IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sergio Santimano
5th September 2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Analytics
  - Predictive Analysis

Introduction

Space Exploration Technologies Corp., commonly referred to as SpaceX, is an American space technology company headquartered at the Starbase development site in Starbase, Texas. Since its founding in 2002, the company has made numerous advances in rocket propulsion, reusable launch vehicles, human spaceflight and satellite constellation technology.

While its competitors spend around an upwards of $165 million, the Falcon 9 costs as low as only $62 million. Most of its saving is duty of their ability to reuse the first stage of launch by landing and using the booster for subsequent launches.

As a data scientist of a startup rivaling SpaceX, the goal of the project is to create a machine learning pipeline to predict the landing outcome of the first stage in the future. This is crucial in bidding the right price to bid against SpaceX.

The questions to be answered:

- How do the variables such as payload mass, launch site, orbit and more affect its success?
- Does rate of successful landing increase over the years?
- What is best algorithm to be used for its binary classification in this case?

Section 1

Methodology

Methodology

Executive Summary

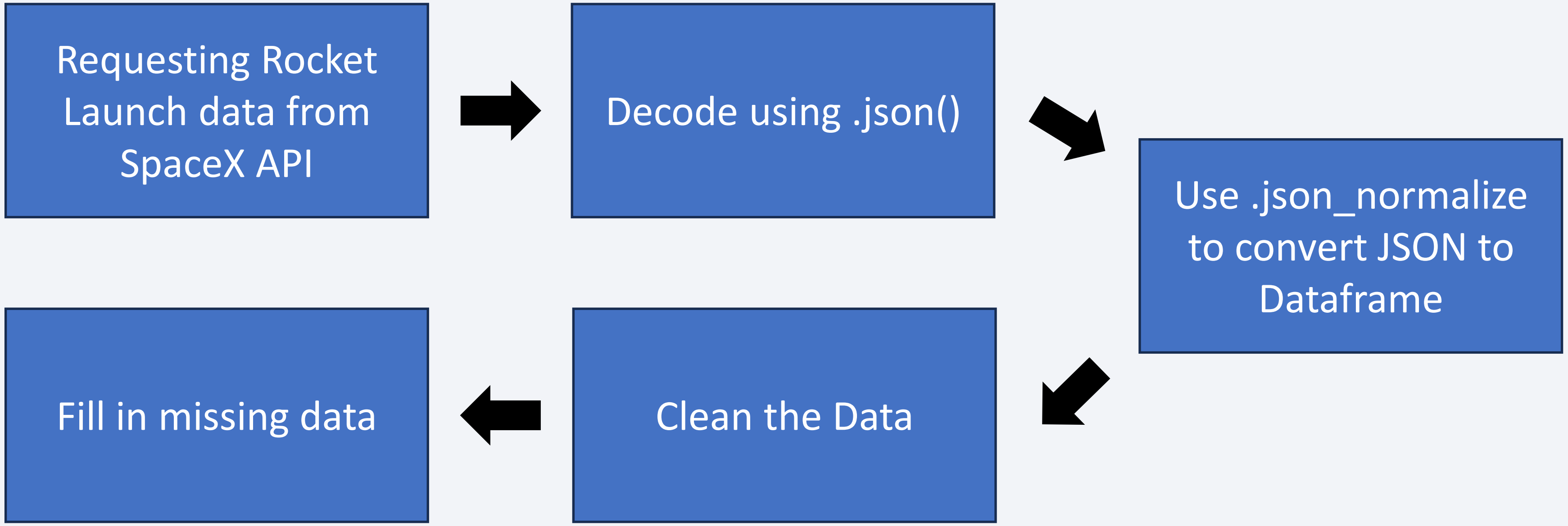
- Data collection methodology:
  - Using SpaceX Rest API
  - Using Web Scrapping from Wikipedia
- Perform data wrangling
  - Filtering the data
  - Dealing with missing values
  - Using One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization 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

- Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia.
- For REST API, its started by using the get request feature. Then, we decoded the response content as JSON and turn it into a pandas Dataframe using json_normalize(). We then cleaned the data and checked for missing values.
- For Web Scraping, we used Beautiful Soup to extract launch records as HTML table, and parse the table to convert it into a pandas Dataframe for further analysis.

