratio

- Here we can see the pie chart with the launch with the highest success rate in launches.



Payload vs Outcome for All Sites

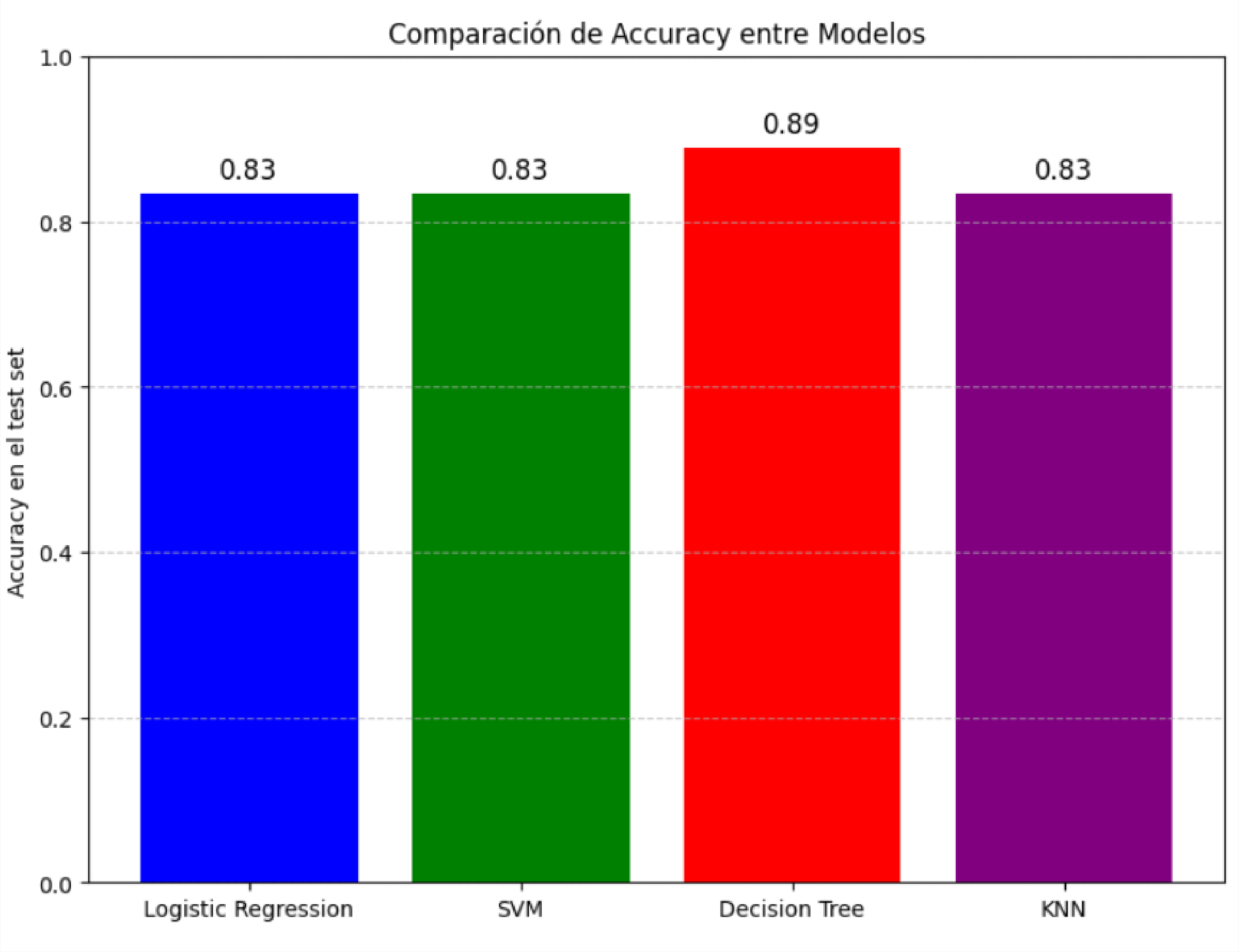




Section 5

Predictive Analysis (Classification)

