IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gabriel Barros Nogueira
São Paulo, Brazil, 2023/03/21

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API
  - Data Collection with web scraping (beautifulsoup)
  - Data wrangling: transforming raw data into a format suitable for analysis(cleaning, organizing, integrating, enriching, transforming and formatting data)
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization (Folium)
  - Machine Learning Prediction (Classification – SVM, Classification Trees and Logistic Regression)

- **Summary of all results**
  - Exploratory Data Analysis result
  - Interactive analytics and dashboard
  - Predictive models for classification

Introduction

An overview of a machine learning project aimed at predicting the cost of rocket launches by determining if SpaceX will reuse the first stage of their Falcon 9 rocket. Space X 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 saving is because Space X can reuse the first stage. The highlights of the success of SpaceX in the commercial space industry, particularly in their ability to reuse the first stage of their rocket and lower the cost of launches.

The project scenario involves working as a data scientist for a new rocket company, Space Y, competing with SpaceX. The goal is to create dashboards and train a machine learning model to predict if the first stage of Falcon 9 will be successfully reused, thereby determining the cost of the launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX APIs: rockets, launchpads and payloads

    and web scraping from Wikipedia
- Perform data wrangling
  - For data wrangling, was calculate the number of launches on each site, and investigate the number and occurrence of each orbit and created a landing outcome label from outcome column.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Created a column for the class, standardized the data, splited into training and test, looking into confusion matrix and accuracy metrics, finding the best hyperparameters.

Data Collection

- Data was collected by APIs from SpaceX, using json to do connections.

https://api.spacexdata.com/v4/rockets/ → BoosterVersion

https://api.spacexdata.com/v4/launchpads → LaunchSite

https://api.spacexdata.com/v4/payloads/ → PayloadData

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

↓

BoosterVersion, PayloadMass ,Orbit ,LaunchSite,Outcome ,Flights
GridFins ,Reused ,Legs,LandingPad ,Block ,ReusedCount ,Serial ,Longitude ,Latitude

Data Collection – SpaceX API

*https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Getting response from API

```
In [106]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_
```

Converting Response to a .json file

```
In [108]: # faz a solicitação da API
response = requests.get(static_json_url)
# verifica se a solicitação foi bem sucedida
if response.status_code == 200:
    # decodifica a resposta em formato JSON
    json_data = response.json()
```

```
    # normaliza os dados do JSON em um DataFrame
    data = pd.json_normalize(json_data)
else:
    # se a solicitação não foi bem sucedida, imprime o status da resposta
    print('Erro:', response.status_code)
```

Assign list to a dictionary then dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

- Data was collected by APIs from SpaceX, using json to do connections.
- Data collection with SpaceX REST calls

```
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
data_falcon9.to_csv("Falcon9_part1.csv", index = False)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1

Data Collection - Scraping

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

2020 [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 △ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 △ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule;[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 △ B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 △ B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								
82	7 March 2020, 04:50[506]	F9 B5 △ B1059.2	CCAFS, SLC-40	SpaceX CRS-20 (Dragon C112.3 △)	1,977 kg (4,359 lb)[507]	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
	Last launch of phase 1 of the CRS contract. Carries *Bartolomeo*, an ESA platform for hosting external payloads onto ISS.[508] Originally scheduled to launch on 2 March 2020, the launch date was pushed back due to a second stage engine failure. SpaceX decided to swap out the second stage instead of replacing the faulty part.[509] It was SpaceX's 50th successful landing of a first stage booster, the third flight of the Dragon C112 and the last launch of the cargo Dragon spacecraft.								
83	18 March 2020, 12:16[510]	F9 B5 △ B1048.5	KSC, LC-39A	Starlink 5 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fifth operational launch of Starlink satellites. It was the first time a first stage booster flew for a fifth time and the second time the fairings were reused (Starlink flight in May 2019).[511] Towards the end of the first stage burn, the booster suffered premature shut down of an engine, the first of a Merlin 1D variant and first since the CRS-1 mission in October 2012. However, the payload still reached the targeted orbit.[512] This was the second Starlink launch booster landing failure in a row, later revealed to be caused by residual cleaning fluid trapped inside a sensor.[513]								
84	22 April 2020, 19:30[514]	F9 B5 △ B1051.4	KSC, LC-39A	Starlink 6 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)

Data Collection - Scraping

Getting response from HTML

```
html_data = requests.get(static_url)
html_data.status_code
```

```
Out[10]: 200
```

Creating BeatifulSoup Object

```
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Finding Tables

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

Getting column names

```
column_names = []

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_al
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is
        if rows.th:
            if rows.th.string:
```

Converting dictionary to dataframe

```
df = pd.DataFrame(launch_dict)
```

```
df.head()
```

ad ss	Orbit	Customer	Launch outcome	N/A	FH 2	FH 3	...	People	Vehicles	Launches by rocket type	Launches by spaceport	Agencies, companies and facilities	Other mission lists and timelines	Version Booster	Booster landing	Date	Time
0	LEO	SpaceX	Success\n	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0003.1	Failure	4 June 2010	18:45

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Within the dataset, there are various instances where the booster failed to land successfully. Some landing attempts were unsuccessful due to accidents; for example, "True Ocean" signifies a successful mission outcome of landing in a specific region of the ocean, while "False Ocean" means an unsuccessful mission outcome of landing in a specific region of the ocean. "True RTLS" indicates a successful mission outcome of landing on a ground pad, while "False RTLS" means an unsuccessful mission outcome of landing on a ground pad. "True ASDS" denotes a successful mission outcome of landing on a drone ship, while "False ASDS" signifies an unsuccessful mission outcome of landing on a drone ship. We primarily translate these outcomes into training labels, where "1" signifies a successful booster landing and "0" signifies an unsuccessful one.

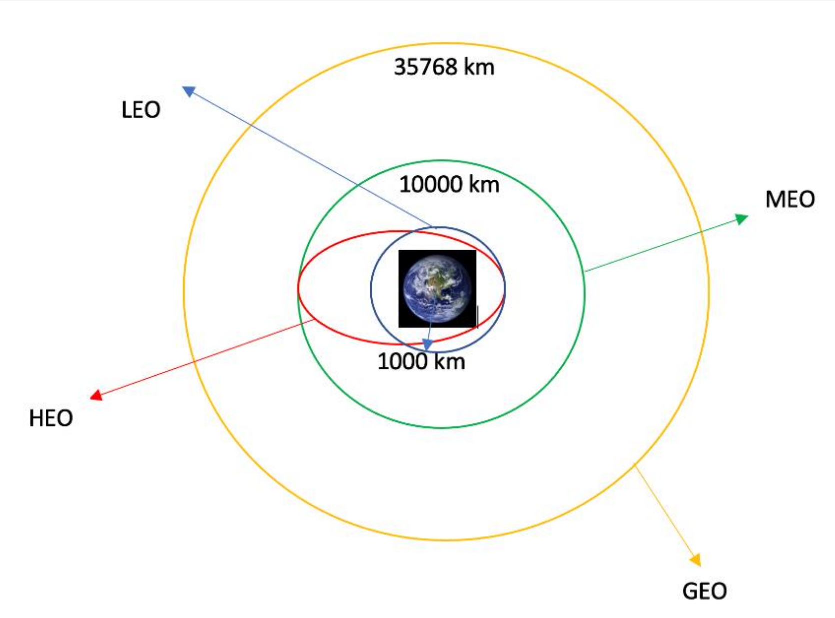


Diagram showing common orbit types SpaceX uses

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

Scatter plots being drawn:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload vs Launch Site
- Orbit vs Flight Number
- Payload vs Orbit Type
- Orbit vs Payload Mass

A scatter plot is a graphical representation of the relationship between two variables. It consists of a grid where each point on the graph represents a unique combination of values for the two variables. Scatter plots are useful for identifying patterns in the data and visualizing the relationship between the variables. By plotting the data points on the graph, it becomes easy to see if there is a correlation between the two variables

Bar graph being drawn:

- Mean vs Orbit

A bar graph is a visual representation of categorical data, which displays the frequency or proportion of each category in a dataset. It consists of a series of bars, where each bar represents a category and its height or length is proportional to the frequency or proportion of that category in the data. Bar graphs are useful for comparing the relative frequencies or proportions of different categories in a dataset. They can also be used to compare the same category across different groups or time periods.

