IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

AKOMATSRI A. Dieu-Donné
July 18th 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX has revolutionized the space industry by offering rocket launches — particularly with the Falcon 9 — at prices as low as $62 million, compared to over $165 million from other providers.This significant cost reduction is largely due to SpaceX's groundbreaking innovation: reusing the first stage of the rocket by landing it safely after launch, allowing it to be used for future missions. Repeating this process drives costs down even further.

As a data scientist at a startup competing with SpaceX, the goal of this project is to develop a machine learning pipeline to predict the landing outcome of the rocket's first stage in future missions.This project is critical for accurately estimating launch costs and determining the right price point to bid competitively against SpaceX.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variables and how it is affecting the outcome.
- The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- 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

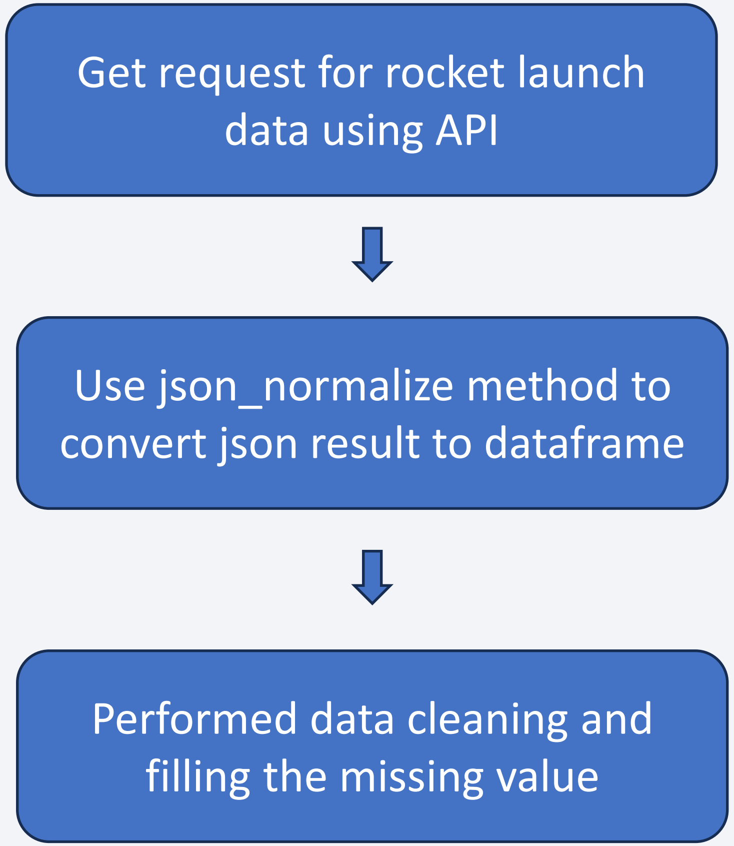
Data collection is the process of gathering and measuring information on targeted variables within a defined system, enabling us to answer relevant questions and evaluate outcomes.In this project, the dataset was collected using two main methods: REST API calls and web scraping from Wikipedia.For the REST API, we initiated the process using a GET request.

The response content was then decoded as JSON and converted into a Pandas DataFrame using the json_normalize() function. After loading the data, we cleaned it by checking for missing values and filling them appropriately.For web scraping, we used BeautifulSoup to extract launch records presented in HTML table format.

The table was parsed and transformed into a Pandas DataFrame to prepare it for further analysis.

Data Collection – SpaceX API

- The initial data was obtained from the `/v4/launches/past` API endpoint.
- Additional data was backfilled from the `rocket`, `launchpad`, `payloads`, and `cores` API endpoints, for records with extant corresponding IDs.



From:

https://github.com/civilwardido/capstone/blob/d451a2537f01aa20d9915c7ae4a87eb451fd7704/notebook_Data_Collection_yJPxhv2oU.ipynb

Data Collection - Scraping

- Web scraping workflow:
  - Send HTTP Get to Falcon 9 Launch page at Wikipedia, to retrieve the HTML source
  - Create a BeautifulSoup object to extract the tables containing launch data
  - Populate a Pandas DataFrame using the extracted columns

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

From:

https://github.com/civilwardido/capstone/blob/ac0187b9602dc19349f9d9ba07bb55061c5c75d0/01-data-collection-webscraping.ipynb

Data Wrangling

- In the dataset, there are several instances where the booster did not successfully land.
- "True Ocean," "True RTLS," and "True ASDS" indicate a successful mission.
- "False Ocean," "False RTLS," and "False ASDS" indicate a failed mission.
  We need to convert the string variables into categorical variables, where "1" represents a successful mission and "0" represents a failed mission.

1. Calculate launches number for each site

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

2. Calculate the number and occurence of each orbit

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate number and occurrence of mission outcome per orbit type

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from Outcome column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5. Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

From:

https://github.com/civilwardido/capstone/blob/6bf922e0b7b47e6025f71fafadc02f3049909403/01-data-collection-webscraping.ipynb

EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit Type
  - Orbit vs. Payload Mass

Scatter plots show relationship between variables. This relationship is called the correlation.

- Bar Graphs
  - Success rate vs. Orbit

Bar graphs show the relationship between numeric and categoric variables.

- Line Graphs
  - Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data

From:

https://github.com/civilwardido/capstone/blob/6bf922e0b7b47e6025f71fafadc02f3049909403/edadataviz.ipynb

EDA with SQL

Show each unique launch site

- Show 5 records where launch site names begin with 'CCA'
- Display the total payload mass carried by boosters launched by 'NASA (CRS)'
- Display the average payload mass carried by the v1.1 Falcon 9 booster
- List the date of the first successful ground landing outcome
- List the booster versions with successful outcomes landing on the drone ship with payloads between kg and 6000kg.
- List the number of successful and failed mission outcomes
- List all of the booster versions that carried the max payload mass
- List the month name, outcome, booster version, and launch site for missions with failure outcomes landing on a drone ship in 2015.
- Show the distribution of outcomes between June 4th, 2010 and March 20th, 2017

From:

https://github.com/civilwardido/capstone/blob/5819471be34dba67e286c0452a835468c75862c2/notebook_Exploratory_Data_Analysis_with_SQL__eqznon1EA.ipynb

Build an Interactive Map with Folium

To visualize the launch data on an interactive map, we utilized the latitude and longitude coordinates of each launch site and placed circle markers at their respective locations, each labeled with the name of the launch site.

Next, we classified the launch outcomes as either failure (0) or success (1), and represented them using red and green markers respectively within a MarkerCluster() for better clarity and grouping.

