IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Maria del Carmen Martín Rodríguez>

<10-12-2024>

Outline

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

Executive Summary

Summary of Methodologies

- Data collection: I made GET requests to retrieve historical Falcon 9 launch data through the SpaceX API.
- Web scraping: Used BeautifulSoup to extract an HTML table of historical Falcon 9 launch records from Wikipedia for further analysis.
- Data Wranling: EDA to find some patterns in the data and determine what would be the label for training supervised models.
- EDA: using Pandas and Matplotlib for visualizations and data preparation.
- EDA: Connected to a Db2 database to load the SpaceX dataset and ran SQL queries to answer key questions about the launches.
- Created interactive visual analyses using Folium to explore geospatial data and launch locations.
- Built an interactive Dash application using Plotly, enabling real-time visualization of SpaceX launch data.
- Machine learning: Trained various models including logistic regression, SVM, decision trees, and KNN, evaluating performance to select the best model using test data.

Summary of all results:

- The visualizations with Folium show that the best launch sites are those **close to the coast and the equator**, in order to avoid urban areas and make the most of the Earth's rotational speed.
- The site with the most successful launches is **KSC LC-39A**, with 76,9% success.
- The **Booster FT version** appears to have the highest rate of successful launches, with loads between 0-4000 kg.
- As **payload mass** increases, first stage landing success decreases.
- The model with the highest precision is **the decision tree**, with an accuracy of **94.9%**.
- The orbits with 100% success rates are HEO, SSO, ES-L1 and GEO, but the one with the highest percentage of successful launches is **SSO**, which has **100% success** for loads between **0-4000kg.**
- Except GTO, all orbits improve their landing success rate with **flight number** and all launch sites also improve their landing success with flight number.
- The success of the mission seems to improve as the years go by.

Introduction

Project background and context

Space Exploration Technologies Corp., known as SpaceX, is an American aerospace manufacturing and space transportation services company headquartered in Hawthorne, California. It was founded in 2002 by Elon Musk with the aim of reducing the costs of traveling to space to facilitate the colonization of Mars. SpaceX has developed several launch vehicles, the Starlink constellation, the Dragon cargo ship and carried astronauts to the International Space Station on Dragon 2.
SpaceX's achievements include the first privately financed liquid-fuel rocket to reach orbit (Falcon 1 in 2008),7 the first private company to launch and recover a spacecraft (Dragon in 2010), the first private in sending a ship to the International Space Station (Dragon in 2012),8 the first powered landing of an orbital rocket (Falcon 9 in 2015), the first reuse of a orbital rocket (Falcon 9 in 2017),
One of the reasons SpaceX is able to reach these milestones is because rocket launches are relatively affordable. SpaceX lists Falcon 9 rocket launches on its website as costing $62 million, while other providers charge more than $165 million per launch. A significant portion of the savings comes from SpaceX's ability to reuse the first stage. Therefore, if we can determine whether the first stage will land, we can estimate the cost of a launch. The second stage helps place the payload into orbit, but most of the work is done by the first stage. This stage is quite large and expensive. Unlike other rocket suppliers, SpaceX's Falcon 9 is capable of recovering the first stage. But sometimes the first stage doesn't land, other times it fails, and other times SpaceX sacrifices the first stage due to mission parameters like payload, orbit, and customer. In this final phase, I will take on the role of a data scientist working for a new rocket company, Space Y, that aims to compete with SpaceX. My job will be to determine the price of each release. To do this, I will collect information about SpaceX and create dashboards for my team. I will also determine if falcon 9 will reuse the first stage. To achieve this, I will train a machine learning model and use publicly available information to predict whether falcon 9 will reuse the first stage.

Problems I want to find answers

- To know if the payload affects the launch success rate.
- To know if the chosen orbit affects the launch success rate.
- To know if the version of the Booster affects the launch success rate.
- To know if the launch site affects the launch success rate.
- To know if the number of flights affects the launch success rate.
- And all of this in order to predict whether Falcon 9 will reuse the first stage and to predict the launch cost.

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 – SpaceX API

Import Libraries and Define Auxiliary Functions

Request and parse the SpaceX launch data using the GET request

We create a dictionary with the variables and then convert it into a Dataframe

Use the json_normalize method to convert the json result to a data frame.

Filter the dataframe to only include Falcon 9 launches

Find missing values and replace them with the column mean using mean() and replace().

Export data to csv .

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api-v2.ipynb

Data Collection - Scraping

1. Import Libraries and Define Auxiliary Functions
2. Execute an HTTP GET method to request the Falcon9 launch HTML page, as an HTTP response.
3. Create a BeautifulSoup object from the HTML response
4. Extract all column/variable names from the HTML table header with find_all().
5. Create a data frame by parsing the launch HTML tables
6. Export data to csv.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Import Libraries and load data.

Calculate the number of launches on each site with value_counts().

Calculate the number and occurrence of each orbit with value_counts().

Calculate the number and occurence of mission outcome of the orbits with value_counts().

Create a landing result label from the Result column, to split landings into successful (1) and unsuccessful (0)

Use mean() on the new column to determine the landing success rate.

Export data to csv.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA with Data Visualization

Import Libraries and load data.

Flight number versus payload mass with catplot().

Visualize the relationship between Flight Number and Launch Site with catplot().

Visualize the relationship between Payload and Launch Site with catplot()

Visualize with barplot() the relationship between success rate of each orbit type

Visualize the relationship between FlightNumber and Orbit type with catplot()

Visualize the relationship between Payload and Orbit type with catplot()

Visualize with lineplot() the launch success yearly trend

Create dummy variables to categorical columns with get_dummies().

Cast all numeric columns to float64 with astype.

Export data to csv.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-dataviz-v2%20(1).ipynb

EDA with SQL

- **SELECT DISTINCT, FROM:** display the names of the unique launch sites in the space mission.
- **SELECT, FROM, WHERE**: display 5 records where launch sites begin with the string 'CCA'.
- **SELECT, SUM(), FROM, WHERE**: Display the total payload mass carried by boosters launched by NASA (CRS).
- **SELECT, AVG(), FROM, WHERE:** Display average payload mass carried by booster version F9 v1.1.
- **SELECT, MIN(), FROM, WHERE:** List the date when the first succesful landing outcome in ground pad was acheived.
- **SELECT, FROM, WHERE, AND, <>:** List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- **SELECT, COUNT(CASE…WHEN…THEN), FROM**: List the total number of successful and failure mission outcomes.
- **SELECT, FROM, WHERE + SUBQUERY:** List the names of the booster versions which have carried the maximum payload mass.
- **SELECT, CASE(WHEN, SUBSTR(), THEN), FROM, WHERE, AND**: List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- **SELECT, COUNT(), FROM, WHERE, BETWEEN, AND, IN, GROUPBY():** Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Folium.map() :** create a map.
- **Folium.Circle() :** create circular marks at launch sites.
- **Folium.map.Marker():**name the sites according to their coordinates on the map.
- **Folium.Popup() :** Create pop-up windows with text by clicking on the marks.
- **DivIcon():** put names next to the circle marks.
- **MarkerCluster():** group the markers of the same area according to their class.
- **MousePosition():** add an object to display coordinates in a corner of the map, depending on where the mouse pointer is positioned.
- **Folium.Marker():** to add the distance from the launch site to another location.
- **Folium.Polyline():** add a line from the launch site to a reference point.
- **Add_child():** add the marks to the map in question.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/lab-jupyter-launch-site-location-v2%20(1).ipynb

Build a Dashboard with Plotly Dash

- **Add a Launch Site Drop-down Input Component**.
  We have four different launch sites, and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate (class=0 vs. class=1).
- **Adds a callback function to render the success pie chart based on the site selected in the dropdown**
  The general idea of this callback function is to get the selected launch site from the dropdown site and render a pie chart displaying the launch success counts.
- **Add a Range Slider to Select Payload.**
  The goal is to know if the variable payload is correlated with the mission outcome. From the control panel point of view, we want to be able to easily select different payload ranges and see if we can identify some visual patterns.
- **Add a callback function to render the success-payload-scatter-chart scatter plot.**
  The goal is to plot a scatterplot with the x-axis as the payload and the y-axis as the launch result (i.e., class column). As such, we can visually observe how the payload can correlate with mission results for selected sites. Additionally, we have color-labeled the version of each Booster at each scatter point so we can observe mission results with different drivers.

