IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cristina Caballero
11/10/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
    - EDA with SQL
    - EDA with visualization
    - Data Visualization with Folium
    - Plotly Dash Visualization
    - Predictive Analysis (machine learning)
- Conclusion
- Appendix

Executive Summary

Summary of methodologies:

The aim of this study is to determine whether a rocket would land successfully or not and study the factors that can affect to it. The methodology used has been:

- Data collection using web scraping
- Data wrangling
- Exploring the data with some visualizations
- Analysis of the data with the help of SQL
- Exploring in a deeper way the launching sites, and visualize them
- Construct models to predict landing outcomes

Summary of all results:

- The launch success has improved over the years
- Most of the launch sites are close to the coast and the Equator
- The results of the models were similar on the test set

Introduction

- Project background and context:

The most successful between all the space companies is SpaceX, it's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space. One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Spaces X's Falcon 9 launch like regular rockets.

- Problems you want to find answers

We would like to predict the rate of successful, and how the different factors affect to the launching outcome.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We used the SpaceX API and we extracted the data with web scraping techniques.
- Perform data wrangling
  - We filtered the data, replace the nulls by the mean, and applied one hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

1. Request the data to the SpaceX API.
2. Decoded the response into a .json and then converted it into a data frame.
3. Filtering the Dataframe to include only Falcon9 launches.
4. Replace the nulls in the field Payload Mass by its mean.
5. Save the data into a .csv

Data Collection – SpaceX API

- We used get request to extract the data from the API, and the cleaned and wrangled the data

- The link is: https://github.com/ccabf/CapstoneProyect/blob/main/week1-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
]: response = requests.get(spacex_url)
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
response.json()
data=pd.json_normalize(response.json())
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=df[df['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

```
# Calculate the mean value of PayloadMass column
meanpay= data_falcon9['PayloadMass'].mean()
meanpay
# Replace the np.nan values with its mean value
gfg['G2'].fillna(value=mean_value, inplace=True)
```