To further analyze the geographical context of the launch sites, we applied the Haversine formula to calculate the distance between each launch site and nearby landmarks. This helped us address key questions such as:

- How close are the launch sites to railways, highways, and coastlines?
- How close are the launch sites to nearby cities?

From:

https://github.com/civilwardido/capstone/blob/4c421b9cf1707cfded8d03541093927e2b124f29/notebook_Interactive_Visual_Analytics_with_Folium_M8uUhCmHY.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard using Plotly Dash, enabling users to explore and interact with the launch data according to their needs.
- Pie charts were created to display the total number of launches by each launch site, providing a clear overview of site activity.
- Scatter plots were also included to illustrate the relationship between launch outcomes and payload mass (kg), categorized by different booster version categories.

From:

https://github.com/civilwardido/capstone/blob/6e9067d0783bc70aecabc6b4513548e50fa2e82b/spacex_dash_app.py

Predictive Analysis (Classification)

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.

From:

https://github.com/civilwardido/capstone/blob/3b2154a5d2846e31c46ed6bd32d01d76194cecbc/notebook_Predictive_Analysis_-_Machine_Learning_Lab_hdUi_lnX5.ipynb

Results

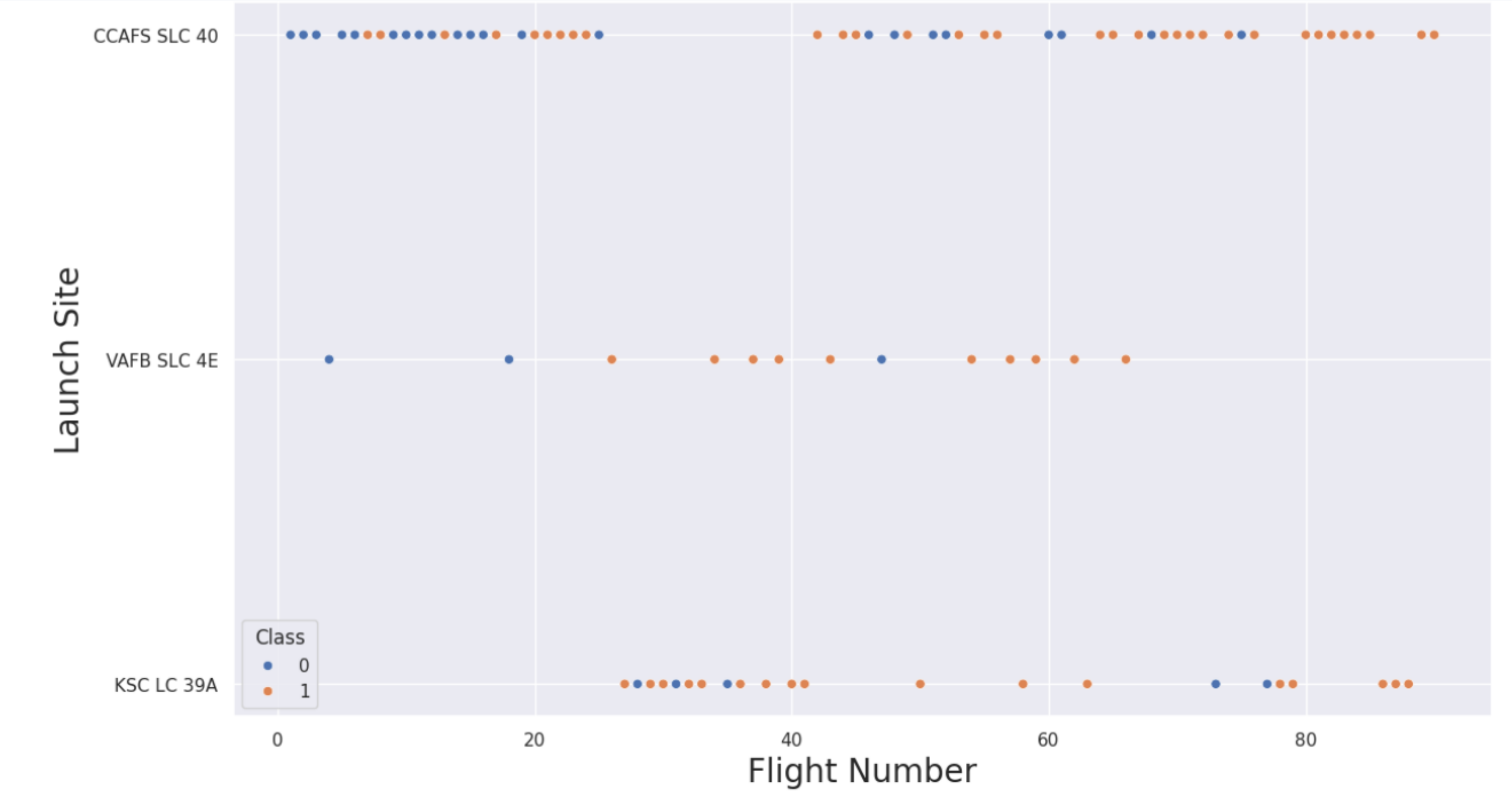
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