Line graph being drawn:

- Succes Rate vs Year

A line graph is a type of graph used to represent data over time or some other continuous variable. In a line graph, data points are plotted on a Cartesian plane with the x-axis representing time or the continuous variable and the y-axis representing the value of the variable being measured.

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

For example of some questions we were asked about the data we needed information about.

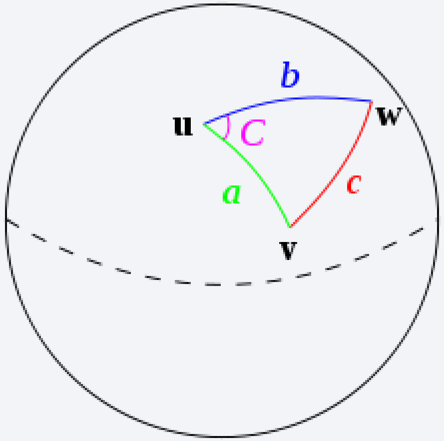
- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the 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

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a *Circle Marker around each launch site with a label of the name of the launch site.*

We assigned the dataframe launch_outcomes(failures, successes) to *classes:*
- ***0* -** Red markers
- ***1* -** Green markers on the map in a MarkerCluster()

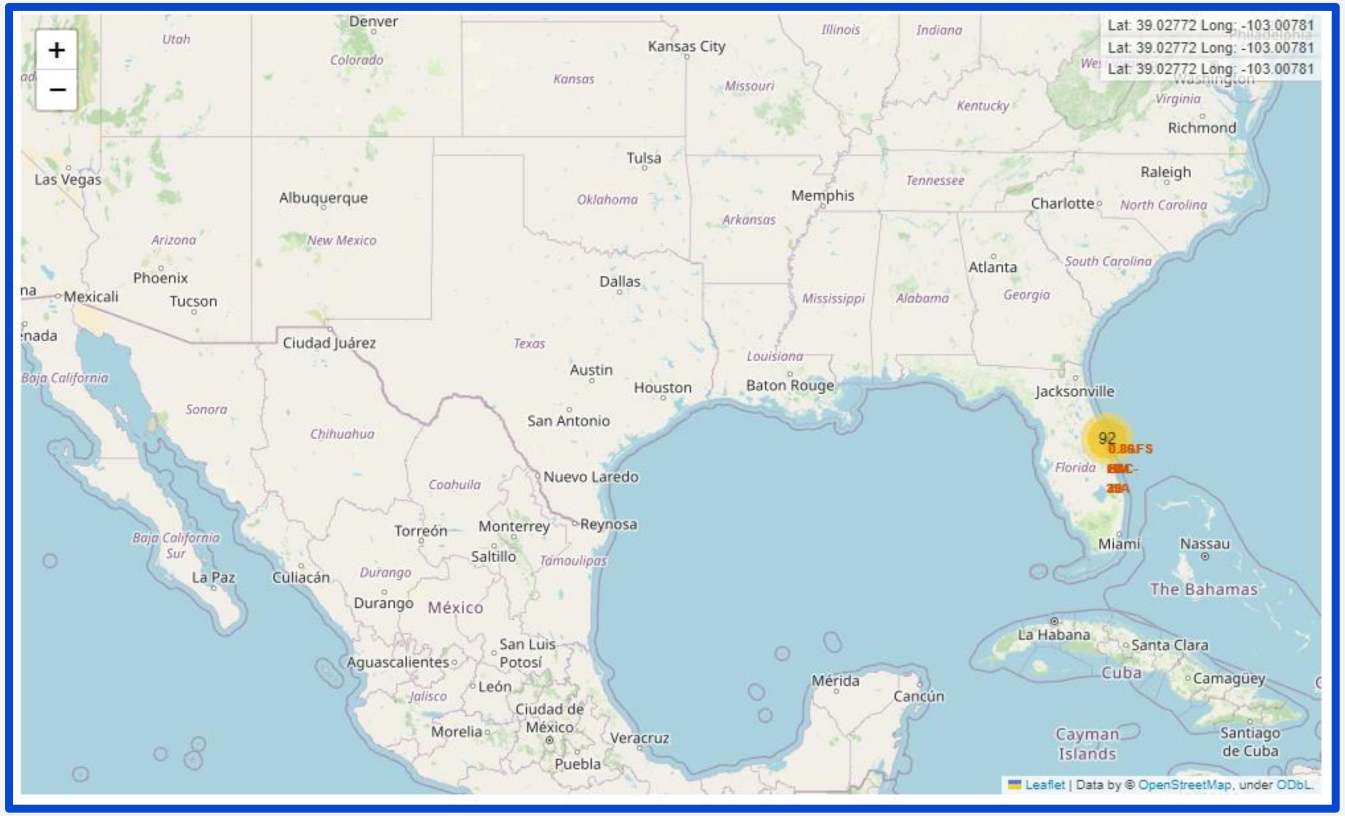


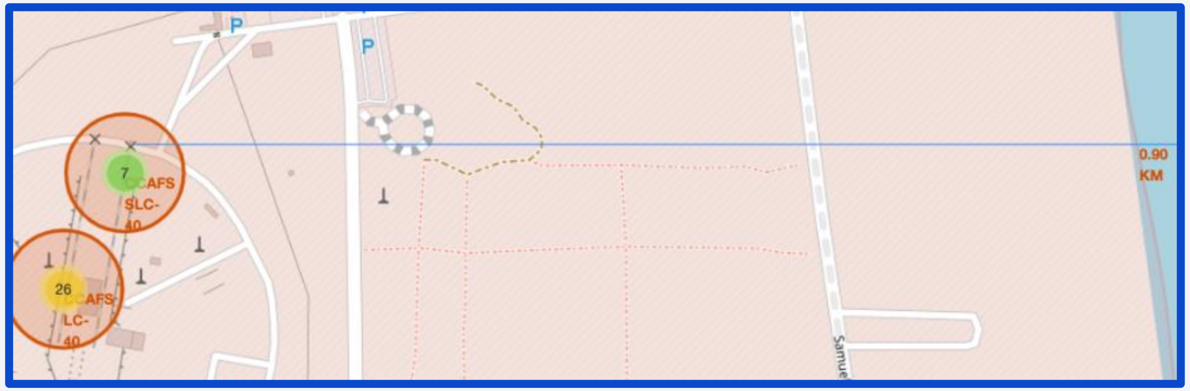
Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns.

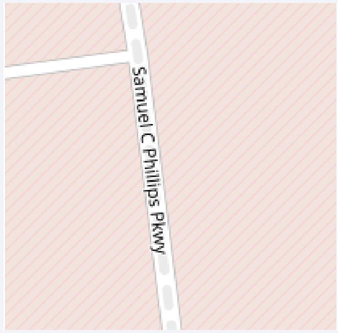
*The formula was developed by British mathematician William Haversian in 1859 for use in marine navigation, and is based on trigonometric principles. It is widely used in applications such as geolocation, GPS, geocoding, and mapping, and is considered one of the most accurate ways to calculate the distance between two points on a spherical surface.

