



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shreyans
15-08-2023



Outline

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

Executive Summary

- Summary of methodologies
 - Data Collection through API and Web Scraping
 - Data Wrangling
 - Exploratory Data Analysis through SQL and Data Visualization
 - Interactive Visual Analytics from Folium
 - Machine Learning Prediction
- Summary of all results
 - Exploratory Data Analysis Result
 - Interactive Analytics
 - Predictive Analysis

Introduction

SpaceX is a revolutionary company who has disrupted the space industry by offering a rocket launch specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollars each. Most of this saving thanks to SpaceX's astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price even further down. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variable 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:
 - I used SpaceX REST API and Wikipedia Web Scraped Data.
- Perform data wrangling
 - I used One-Hot-Encoding for categorical Variables
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using 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. Then, we decoded the response content as Json and turn it into a pandas data frame using `json_normalize()`. We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas data frame for further analysis

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook_Data_Collection_yJPxhv2oU.ipynb

Get request for rocket launch data using API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- [https://github.com/mehtashreyans3602/IBM-Data-Science-Capstone/blob/main/notebook Data Collection with Web Scraping nl89VIRCE.ipynb](https://github.com/mehtashreyans3602/IBM-Data-Science-Capstone/blob/main/notebook%20Data%20Collection%20with%20Web%20Scraping%20nl89VIRCE.ipynb)

Request the Falcon9
Launch Wiki page from url

Create a BeautifulSoup
from the HTML response

Extract all column/variable
names from the HTML
header

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 on each site, then calculate the number and occurrence of mission outcome per orbit type.

We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV

https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook_Data_Wrangling_9HnvfsJ5G.ipynb

EDA with Data Visualization

- We first started by using scatter graph to find the relationship between the attributes such as between:
 - Payload and Flight Number.
 - Flight Number and Launch Site.
 - Payload and Launch Site.
 - Flight Number and Orbit Type.
 - Payload and Orbit Type.

Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes. Bar graphs Attribute to that. To see the Trend we use the Line Chart, here we use it to launch success trend.

[https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook Exploratory Data Analysis with Visualisation Lab jJkKVG6F1.ipynb](https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook%20Exploratory%20Data%20Analysis%20with%20Visualisation%20Lab%20jJkKVG6F1.ipynb)

EDA with SQL

- Using SQL, we had performed many queries to get better understanding of the dataset, Ex: - Displaying the names of the launch sites.
 - Displaying 5 records where launch sites begin with the string 'CCA'.
 - Displaying the total payload mass carried by booster launched by NASA (CRS).
 - Displaying the 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 sites names for in year 2015.
 - Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order

Build an Interactive Map with Folium

To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.

I then assigned the data frame `launch_outcomes(failure,success)` to classes 0 and 1 with Red and Green markers on the map in `MarkerCluster()`.

I then used the Haversine's formula to calculate the distance of the launch sites to various landmarks to find answers to the questions of:

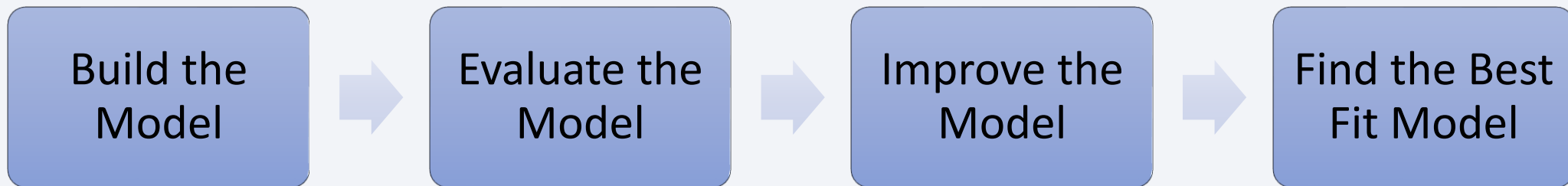
- How close the launch sites with railways, highways and coastlines?
- How close the launch sites with nearby cities?

[https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook Interactive Visual Analytics with Folium M8uUhCmHY.ipynb](https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/notebook%20Interactive%20Visual%20Analytics%20with%20Folium%20M8uUhCmHY.ipynb)

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- <https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/SpacexDash.py>

Predictive Analysis (Classification)



<https://github.com/mehtashreyans3602/IBM-Data-Science-Capestone/blob/main/SpacexDash.py>