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

We create for the variable Y an array from the 'Class' column using to_numpy()

We assign our set of dummy variables to our variable X and standardize the data with StandardScaler()

We use train_test_split() to split the data into test and training data

We created four objects: one logistic regression, another SVM, a decision tree classifier and another KNN.

We create the GridSearchCV() objects, with cv=10

We train each model with the fit() method

We adjust the object to find the best parameters with best_params_

We calculate the average precision obtained with the parameters of each model using best_score_

We calculate the precision of each model with the score() method.

We generated the confusion matrix to evaluate the models.

We use best_score_ to know which model has the best performance

https://github.com/Mai-de-jerez/IBM_Applied_Data_Science_Capstone/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1%20(1).ipynb

Results

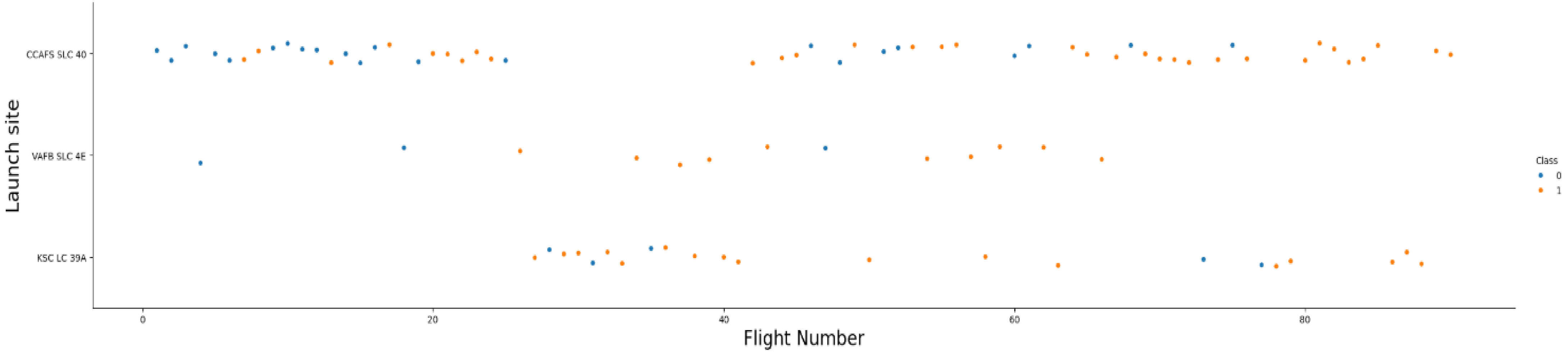
- Exploratory data analysis results
  - When the flight number increases, the success rate of each type of orbit improves.
  - The orbit with the highest percentage of successful launches is SSO, with 100% success,
- Interactive analytics demo in screenshots:
  - The site with the highest percentage of successful launches is KSC LC-39A.
  - The payload mass with the highest rate of successful launches is between 0 and 5000kg.
  - The FT version of the propellant with loads between 0 and 5000kg is the one with the highest success rate.
- Predictive analysis results
  - The best model is the decision tree with an accuracy of 94.4%.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch site",fontsize=20)
plt.show()
```

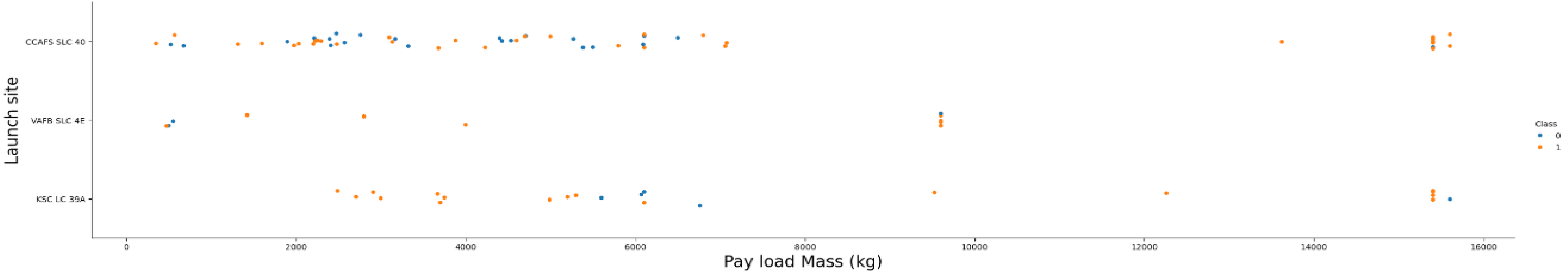


I would say that the success rate improves with the number of flights for the three locations. But clearly CCAFS SLC 40 is the one that improves the most with the number of flights. In the other cases it is not so clear. CCAFS SLC-40 has a lower success rate.

Payload vs. Launch Site

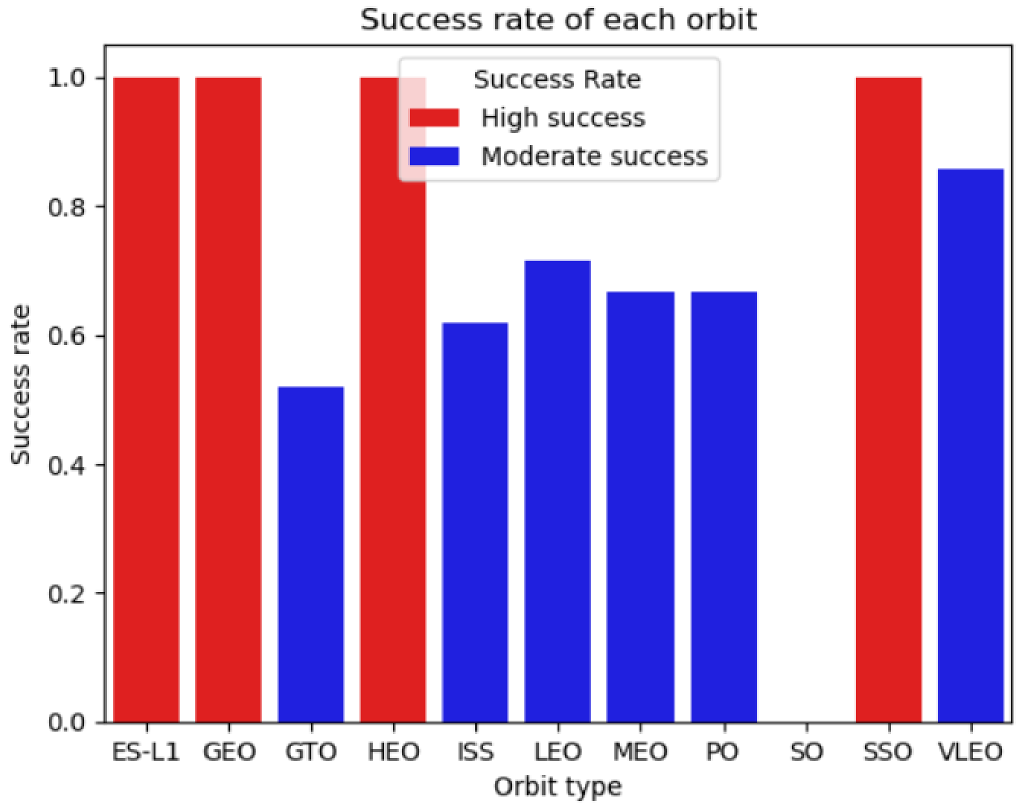
We also want to observe if there is any relationship between launch sites and their payload mass.

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch site",fontsize=20)
plt.show()
```



