IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ume Habiba>
<3/7/2025>

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

Introduction

Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

- What factors determine if the rocket will land successfully?

- The interaction amongst various features that determine the success rate of a successful landing.

- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

The data was collected using various methods

Data collection was done using get request to the SpaceX API.

- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().

- We then cleaned the data, checked for missing values and fill in missing values where necessary.

- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.

- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- **The link to the notebook is**
- https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/ffd67b30864dc3de743fccdf4385961655f89afa/Data%20Collection%20API.ipynb

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)

In [8]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/

In [9]: response.status_code

Out[9]: 200

In [10]: # Use json_normalize meethod to convert the json result into a dataframe
         data=pd.json_normalize(response.json())

In [11]: # Displaying first 5 rows
         data.head()

Out[11]:
```

static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships

Data Collection - Scraping

- **We applied web scrapping to**
- **webscrap Falcon 9 launch records**
- **with BeautifulSoup**
- **We parsed the table and**
- **converted it into a pandas**
- **dataframe.**
- The link to the notebook is
- https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/ffd67b30864dc3de743fccdf4385961655f89afa/Data%20Collection%20with%20Web%20Scraping.ipynb

```
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]: 200

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>

In [8]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

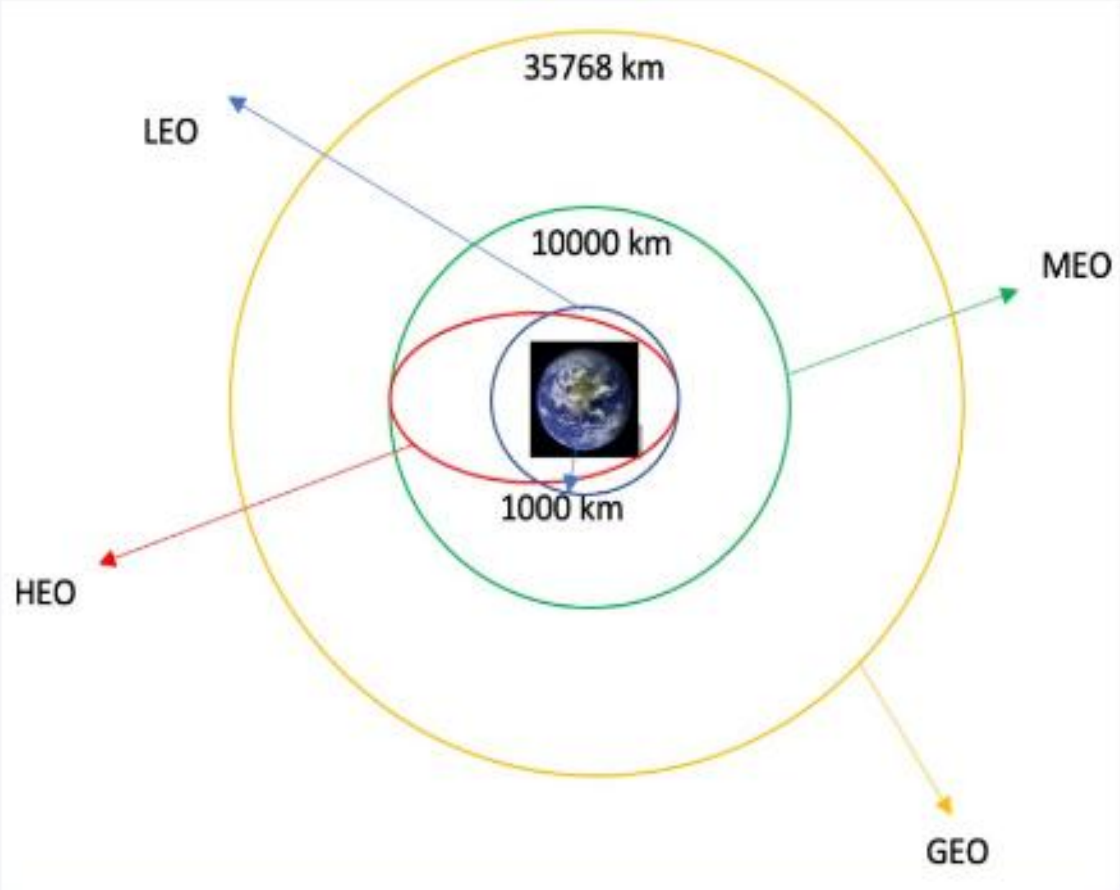
In [9]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
```

Data Wrangling

- We performed exploratory data analysis
- and determined the training labels.
- We calculated the number of launches at
- each site, and the number and
- occurrence of each orbits
- We created landing outcome label from
- outcome column and exported the
- results to csv.

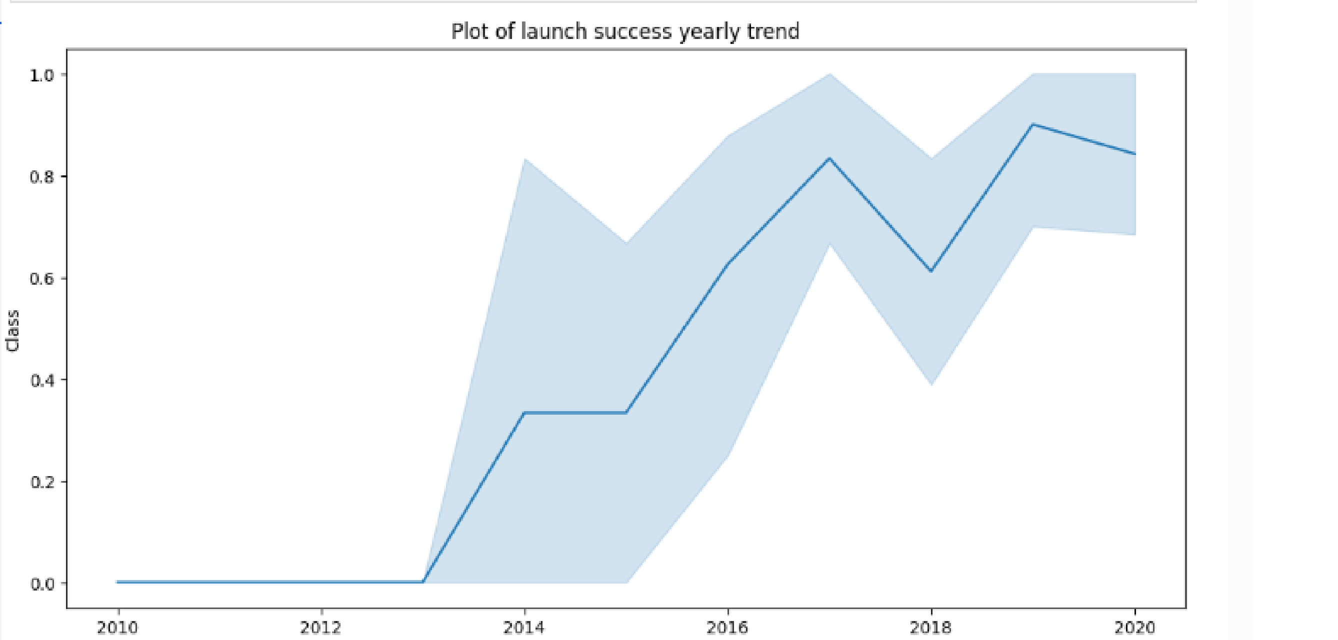
- https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/ffd67b30864dc3de743fccdf4385961655f89a
/Data%20Wrangling.ipynb

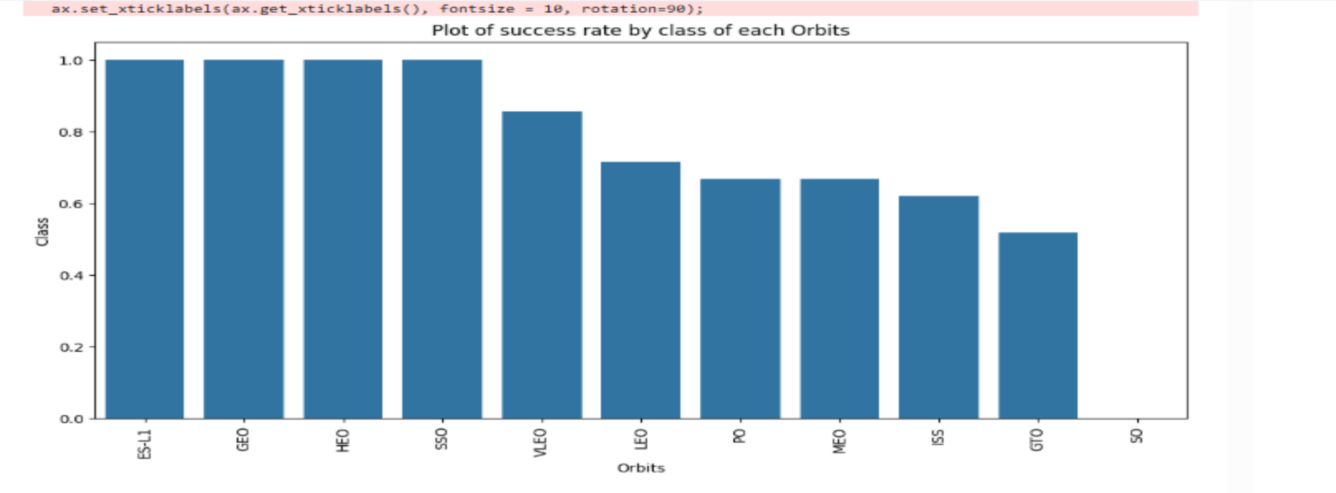


EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/ffd67b30864dc3de743fccdf4385961655f89afa/EDA%20with%20Data%20Visualization.ipynb