Results

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

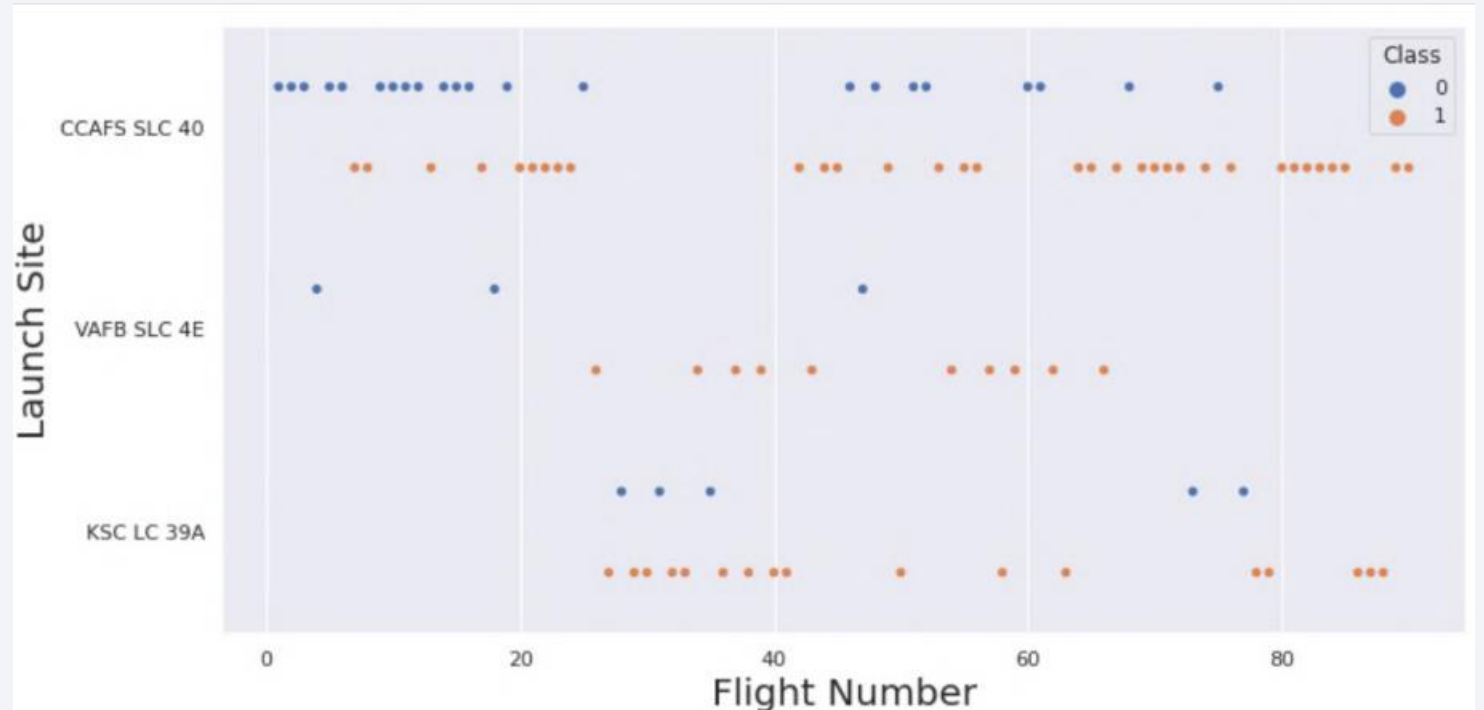
The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is dynamic and technological.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

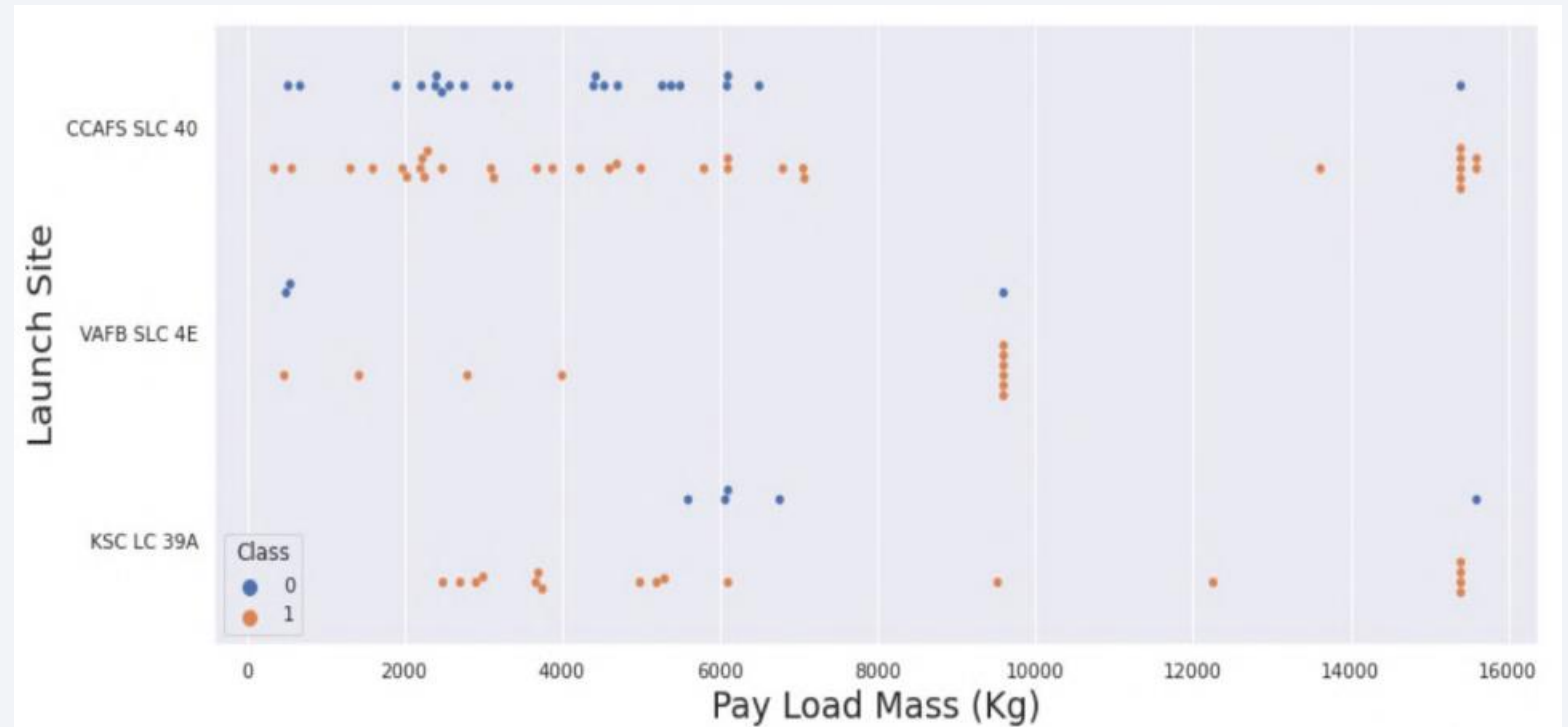
This scatter plot shows that the larger the flights amount of the launch site, the greater the success rate will be. However, site CCAFS SLC40 shows the least pattern of this.