Data Collection - Scraping

- We extracted the Data from a statics url using Beautiful Soup.
- Then we converted it into a table and transform it into a dataframe.
- https://github.com/ccabf/CapstoneProyect/blob/main/week1-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&ol
```

Next, request the HTML page from the above URL and get a `response` object

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).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup_object = BeautifulSoup(response,  "lxml")
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Data Wrangling

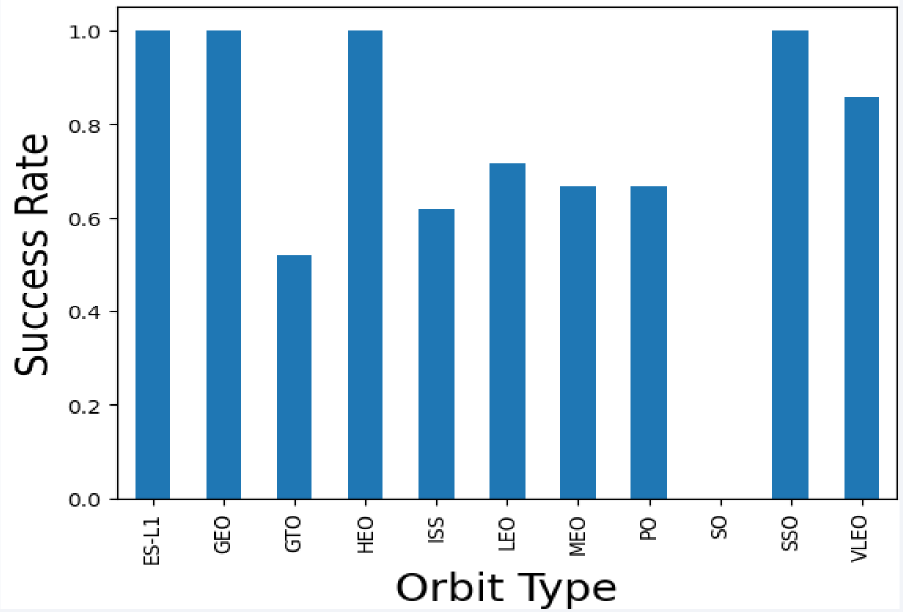
- First we calculate the percentage of nulls on each of the fields.
- We calculated the number of launches on each site.
- We determined the number and occurrence of each orbit.
- We calculated the number and occurrence of mission outcomes of the different orbits.
- Lastly, we created a new column named Class to determine whether the launch was successful or not.

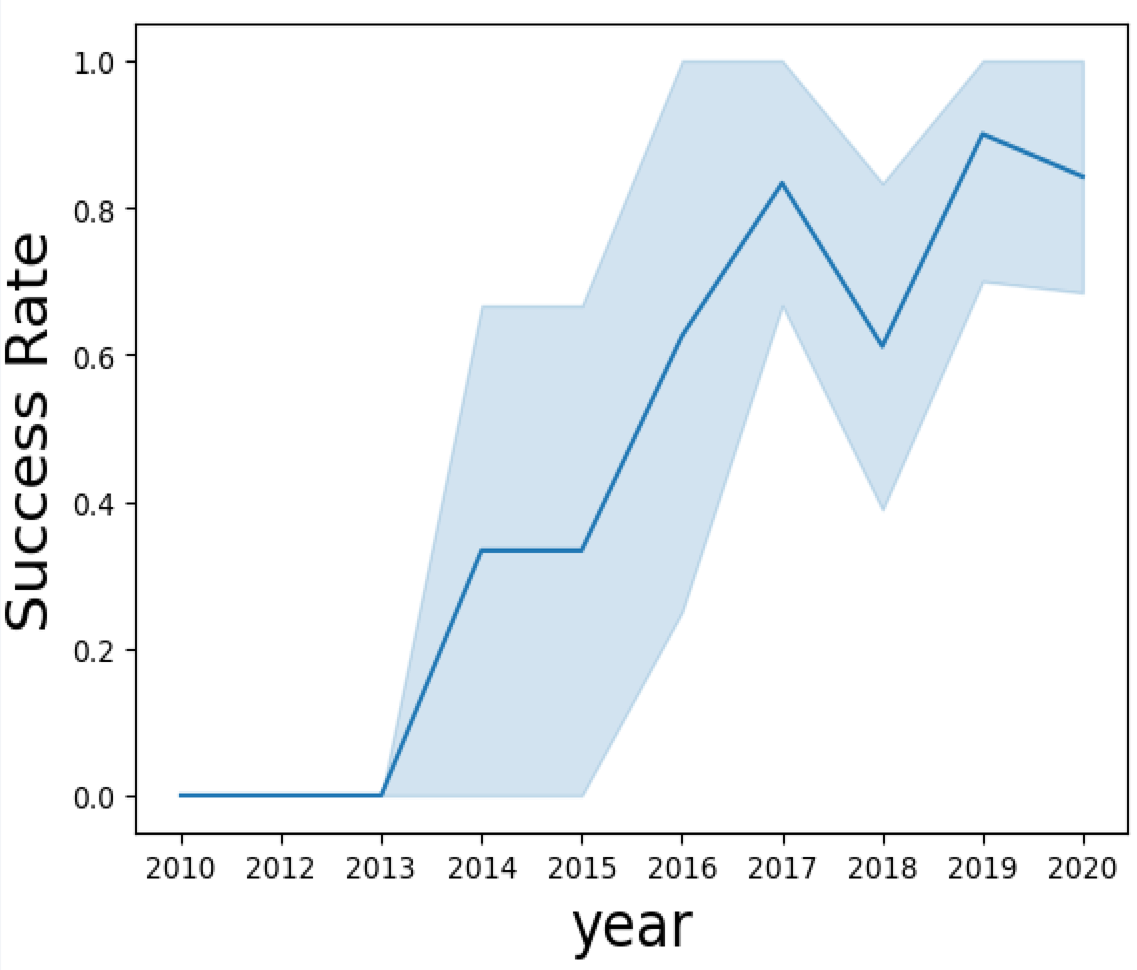
https://github.com/ccabf/CapstoneProyect/blob/main/week1-Data%20wrangling.ipynb

EDA with Data Visualization

- We analyzed the success rate trough other variables as the orbit type or the trend of the success over the years

https://github.com/ccabf/CapstoneProyect/blob/main/week2-eda-sql.ipynb





EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- We added the launch sites as circles first.