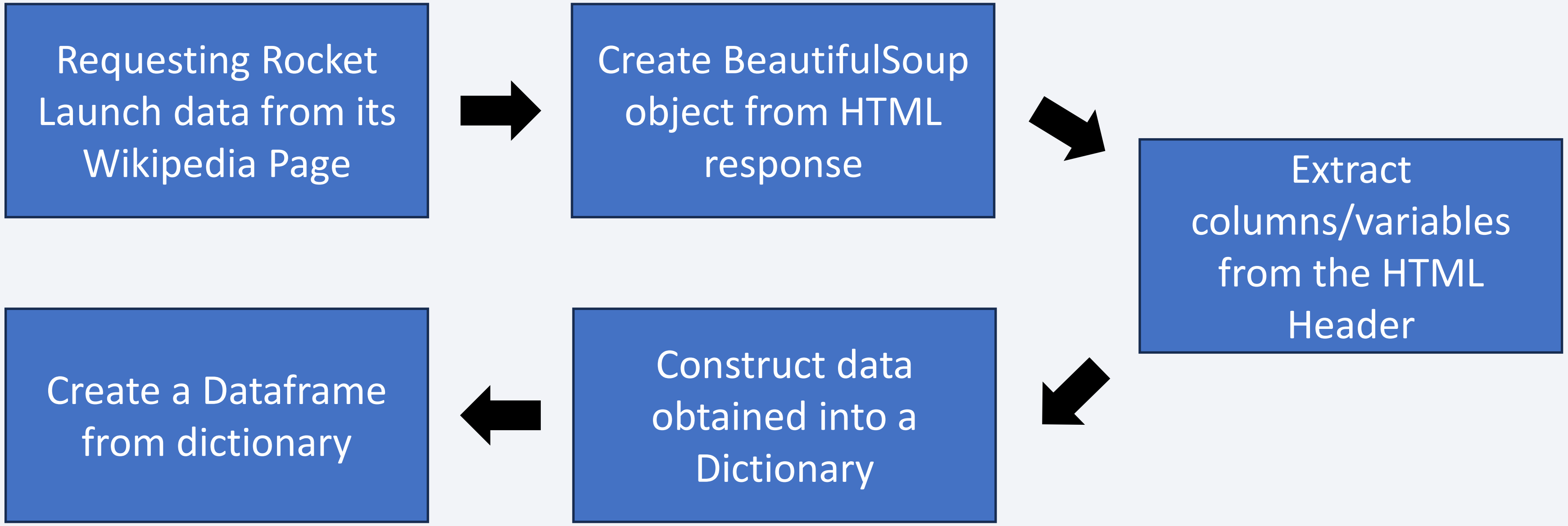
Data Collection – SpaceX API

GitHub Link



Data Collection - Scraping

GitHub Link



Data Wrangling

GitHub Link

- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis.
- We will first calculate the number of launches of each launch site. Then calculate the number and occurrence of mission outcomes per orbit type.
- Then create a landing outcome label from the Outcoe column. This will make it easier for further analysis, visualization and ML.
- We mainly convert these outcomes into labels with “1” meaning the booster successfully landed while “0” means it was not successful.

Perform exploratory Data Analysis and determine training labels

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from ‘Outcome’ column

EDA with Data Visualization

GitHub Link

- Scatter plots show the relationship between variables. If a relationship exists, they could be used in machine learning model.
- Bar charts show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value.
- Line Charts show trends in data over time.
- **Plotted Charts:** Flight Number vs. Payload Mass, Flight Numbber vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs Orbit Type and Success Rate Yearly Trend.

EDA with SQL

GitHub Link

- Performed SQL Queries
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string ‘CCA’
  - 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 when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship 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 failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
  - Ranking 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

Build an Interactive Map with Folium

GitHub Link

- Markers of all Launch Sites:
  - Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
  - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.
- Colored Markers of the launch outcomes for each Launch Site:
  - Added colored Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.
- Distances between a Launch Site to its proximities
  - Added colored Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

Build a Dashboard with Plotly Dash

GitHub Link

- Launch Sites Dropdown List:
  - Added a dropdown list to enable Launch Site selection
- Explain why you added those plots and interactions
  - Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected
- Slider of Payload Mass Range:
  - Added a slider to select Payload range
- Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:
  - Added a scatter chart to show the correlation between Payload and Launch Success.

Predictive Analysis (Classification)

GitHub Link

Building the Model
- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model
- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- 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.