Payload vs. Launch Site

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

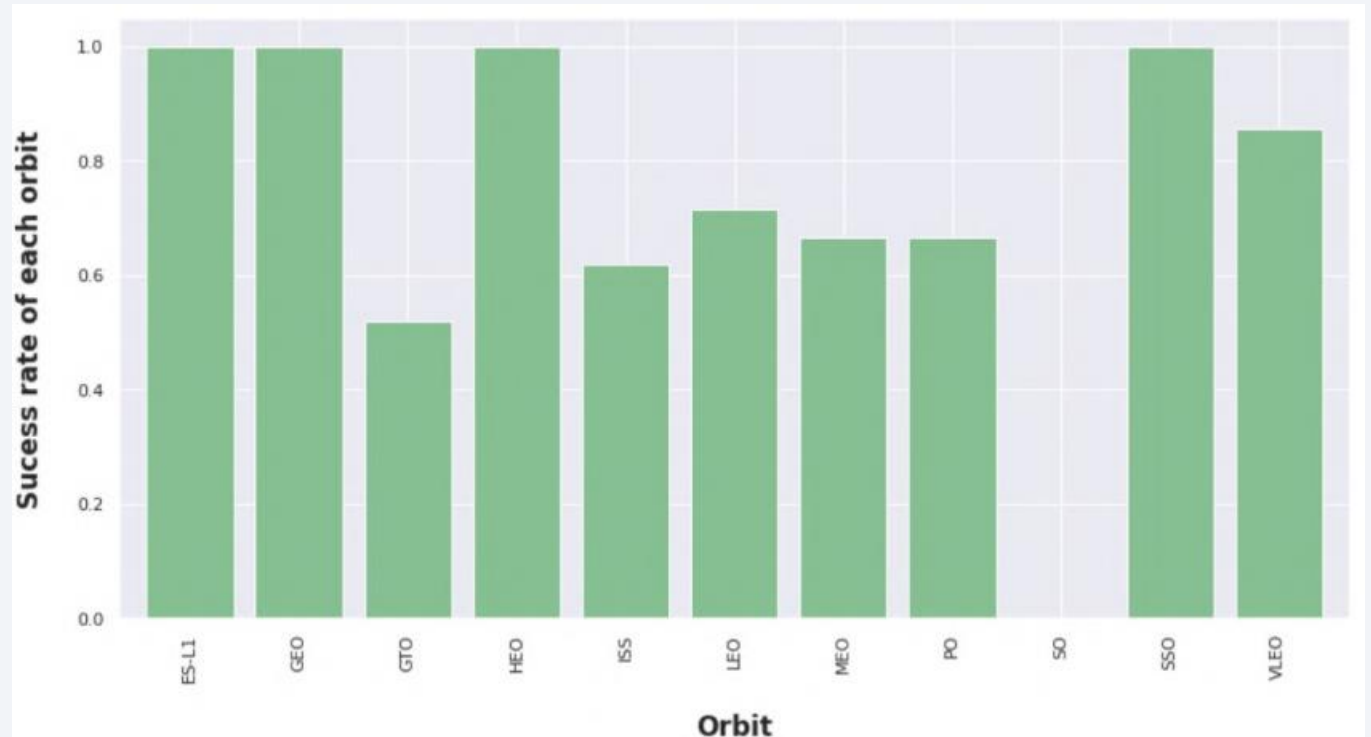
However, there is no clear pattern to say the launch site is dependent to the payload mass for the success rate.



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.

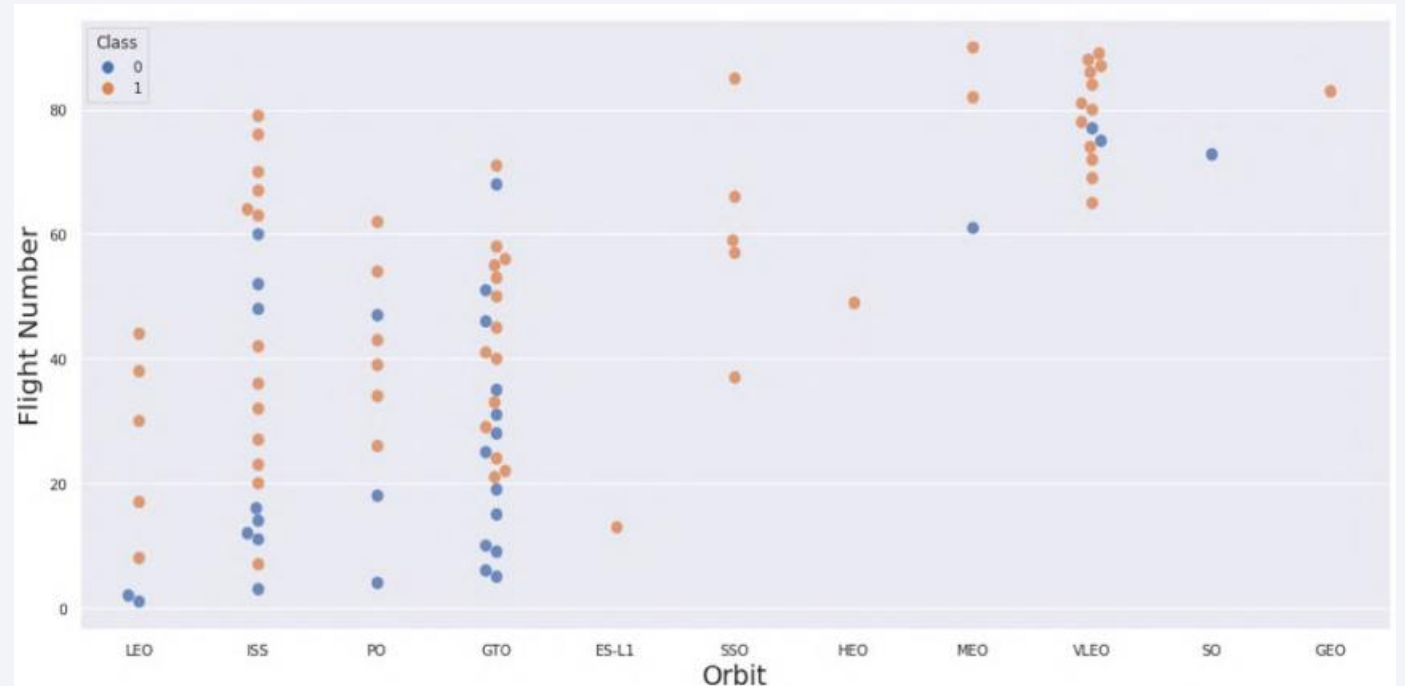
However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



