IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rafael Serafim Agum
29/02/24

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Project history and context**
- **Problems you want to find answers to**

Seção 1

Methodology

Methodology

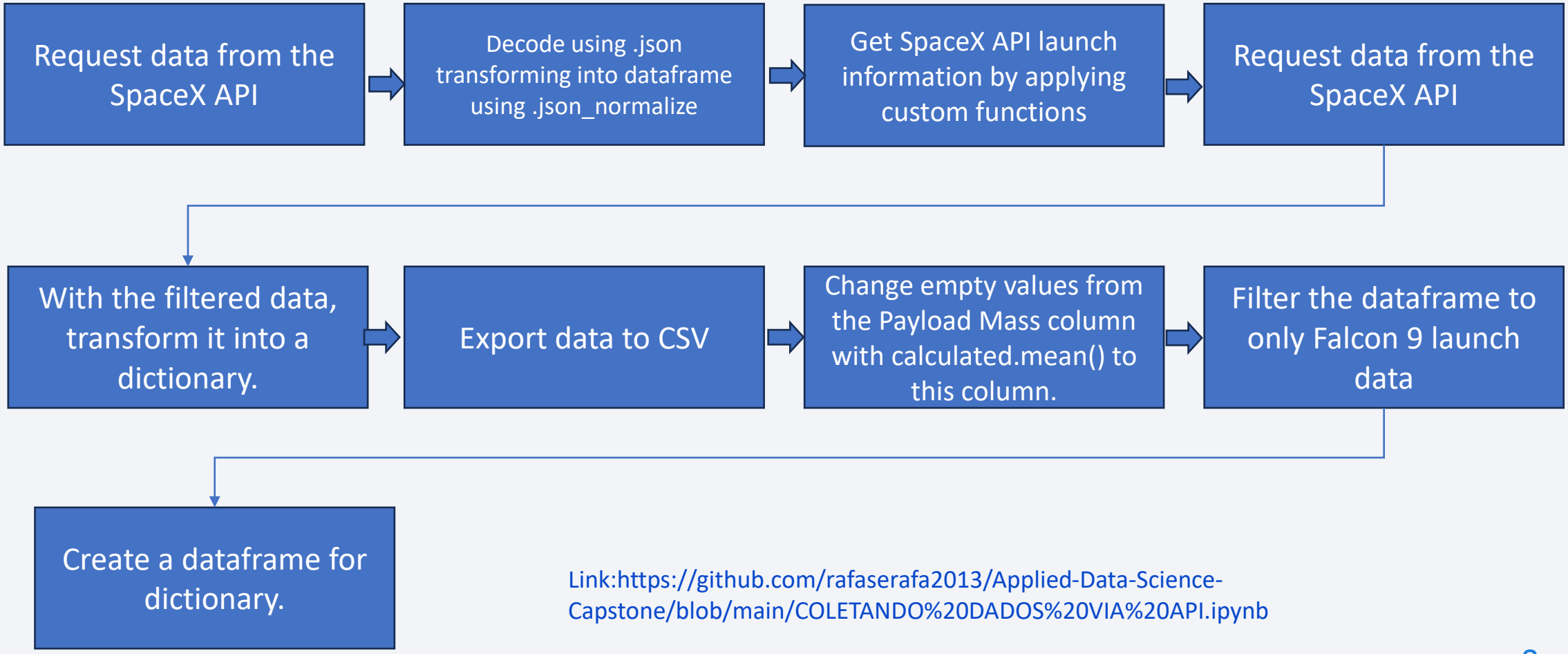
Sumário Executivo

- Methodology for data collection:
  - Data was collected using SpaceX's REST API and data scraping via Wikipedia
- Data Wrangling
  - Data was processed using one-hot coding for categorical features.
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analytics using machine learning models
  - Construction, adjustment and evaluation of classification models to ensure the best results.

Data collect

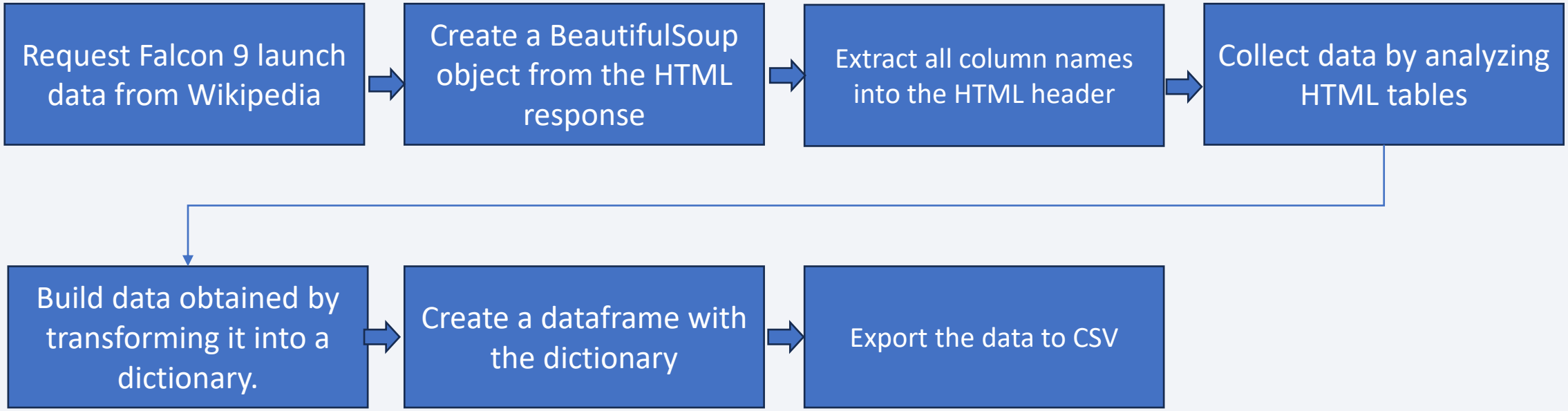
- Data collection is the process of gathering and measuring information about variables in an established system, which then allows you to answer questions and evaluate results. As mentioned, the dataset was collected by RESTAPI and Web Scrapping from Wikipedia.
- For REST API, it is initiated using get request. Then, we decode the content response as Json and transform it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values, and filled in whatever was needed.
- For web scraping, we will use BeautifulSoup to extract the release records as HTML table, parse the table and convert it to a pandas dataframe for further analysis.

Data collection via SpaceX API



Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/COLETANDO%20DADOS%20VIA%20API.ipynb

Data Collection via Data Scraping



Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/COLETANDO%20DADOS%20VIA%20WEB%20SCRAPPING.ipynb

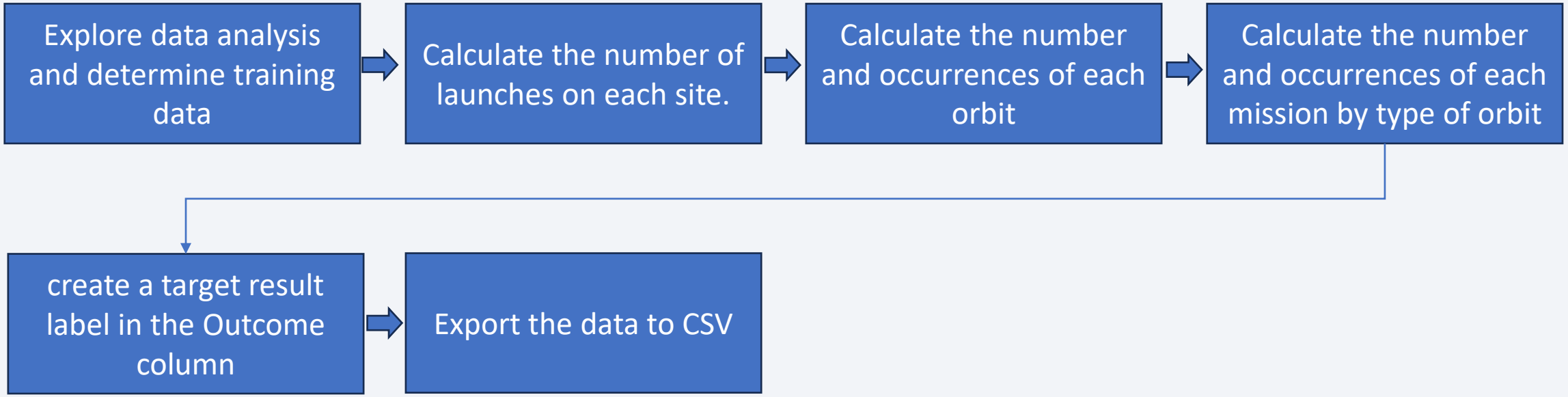
Data Wrangling

- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).

- We will first calculate the number of launches at each location and then calculate the number and occurrence of mission outcome by orbit type.