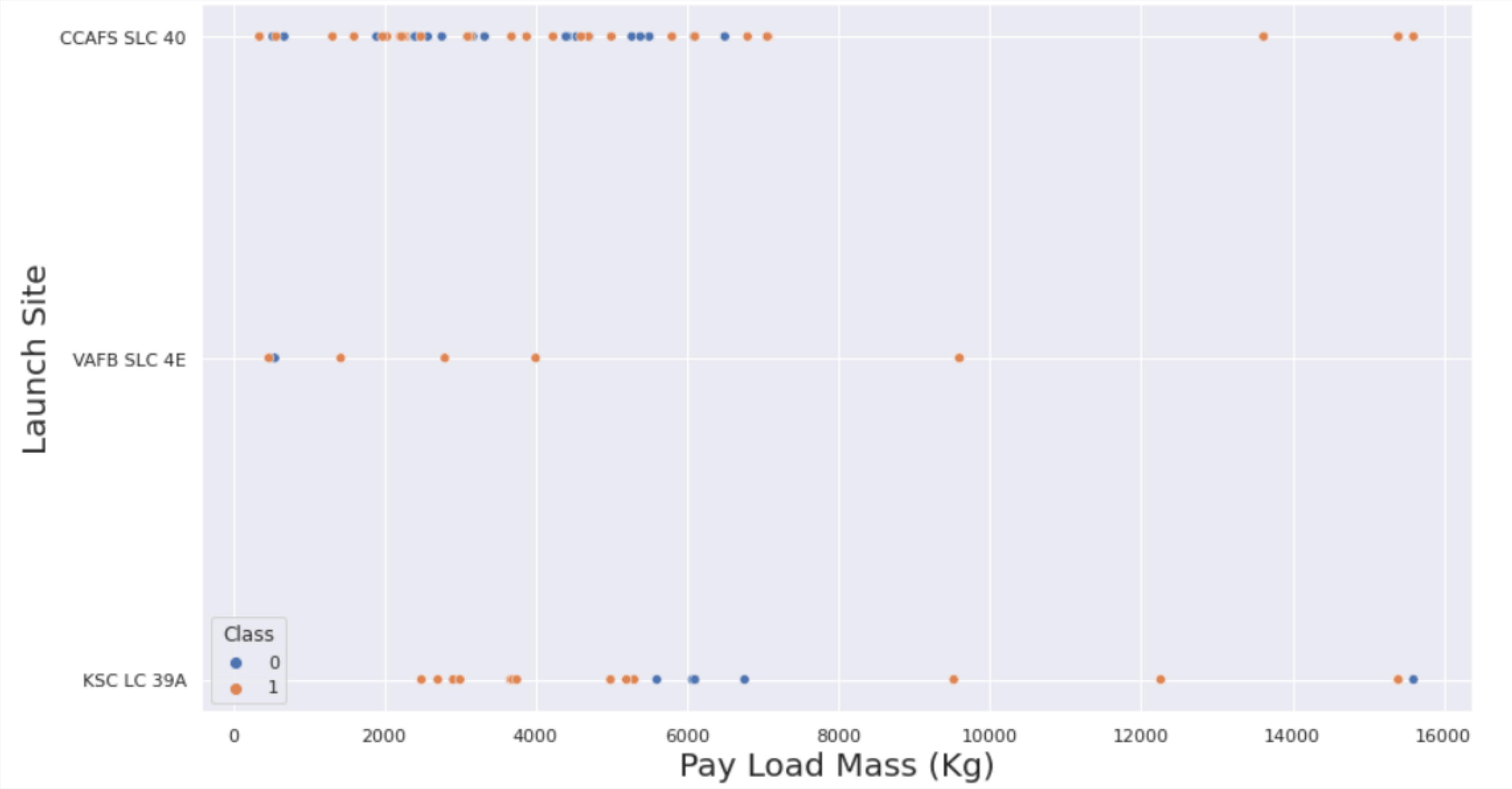
- This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be.
- However, site CCAFS SLC40 shows the least pattern of this.



Payload vs. Launch Site

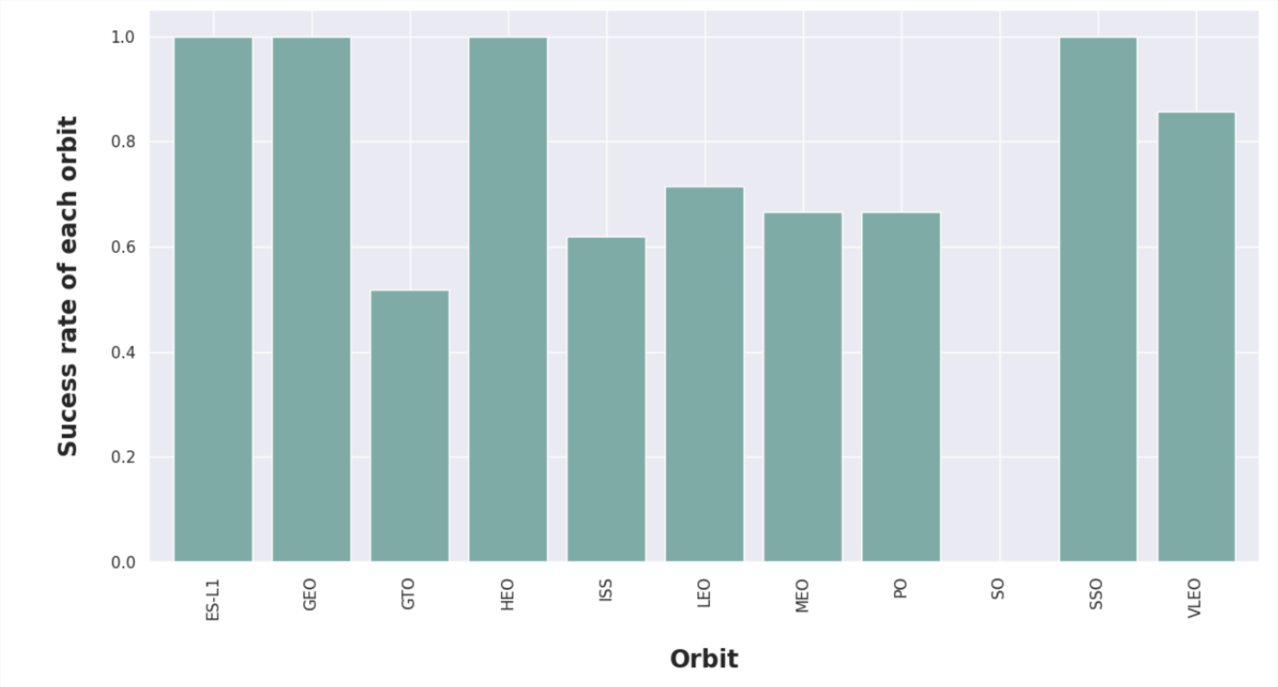
- This scatter plot shows once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.

- However, there is no clear pattern to say the launch site is dependent to the pay load mass for the success rate.