Flight Number vs. Orbit Type

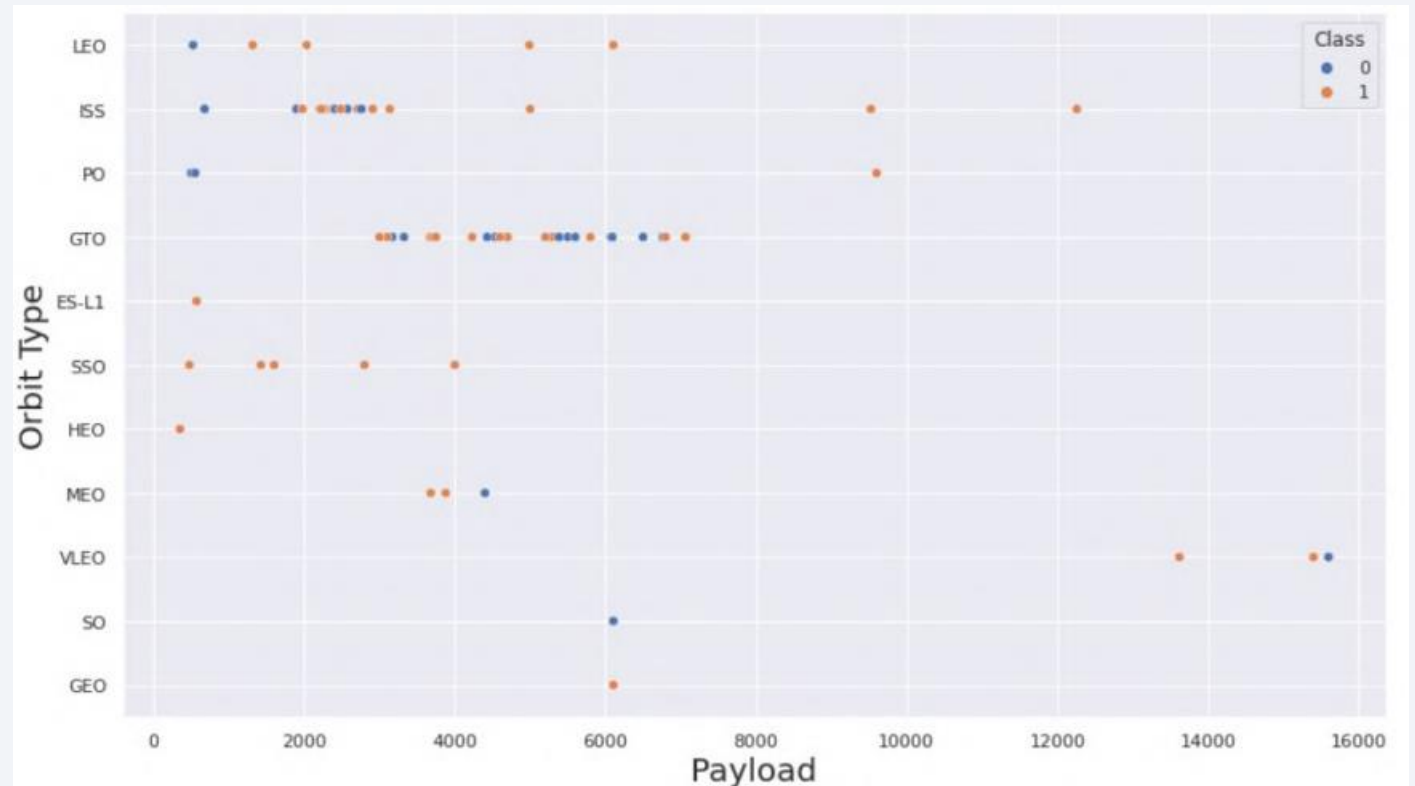
This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.

Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.