- Then we create a target result label from the result column. This will make it easier for future analysis, visualization and machine learning. Lastly, we will export the result to a CSV.

Data Wrangling



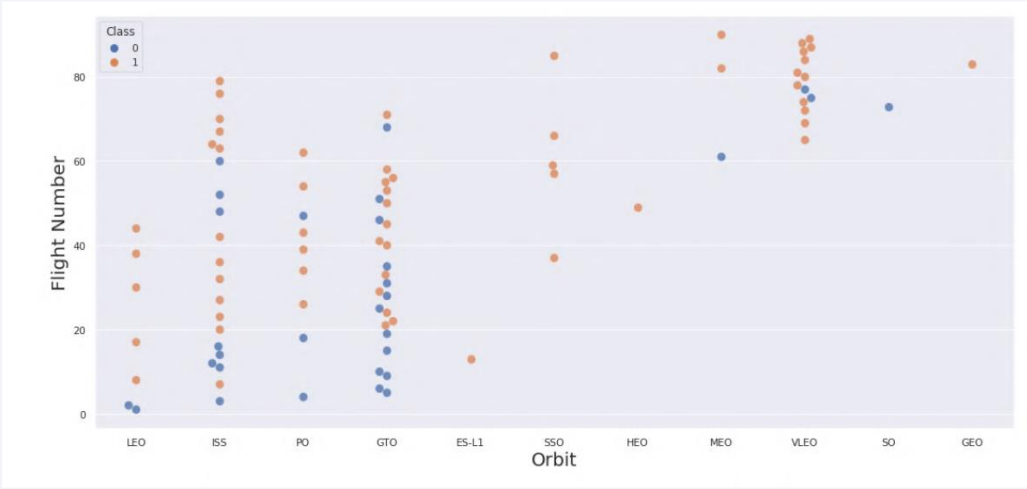
Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/DISPUTA%20DE%20DADOS.ipynb

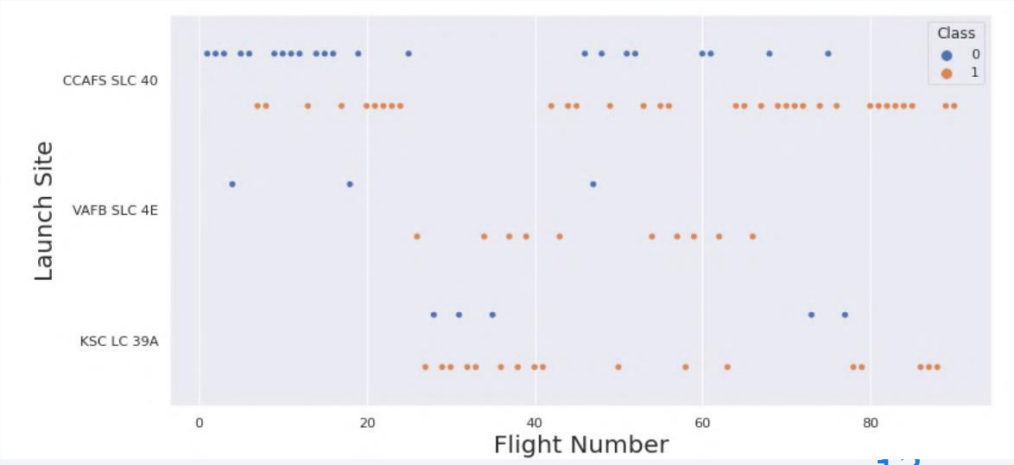
EDA with data visualization

We start by using the scatterplot to find the relationship between attributes, such as between:

- Payload and Flight Number.
- Flight number and launch location.
- Payload and launch site.
- Flight number and orbit type.
- Payload and orbit type.

Scatterplots show the dependence of attributes on each other. Once a pattern is determined from the charts, it is very easy. See which factors most affect the success of landing results.

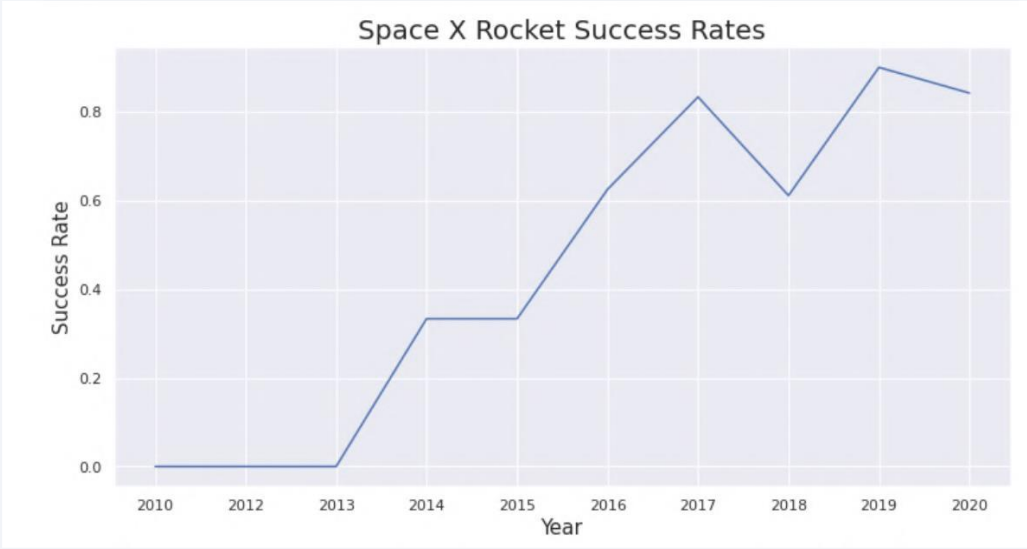


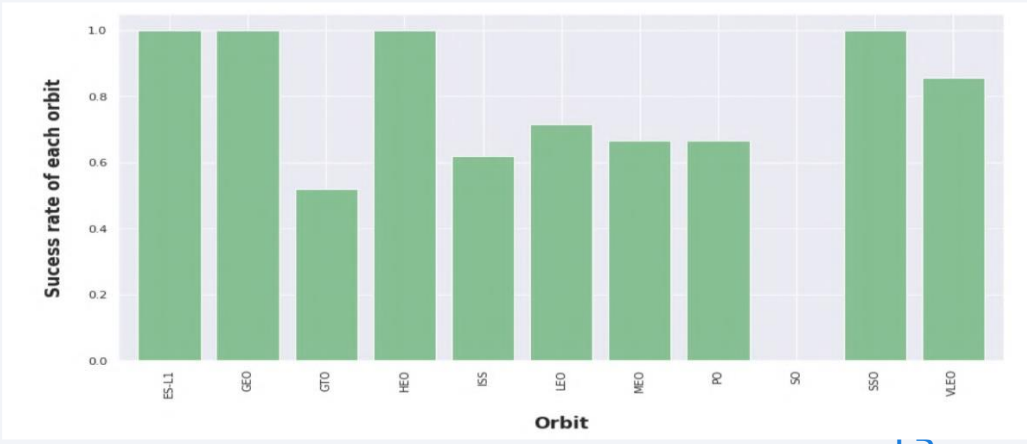


Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/EDA%20COM%20VISUALIZA%C3%87%C3%83O%20DE%20DADOS.ipynb

EDA with data visualization

This way, we were able to find relationships using the scatter plot. We will then use other visualization tools such as bar charts and line plots for exploratory analysis. Bar charts are one of the easiest ways to interpret the relationship between attributes. In this case, we will use the bar chart to determine which orbits have the highest probability of success. We then use the line graph to show trends or pattern of the attribute over time which in this case is used to see the trend of annual successful launch. We then use feature engineering to be successfully used in predicting the future module by creating variables for categorical columns.





Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/EDA%20COM%20VISUALIZA%C3%87%C3%83O%20DE%20DADOS.ipynb

EDA with SQL

Using SQL, we will perform many queries to better understand the data set, Ex:
- Displaying launch site names.
- Displaying 5 records where launch locations begin with the string ‘CCA’.
- Displaying the total mass of payload carried by the NASA-launched booster (CRS).
- Displaying the average mass of cargo carried by the F9 v1.1 booster version.
- List the date the first successful landing on the ground was achieved.
- Listing the names of boosters that are successful on drone ships and have payload mass greater than 4,000 but less than 6,000.
- Listing the total number of successful and failed mission results.
- Listing the names of booster_versions that carried the maximum payload mass.
- Listing failed drone ship landing_outcomes, their booster versions and launch locations names for the year 2015.
- Sorting the count of results or landing successes between the date 06/04/2010 and 03/20/2017, in descending order.

Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/EDA%20COM%20SQL.ipynb

Building an interactive map with Folium

Markers for all launch sites:
- Added NASA Johnson Space Center circled label, pop-up label and text label using your latitude and longitude coordinates as your starting location.
- Added circled markers, pop-up label, and text label of all launch sites using their latitude and longitude coordinates to show their geographic locations and proximity to the Equator and coasts.