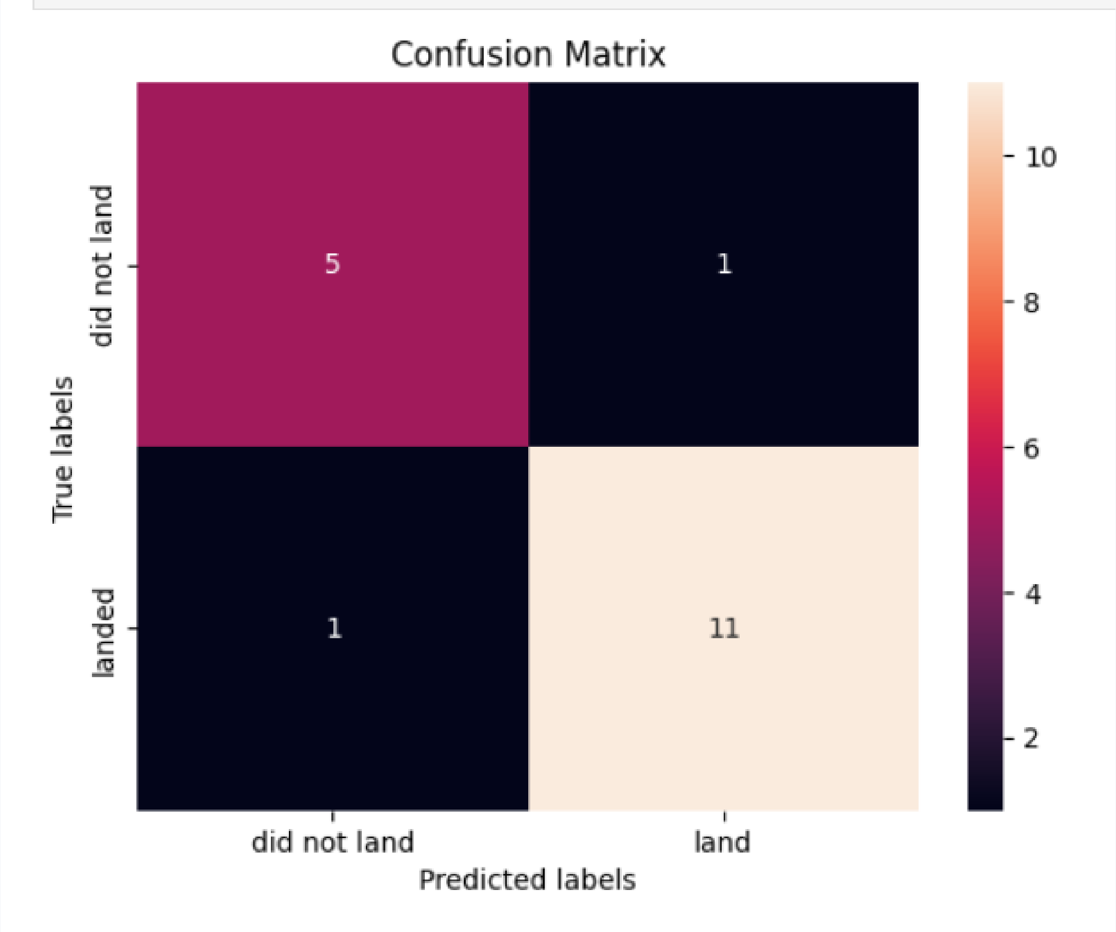
Classification Accuracy



- The model with the highest classification accuracy is the Decision Tree.

Confusion Matrix

- A higher True Positive count (11) suggests the model is fairly good at predicting successful landings.
- The presence of False Positives (1) and False Negatives (1) indicates some misclassifications, though the error rate appears low.



Conclusions

- For this project, I used different syntaxes and worked with different models.
- Data science is definitely an area where you need to dedicate a lot of time to manipulating and analyzing data to present it in this PowerPoint presentation.

Appendix

```
# Show the head of the dataframe
print(launch_df.head())
```

```
   FlightNumber        Date BoosterVersion  PayloadMass Orbit  \
0             1  2006-03-24       Falcon 1         20.0   LEO
1             2  2007-03-21       Falcon 1          NaN   LEO
2             4  2008-09-28       Falcon 1        165.0   LEO
3             5  2009-07-13       Falcon 1        200.0   LEO
4             6  2010-06-04       Falcon 9          NaN   LEO

        LaunchSite    Outcome  Flights  GridFins  Reused   Legs LandingPad  \
0  Kwajalein Atoll  None None        1     False   False  False       None
1  Kwajalein Atoll  None None        1     False   False  False       None
2  Kwajalein Atoll  None None        1     False   False  False       None
3  Kwajalein Atoll  None None        1     False   False  False       None
4     CCSFS SLC 40  None None        1     False   False  False       None

   Block  ReusedCount    Serial   Longitude   Latitude
0    NaN            0  Merlin1A  167.743129   9.047721
1    NaN            0  Merlin2A  167.743129   9.047721
2    NaN            0  Merlin2C  167.743129   9.047721
3    NaN            0  Merlin3C  167.743129   9.047721
4    1.0            0     B0003  -80.577366  28.561857
```

```
data_falcon9.isnull().sum()
```

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

Thank you!