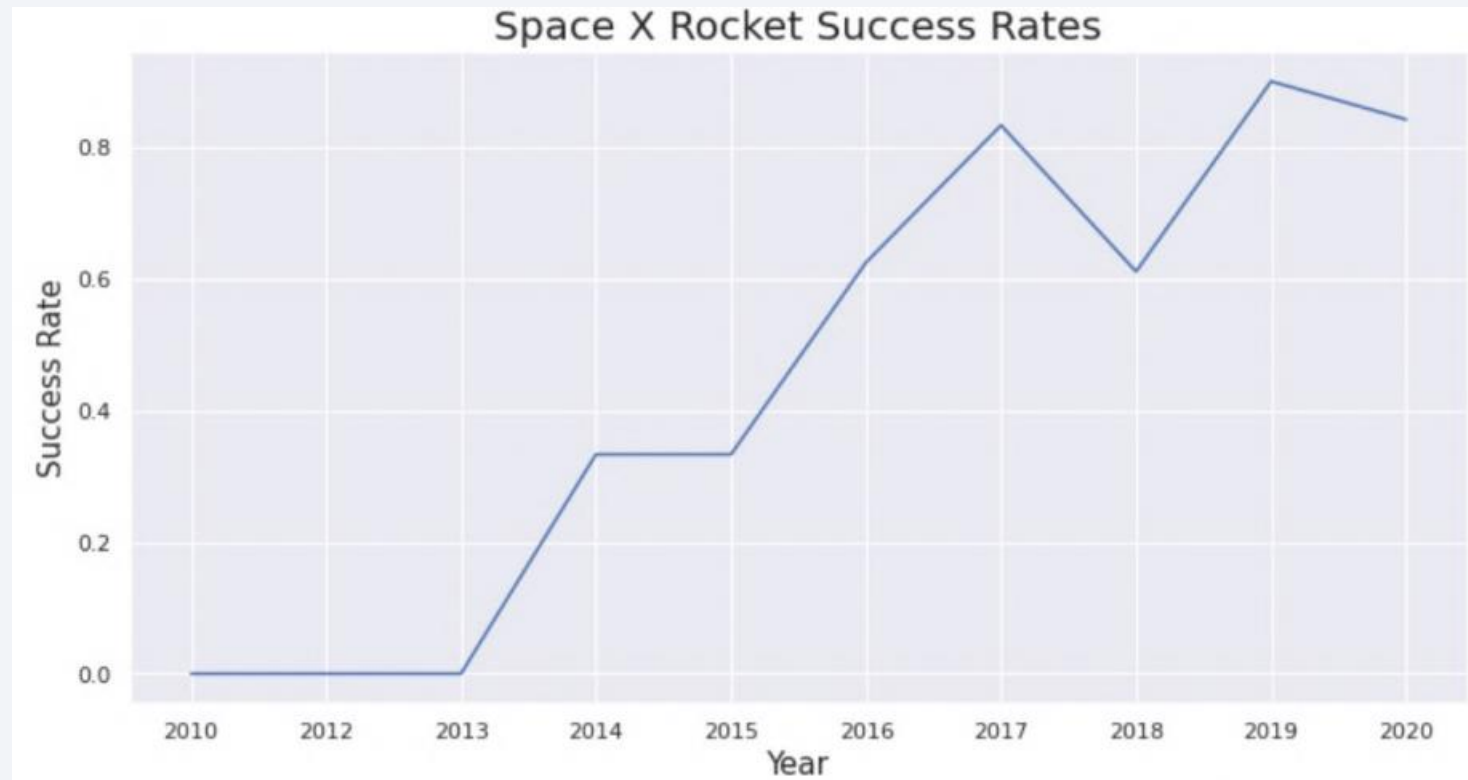
Payload vs. Orbit Type

Heavier payload has positive impact on LEO, ISS and PO 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 figure clearly depicts an increasing trend from the year 2013 until 2020. If this trend continues for the next year onward, the success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

I have used DISTINCT Keyword to filter out the Launch Site Names here.

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
* ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3  
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

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

Launch Site Names Begin with 'CCA'

I have used the LIKE to display 5 records where launch sites begin with 'CCA'

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

```
task_2 = '''
SELECT *
FROM SpaceX
WHERE LaunchSite LIKE 'CCA%'
LIMIT 5
'''

create_pandas_df(task_2, database=conn)
```

	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

I have 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

I have 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.clogj3  
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