Colored launch result markers for each launch location:
- Added colored markers of successful (green) and failed launches (red) using Marker Cluster to identify which launch locations have relatively high success rates.

Distances between a Launch Site and its surroundings:
- Added colored lines to show distances between the KSC LC-39A launch site (example) and its surroundings such as Railway, Highway, Coastline and Nearest City.

Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/ANALISE%20DE%20INTERA%C3%87%C3%83O%20VISUAL%20COM%20FOLIUM.ipynb

Building a dashboard with Plotly Dash

We will build an interactive dashboard with Plotly dash that allows the user to allocate data according to their needs.

We will plot pie charts showing the total launches of certain sites.

We will then plot a scatterplot showing the relationship with Result and Payload Mass (Kg) for the different reinforcement versions.

Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/RAFASERAFA.APP.py

Predictive Analytics (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform data and then split into training data and test data.
- Decide which type of ML to use.
- define the parameters and algorithms for GridSearchCV and fits it to the dataset.

Evaluating the model

- Check the accuracy of each model.
- Obtain adjusted hyperparameters for each type of algorithm.
- plot the confusion matrix.

Improving the model

- Use feature engineering and algorithm tuning

Find the best model

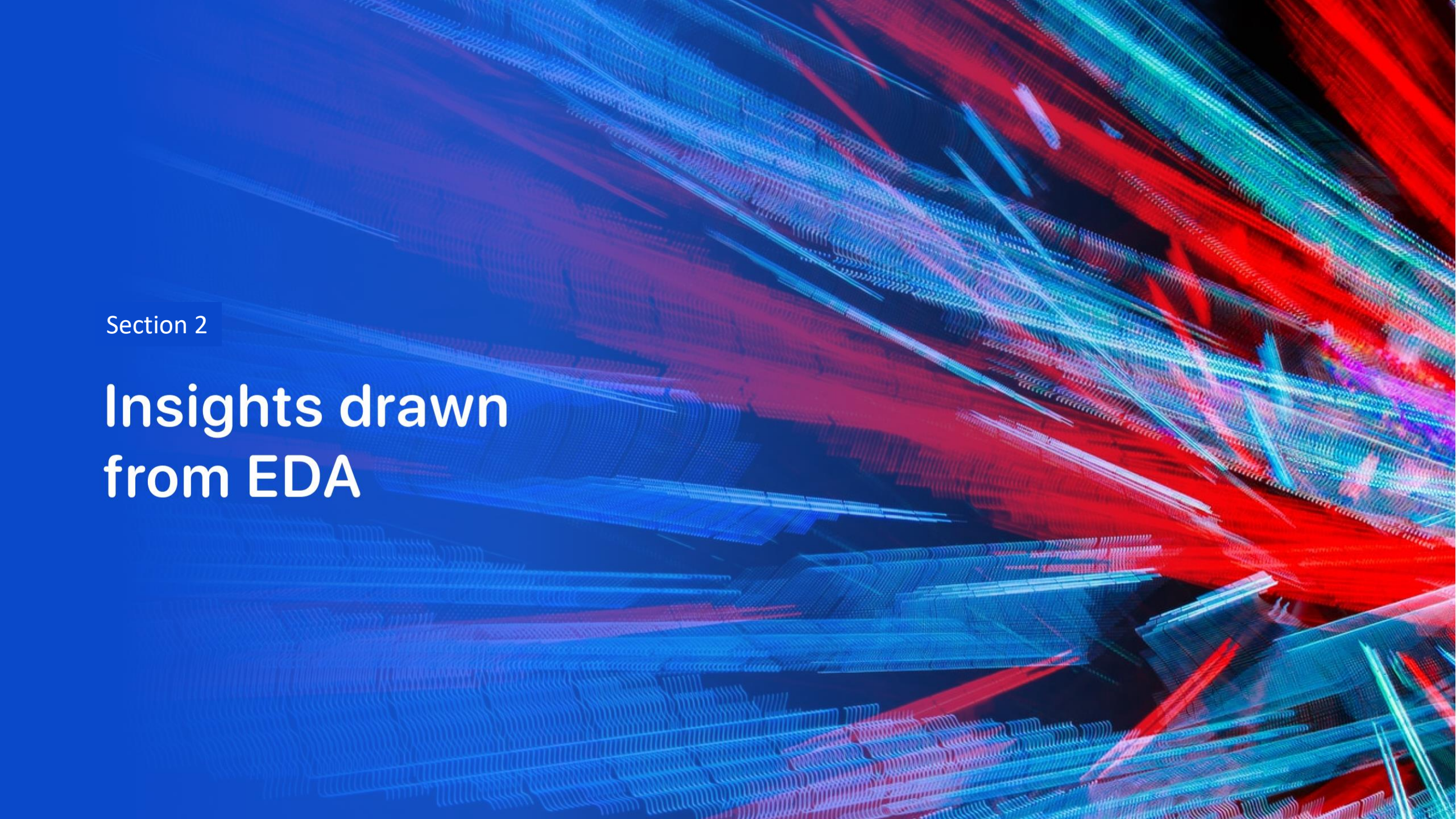
- The model with the best accuracy score will be the best performing model.

Link:https://github.com/rafaserafa2013/Applied-Data-Science-Capstone/blob/main/RAFASERAFA.APP.py

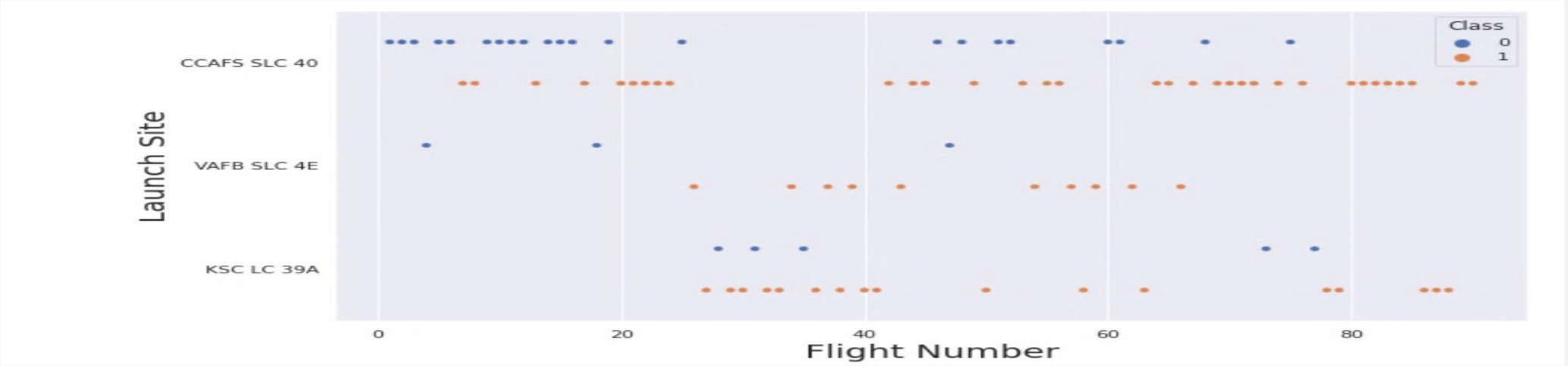
Results

The results will be categorized into 3 main results, which will be:

- Results of exploratory data analysis.
- Demonstration of interactive analysis in screenshots.
- Predictive analysis results.


Section 2
Insights drawn from EDA

Flight numbers and launch pad



This scatterplot shows that the higher the value of flights from the launch site, the higher the success rate. However, the CCAFS SLC40 website shows the standard minimum of this.

Points to note:

- All the first flights failed, while the last flights were successful.
- The CCAFS SLC 40 launch site holds about half of all launches.
- VAFB SLC 4E and KSC LC 39A had higher success rates.
- It can be assumed that each new launch has a higher success rate.