- Then we added the launches outcome in different colors, if it has been successful, it would be green, and if not, red.
- We identified which launch sites have considerably high success rate.
- We estimated the distances to different points like the coast, railways, highways and cities.

https://github.com/ccabf/CapstoneProyect/blob/main/week3-folium-launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- We have used Plotly to build an interactive dashboard.
- We used pie charts to show the total number of launches on different sites.
- By scatter plots we have represented the relationship between the Outcome and Payload mass (Kg) for different booster versions.
- https://github.com/ccabf/CapstoneProyect/blob/main/app-plotly.py

Predictive Analysis (Classification)

- We loaded the data and separated it into train and test sets.
- We applied different models as: Logistic regression, support vector machine, decision tree, and Kneighbors.
- We calculated the accuracy of each model.

https://github.com/ccabf/CapstoneProyect/blob/main/week4-Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

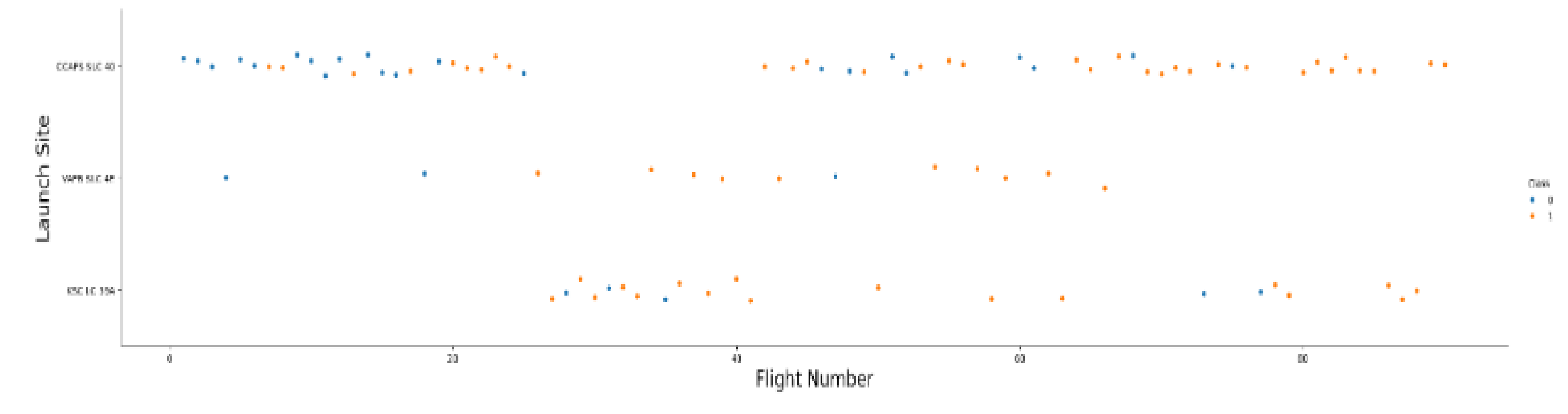
Results

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

Section 2

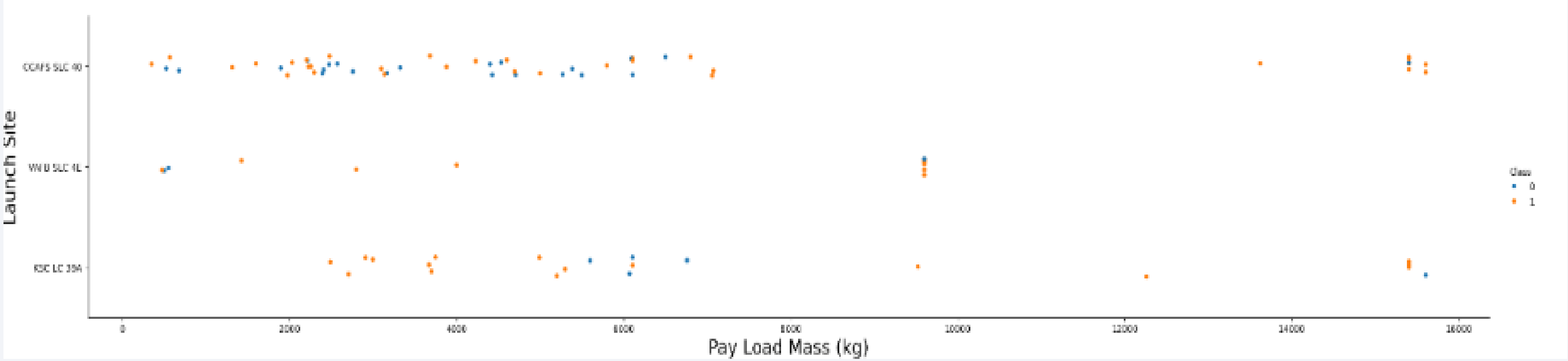
Insights drawn from EDA

Flight Number vs. Launch Site