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000). CCAFS SLC-40 shows relatively good performance for lighter loads, with a tendency to have more successes as the program progresses. KSC LC-39A appears to be best suited for missions with large payloads, with a significant proportion of successes even at the upper end of payload. VAFB SLC-4E: This site has a smaller number of missions in the data set and an apparently higher proportion of failure

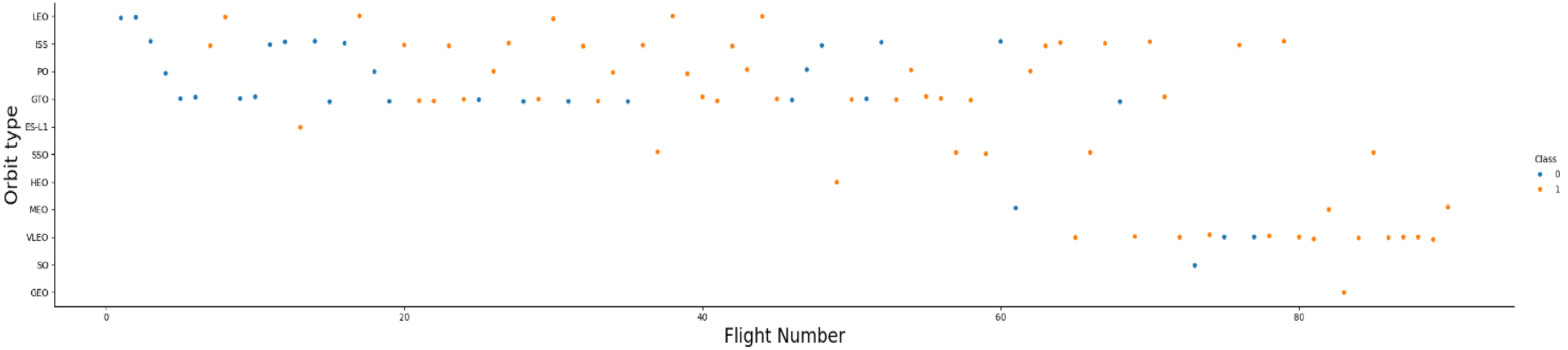
Success Rate vs. Orbit Type



Highly successful orbits: ES-L1, GEO, HEO, and SSO have impeccable performance. Most challenging orbits: GTO has the lowest success rate. LEO and VLEO: their success rate is moderate.

Flight Number vs. Orbit Type

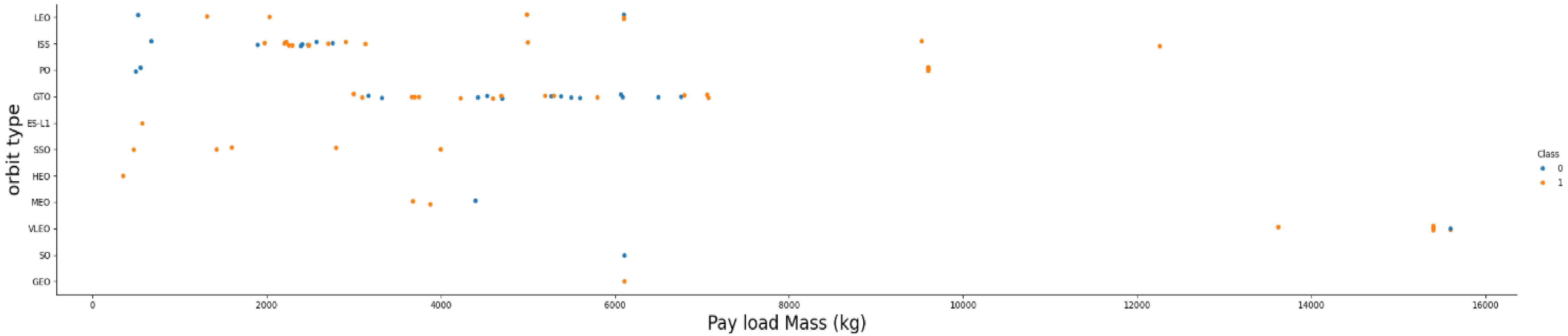
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit type",fontsize=20)
plt.show()
```



In the LEO orbit, Success appears related to the number of flights. In VLEO, ISS and PO success also seems to improve with the number of flights. However, there seems to be no relationship between the flight number when in GTO orbit, its success rate is also low. The orbits with the highest success rates are LEO, VLEO and especially SSO

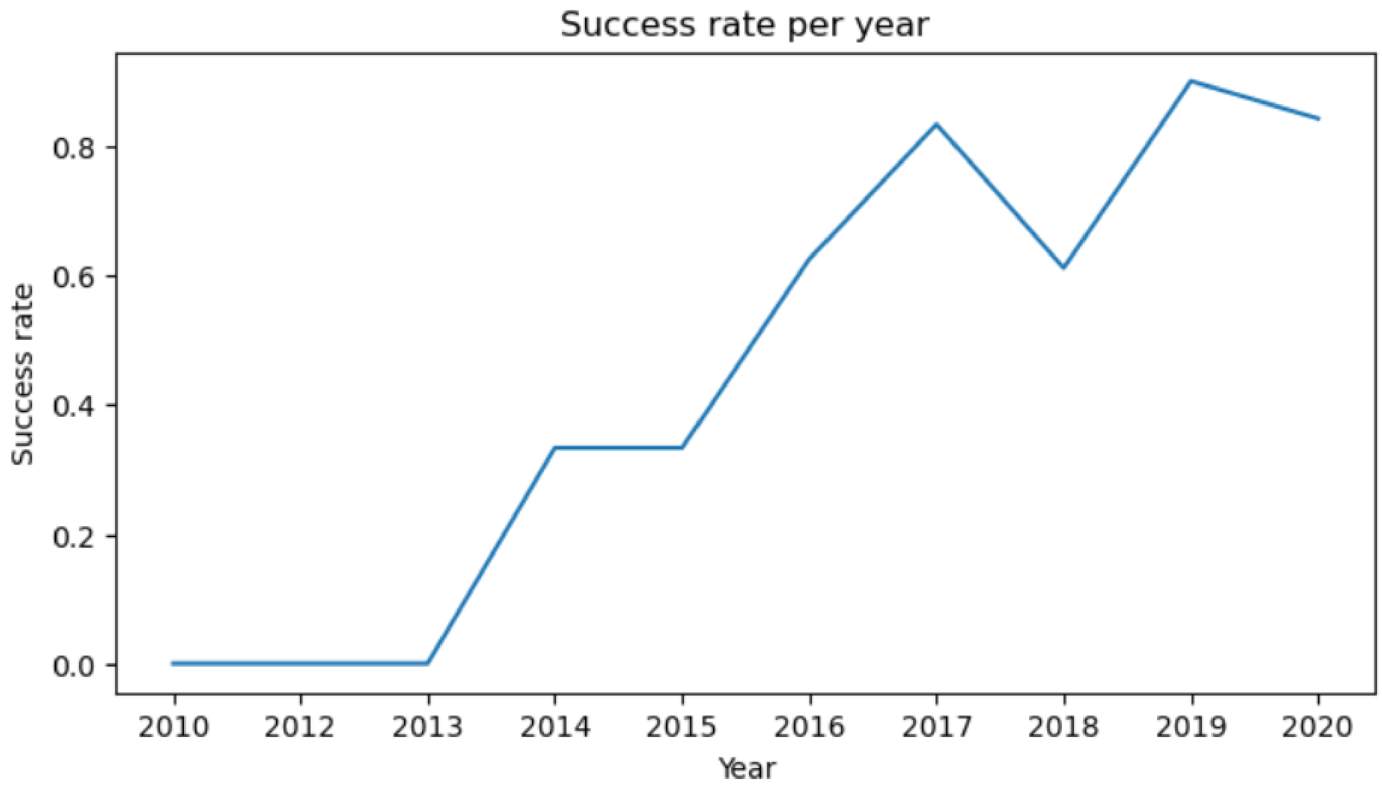
Payload vs. Orbit Type