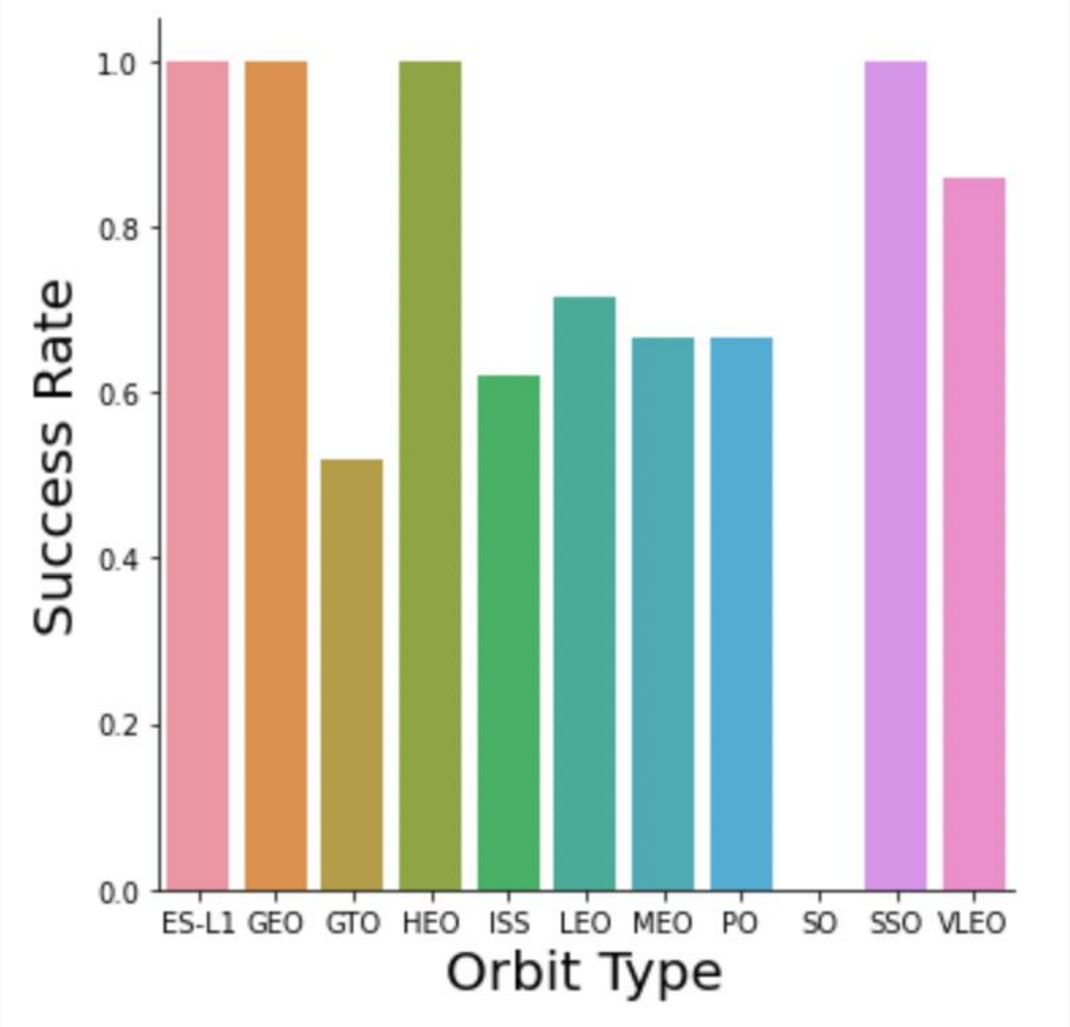
Payload vs. Launch site

This scatterplot shows that the mass of the payload is more than 7,000 kg, and the rate success probability will be highly increased. However, there is no clear standard to say whether site launch is dependent on payload mass for return success.

Other points to note:

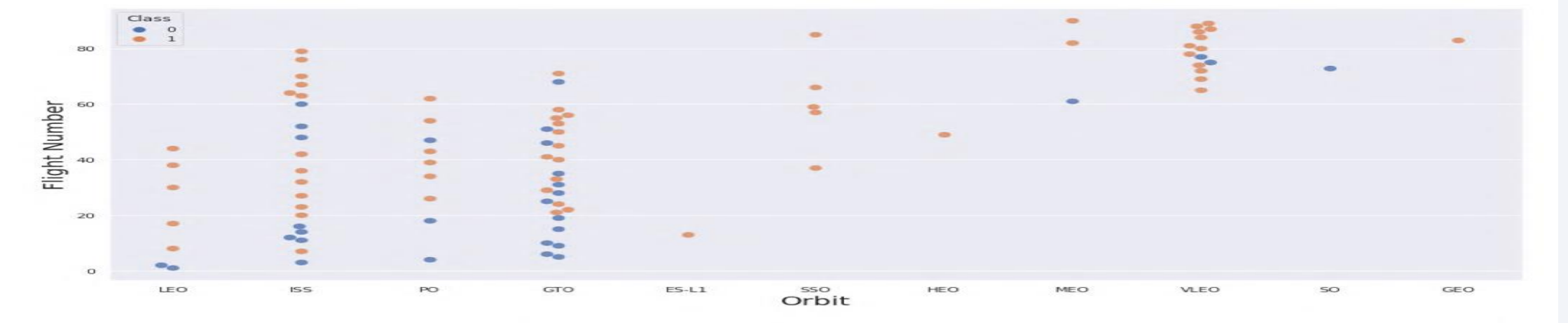
- For each launch site, the greater the mass of the payload, the greater the success to assess.
- Most launches with a useful mass greater than 7,000 kg were successful.
- The KSC LC 39A also has a 100% success rate for payload mass less than 5,500 kg.

Success Rate vs Orbit Type



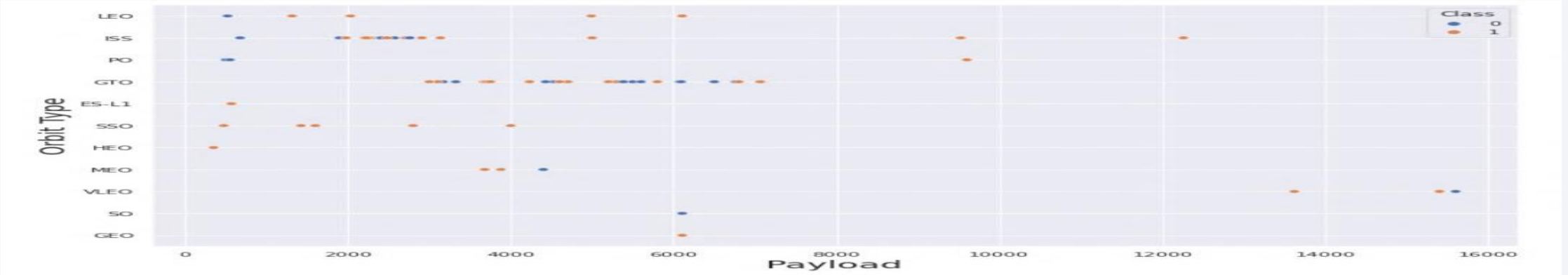
- Orbits with 100% success rate:
- ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate:
- ONLY
- Orbits with success rate between 50% and 85%:
- GTO, ISS, LEO, MEO, PO

Number of flights and type of orbits



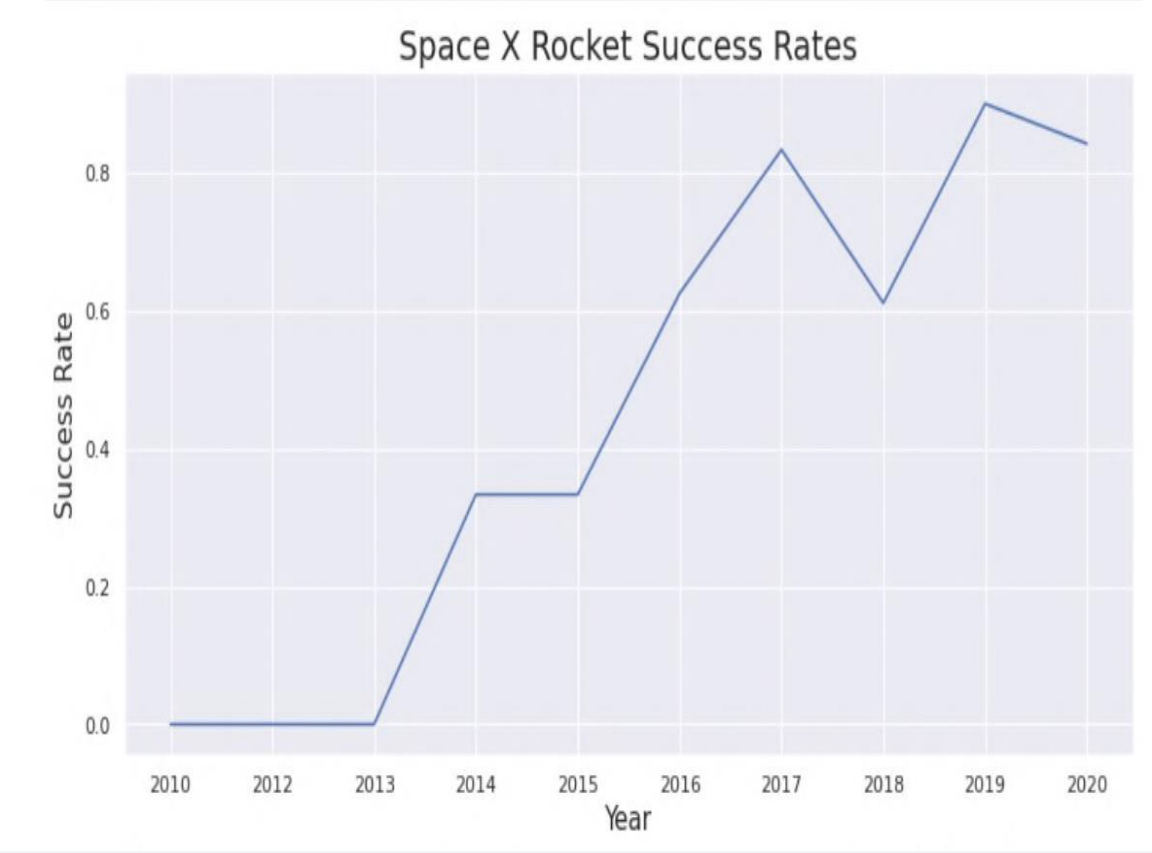
This scatterplot shows that generally, the greater the number of flights of each orbit, the higher the success rate (especially LEO orbit), except for GTO Orbit which showed no relationships between attributes. The Orbit that has only 1 occurrence should also be excluded from the list above as it requires more data to reach a conclusion.