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash







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.

http://localhost:8888/notebooks/Desktop/FInal_project_IBM/lab_jupyter_launch_site_location.jupyterlite.ipynb#

Predictive Analysis (Classification)

- Load our dataset into Numpy and Pandas dataframe
- Data Wrangling
- Split our data into training, validation and test sets
- Build diferente models using SVM, Decision tree and logistic regression

- Check the accuracy and F-score for each model
- Get tuned hyperparameters for each type of algorithms
- Plot confusion Matrix

https://github.com/Gabrielbnog/Final_Project_DS_IBM/blob/main/SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

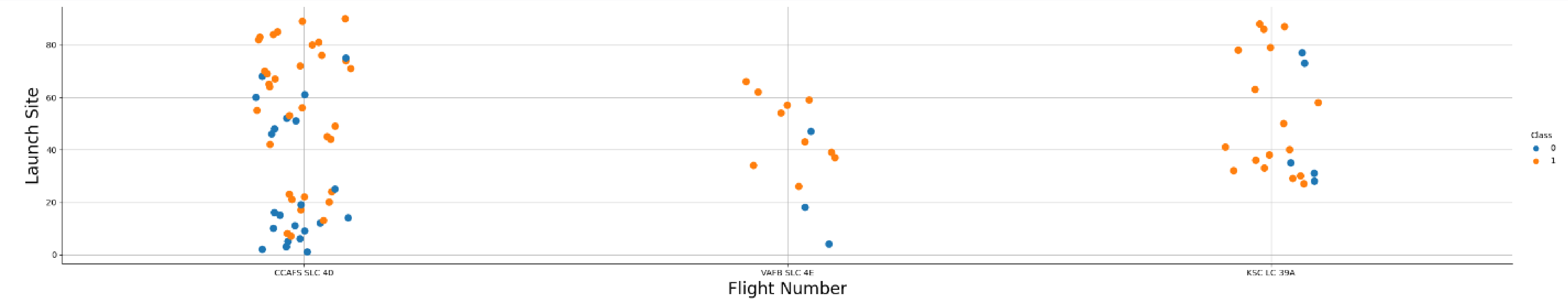
Results

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

Section 2

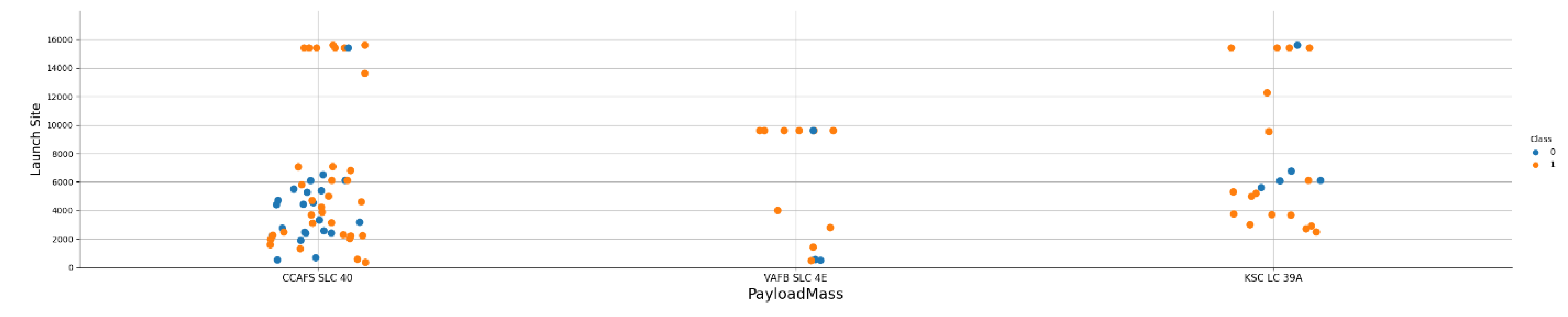
Insights drawn from EDA

Flight Number vs. Launch Site



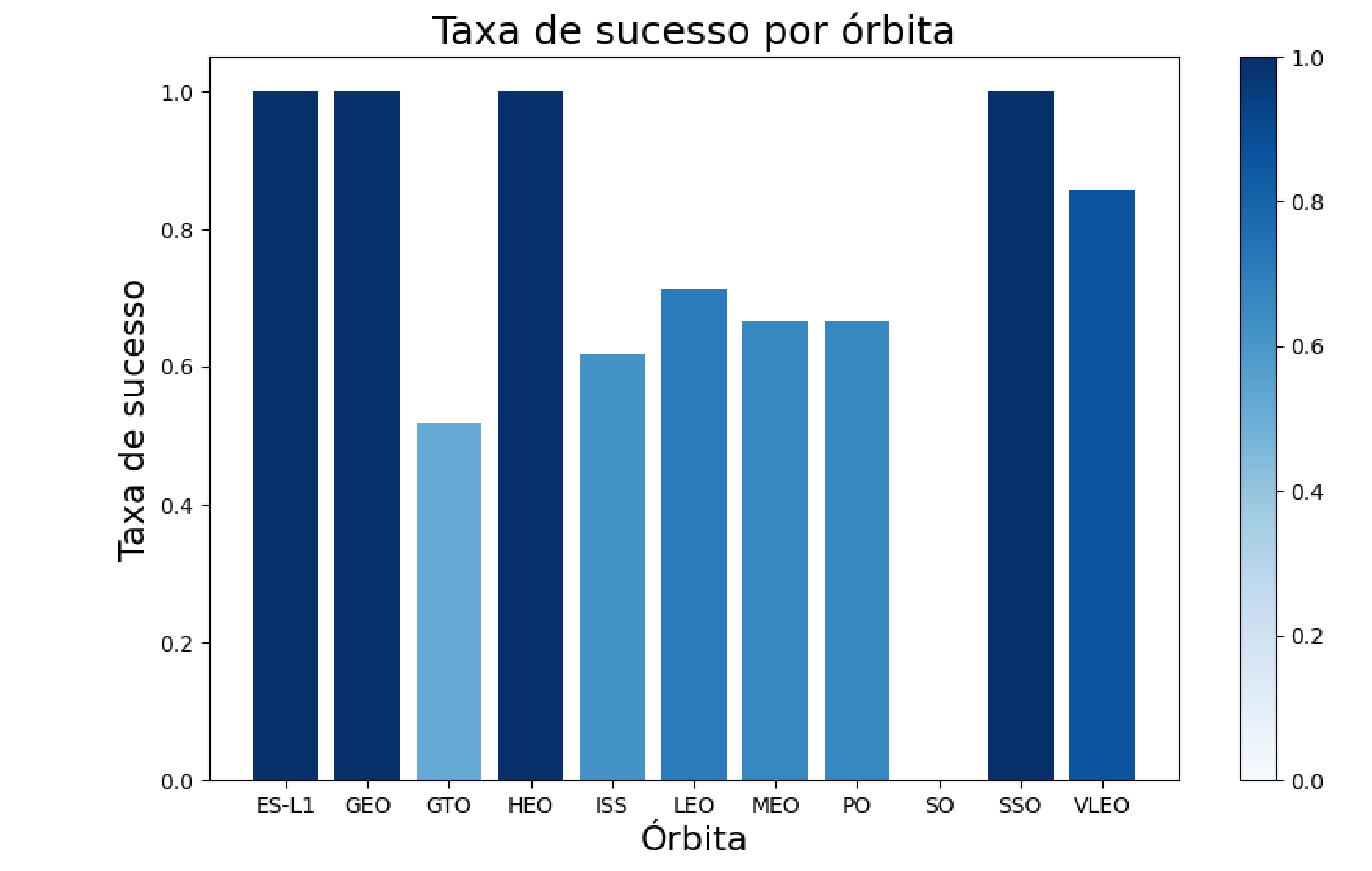
The more amount of flights at a launch site the greater the success rate at a launch site.