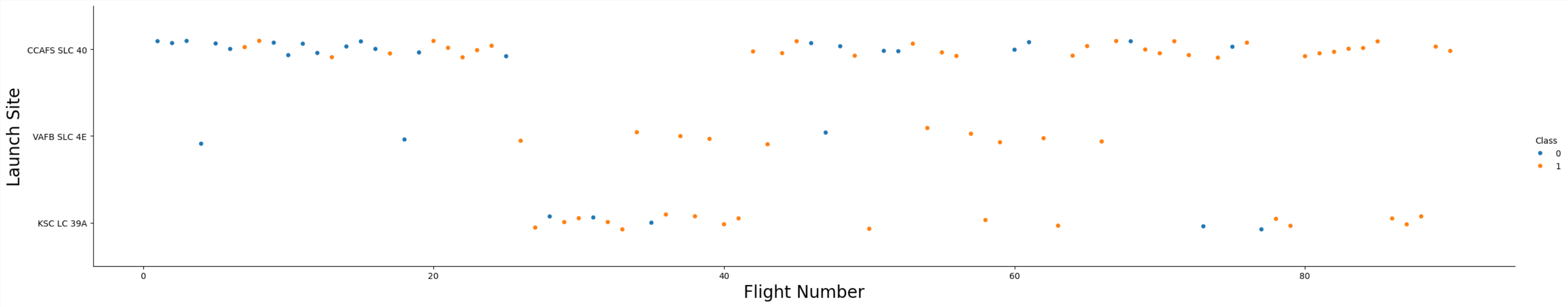
Results

- The results will be categorized to 3 main results which is:
    - Exploratory data analysis results
    - Interactive analytics demo in screenshots
    - Predictive analysis results

Section 2

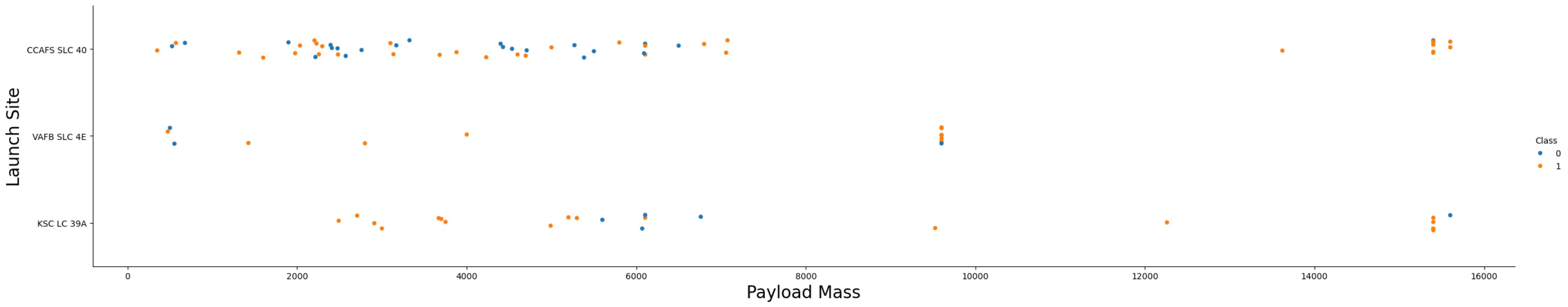
Insights drawn from EDA

Flight Number vs. Launch Site



- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- It can be assumed that each new launch has a higher rate of success.