We can determine that the higher the number of flightts from a launch site, the grater the probability of success

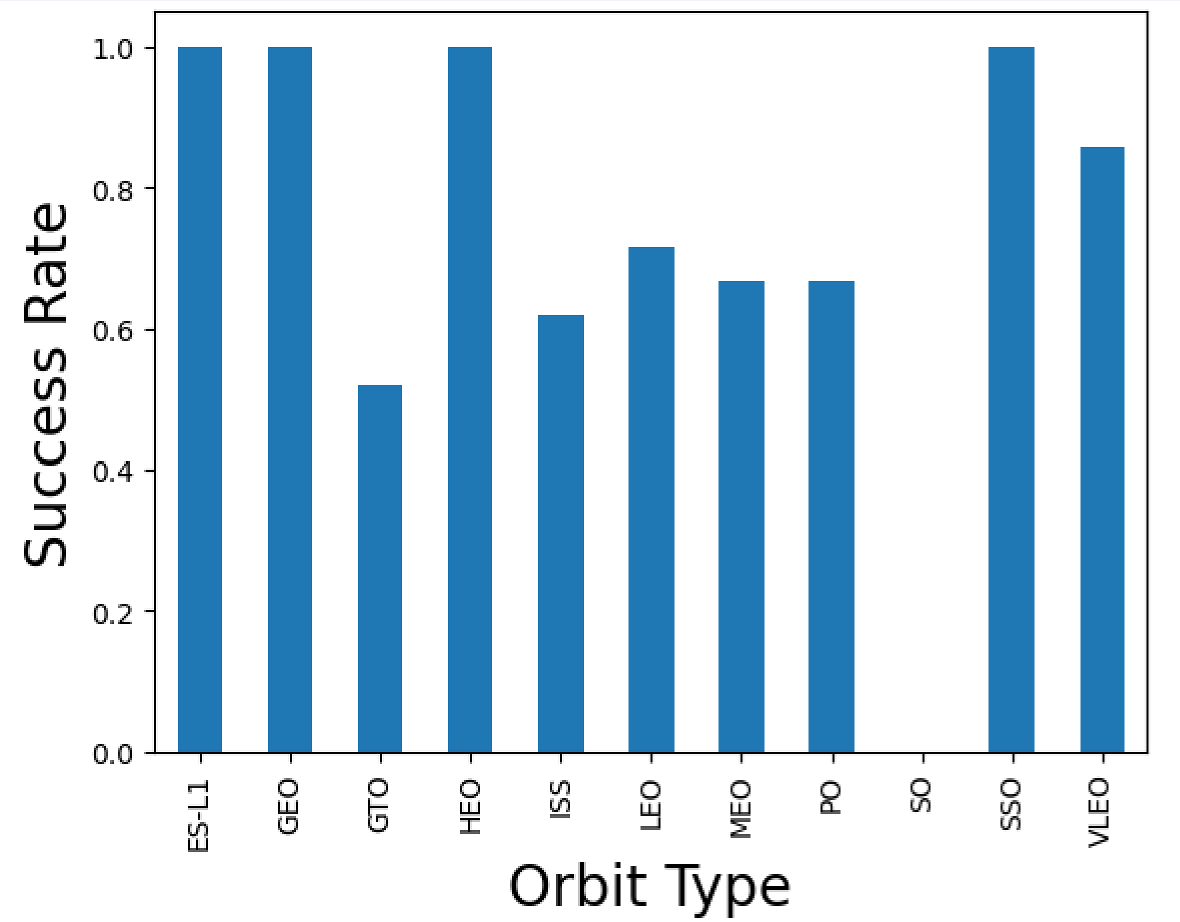
Payload vs. Launch Site



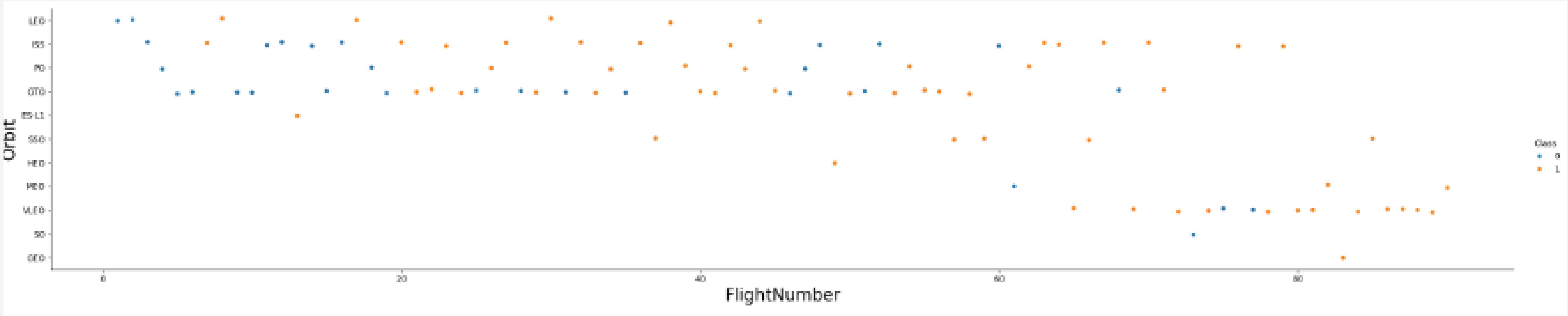
We have observed that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type

- Some of the orbits like the GEO, ES-L1, HEO and SSO have a success rate of 1, the 100% of the launches are successful.
- There are others like the GTO that has a minor rate.

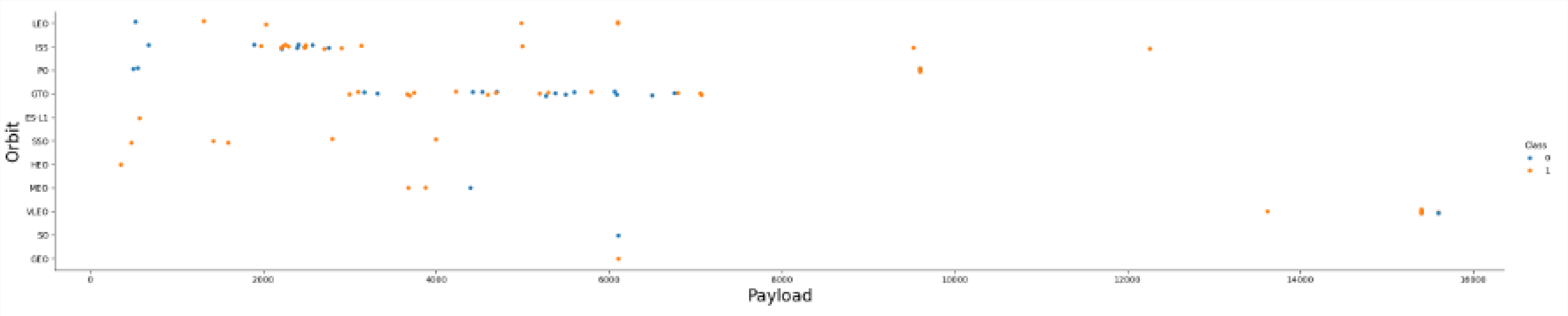


Flight Number vs. Orbit Type



In this plot we can see the Flight number represented against the Orbit type. For some orbits like the LEO the number of flight affect to the success of the Flight, whereas on other orbits we can not see such relationships.

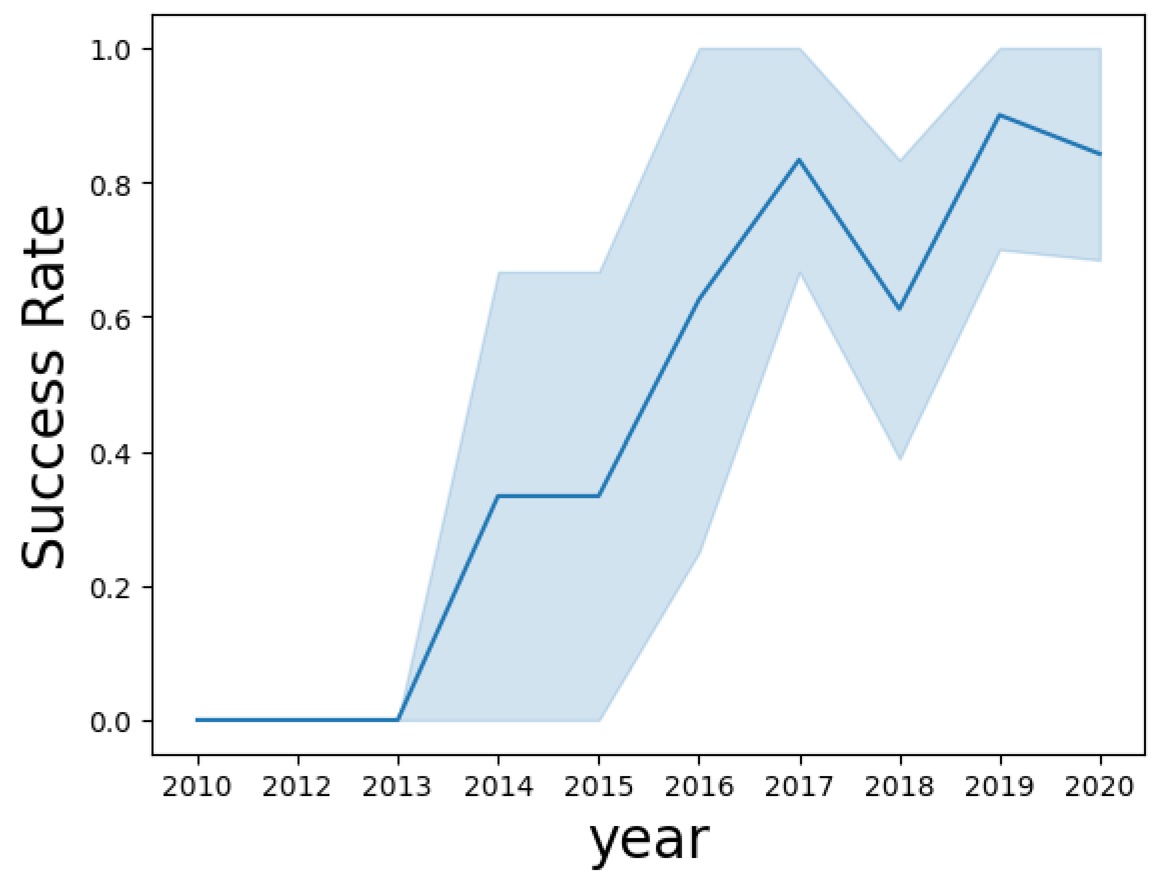
Payload vs. Orbit Type



We have plotted the Payload against the orbit. We can see that in orbits as the LEO or the PO there is a clear relationship between these two variables, with small payloads, there is no success in the landing, whereas the exit od the landing is more probable when working with heavy payloads.

Launch Success Yearly Trend

Considering the success rates, we can clearly see a significant increase from 2013 until 2020.



All Launch Site Names

The DISTINCT command display the unique values for the LAUNCH_SITE field.

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