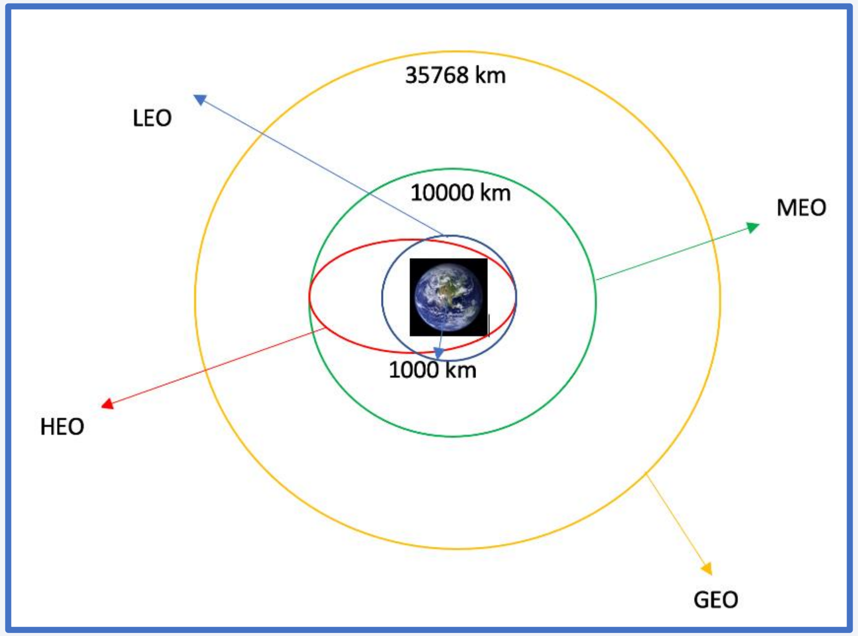
Payload vs. Launch Site



The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

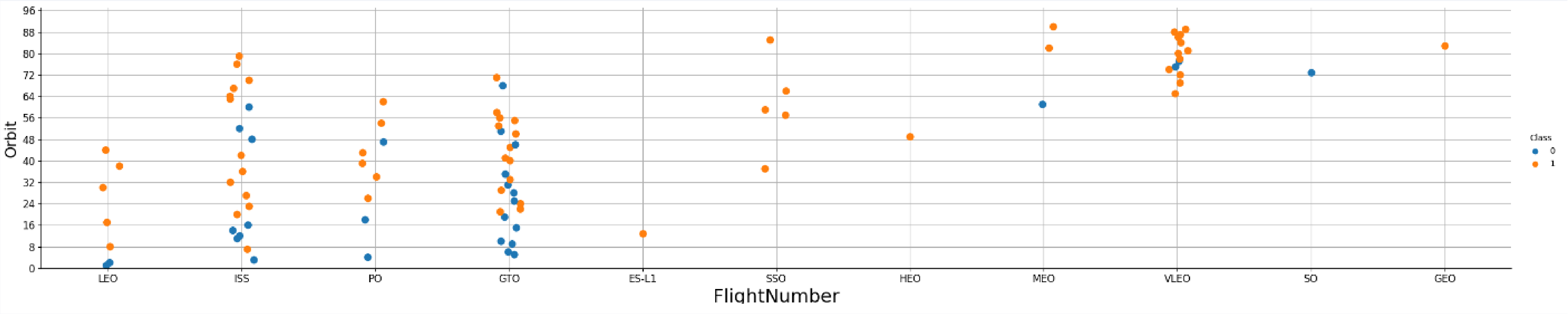
Success Rate vs. Orbit Type





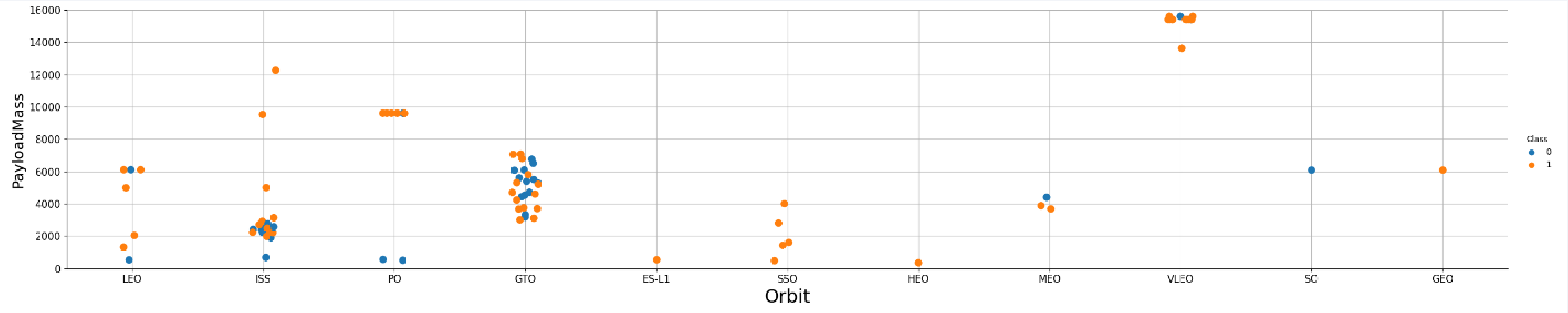
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Flight Number vs. Orbit Type



We should see 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.

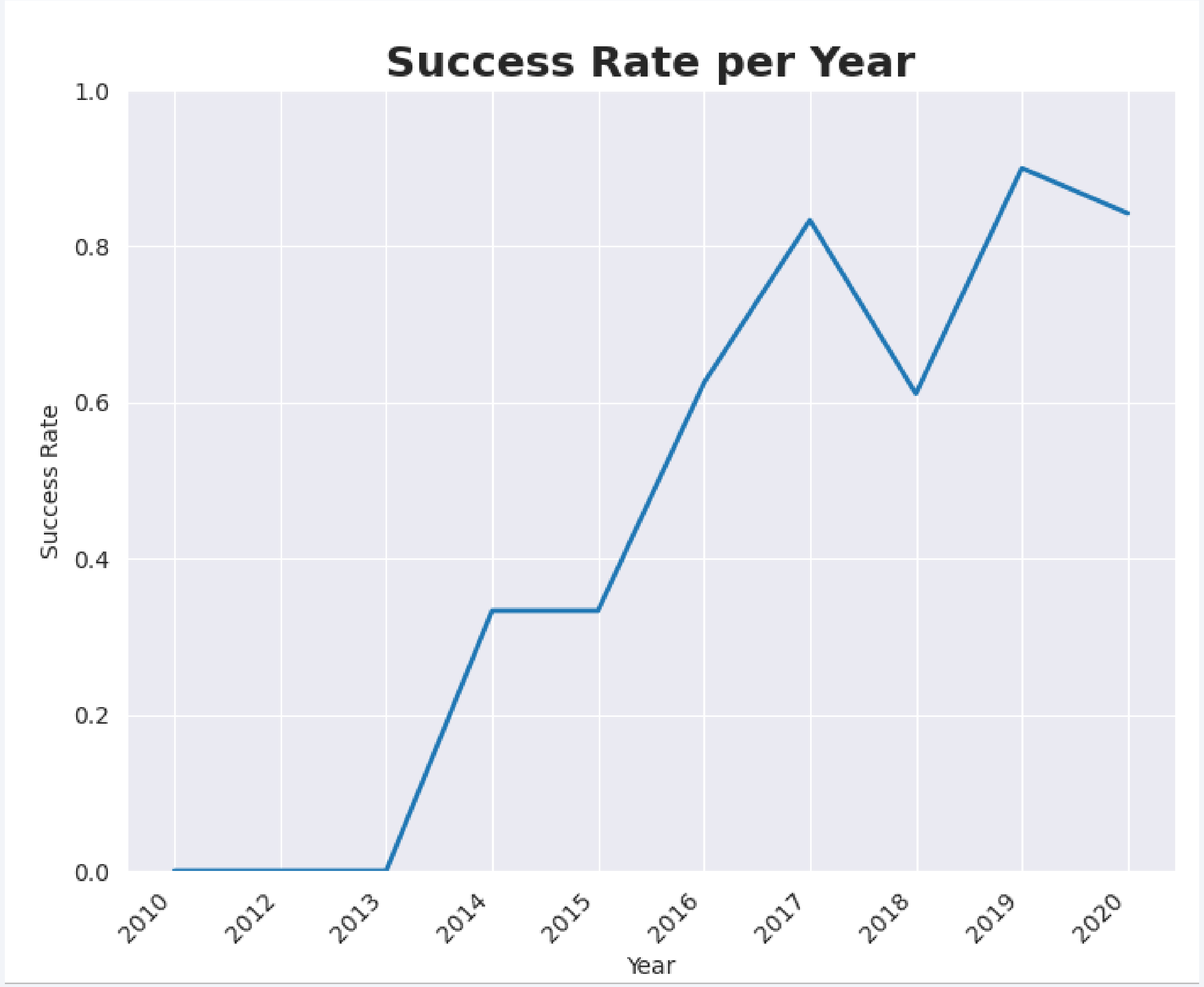
Payload vs. Orbit Type



We should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

We can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTBL
```

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

Using the word *DISTINCT* in the query means that it will only show Unique values in the *Launch_Site* column from *SPACETBL*

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL \
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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Using the word *LIMIT 5* in the query means that it will only show 5 records from *SpaceXTBL* and *LIKE* keyword has a wild card with the words *'CCA%'* the percentage in the end suggests that the Launch_Site name must start with KSC.

Total Payload Mass by Customer NASA (CRS)