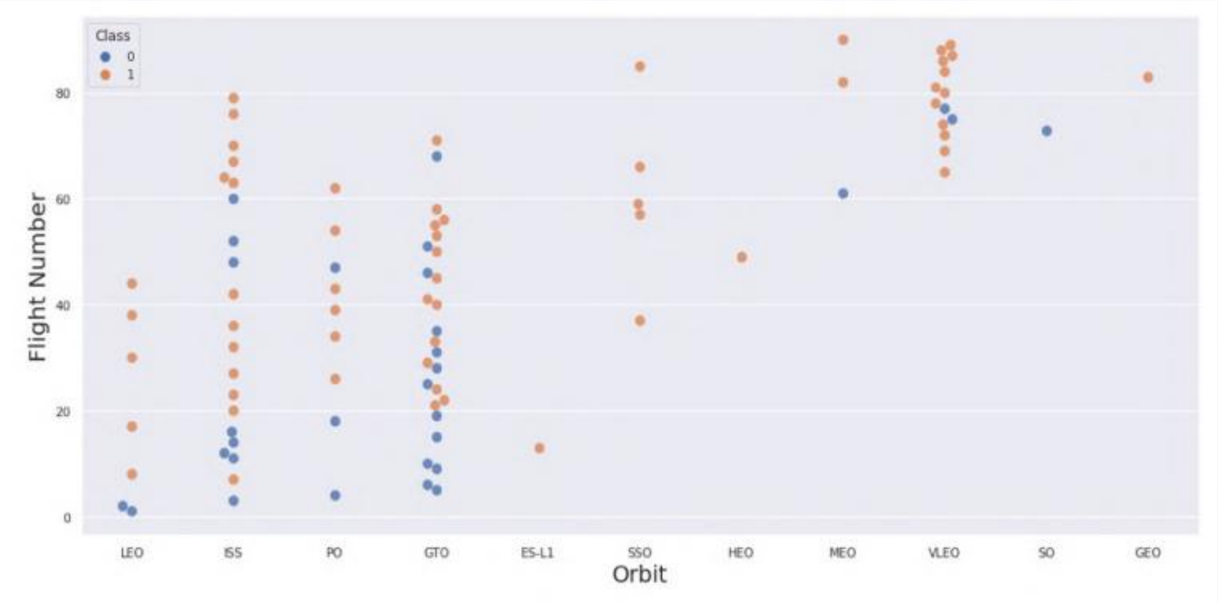
Success Rate vs. Orbit Type

- This figure depicted the possibility of the orbits to influences the landing outcomes as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success.
- However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



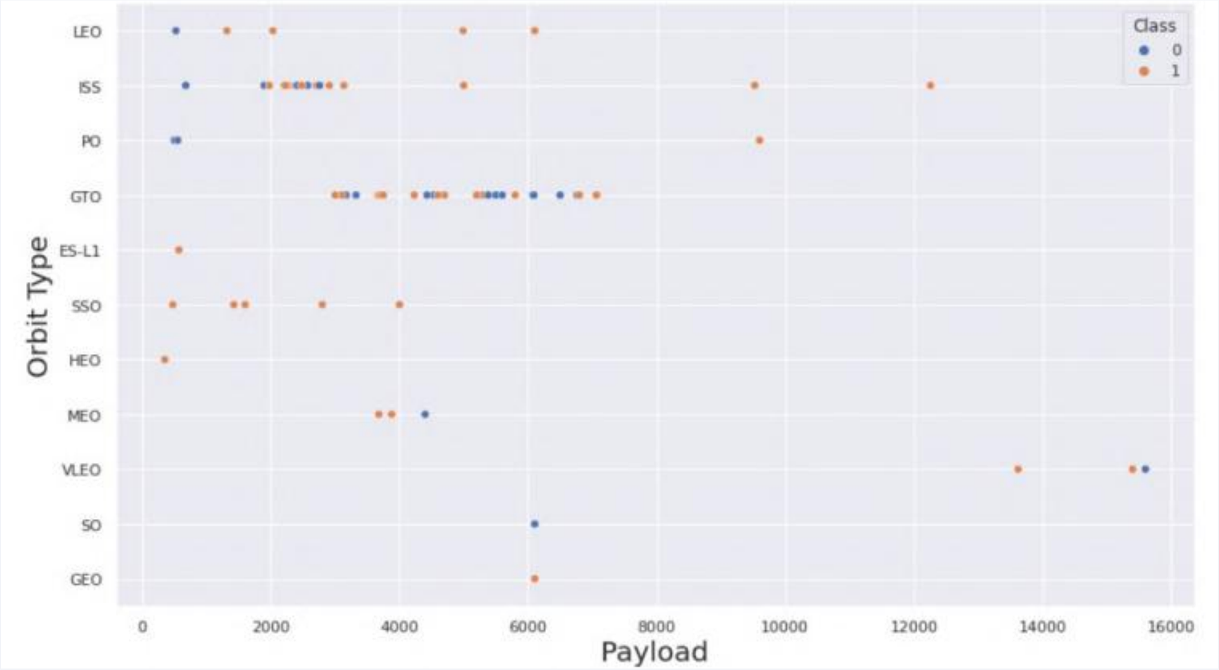
Flight Number vs. Orbit Type

- This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.
- Orbit that only has 1 occurrence should also be excluded from above statement as it’s needed more dataset.



Payload vs. Orbit Type

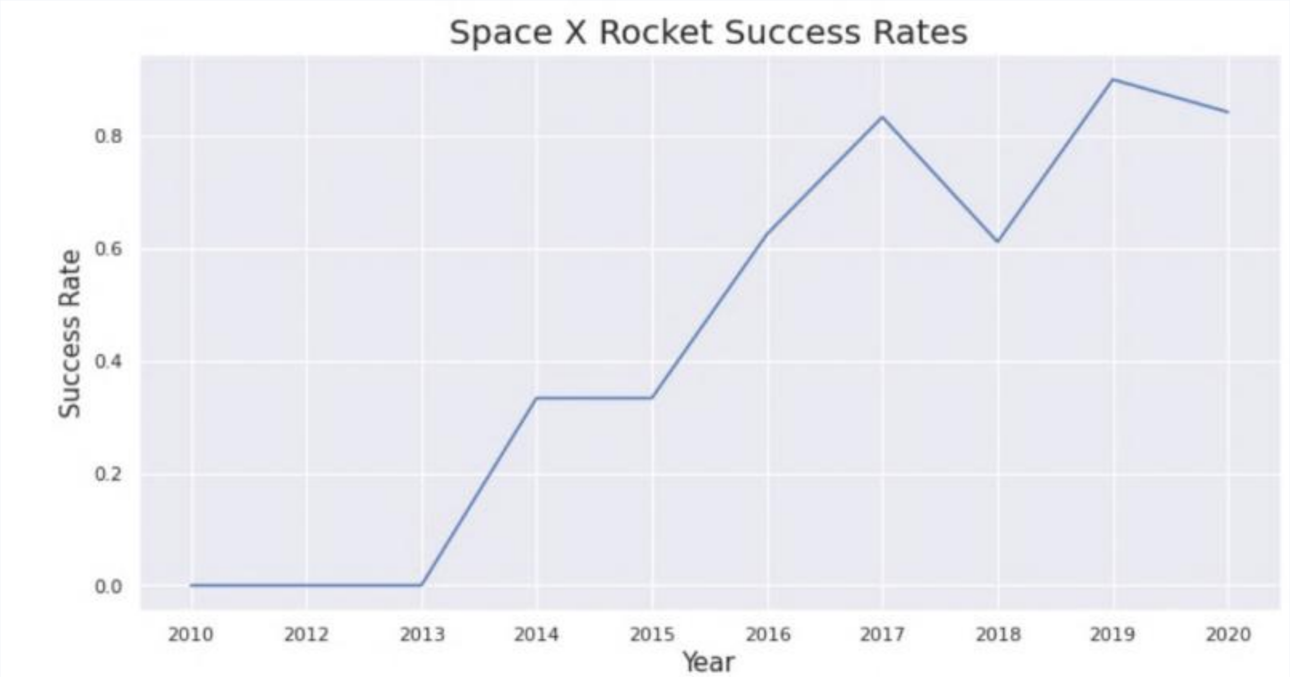
- Heavier payload has positive impact on LEO, ISS and P0 orbit. However, it has negative impact on MEO and VLEO orbit.
- GTO orbit seem to depict no relation between the attributes.
- Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.