Payload vs Orbit Type



Larger payload has positive impact on LEO, ISS and P0 orbit. However, it has a negative impact on MEO and VLEO orbit. The GTO orbit does not seem to represent a relationship between the attributes. Meanwhile, again the SO, GEO and HEO orbits need more data sets to observe any patterns or trends.

Annual trend of successful launches



This clearly represented figure has an increasing trend from 2013 to 2020. If this trend continues during the next year onwards, the success rate will be constant and will increase until reaching 1/100% Success Rate. It was only in 2017 to 2018 that there was a small drop, then recovering.

All launch pad names

```
In [5]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[5]:

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

We use the DISTINCT keyword to only show unique launch locations from SpaceX data.

Launch pad names that start with 'CCA'

We use the following query below to display 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total payload mass

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Total Payload Mass by NASA (CRS)
45596

We calculated the total payload carried by NASA boosters with a result of 45596 using the above query.

Average payload by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Average Payload Mass by Booster Version F9 v1.1
2928

We calculated the average mass of the payload carried by the F9 v1.1 booster version with the result of 2928.

First successful Earth landing date

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

We use the min() function to find the result. We note that the date of the first successful land landing result was December 22, 2015.

Pouso bem-sucedido de navio drone com carga útil entre 4.000 e 6.000

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datab
ases.appdomain.cloud:32731/bludb
Done.
```

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

The WHERE clause was used to filter boosters that successfully landed on a drone ship and the AND condition was applied to determine the successful landing with payload mass greater than 4,000, but less than 6,000 and found the above values.

Total number of successful and failed mission results

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Failure Mission
1

Characters like '%' were used to filter WHERE MissionOutcome was a success or failure.

Maximum payload transport by boosters

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

It was determined the reinforcement that has took the maximum payload using a subquery in WHERE clause and the MAX() function.

2015 release records

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.
databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

We used combinations of the WHERE, LIKE, AND and BETWEEN clause to filter the failed landing results on drone ships, their booster versions and launch site names for the year 2015.