```python
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("orbit type",fontsize=20)
plt.show()
```



SSO and LEO work better with light loads, GTO is a complicated orbit, VLEO and ISS work better with heavier loads. The highest success rate is for SSO with loads between 0 and 4000kg.

Launch Success Yearly Trend



You can see that the success rate since 2013 kept increasing until 2017 (stable in 2014) and after 2015 it started increasing. There was a small decrease from 2017 to 2018.

All Launch Site Names

```
%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

Here we can see the name of the four launch locations.

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

All five missions were successful, 4 of 5 customers were NASA, all payloads were light, and the orbit is always LEO.

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass,Customer
FROM SPACEXTABLE
WHERE Customer='NASA (CRS)'
```

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

total_payload_mass	Customer
45596	NASA (CRS)

Total mass of payload carried by NASA-launched boosters (CRS): 45596

Average Payload Mass by F9 v1.1

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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS total_payload_mass
FROM SPACEXTABLE
WHERE Booster_Version='F9 v1.1'
```

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

total_payload_mass
2928.4

Average payload mass carried by the F9 v1.1 booster version : 2928.4

First Successful Ground Landing Date

```
%%sql
SELECT MIN(Date) AS first_success_date
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
```

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

first_success_date
2015-12-22

Date on which the first successful landing on a terrestrial platform was achieved: 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000
```

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

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

Names of thrusters that are successful in unmanned spacecraft and have a charge mass greater than 4000 but less than 6000:

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT
    COUNT(CASE WHEN Mission_Outcome LIKE '%Success%' THEN 1 END) AS Total_Success,
    COUNT(CASE WHEN Mission_Outcome LIKE '%Failure%' THEN 1 END) AS Total_Failure
FROM SPACEXTABLE
```

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

Total_Success	Total_Failure
100	1

Total number of successful mission results : 100
Total number of failure mission results: 1

Boosters Carried Maximum Payload

```
%%sql
SELECT booster_version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
)
```

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

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

Names of booster versions that have carried the maximum payload mass:

F9 B5 B1048.4

F9 B5 B1049.4

F9 B5 B1051.3

F9 B5 B1056.4

F9 B5 B1048.5

F9 B5 B1051.4

F9 B5 B1049.5

F9 B5 B1060.2

F9 B5 B1058.3

F9 B5 B1051.6

F9 B5 B1060.3

F9 B5 B1049.7

2015 Launch Records

```
%%sql
SELECT
    CASE
        WHEN substr(Date, 6, 2) = '01' THEN 'Enero'
        WHEN substr(Date, 6, 2) = '02' THEN 'Febrero'
        WHEN substr(Date, 6, 2) = '03' THEN 'Marzo'
        WHEN substr(Date, 6, 2) = '04' THEN 'Abril'
        WHEN substr(Date, 6, 2) = '05' THEN 'Mayo'
        WHEN substr(Date, 6, 2) = '06' THEN 'Junio'
        WHEN substr(Date, 6, 2) = '07' THEN 'Julio'
        WHEN substr(Date, 6, 2) = '08' THEN 'Agosto'
        WHEN substr(Date, 6, 2) = '09' THEN 'Septiembre'
        WHEN substr(Date, 6, 2) = '10' THEN 'Octubre'
        WHEN substr(Date, 6, 2) = '11' THEN 'Noviembre'
        WHEN substr(Date, 6, 2) = '12' THEN 'Diciembre'
    END AS Month,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
AND Landing_Outcome = 'Failure (drone ship)'
```

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

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

Names of the months, results of unsuccessful landing on the unmanned ship, booster versions and launch site for the months of the year 2015:

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

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

```
%%sql
SELECT COUNT(landing_outcome) AS total,Landing_Outcome
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
and Landing_Outcome IN ('Failure (drone ship)','Success (ground pad)')
GROUP BY Landing_Outcome
```

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

total	Landing_Outcome
5	Failure (drone ship)
3	Success (ground pad)

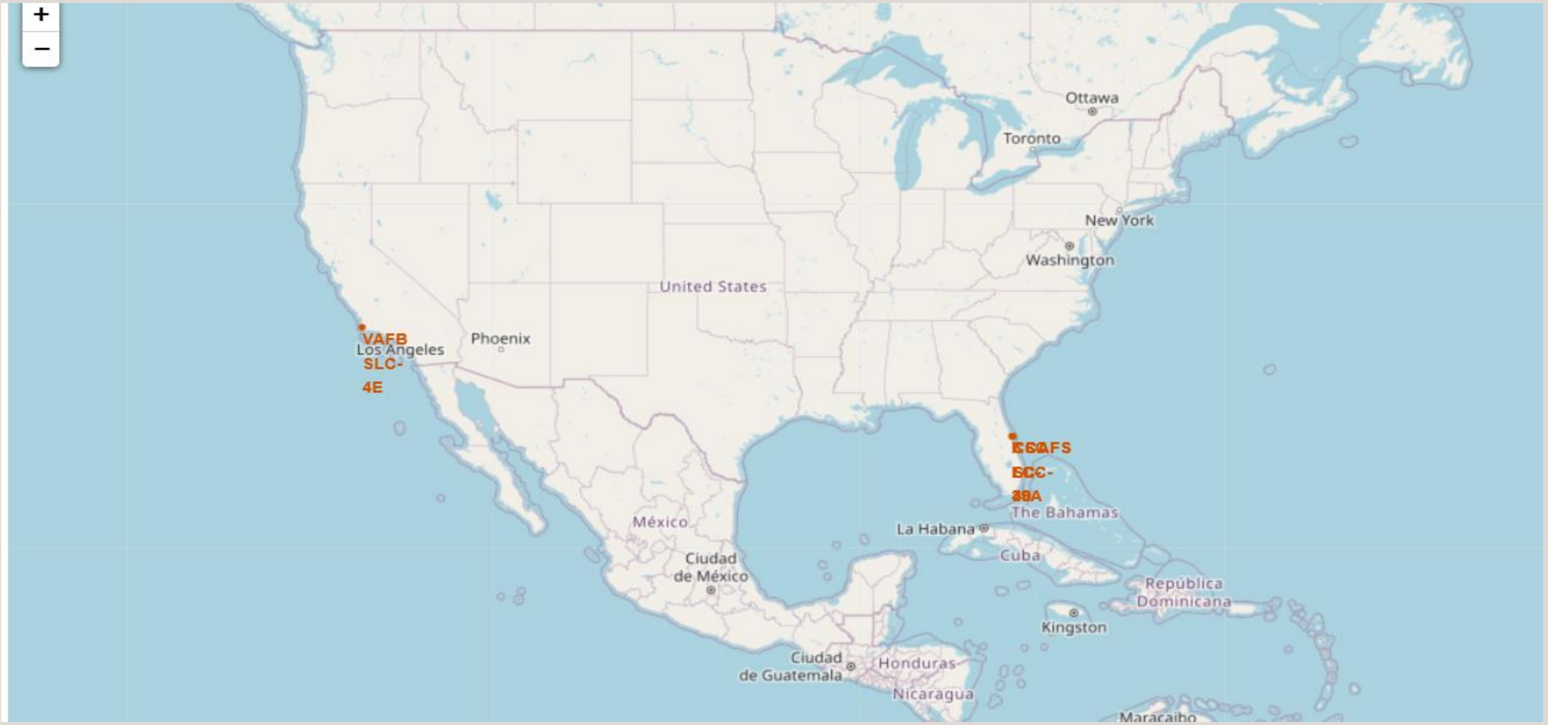
Count of landing results, between the date 2010-06-04 and 2017-03-20, in descending order:

Failure (drone ship): 5
Success (ground pad) : 3

Section 3

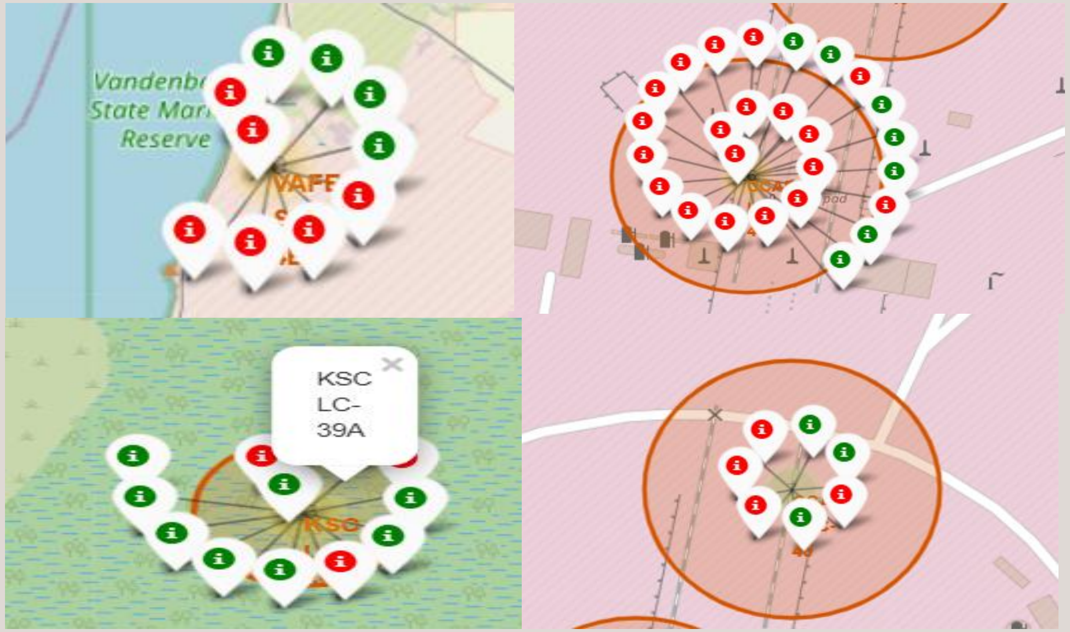
Launch Sites Proximities Analysis

Launch sites



All launch sites are close to the equator to take full advantage of the Earth's considerable rotation speed. The launch sites are near the coast because if during the rocket's ascent trajectory, the rocket explodes and falls back to earth, it is better for it to fall in an unpopulated area, and that is easier near the sea.

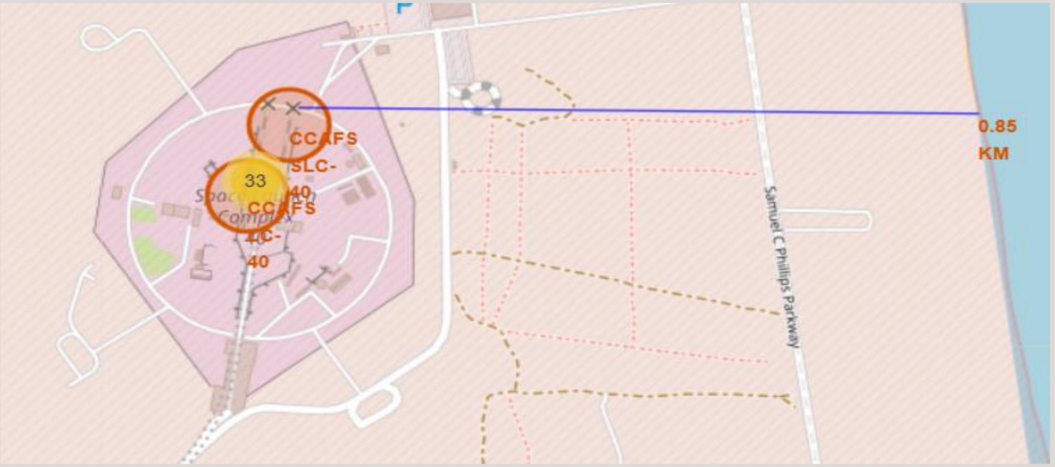
Number of launches per site and success rate

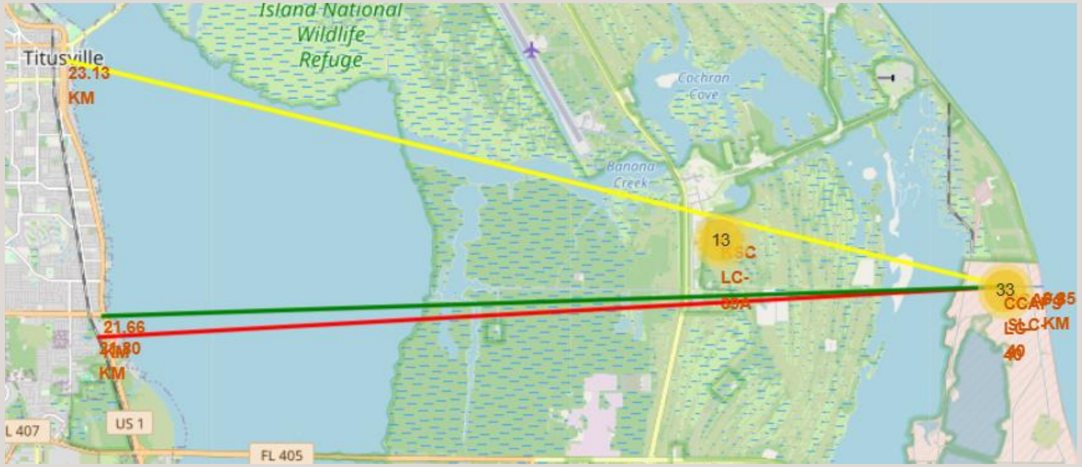


If we look at the first map, we can see the number of pitches grouped by zone.