Launch Success Yearly Trend

This figures clearly depicted and increasing trend from the year 2013 until 2020.

If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

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

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

Launch Site Names Begin with 'CCA'

We used the query above 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

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

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

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

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) 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

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 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

Total Number of Successful and Failure Mission Outcomes

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

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.clogj3sd0tgtu0lqde00.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.clogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Failure Mission
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 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

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 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

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 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.clogj3sd0tgtu0lqde00.databases.appdomain.c
loud:32731/bludb
Done.
```

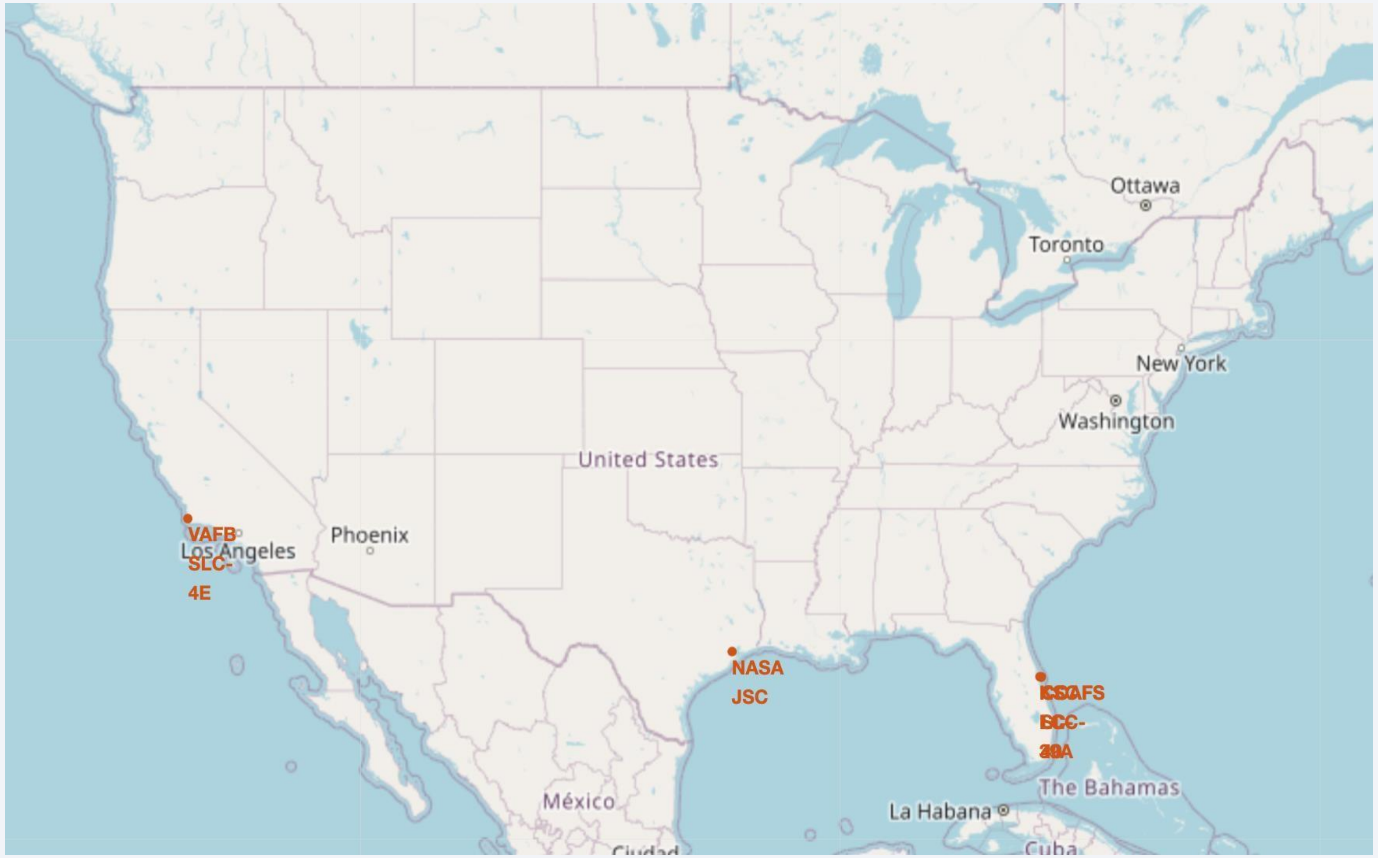
Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

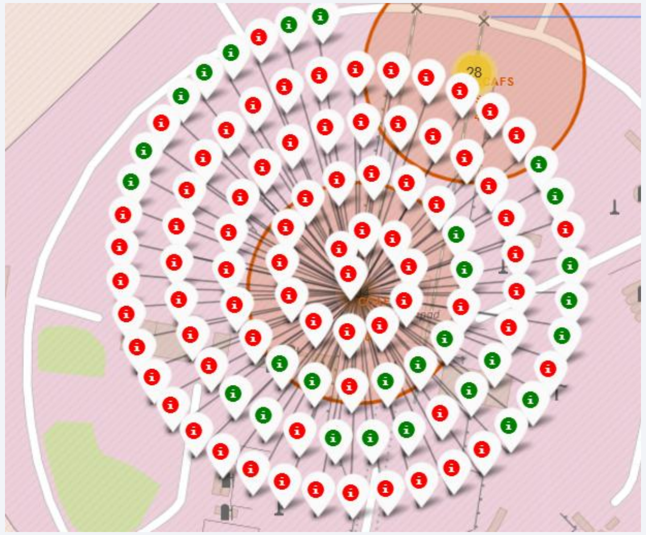
Location of all the Launch Sites

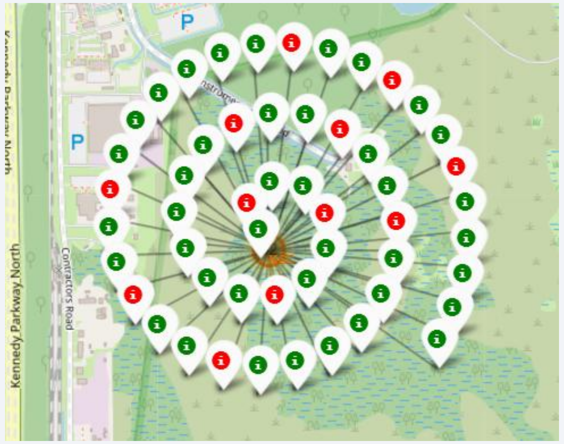
We can see that all the SpaceX launch sites are located inside the United States



Markers showing launch sites with color labels

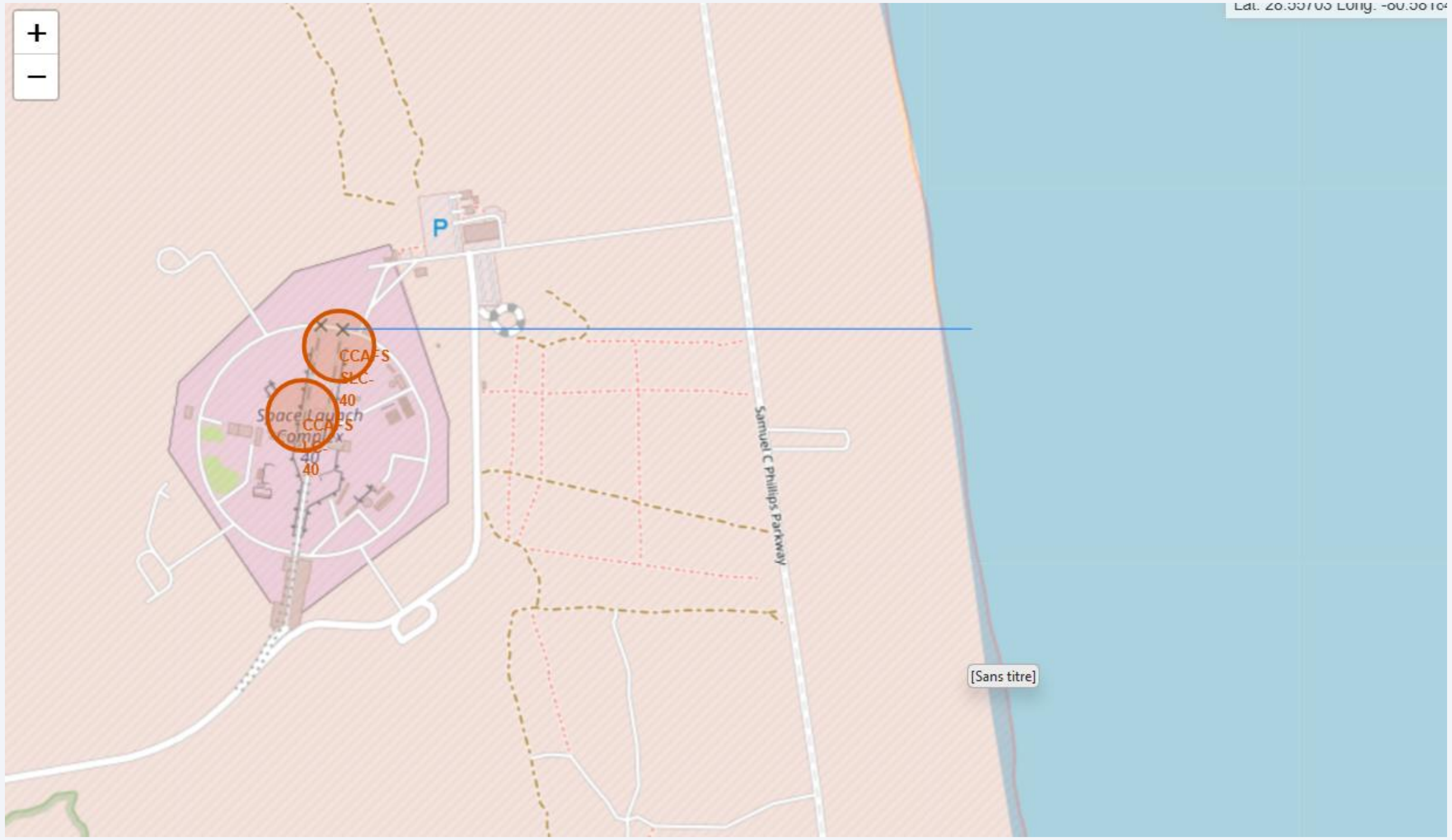
We can see the markers showing launch site with color label





Launch Sites Distance to Landmarks

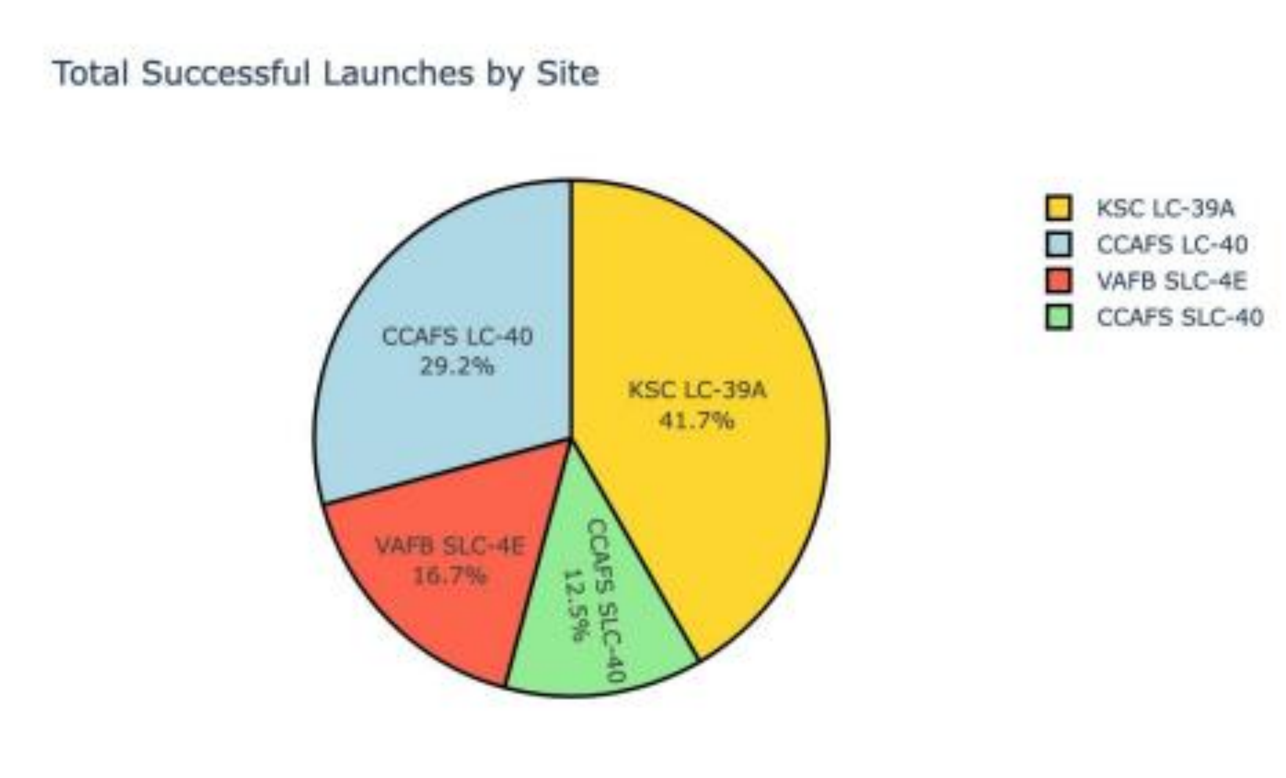
We can see PolyLine between a launch site to the selected coastline point



Section 4

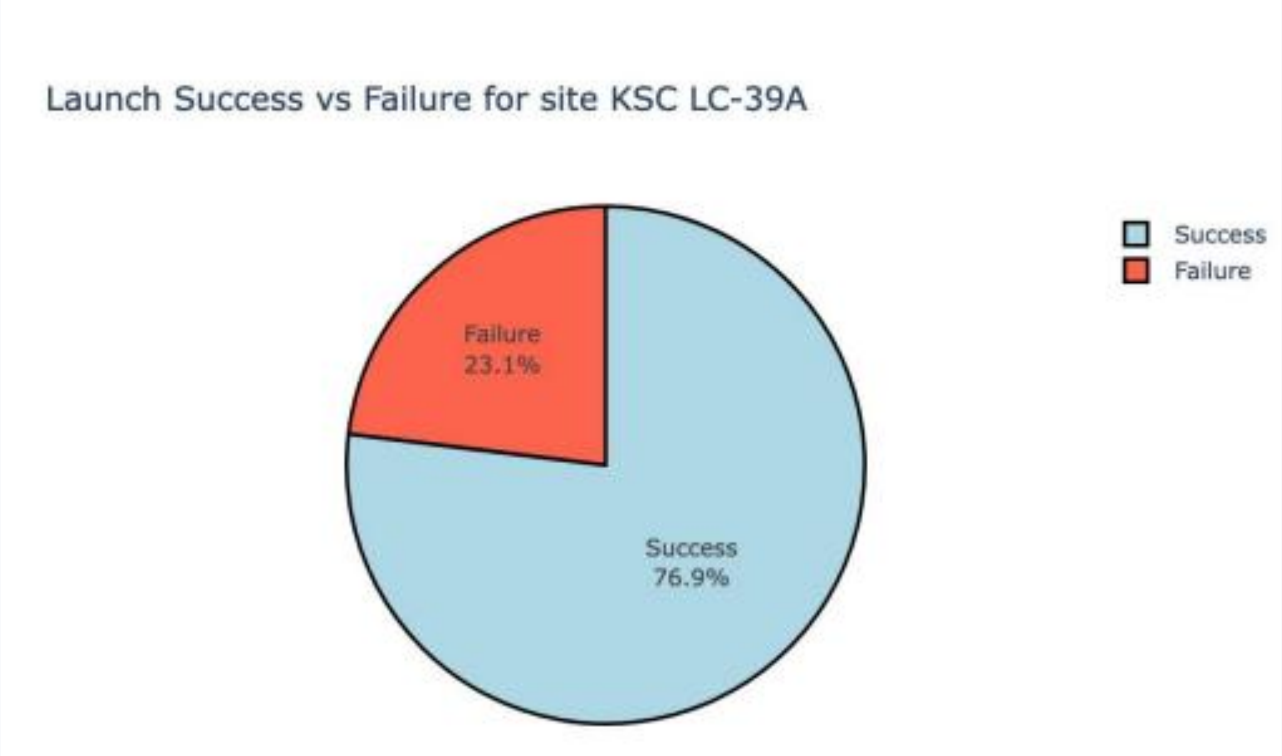
Build a Dashboard with Plotly Dash

The success percentage by each sites.



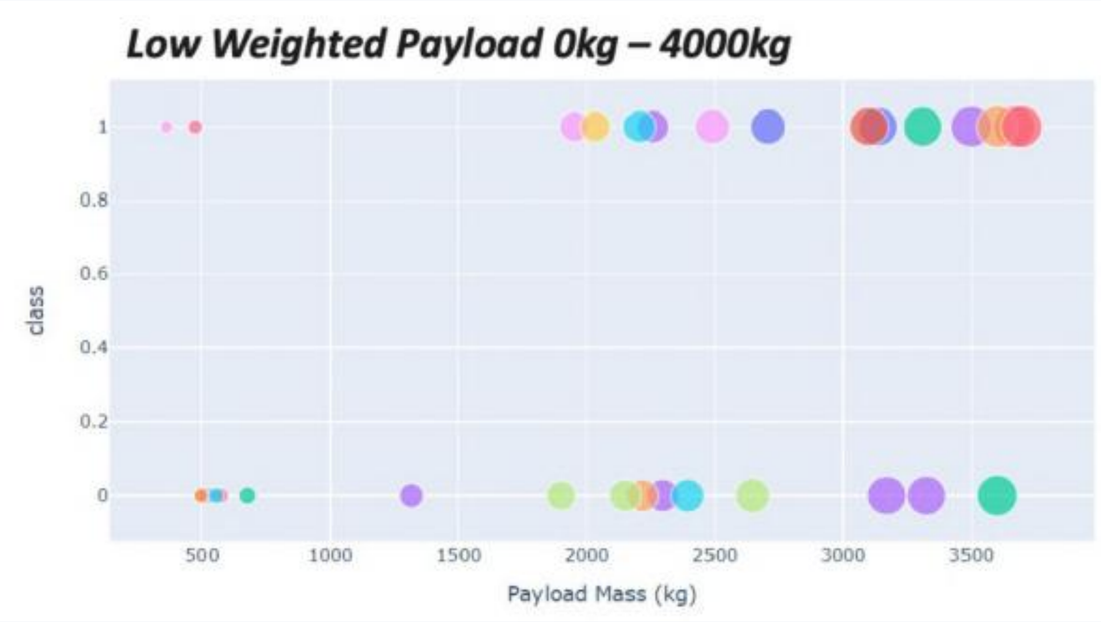