```
 * 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'

The command LIKE is used to search results that have something in common with the string you write. ‘CCA%’ means that search anything that starts 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
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

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

With the command SUM we can aggregate the payload.

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

- AVG is used to calculate the average of a field.

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT MIN(Date)  FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

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

MIN(Date)
2015-12-22

With MIN(Date) we can extract from the data the first successful landing outcome in the ground (also filtering the landing outcome with success ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Boosters Carried Maximum Payload

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT Booster_Version FROM SPACEXTBL 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

Here there are the list of the boosters that have the requisites mentioned.

2015 Launch Records

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

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

Here there are the total number of success and failure flights.

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

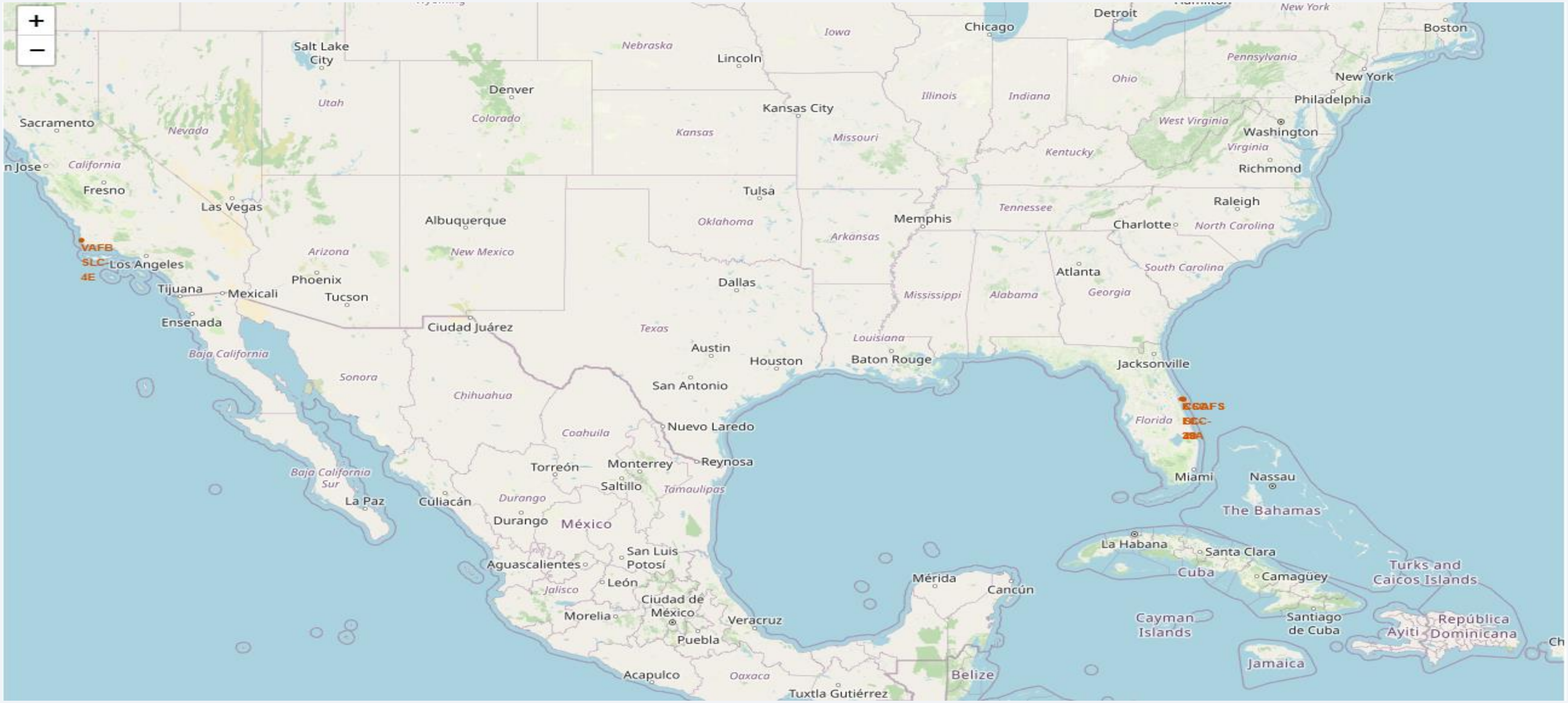
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Present your query result with a short explanation here