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving
- the jupyter notebook.
- � We applied EDA with SQL to get insight from the data. We wrote queries to
- find out for instance:
- The names of unique launch sites in the space mission.
- - The total payload mass carried by boosters launched by NASA (CRS)
- - The average payload mass carried by booster version F9 v1.1
- - The total number of successful and failure mission outcomes
- - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/ffd67b30864dc3de743fccdf4385961655f89afa/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- **We marked all launch sites, and added map objects such as markers, circles, lines**
- **to mark the success or failure of launches for each site on the folium map.**
- **We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0**
- **for failure, and 1 for success.**
- **Using the color-labeled marker clusters, we identified which launch sites have**
- **relatively high success rate.**
- **We calculated the distances between a launch site to its proximities. We answered**
- **some question for instance:**
- \- Are launch sites near railways, highways and coastlines.
- \- Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and
- Payload Mass (Kg) for the different booster version.
- The link to the notebook is
- https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/0fb7e62906591b9078d7e5d8a066ba8a200efbd2/app.py

Predictive Analysis (Classification)

- **We loaded the data using numpy and pandas, transformed the data, split our**
- **data into training and testing.**
- **We built different machine learning models and tune different**
- **hyperparameters using GridSearchCV.**
- **We used accuracy as the metric for our model, improved the model using**