Landing and classification results between 06/04/2010 and 03/20/2017

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu01qde00.databases.appdomain.cloud:32731/bludb
Done.
```

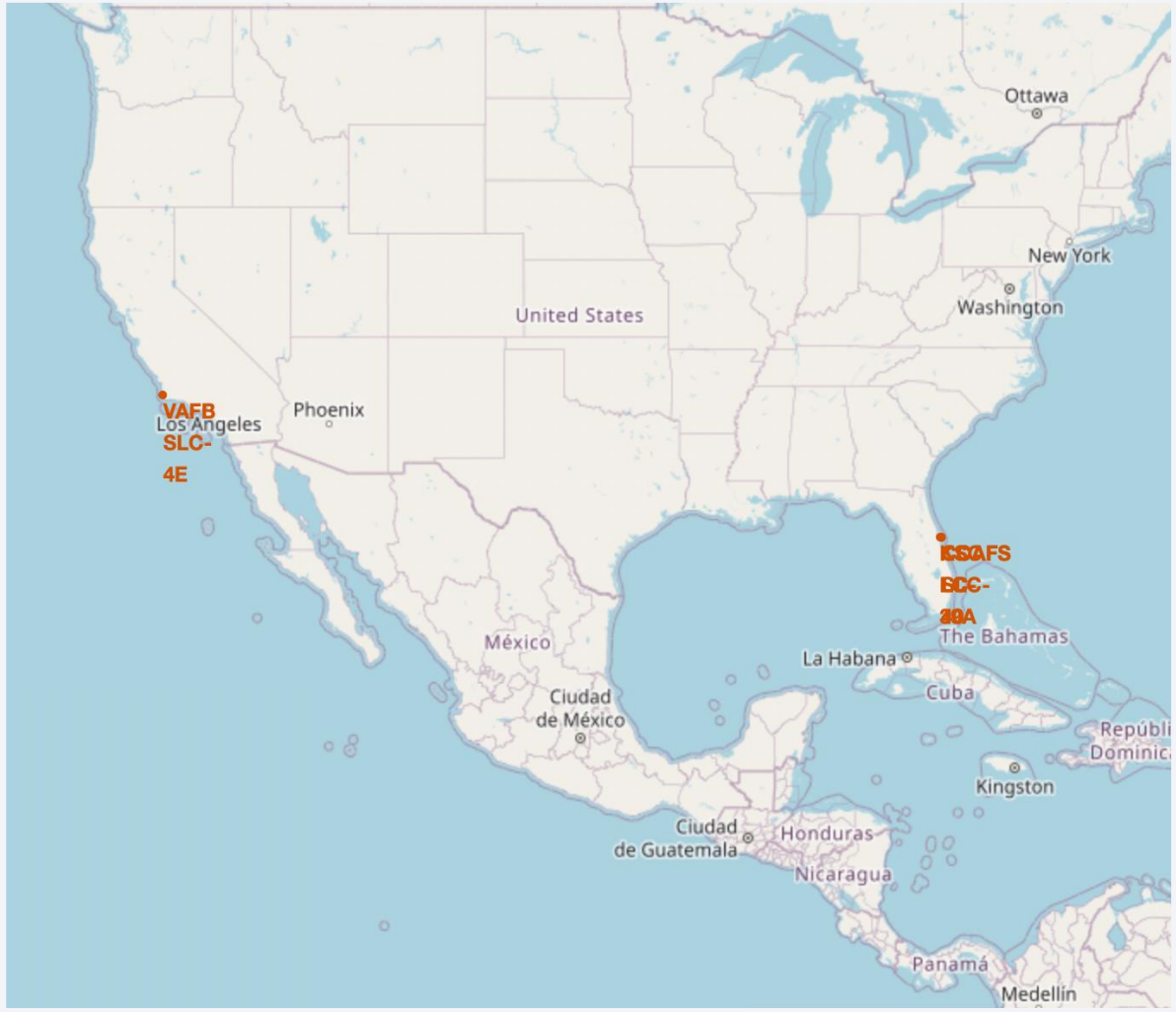
Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

This figure shows how the classification of the count of landing results as drone ship failure or ground base success between the dates 06/04/2010 and 03/20/2017 in descending order resulted.

Section 3

Launch Sites Proximities Analysis

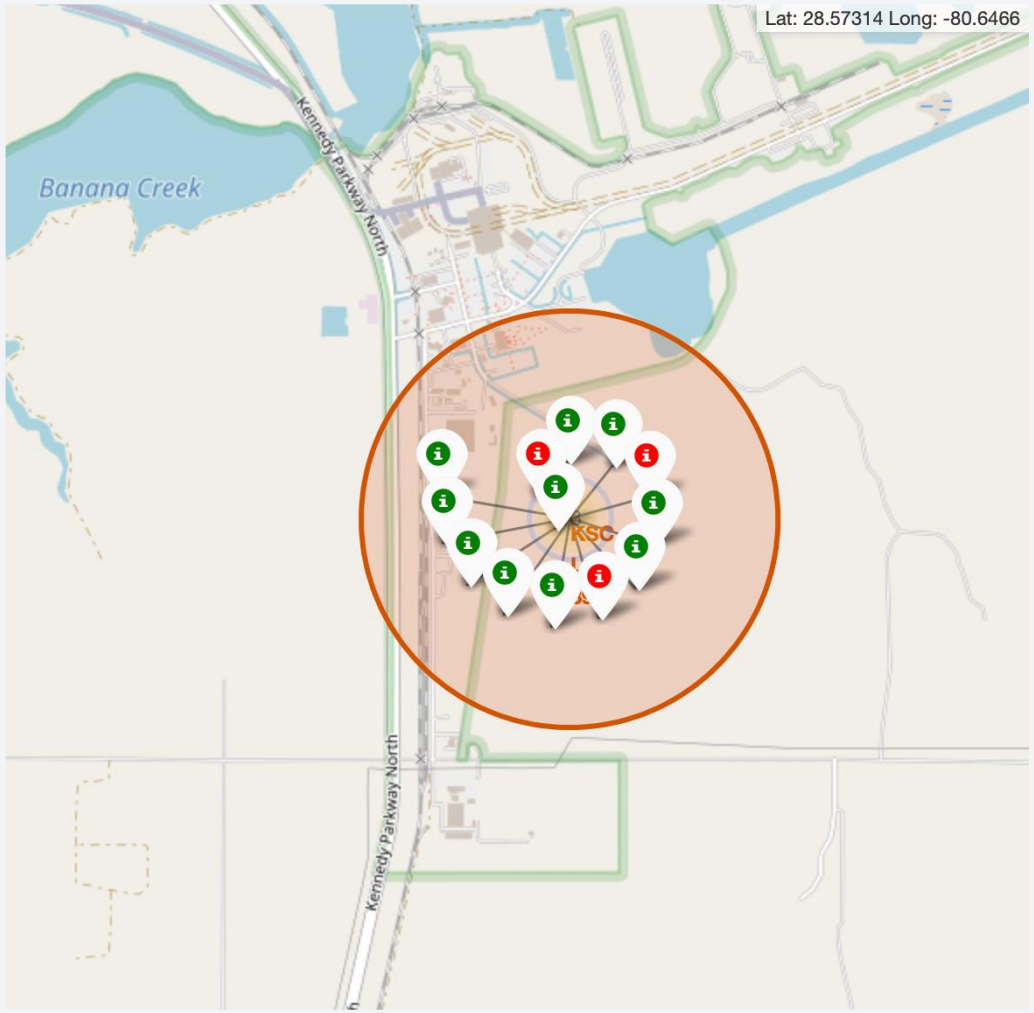
Launch pads marked on the map

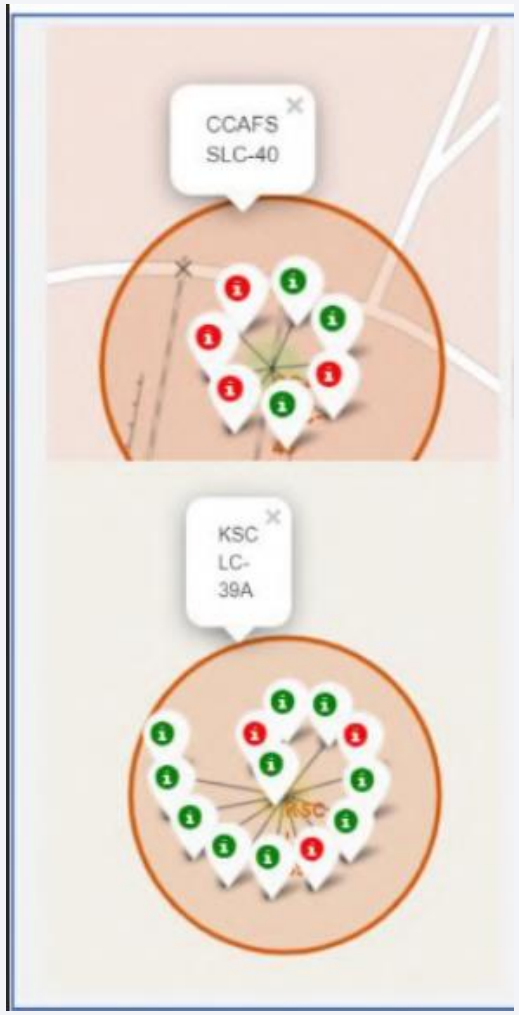


Most launch sites are close to the Equator Line. The Earth is moving faster around the equator than anywhere else on the Earth's surface. Anything on the Earth's surface at the equator moves at 1670 km/hour.

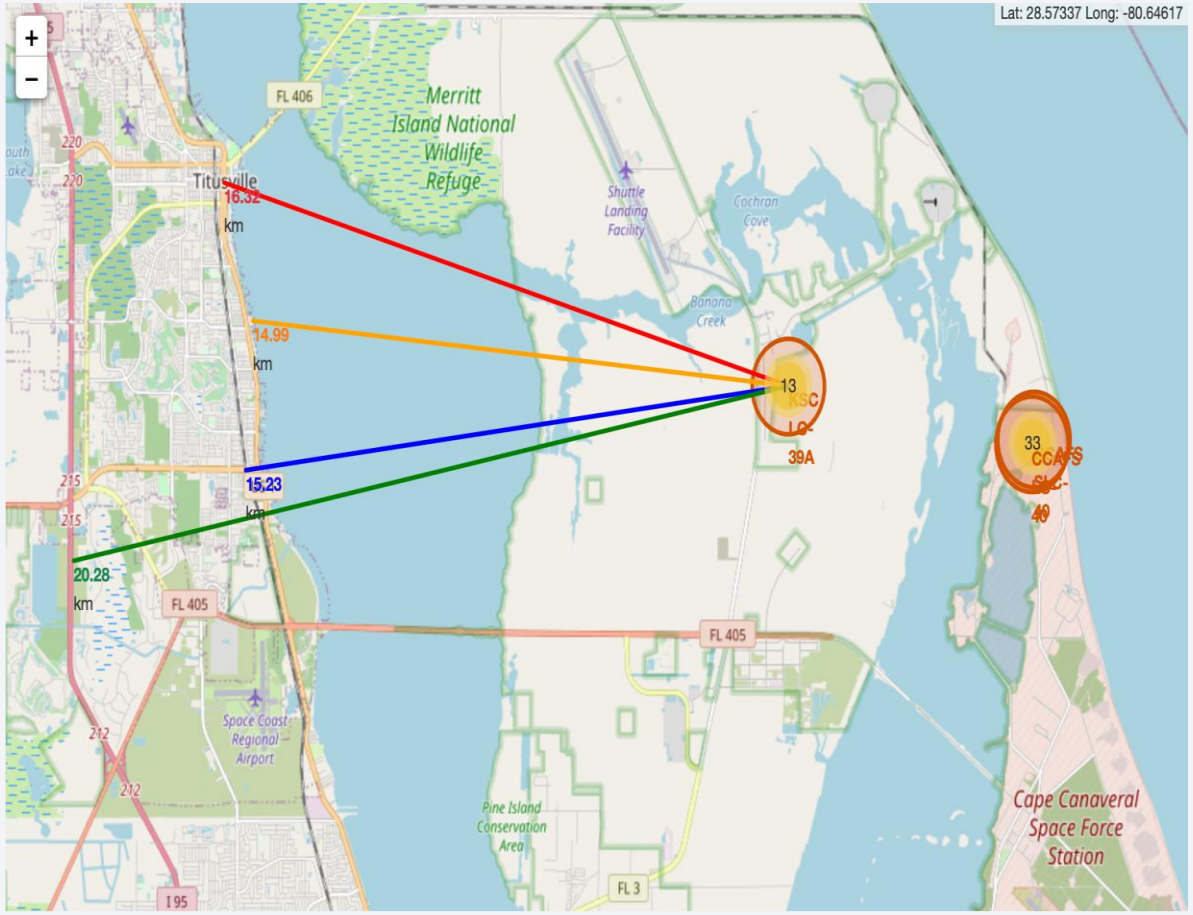
This speed will help the spacecraft keep up a speed good enough to remain in orbit. All launch sites are very close to the coast, while launching rockets toward the ocean minimizes the risk of having any debris land or explode near people.