```
%sql select Customer, sum(PAYLOAD_MASS__KG_) from SPACEXTBL \
where Customer = 'NASA (CRS)' \
group by Customer
```

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

Customer	sum(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

Using the function *SUM* summates the total in the column *PAYLOAD_MASS_KG_*
The *WHERE*clause filters the dataset to only perform calculations on *Customer NASA (CRS)*

Average Payload Mass by F9 v1.1

```
%sql select Booster_Version, avg(PAYLOAD_MASS__KG_) from SPACEXTBL \
where Booster_Version = 'F9 v1.1' \
group by Booster_Version
```

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

Booster_Version	avg(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

Using the function *AVG* works out the average in the column *PAYLOAD_MASS_KG_*
The *WHERE*clause filters the dataset to only perform calculations on *Booster_version F9 v1.1*

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

2015-12-22

Using the function *MIN* works out the minimum date in the column *Date*
The *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

```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)'
and payload_mass__kg_ between 4000 and 6000
```

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

Selecting only *Booster_Version*
The *WHERE*clause filters the dataset to *Landing_Outcome = Success (drone ship)*
The *AND* clause specifies additional filter conditions
Payload_MASS_KG_*>4000AND*Payload_MASS_KG_<6000

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL \
group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL\
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
[illegible]	[illegible]	CCAFS LC-	[illegible]	[illegible]	[illegible]	SKY Perfect JSAT	[illegible]	[illegible]

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

```
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

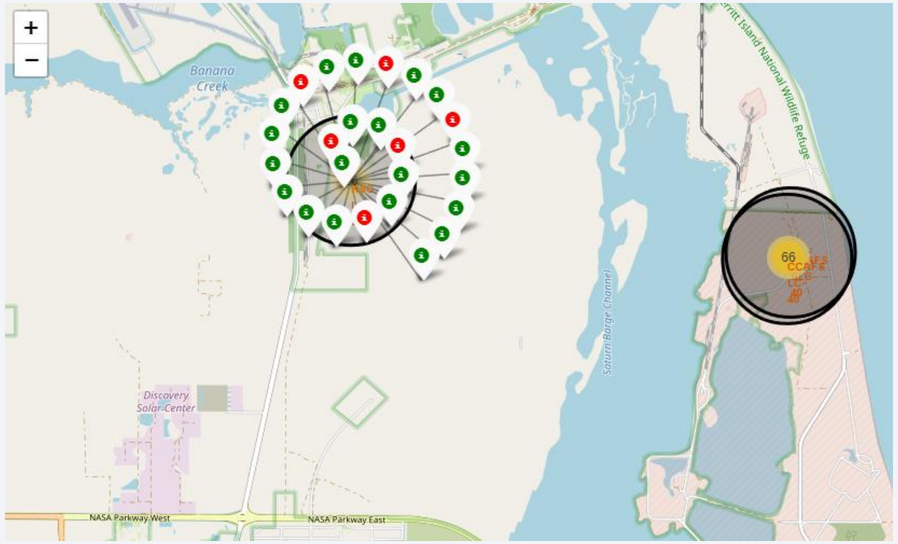
All launch sites global map markers

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

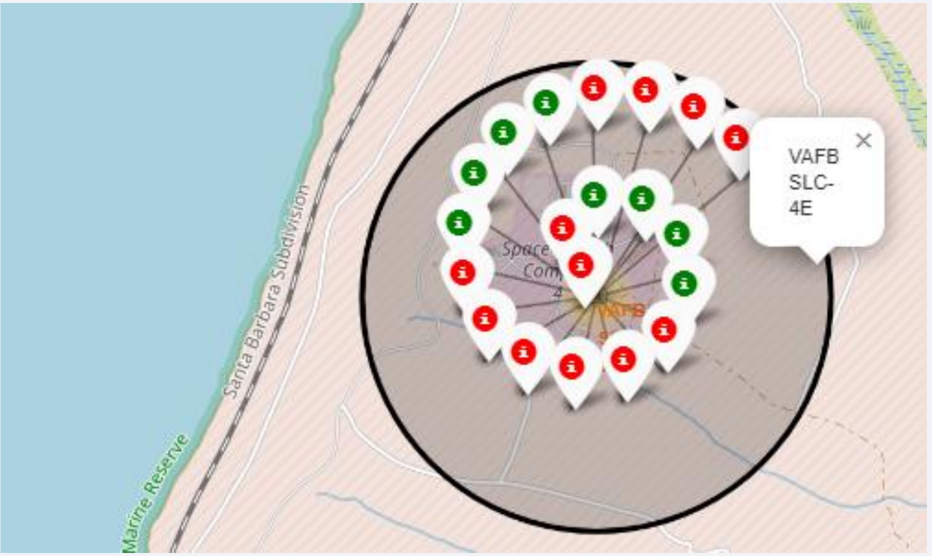
Colour labelled markers

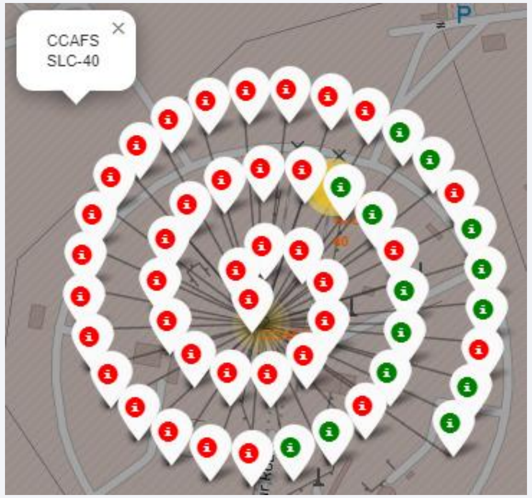
Green Marker shows successful launches and
Red Marker shows failures