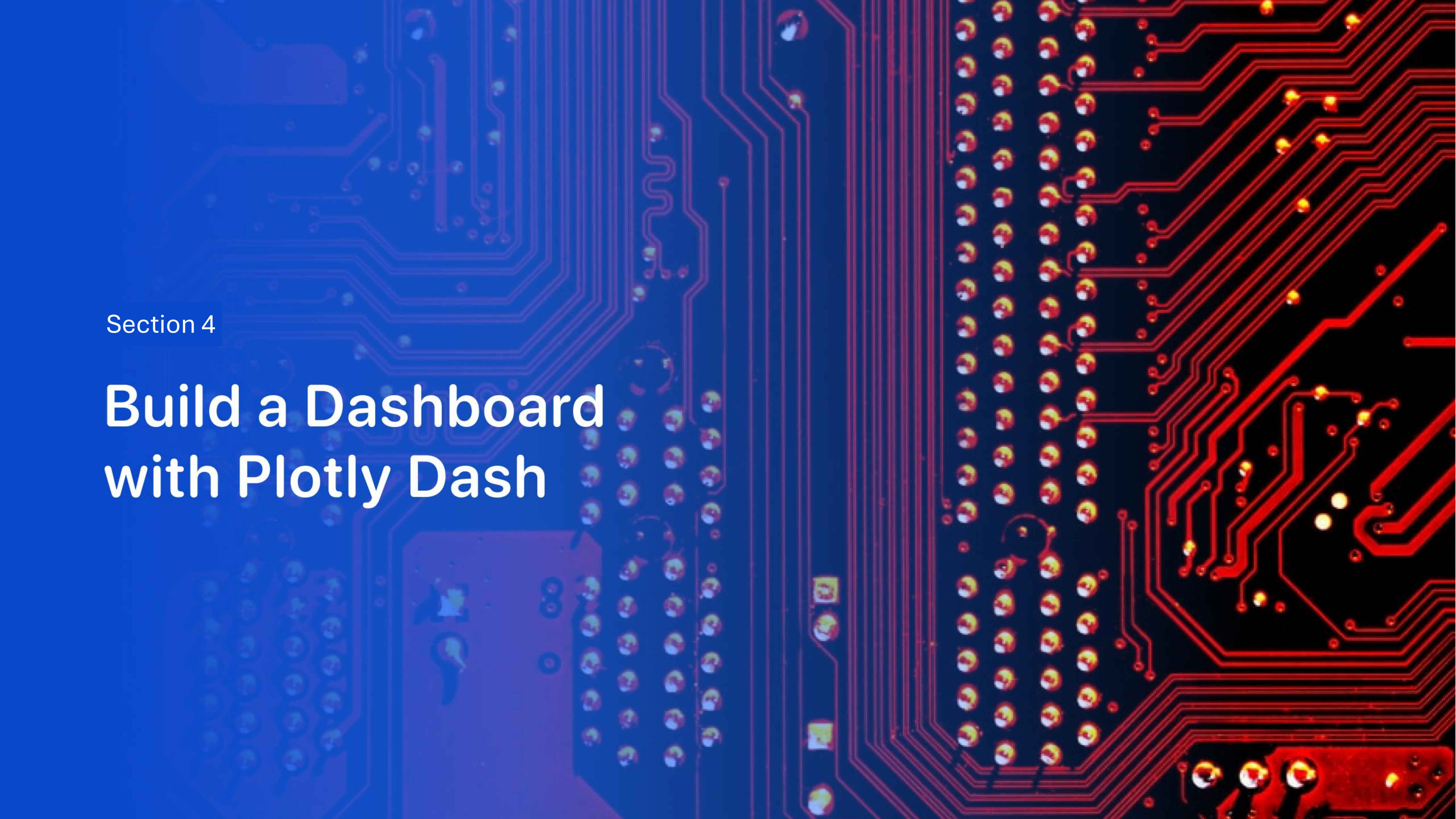
If we look at the second map, we can see that by clicking on the mark, we are shown the result of each launch, green for the successful ones and red for the unsuccessful ones. And we can also easily see that KSC LC 39-A is the site with the most successful launches.

Distance between the launch site and a reference point





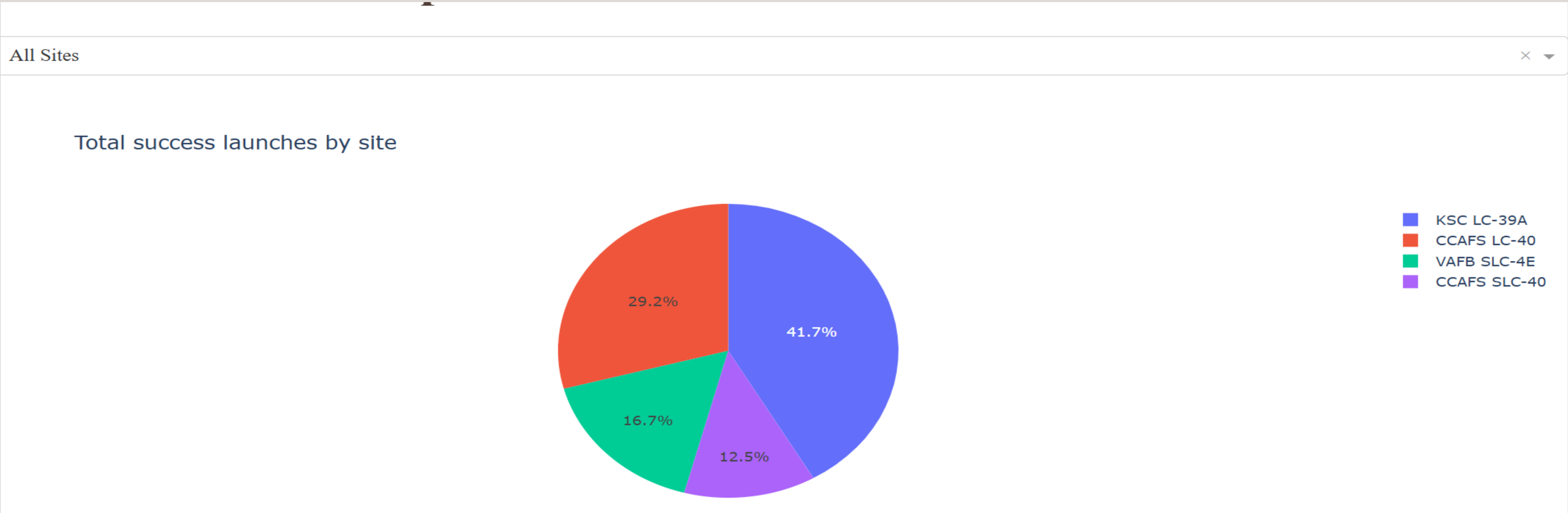
The launch site is close to the coast (0.85 km), but is far from cities, railways and highways, maintaining a safe distance of more than 20 km.



Section 4

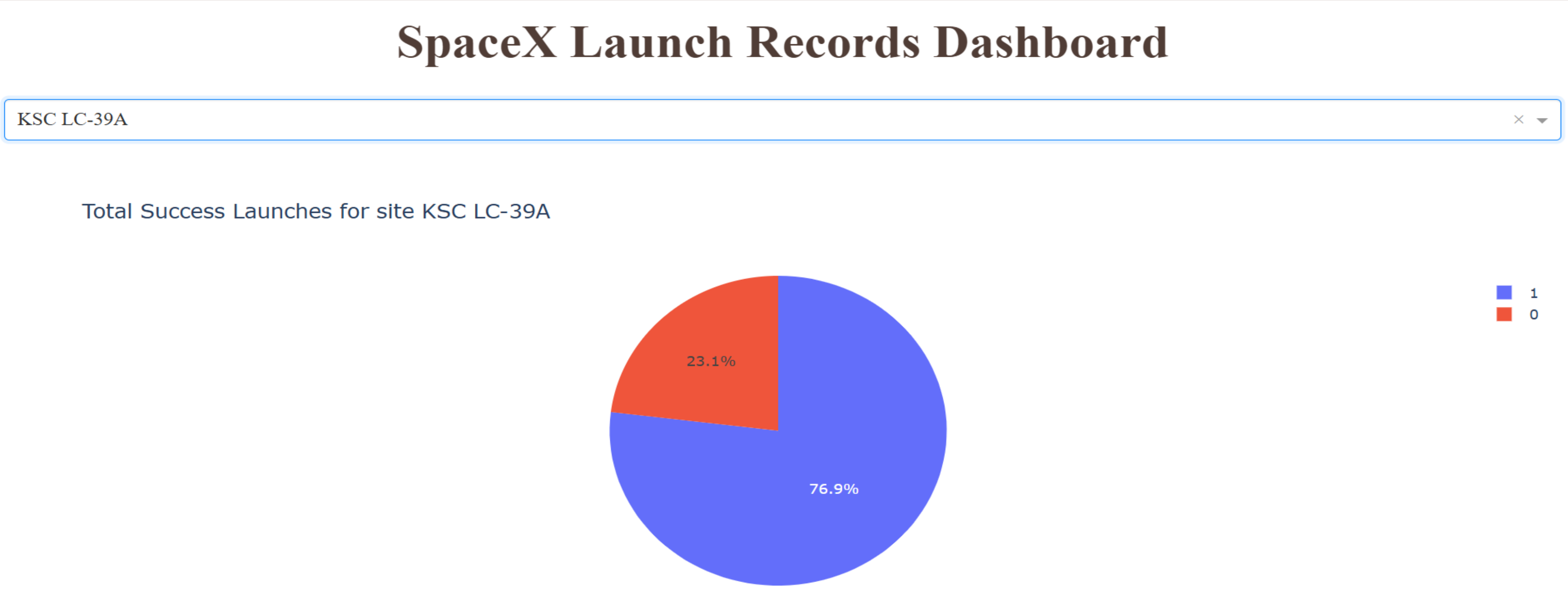
Build a Dashboard with Plotly Dash

Launch success count for all sites



We can clearly see that the site with the highest number of launches is KSC LC-39A