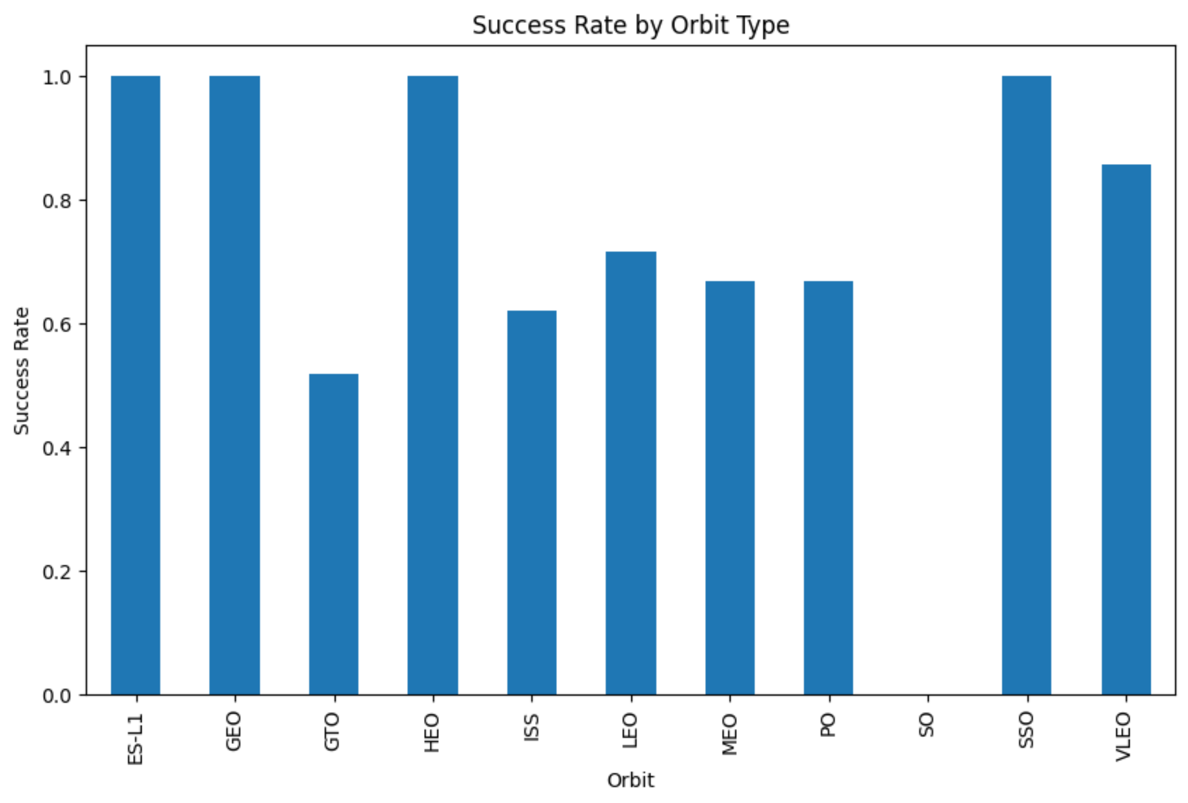
Payload vs. Launch Site



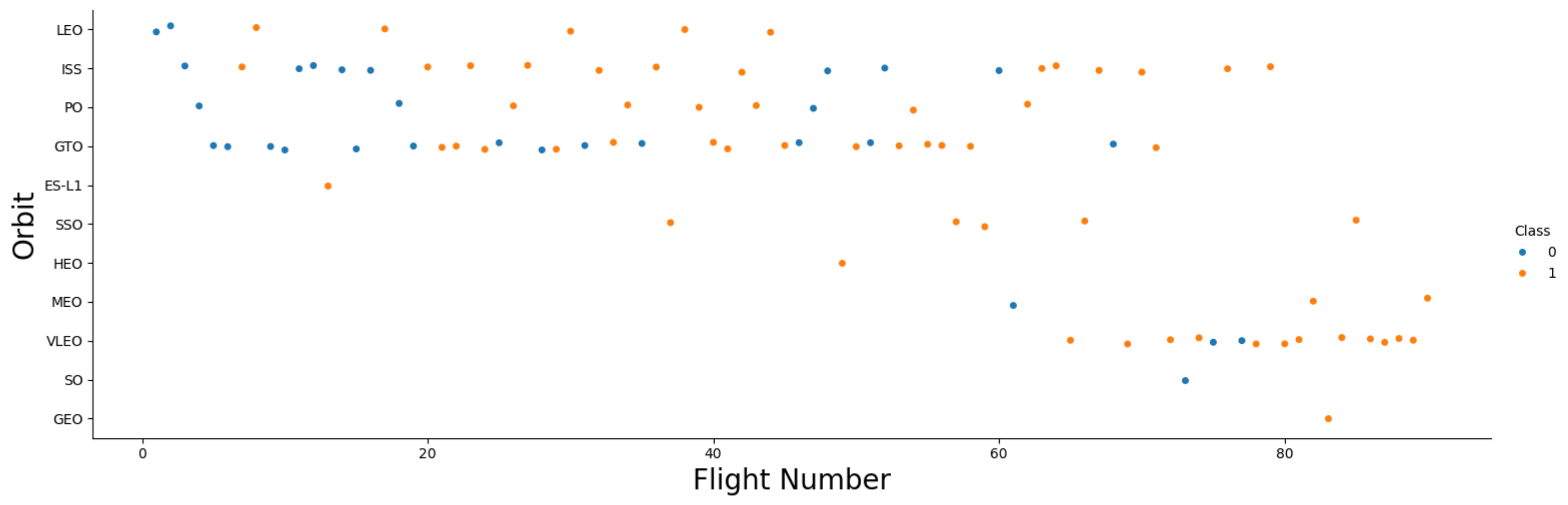
- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