Markers showing launch locations with colored labels





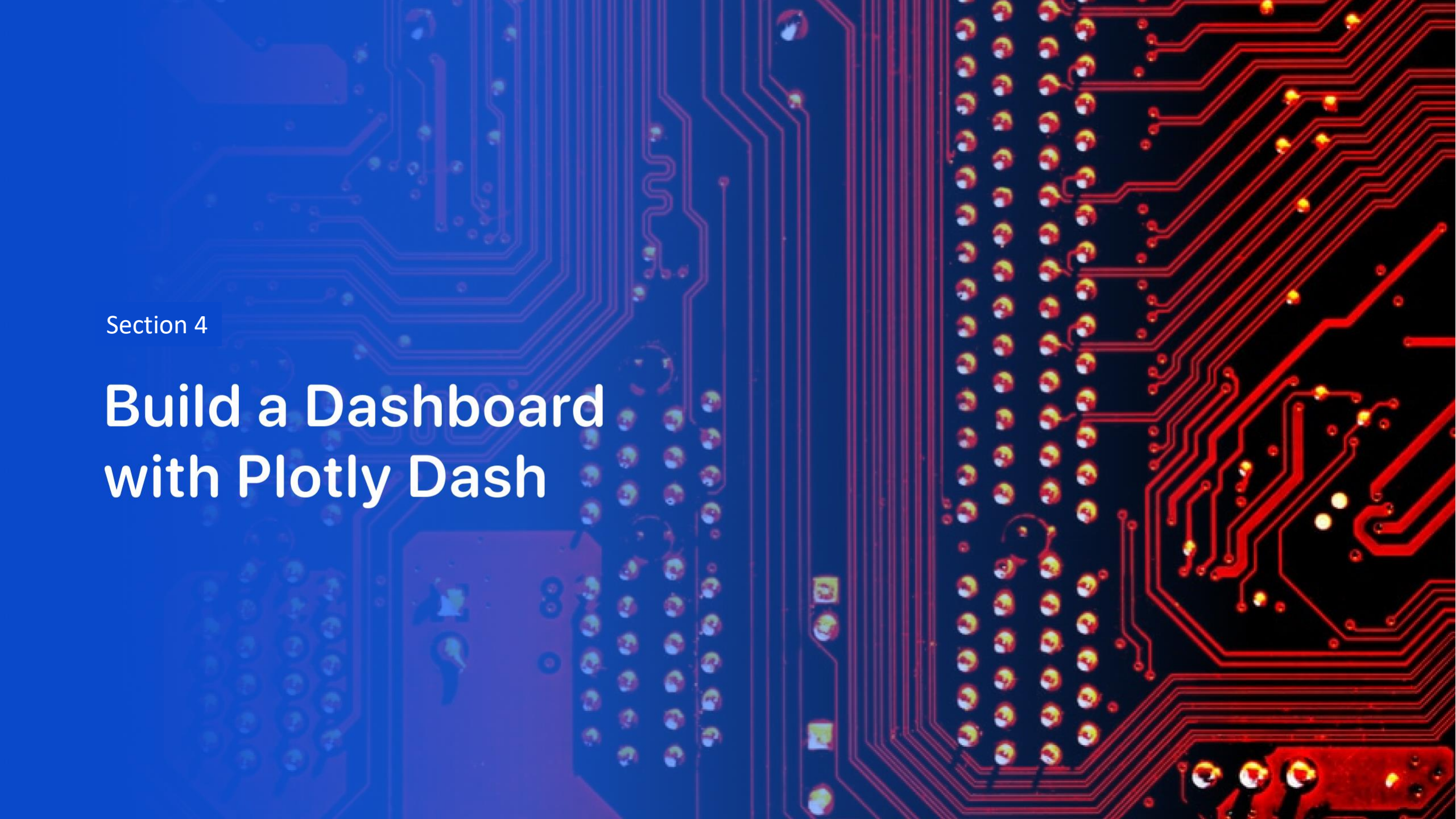
Distance from launch sites to landmarks



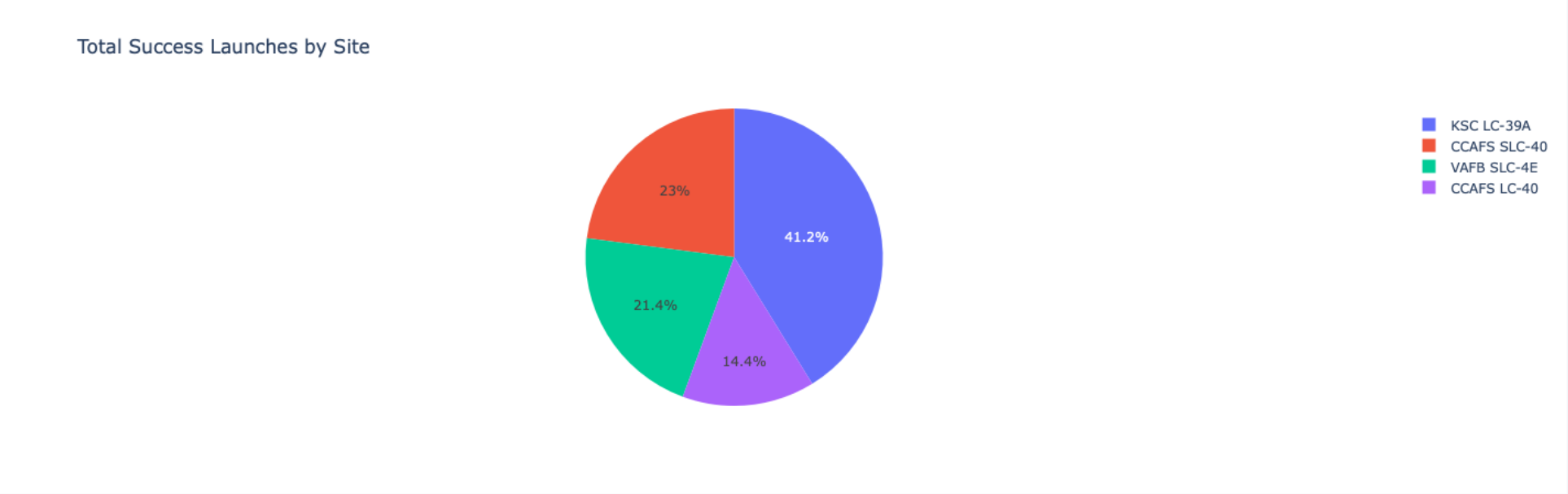
Visual analysis of the launch
KSC LC-39A website we can clearly see that that's it:
- relatively close to the railway (15.23 km)
- relatively close to the highway (20.28 km)
- relatively close to the coast (14.99 km)

• Additionally, the KSC LC-39A launch site is relatively close to your nearest city Titusville (16.32 km).

• Rocket failed with its high speed canister cover distances like 15-20 km in a few seconds. Could potentially be dangerous for populated areas.

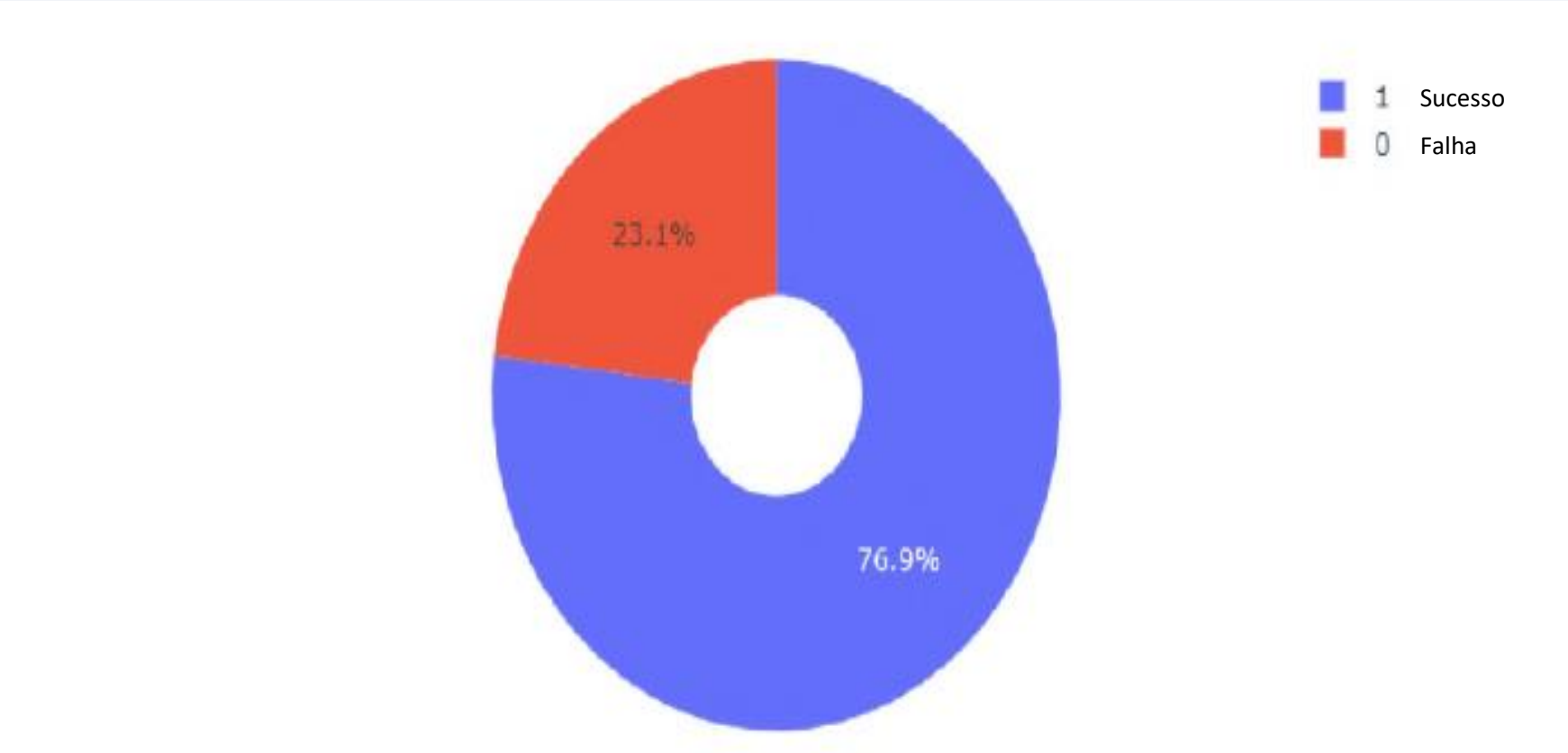

Section 4
Build a Dashboard with Plotly Dash

Launch success count for all sites

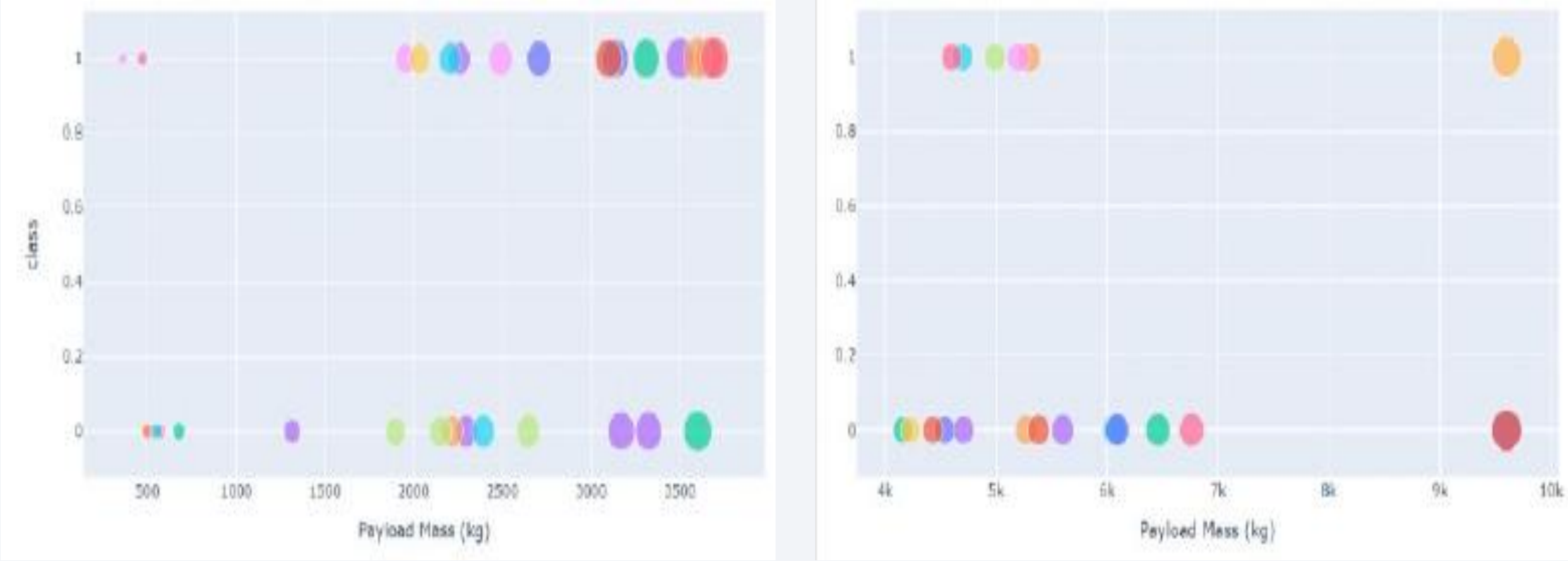


The graph clearly shows that of all the locations, KSC LC-39A has the highest utilization of successful launches.

Highest launch success rate: KSC LC-39A



Scatterplot of payload vs launch result



We can see that the success rate for low weight payloads is higher than that for high weight payloads.

Section 5

Predictive Analysis (Classification)