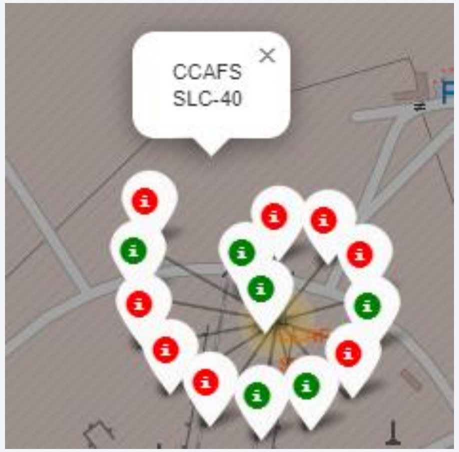
Florida Launch Sites



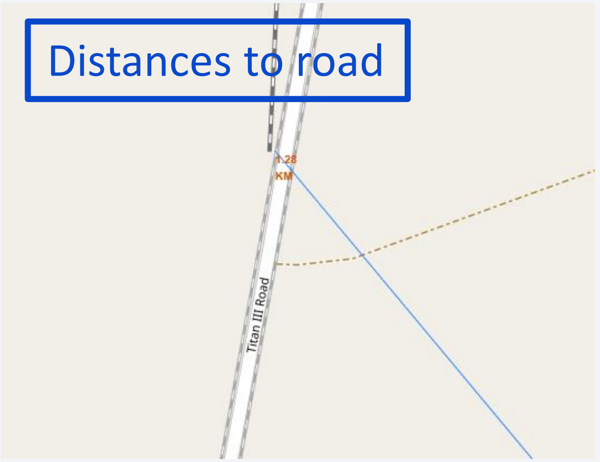
California Launch Sites

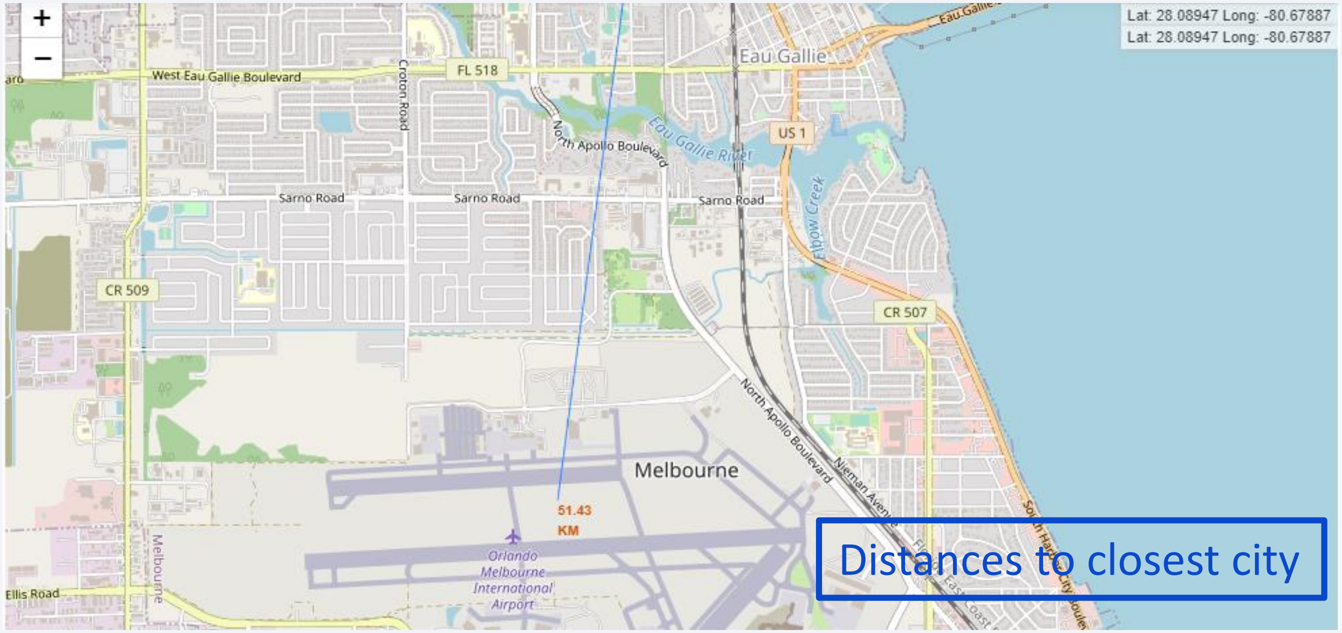






Working out Launch sites distances to landmarks



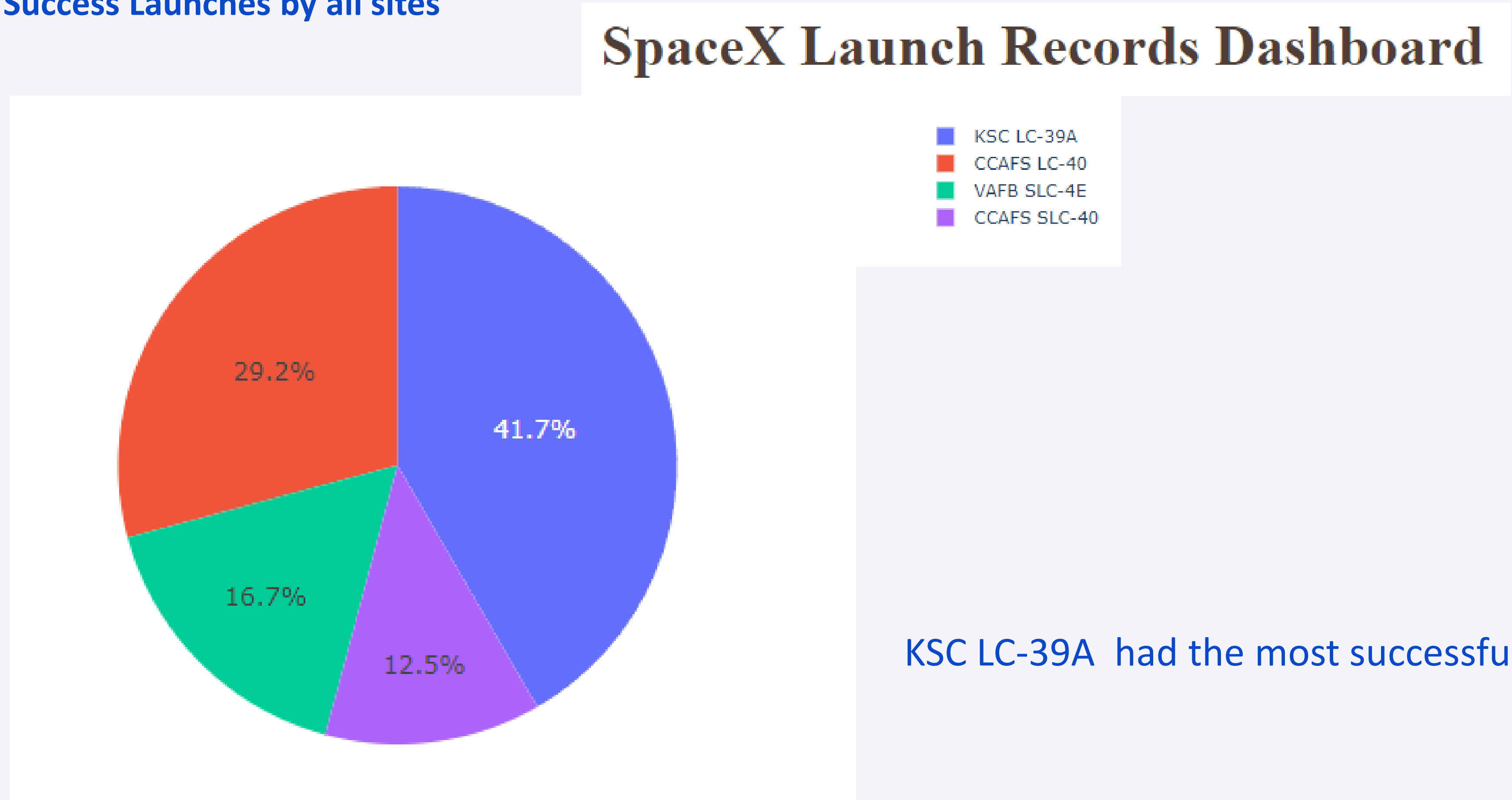


Section 4

Build a Dashboard with Plotly Dash

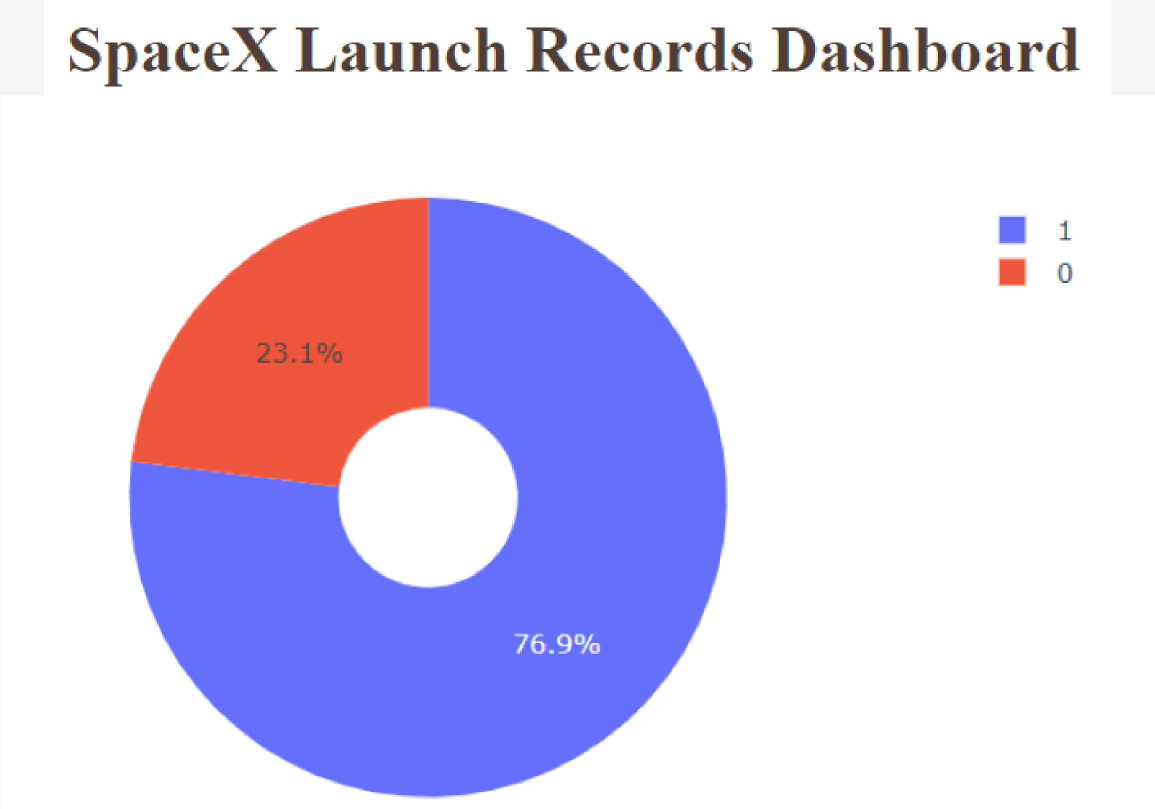
Dashboard – Pie chart showing the success percentage achieved by each launch site