- KSC LC-39A experienced the highest proportion of successful landings, followed by CCAFS LC-40.
- VAFB SLC-4E and CCAFS SLC-40 the lowest,

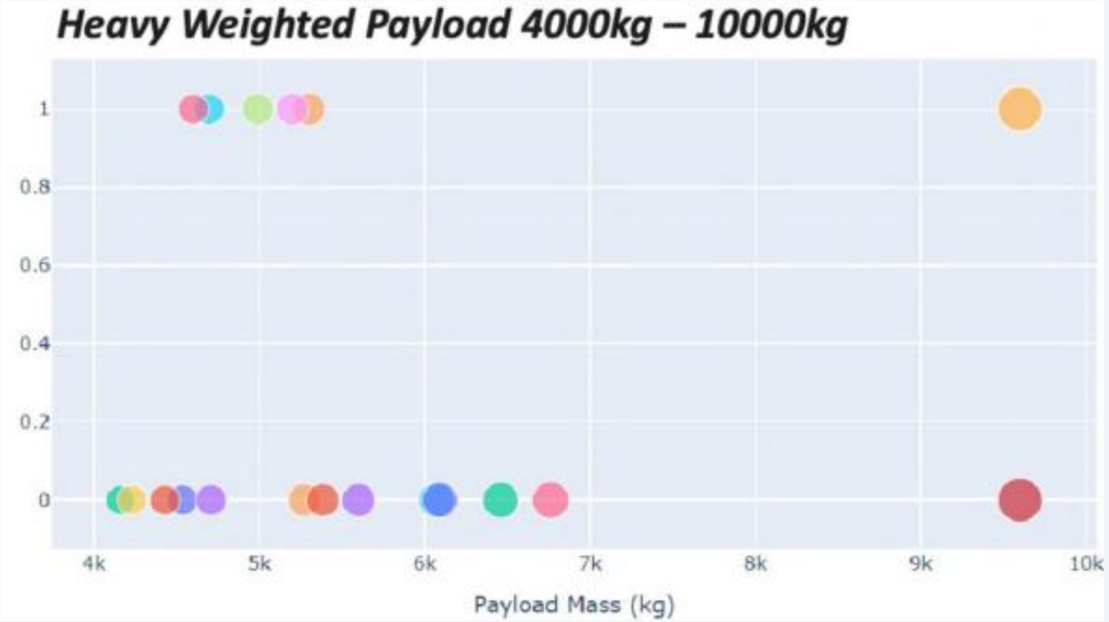
The highest launch-success ratio: KSC LC-39A



- KSC LC-39A had the highest ratio of successful landings

Payload vs Launch Outcome Scatter Plot





With a payload mass between 3,000kg and 5,000kg, v1.1 boosters performed the worst.

In the same payload range, B4 and B5 boosters had the best success rate, followed by FT.

Section 5

Predictive Analysis (Classification)

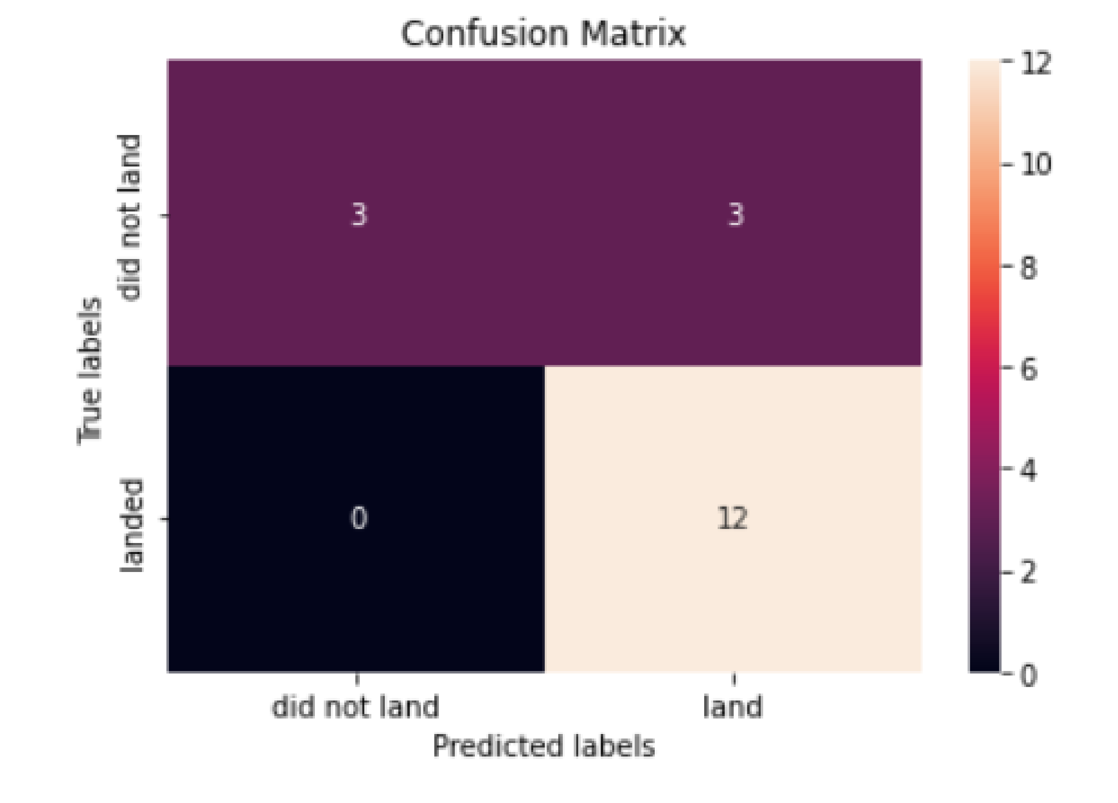
Classification Accuracy

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



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

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!