Section 3

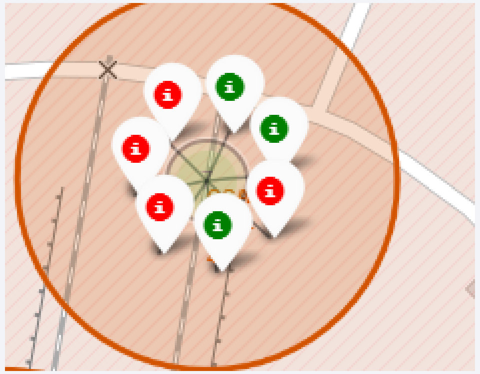
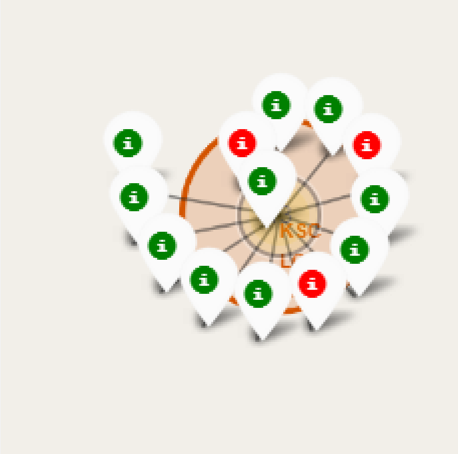
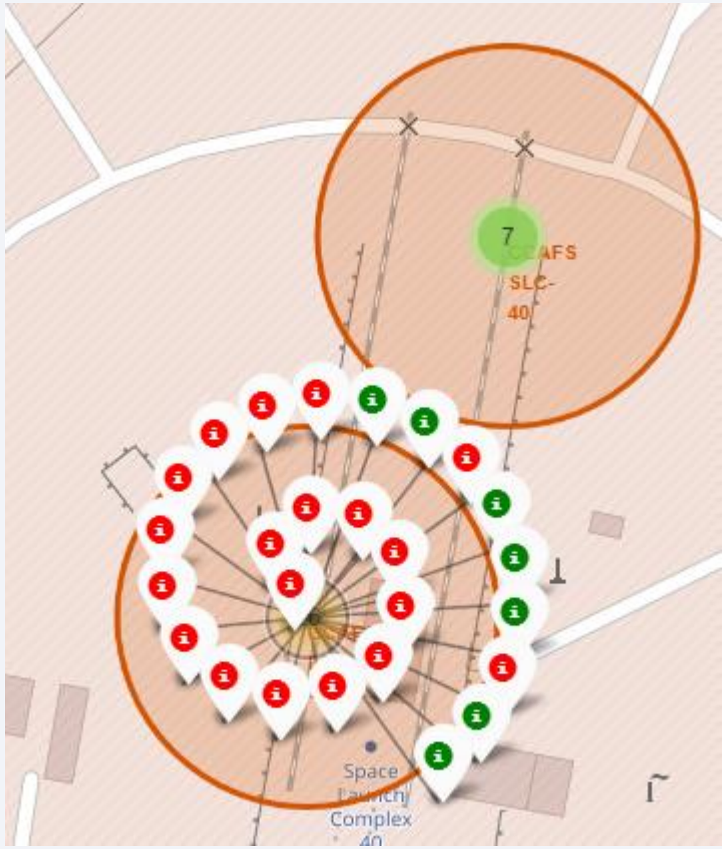
Launch Sites Proximities Analysis

Launch sites global markers

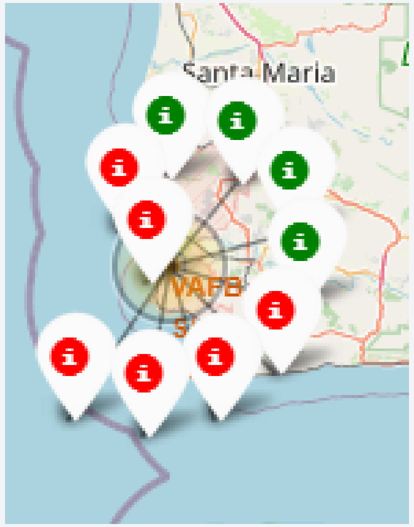


The launch sites are located in the United States.

Markers with color labels

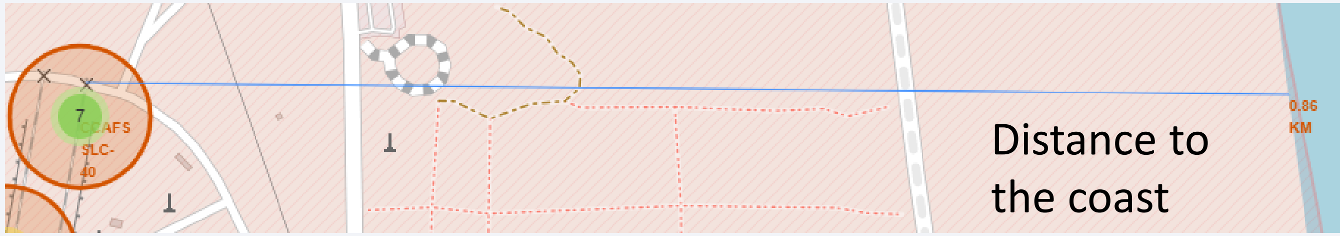
Florida Launch Sites.