Success Rate vs. Orbit Type

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

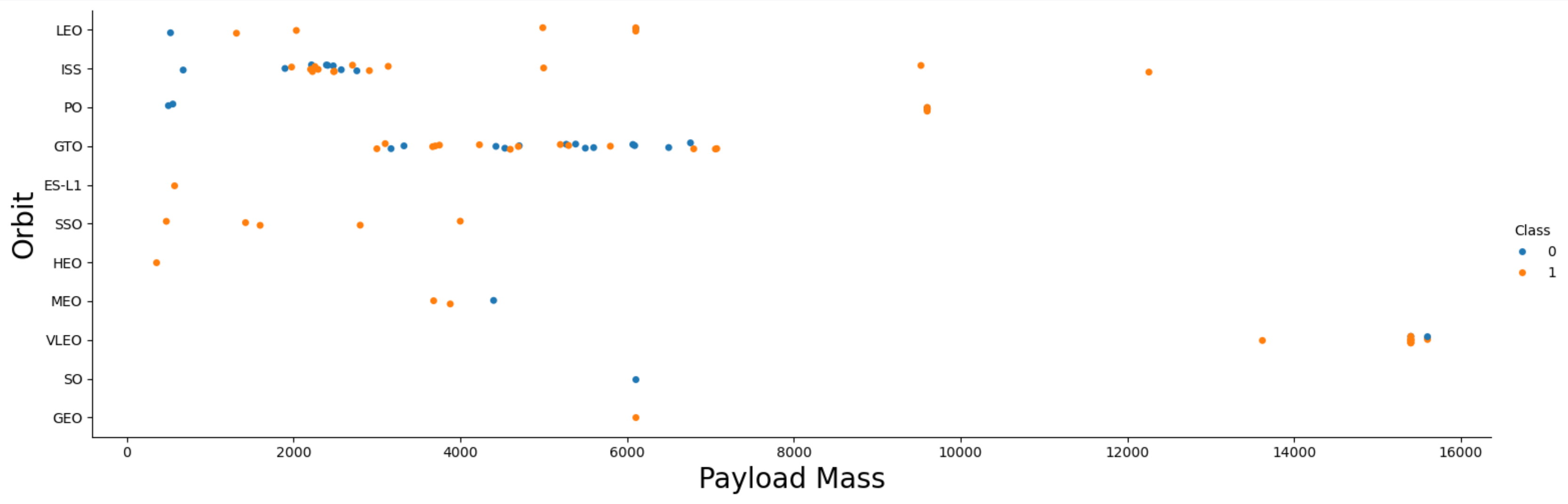


Flight Number vs. Orbit Type



- 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.

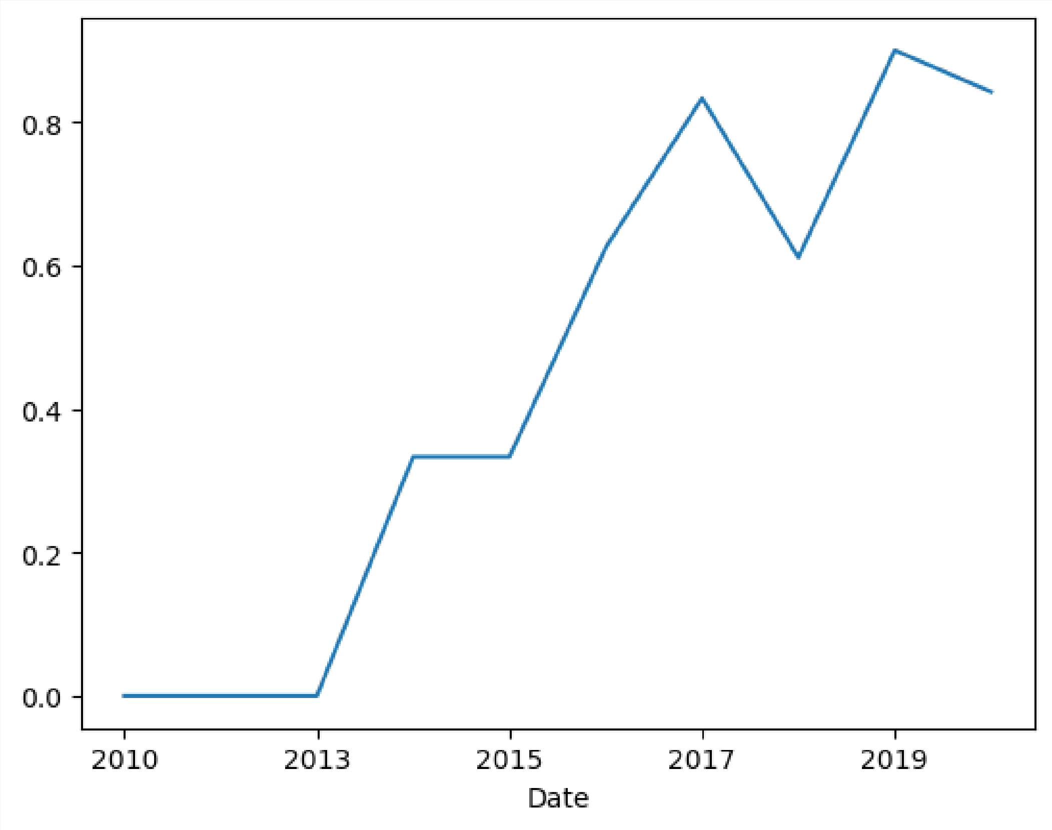
Payload vs. Orbit Type



- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020.



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- We used the query above to display 5 records where launch sites 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

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer" == 'NASA (CRS)'
```

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

sum("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.66

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" like 'F9 v1.1%'
```

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

avg("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- We use the min() function to find the result We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min("Date") from SPACEXTABLE where "Landing_Outcome" == 'Success (ground pad)
```

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

min("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%%sql
select "Booster_Version"
from SPACEXTABLE
where "Landing_Outcome" = 'Success (drone ship)'
  and "PAYLOAD_MASS__KG_" between 4000 and 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like ‘%’ to filter for WHERE Mission_Outcome was a success or a failure.