Total Success Launches by all sites



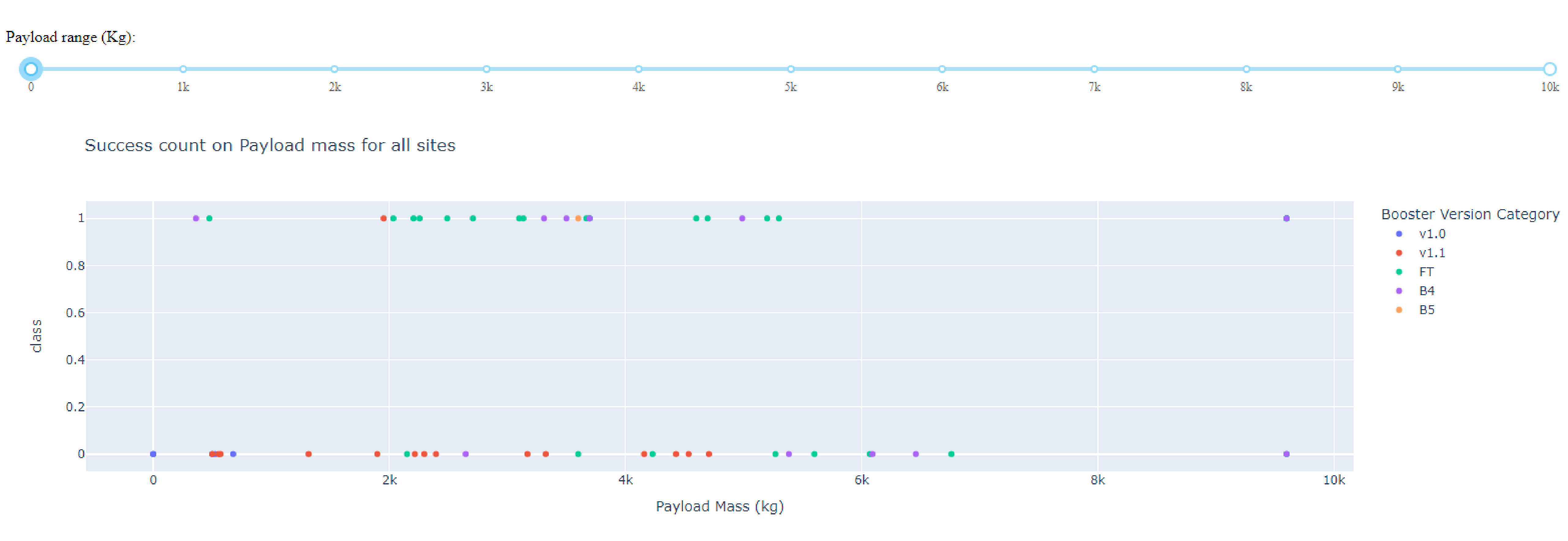
KSC LC-39A had the most successful launches from all sites

Dashboard – Pie chart for the launch site with highest launch success ratio



KSC LC-39A achieved a 76,9% success rate while getting a 23,1% failure rate

Dashboard – Payload vs Launch Outcome for different booster version category



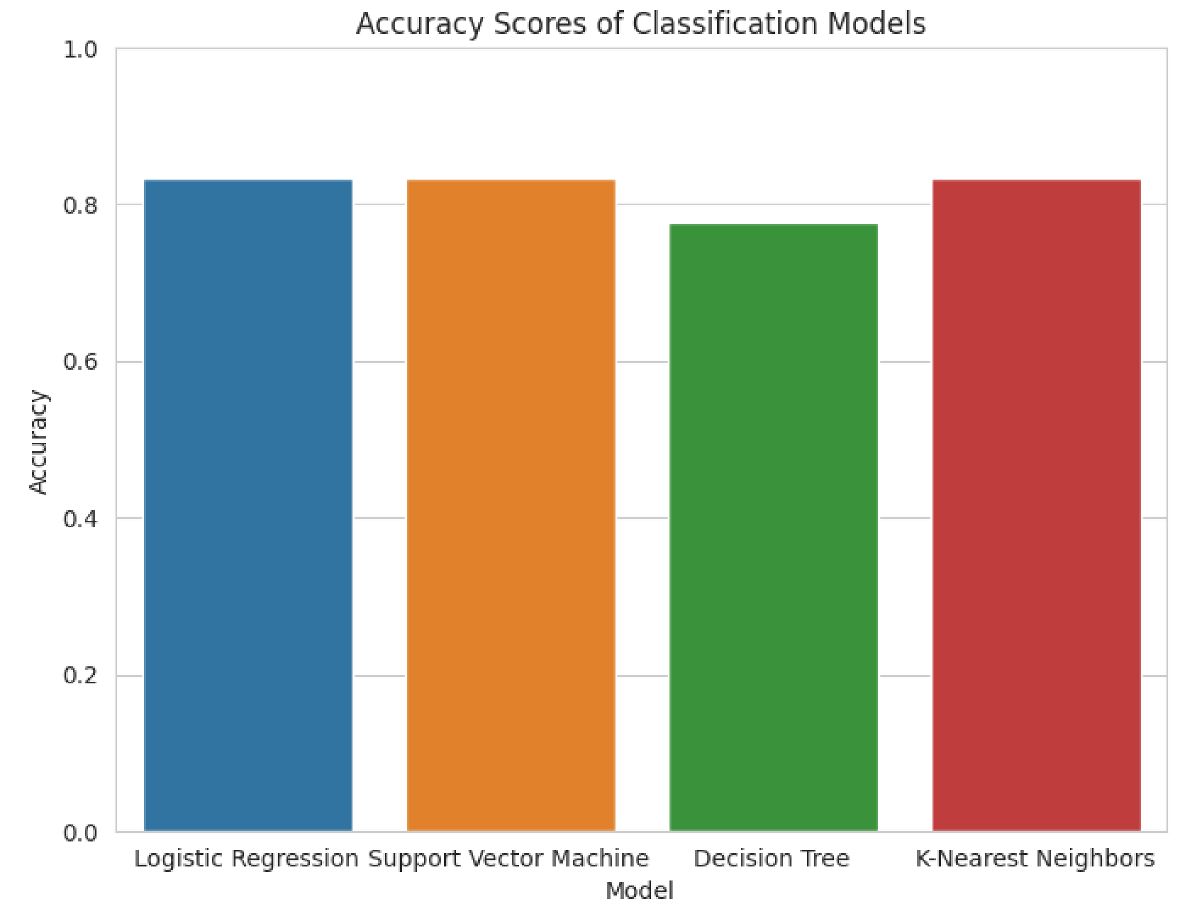
Section 5

Predictive Analysis (Classification)