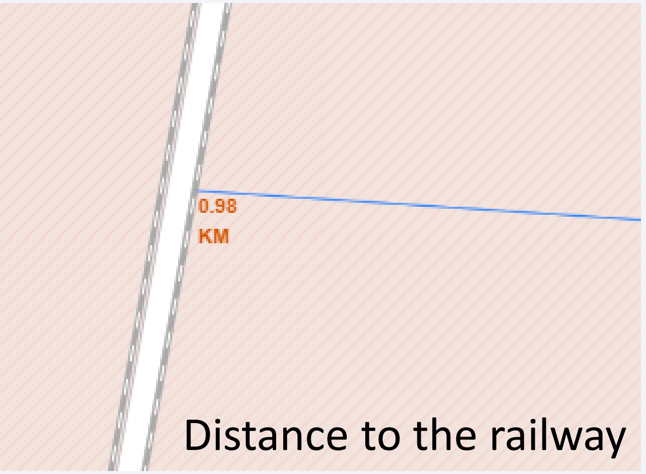
California

A green marker = successful launch.
Red marker = failure

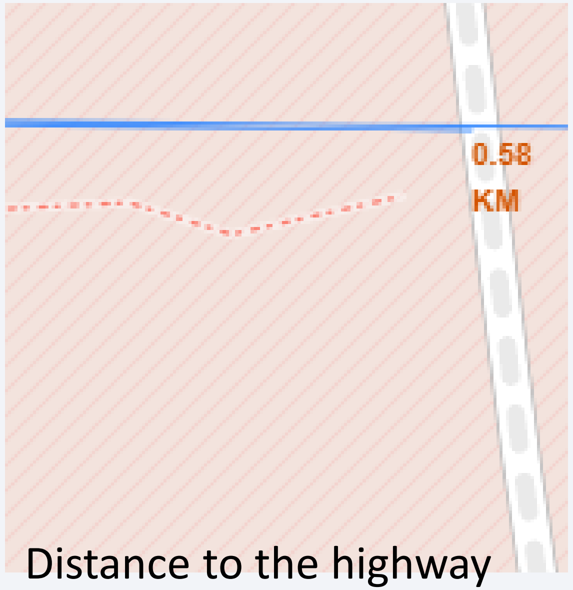
Landmarks distance



Distance to the coast



Distance to the railway



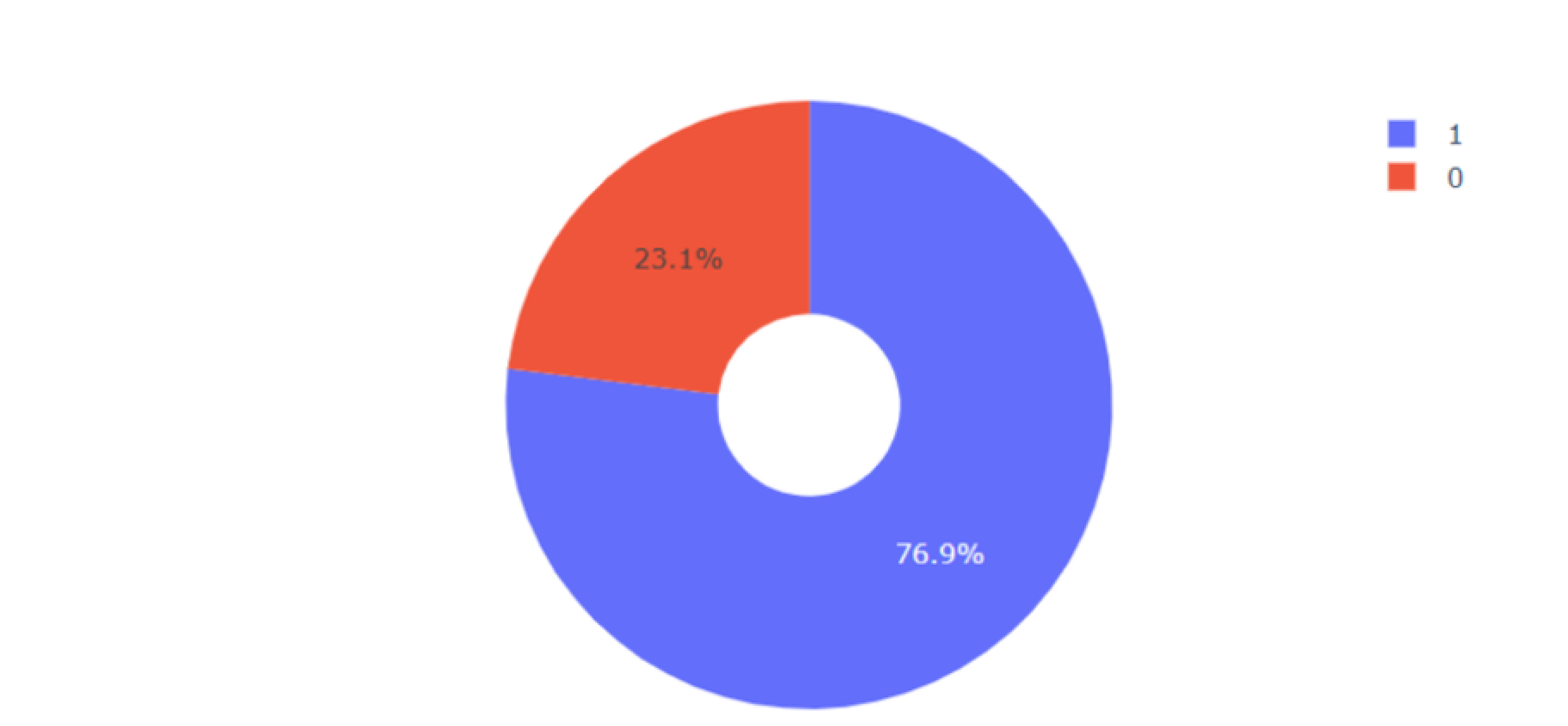
Distance to the highway

Distance to the city

Section 4

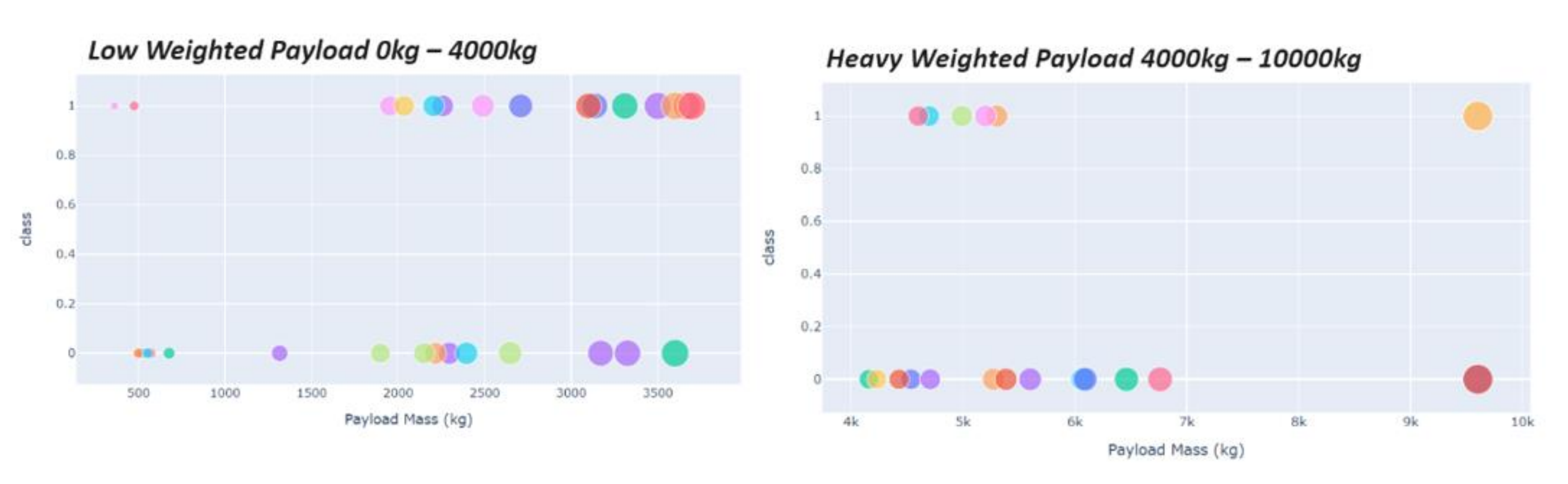
Build a Dashboard with Plotly Dash

Pie chart with the success rate of the Launch site with higher success ratio



KSC LC-39A has achieved a 76.9% success rate

Scatter plot of payload vs the launch outcome



Section 5

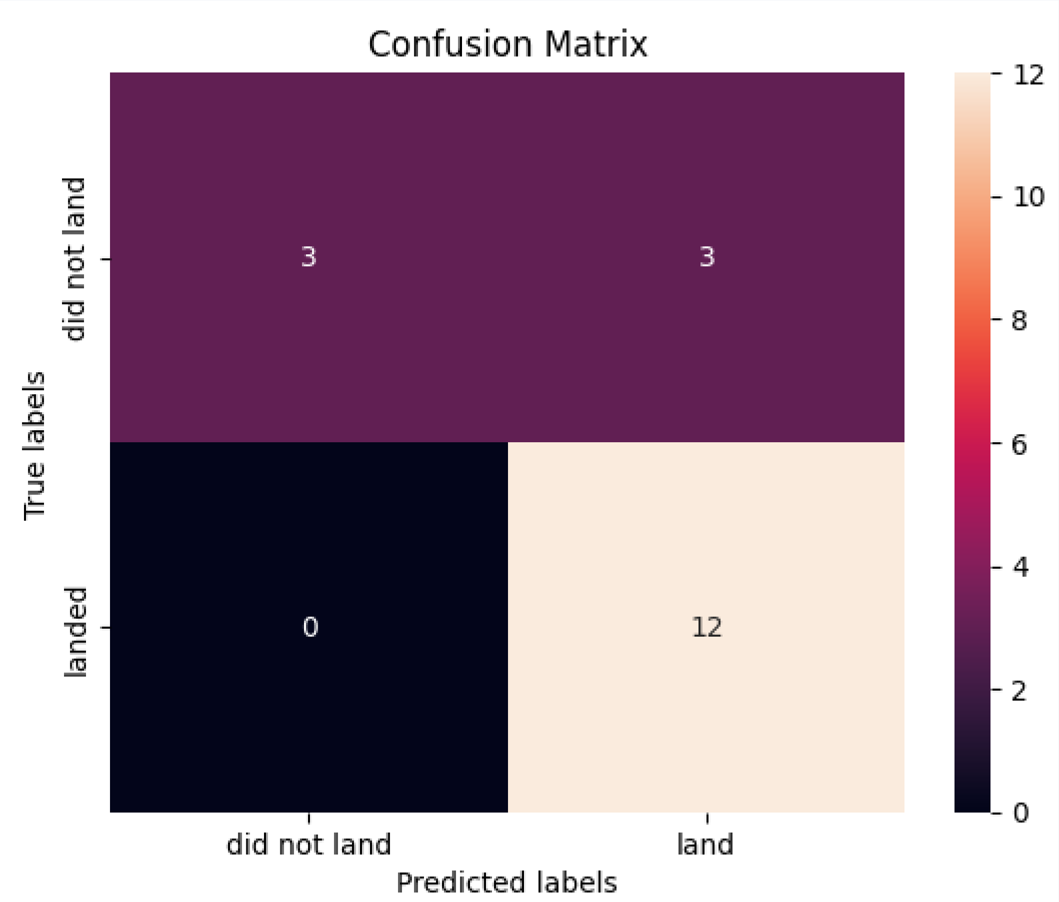
Predictive Analysis (Classification)

Classification Accuracy

```
The best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

The decision tree is the highest classification accuracy.

Confusion Matrix



This is the confusion matrix of the decision tree. There is an important problem like the failed launchings that have been classified as a successful landing.

Conclusions

- All the launching sites are located close to some landmarks as the coast, highways or railways, but far away from the cities.
- There has been a significant improvement in the launching success rate from 20113 to 2020.
- KSC LC-39A has the biggest success rate of any site.
- Comparing the different orbits, we have noted that the GEO, ES-L1, HEO and SSO have a success rate of 1, the 100% of the launches are successful, whereas others, like the GTO, have a minor rate.
- The decisión tree is the model that fits the best in this case.

Thank you!