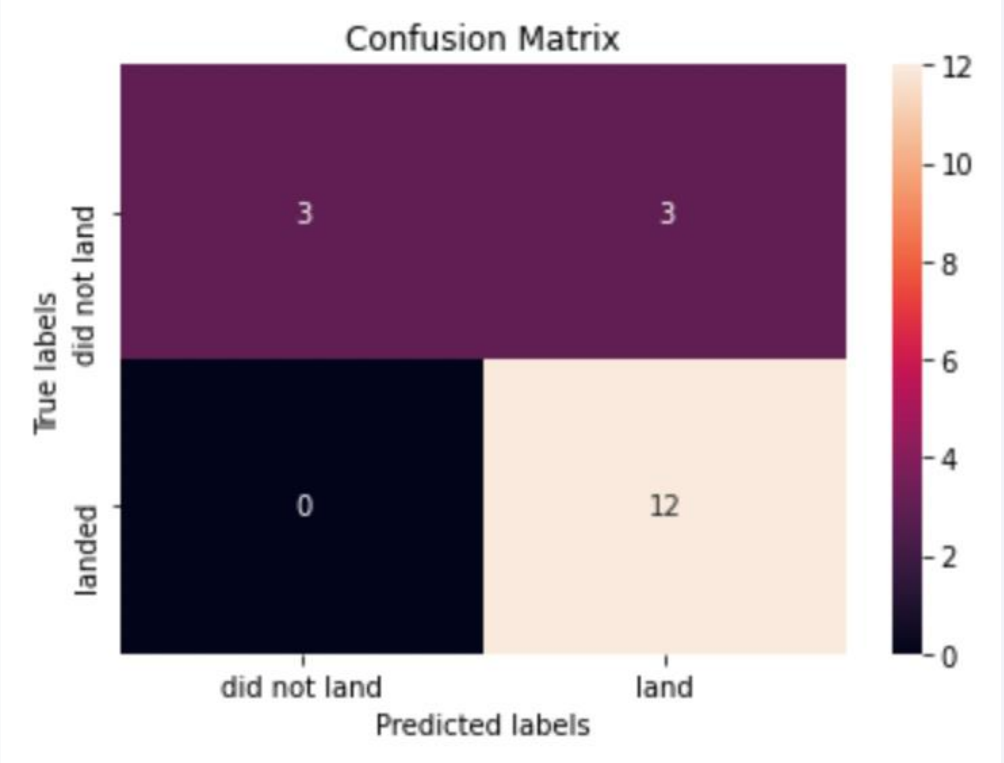
Classification Accuracy

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

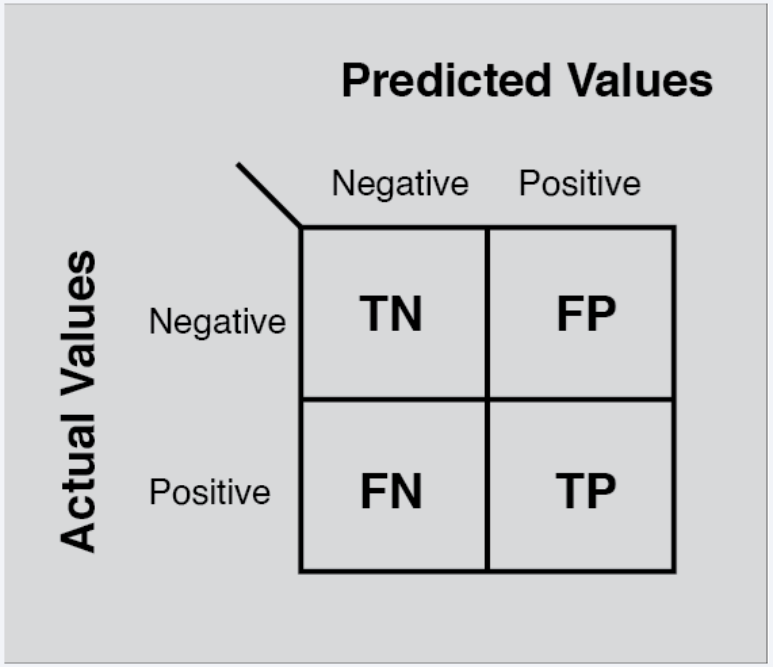
```
Best Algorithm is Tree with a score of 0.9017857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_sampl
es_split': 10, 'splitter': 'random'}
```

As we can see, using the code above we can identify that the best algorithm to use is the Tree Algorithm, which has the highest classification accuracy.

Confusion matrix



Confusion matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between different classes. The biggest problem is false positives, that is, an unsuccessful landing marked as a successful landing by the classifier.

Conclusions

- Decision Tree Model is the best algorithm for this data set.
- Launches with a low payload mass show better results than launches with a higher payload mass.
- Most launch sites are close to the equator and all sites are very close to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest launch success rate of all sites.
- ES-L1, GEO, HEO and SSO orbits have 100% success rate.

Thank you very much!