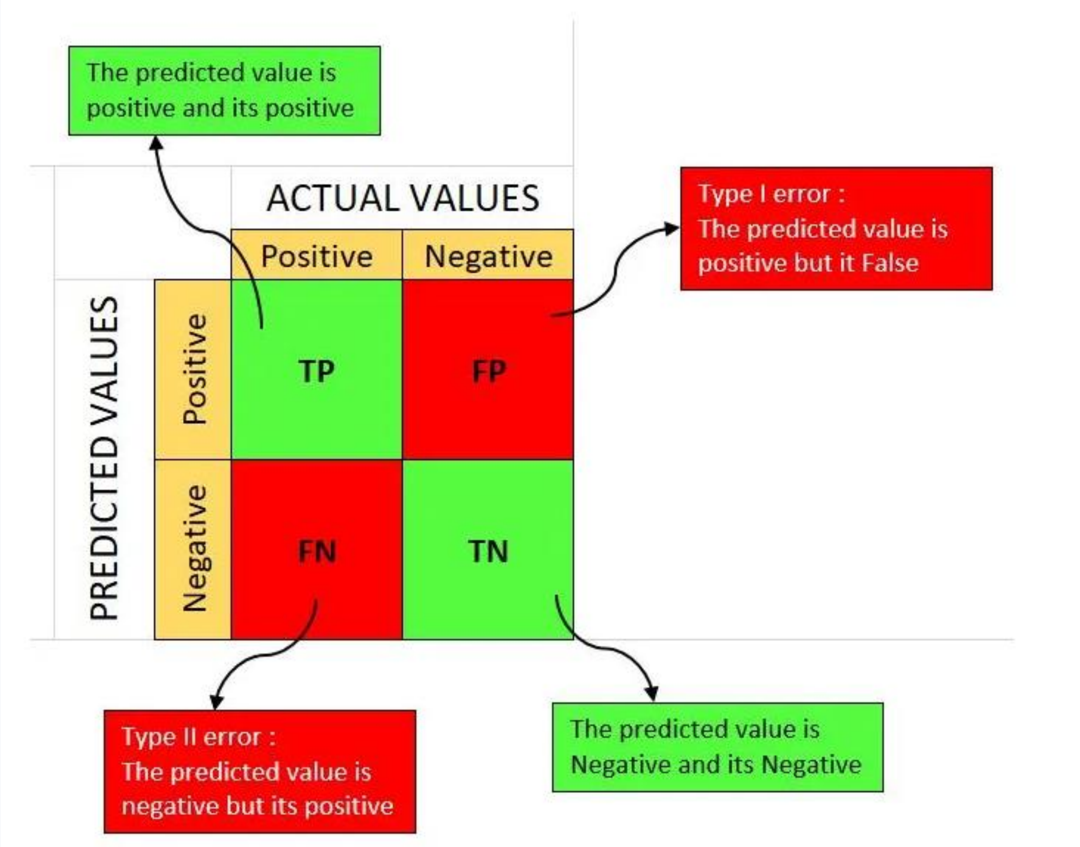
Classification Accuracy

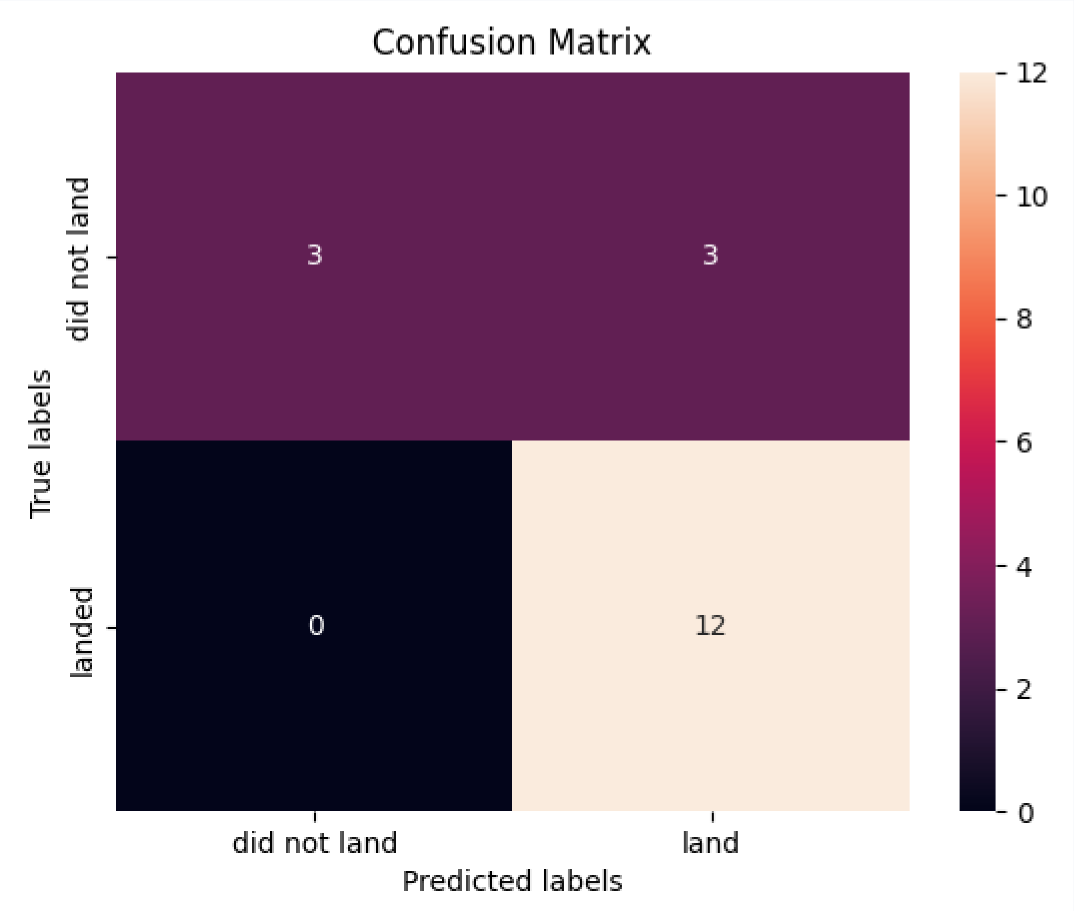
	Model	Accuracy
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333
2	Decision Tree	0.777778
3	K-Nearest Neighbors	0.833333

The best model looking only for accuracy,
that not is a good way to measure the best model,
could be LR, SVM or KNN.



Confusion Matrix





Confusion matrix for KNN

Conclusions

- **The LR, SVM and KNN have the same performance, when we look only to accuracy. A good way to improve the analysis is looking to a F1-Score.**
- **Low weighted payloads perform better than the heavier payloads**
- **The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches**
- **We can see that KSC LC 39A had the most successful launches from all the sites**
- **Orbit GEO,HEO,SSO,ES L1 has the best Success Rate**

Thank you!