Average Payload Mass by Booster Version F9 v1.1

2928

First Successful Ground Landing Date

I have used 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 Successful Landing Outcome in Ground Pad"  
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
* ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3  
sd0tgtu01qde00.databases.appdomain.cloud:32731/bludb  
Done.
```

First Successful Landing Outcome in Ground Pad
--

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

I 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.clogj3sd0tgtu0lqde00.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

I used wildcard like '%' to filter for WHERE Mission Outcome 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

I 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.clogj3sd0tgtu01qde00.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

I 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.clogj3sd0tgtu0lqde00.
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 2017-03-20.

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.clogj3sd0tgtu01qde00.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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

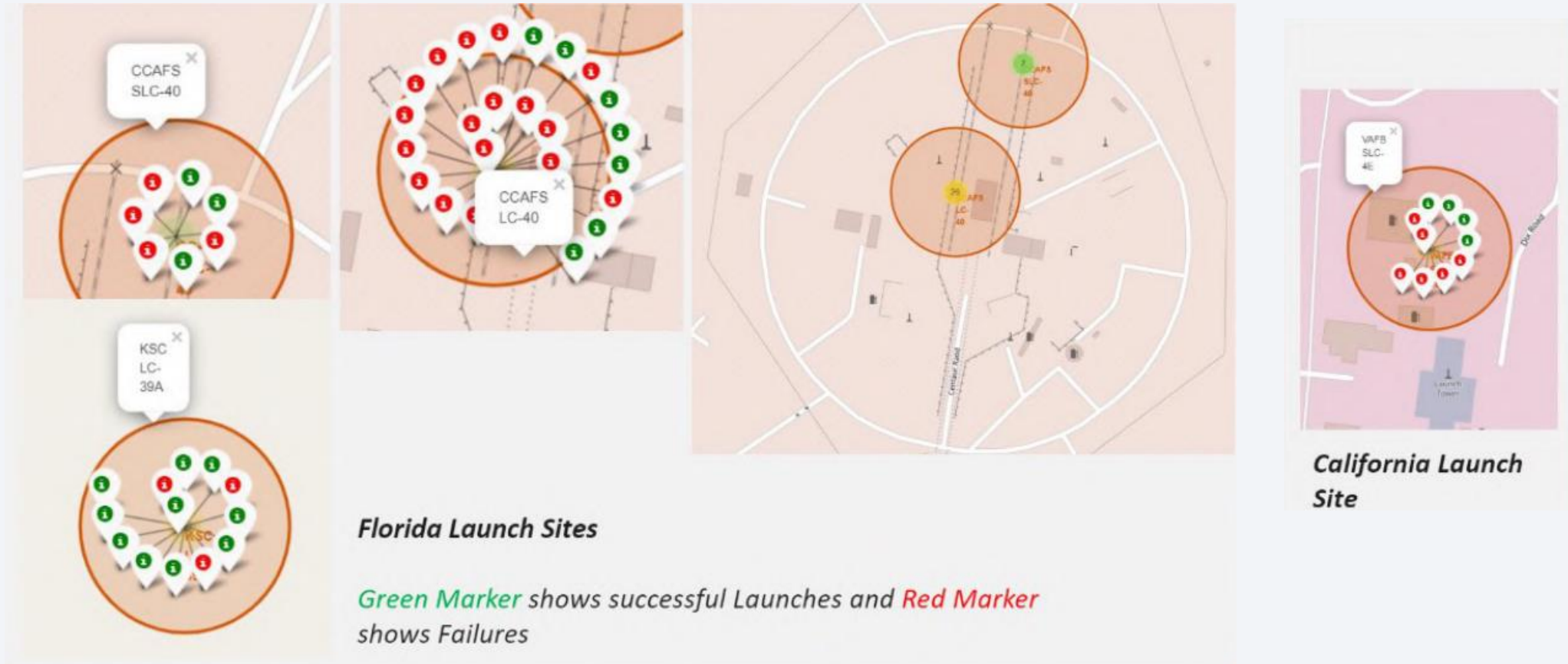
Section 3

Launch Sites Proximities Analysis

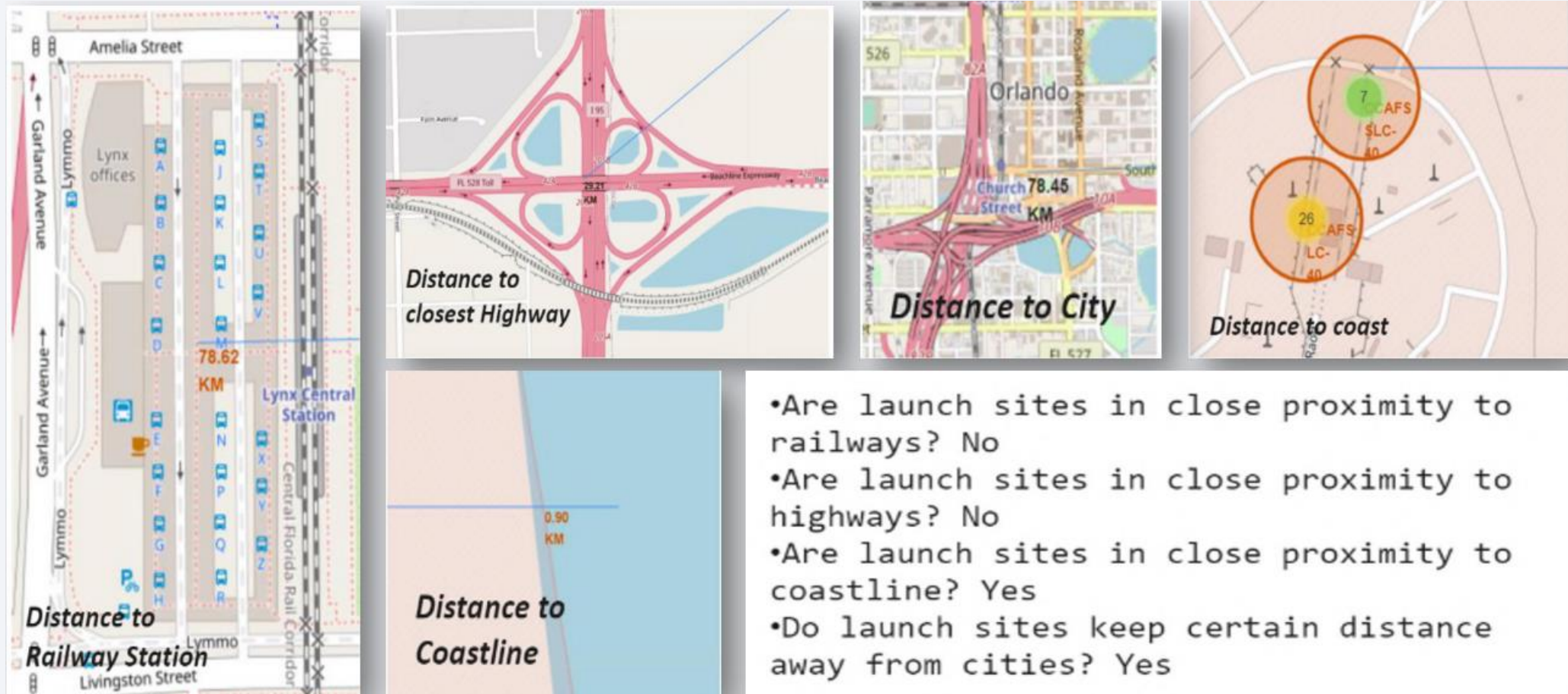
Site Locations in USA



Markers showing launch sites with color labels



Launch Sites Distance to Landmarks

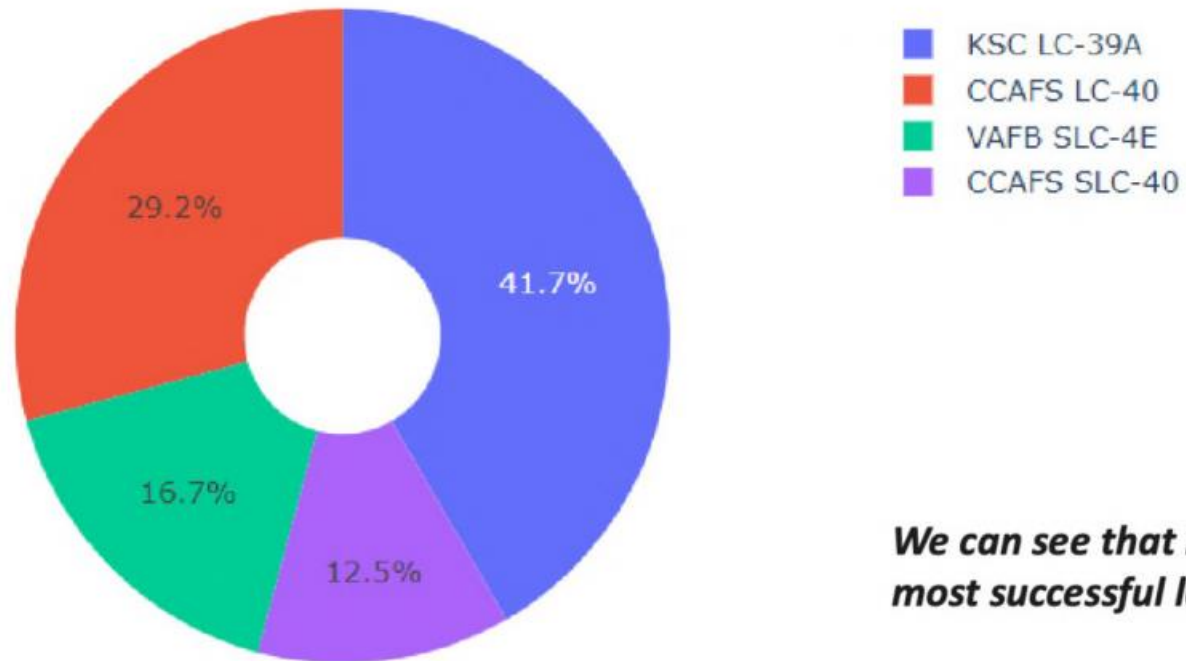




Section 4

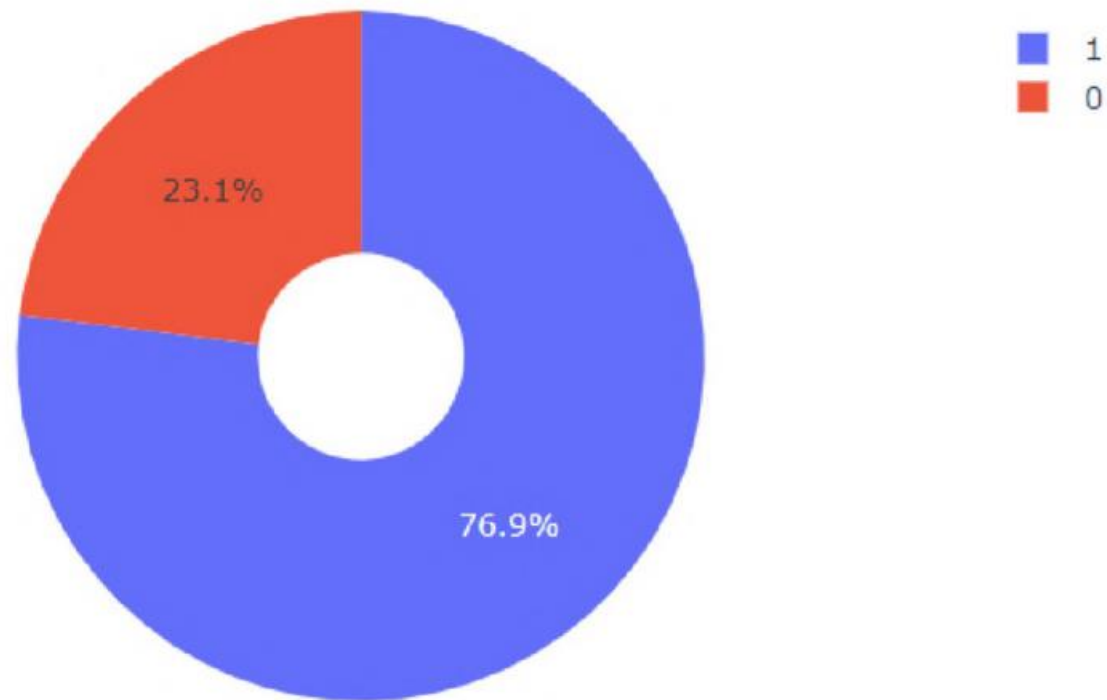
Build a Dashboard with Plotly Dash