Launch site with highest launch success ratio



Indeed we can verify that the place with the highest rate of successful launches is KSC LC-39A, with a success rate of 76.9%. (Blue= successful, red= not successful)

Relationship between the choice of payload and Booster version with mission success

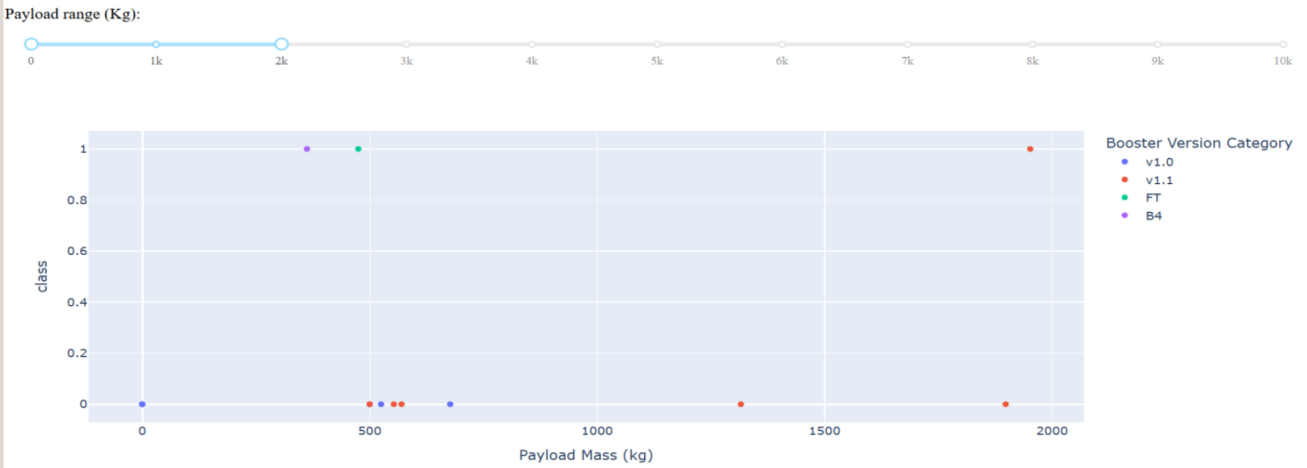
Payload ranges with the lowest success rate:

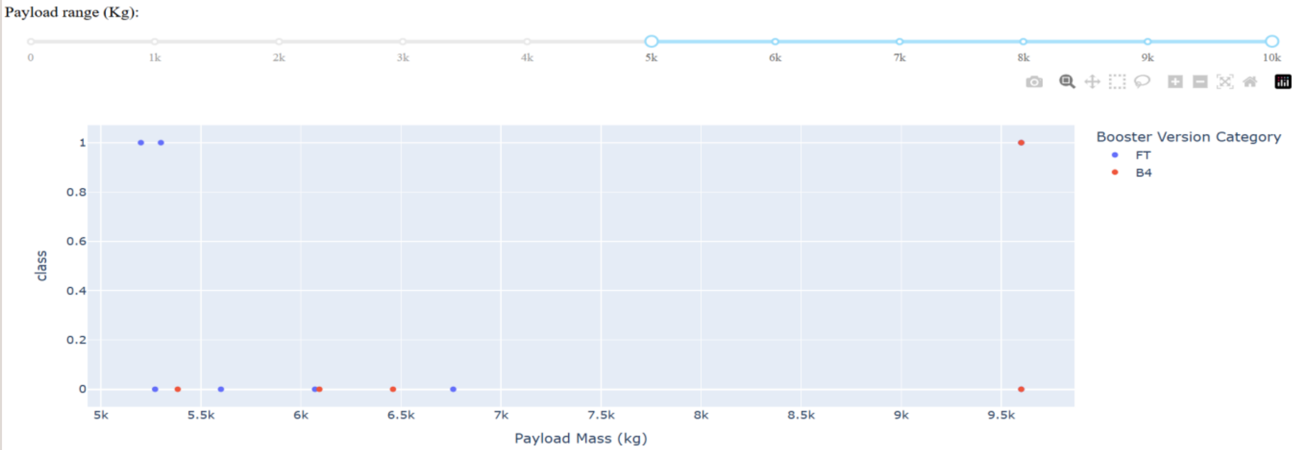
0-2000kg
5000-10000kg.

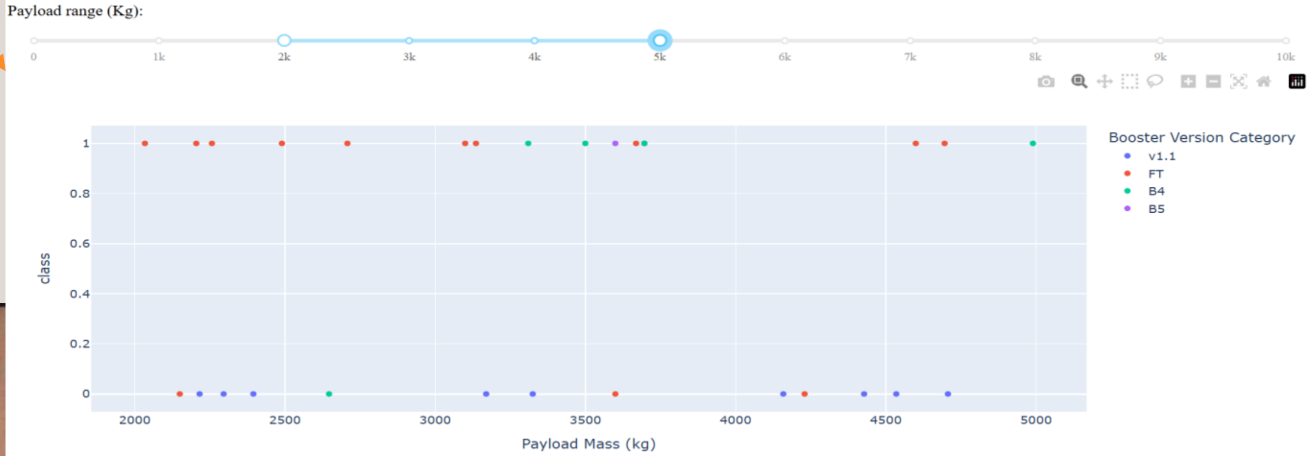
Payload range with highest launch success rate:

2000-5000kg.