```
%%sql
select "Mission_Outcome", count("Mission_Outcome")
from SPACEXTABLE
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

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

```
%%sql
select "Booster_Version", "PAYLOAD_MASS__KG_"
from SPACEXTABLE
where "PAYLOAD_MASS__KG_" = (
    select max("PAYLOAD_MASS__KG_")
    from SPACEXTABLE)
```

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

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

2015 Launch Records

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%%sql
select
  substr(Date, 6, 2) as Month,
  "Booster_Version",
  "Launch_Site",
  "Landing_Outcome"
from SPACEXTABLE
where substr(Date, 0, 5) = '2015'
  and "Landing_Outcome" like 'Failure%';
```

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

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

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

```
%%sql
select "Landing_Outcome", count(*) as Outcome_Count
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by "Landing_Outcome"
order by Outcome_Count desc;
```

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

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

Section 3

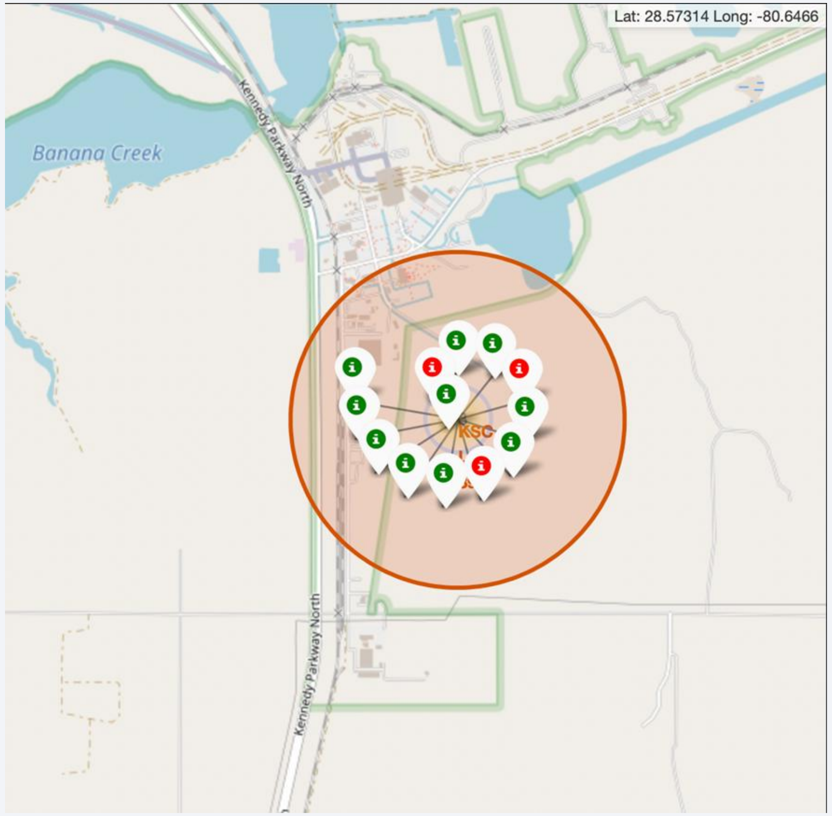
Launch Sites Proximities Analysis

Location of the Launch Sites

- We can see that all the SpaceX launch sites are located inside the United States

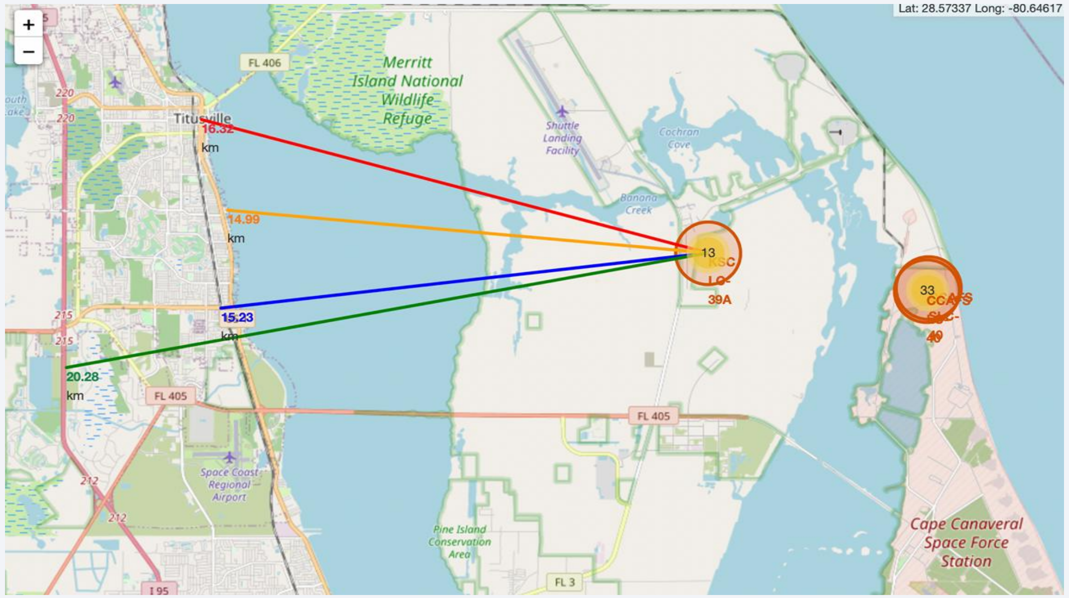
Markers showing launch sites with color labels

- From the colour-labeled markers we should be able to easily identify which launch sites have relatively high success rates.
  - Green Marker = Successful Launch
  - Red Marker = Failed Launch
- Launch Site KSC LC-39A has a very high Success Rate.



Launch Sites Distance to Landmarks

- From the visual analysis of the launch site KSC LC-39A we can clearly see that it is: - relative close to railway (15.23 km) - relative close to highway (20.28 km) - relative close to coastline (14.99 km)
- Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).
- Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.

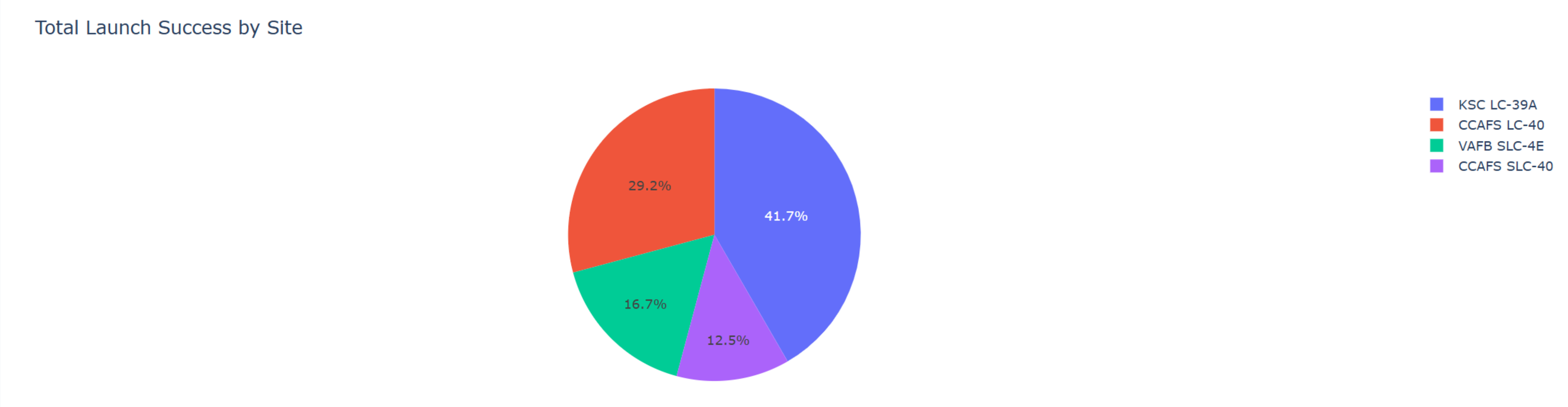


Section 4

Build a Dashboard with Plotly Dash

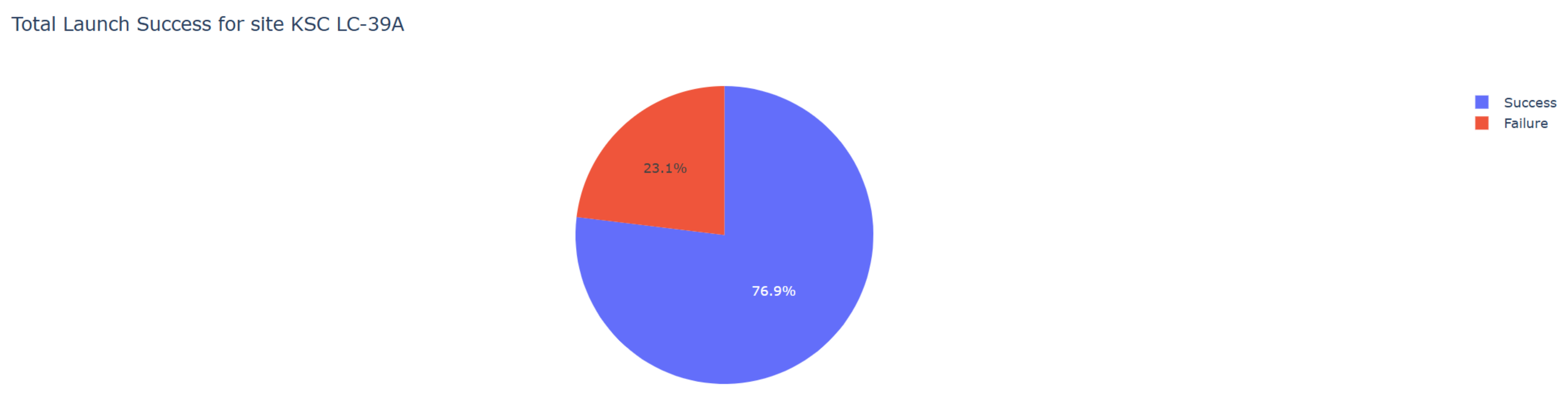
Launch success count for all sites

- The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.



Launch site with highest launch success ratio

- KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.



Payload Mass vs. Launch Outcome for all sites

- The charts show that payloads between 2000 and 5500 kg have the highest success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

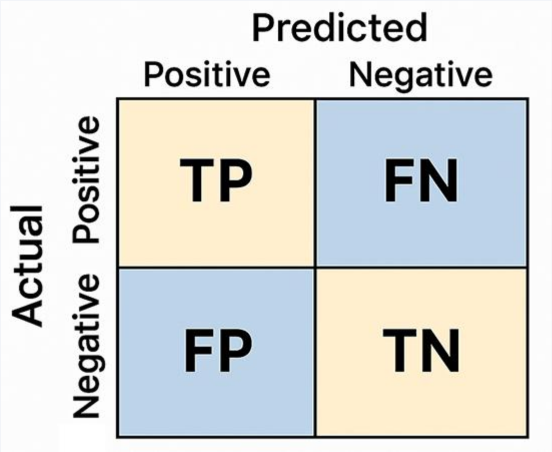
- As we can see, by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy.

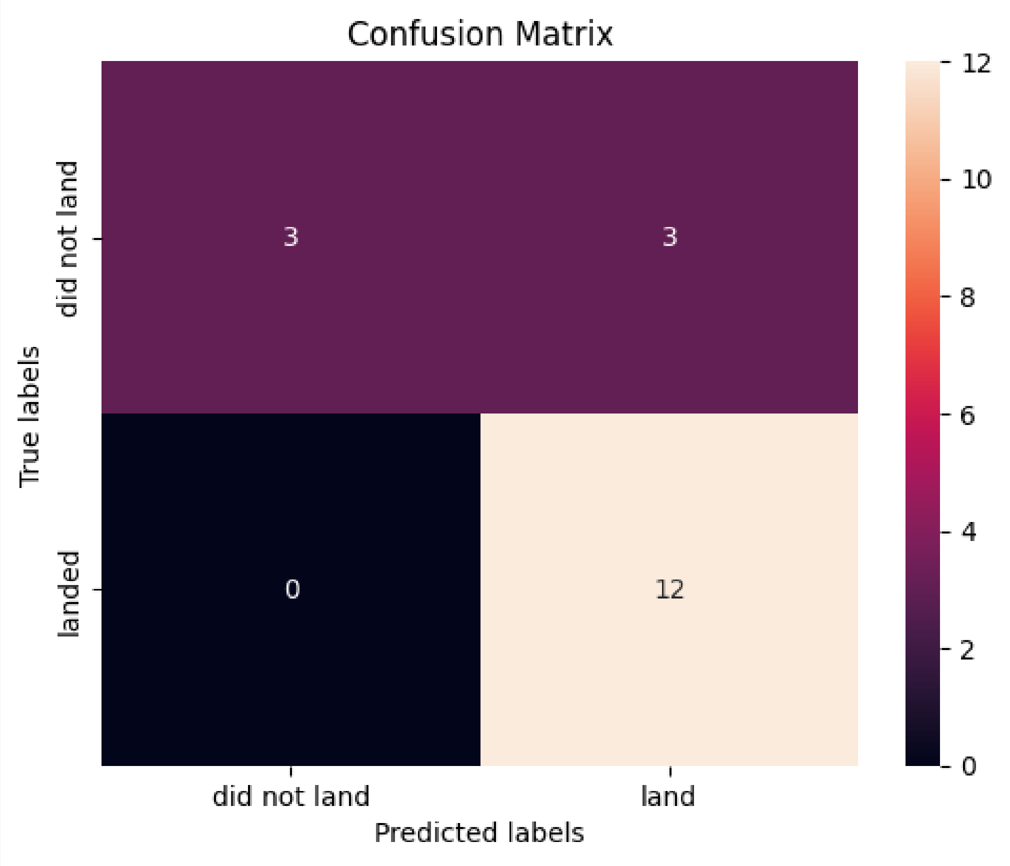
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params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

Confusion Matrix

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.





Conclusions

We can conclude that:

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!