Success Percentage of Each Site



We can see that KSC LC-39A had the most successful launches from all the sites

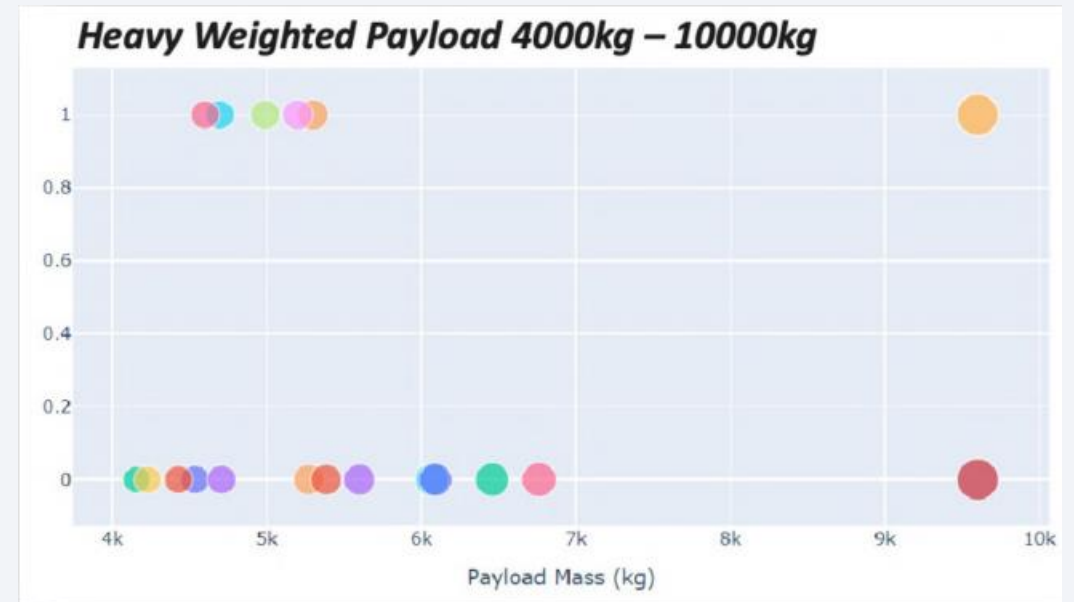
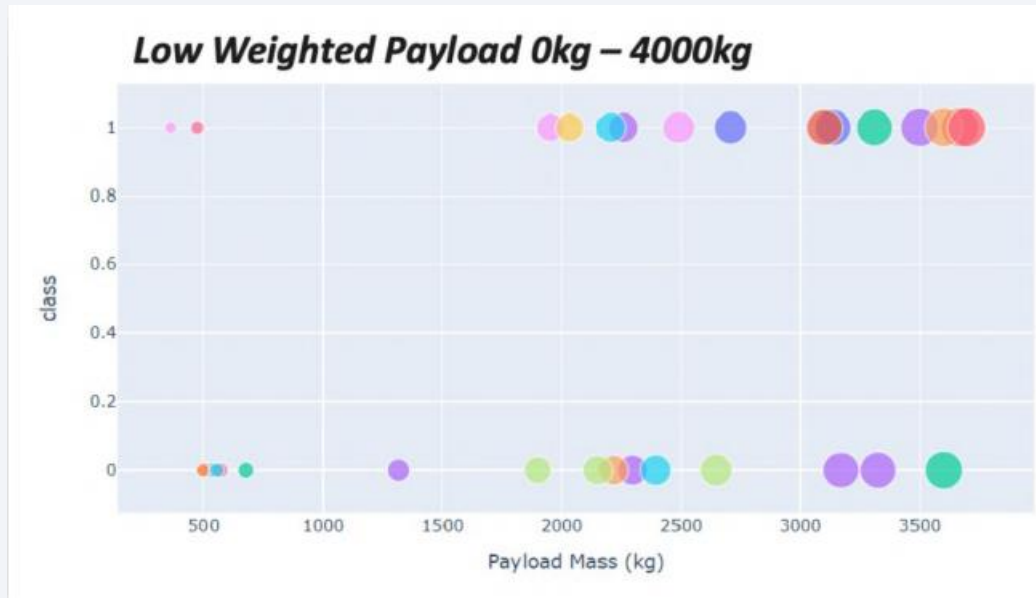
Highest Launch Success Ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot

We can see that all the success rate for low weighted payload is higher than heavy weighted payload



Section 5

Predictive Analysis (Classification)

Classification Accuracy

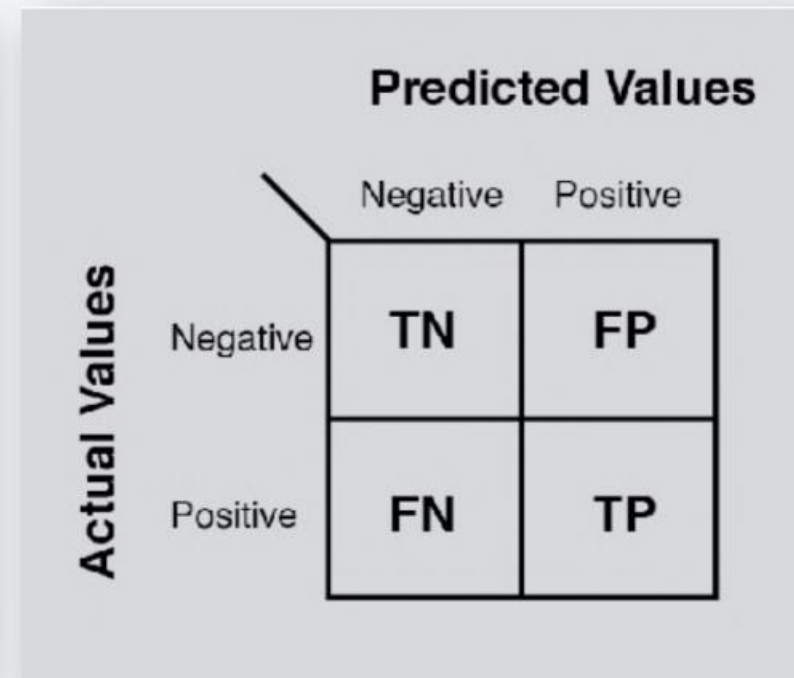
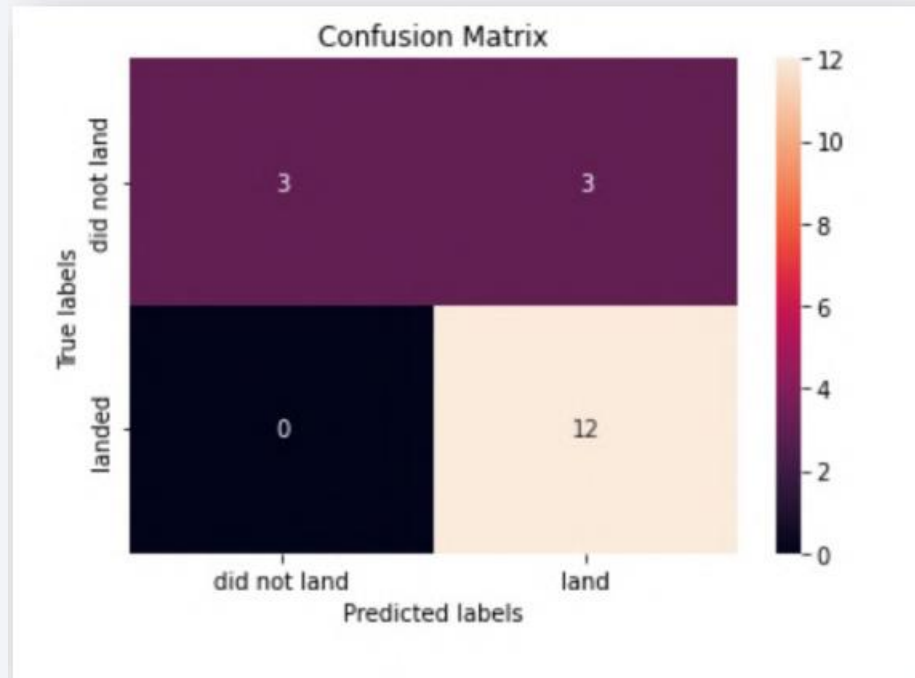
- As we can see, by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy.

```
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_samples_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

I 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!