Booster version with the highest successful launch rate:
FT





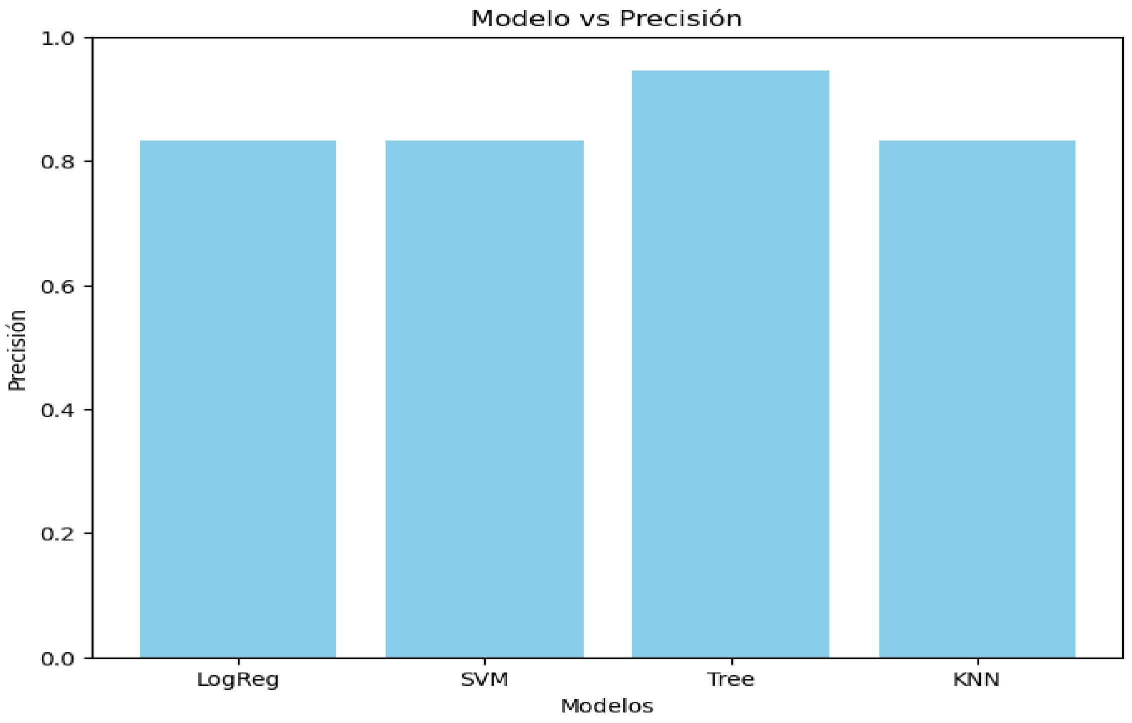


Section 5

Predictive Analysis (Classification)

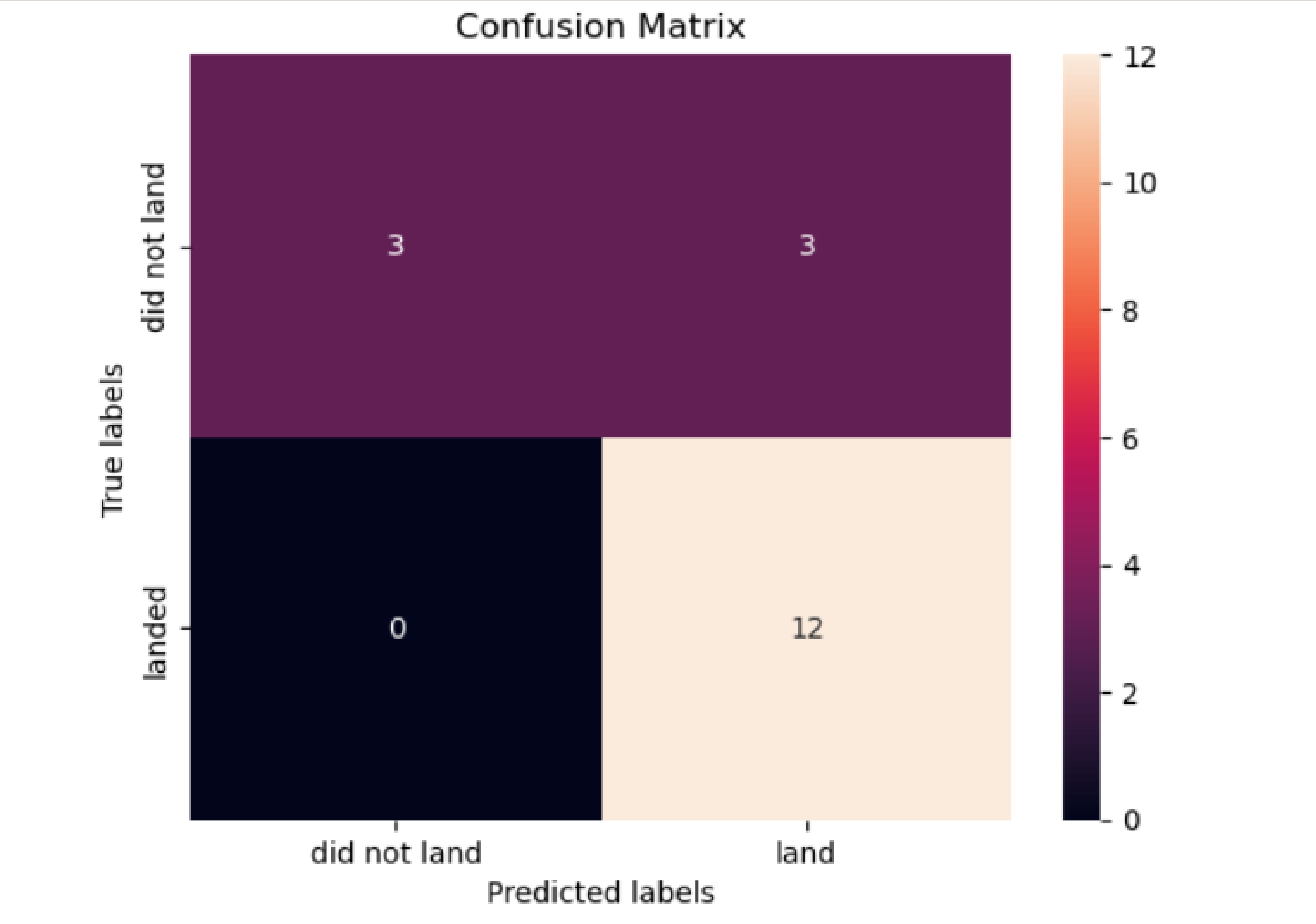
Classification Accuracy

In [38]:
```python
plt.figure(figsize=(8, 6))
plt.bar(df_report['Model'], df_report['Accuracy'], color='skyblue')
plt.title('Modelo vs Precisión')
plt.xlabel('Modelos')
plt.ylabel('Precisión')
plt.ylim([0, 1])
plt.show()
```



The decision tree has the best precision: 94.4%.

Confusion Matrix



The confusion matrix is similar for all models.
Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the problem is false positives.

<u>Overview:</u>
True Postive - 12 (True label is landed, Predicted label is also landed)

False Postive - 3 (True label is not landed, Predicted label is landed)

Conclusions

- **Launch site**: the site with the highest rate of successful launches is KSC LC-39A
- **Orbit:** SSO is the orbit with the highest percentage of successful launches, and has a 100% success rate, with loads between 0-4000kg.
- **Payload mass(kg):** between 1000 and 5000 kg the rate of successful launches is higher.
- **Flight number:** As the number of flights increases, the number of successful launches usually improves, at all launch sites, and in all orbits except GTO.
- **Yearly trend**: The successful launch rate for falcon 9 has improved over the years.
- **Booster version**: FT is the version of the booster with the highest rate of successful launches, with loads between 0-5000kg.
- **Launch site location**: the best launch sites are those close to the coast and the equator.
- **Model:** The best model is the classifier decision tree, with an accuracy of 94.4%.

Appendix

- Resources:
  - https://api.spacexdata.com
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

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)

Thank you!