- **feature engineering and algorithm tuning.**
- **We found the best performing classification model.**
- The link to the notebook is
- https://github.com/umehabiba28/IBM-Data-Science-Capstone/blob/0fb7e62906591b9078d7e5d8a066ba8a200efbd2/Machine%20Learning%20Prediction.ipynb

Results

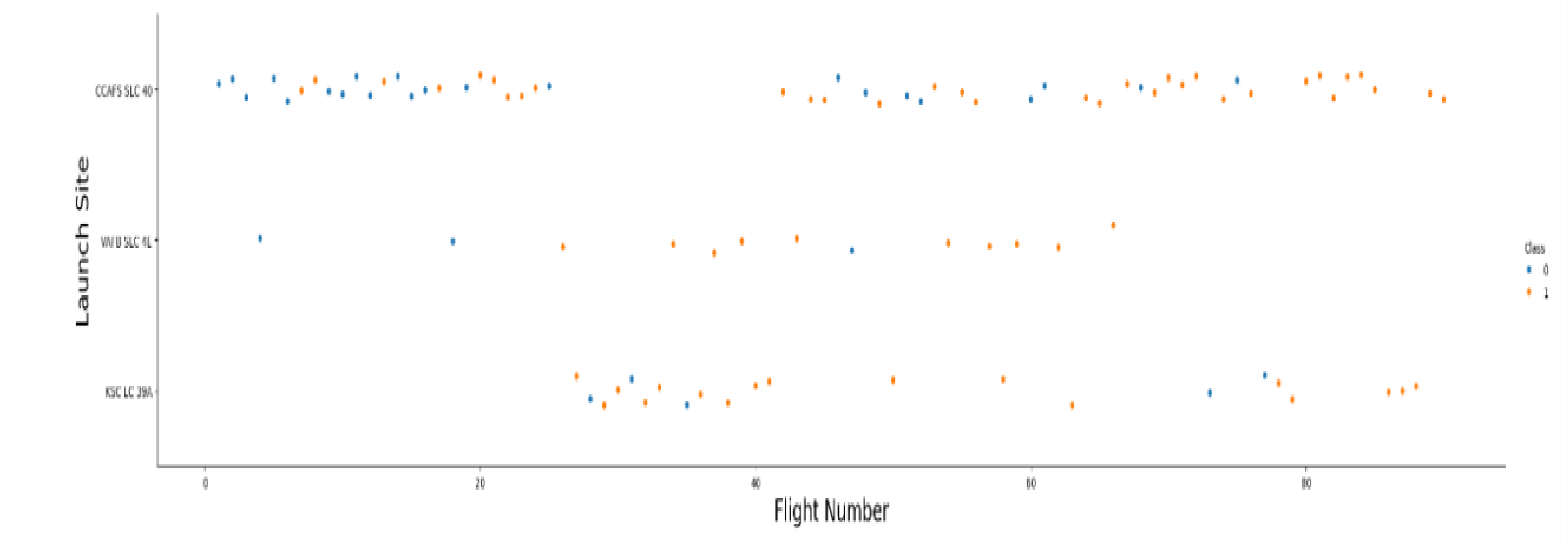
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

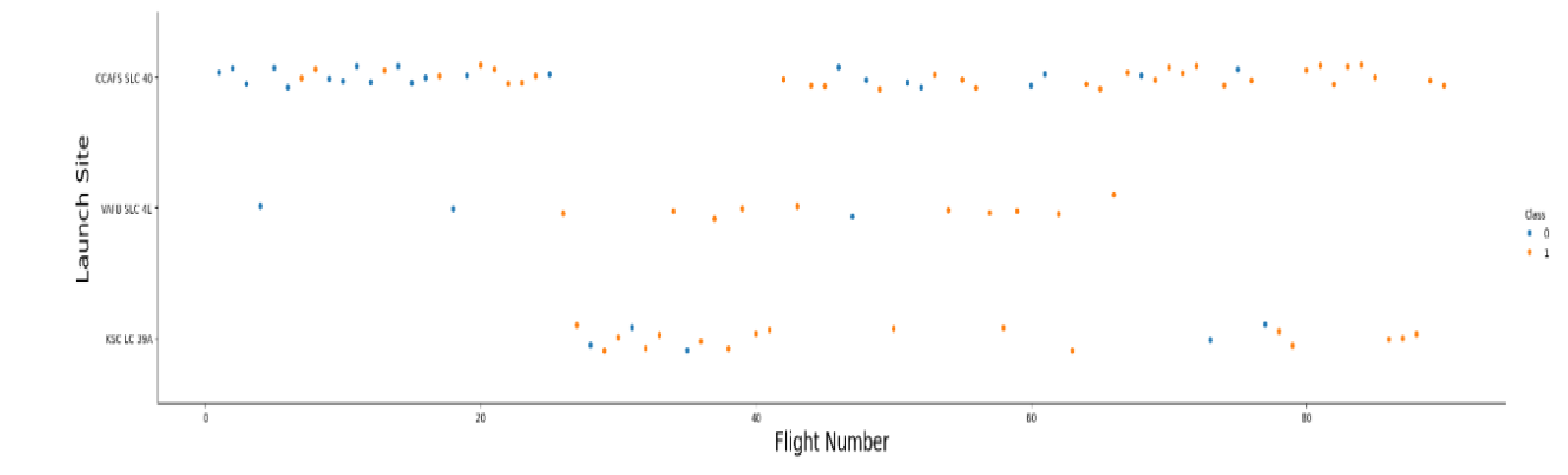
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the
- greater the success rate at a launch site.



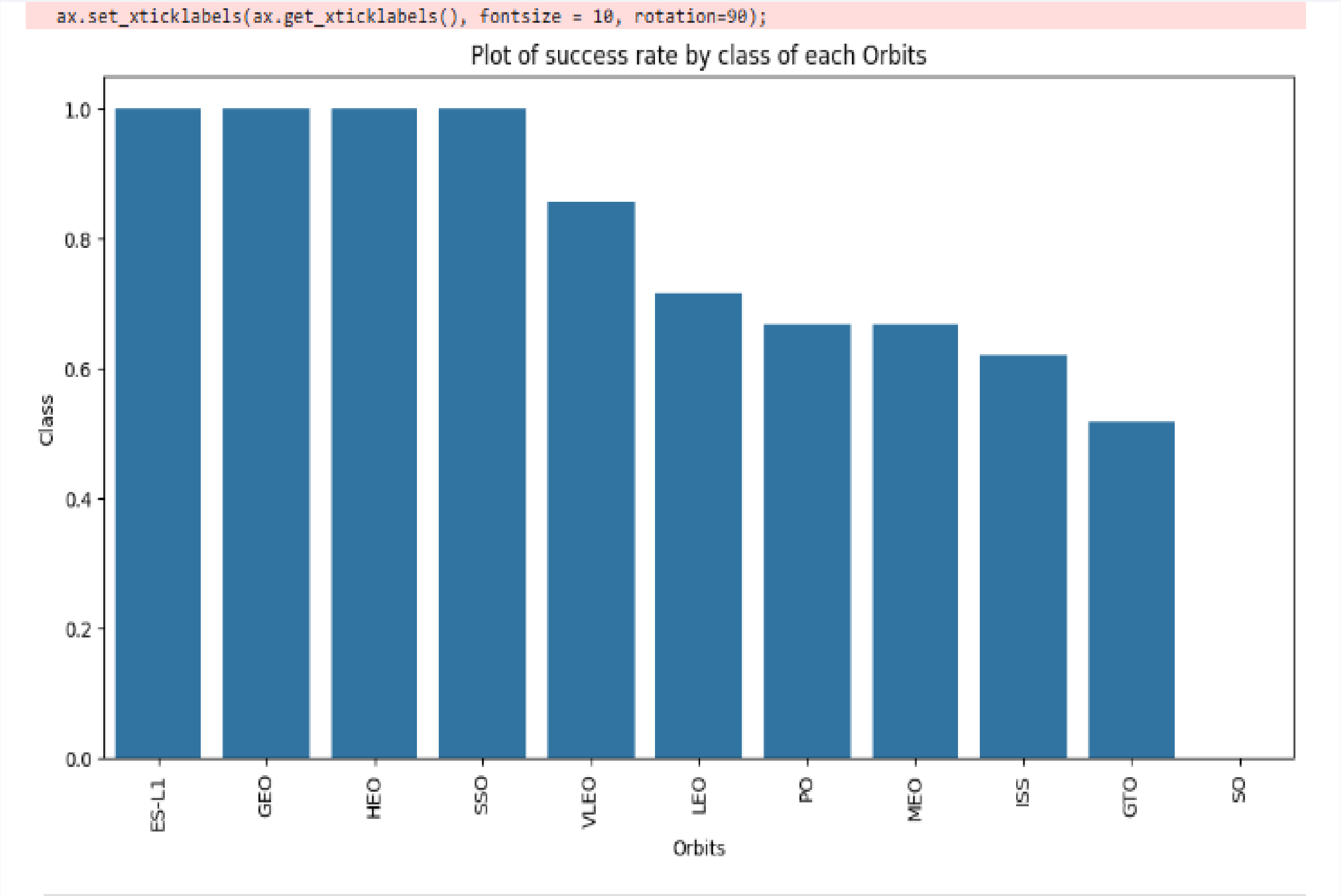
Payload vs. Launch Site

- Payload vs. Launch Site



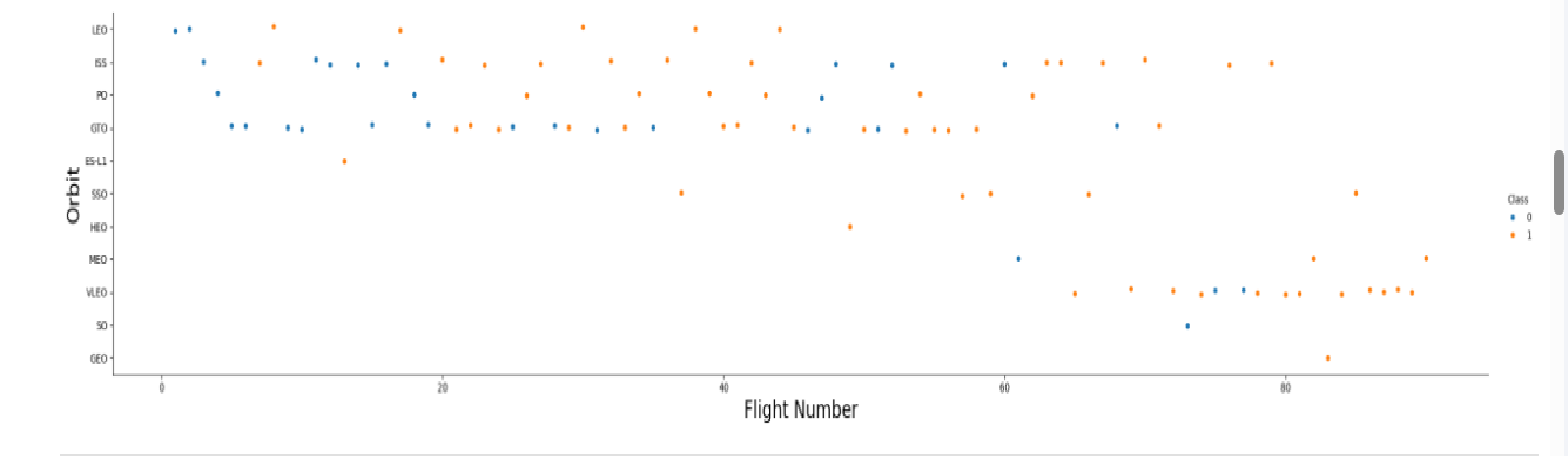
Success Rate vs. Orbit Type

From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



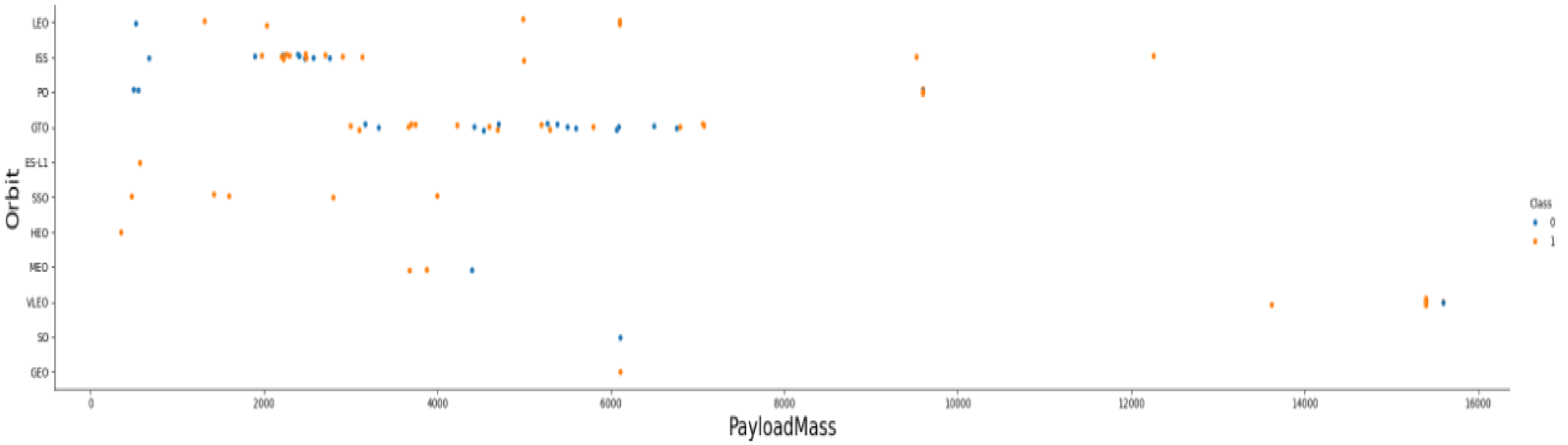
Flight Number vs. Orbit Type

The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



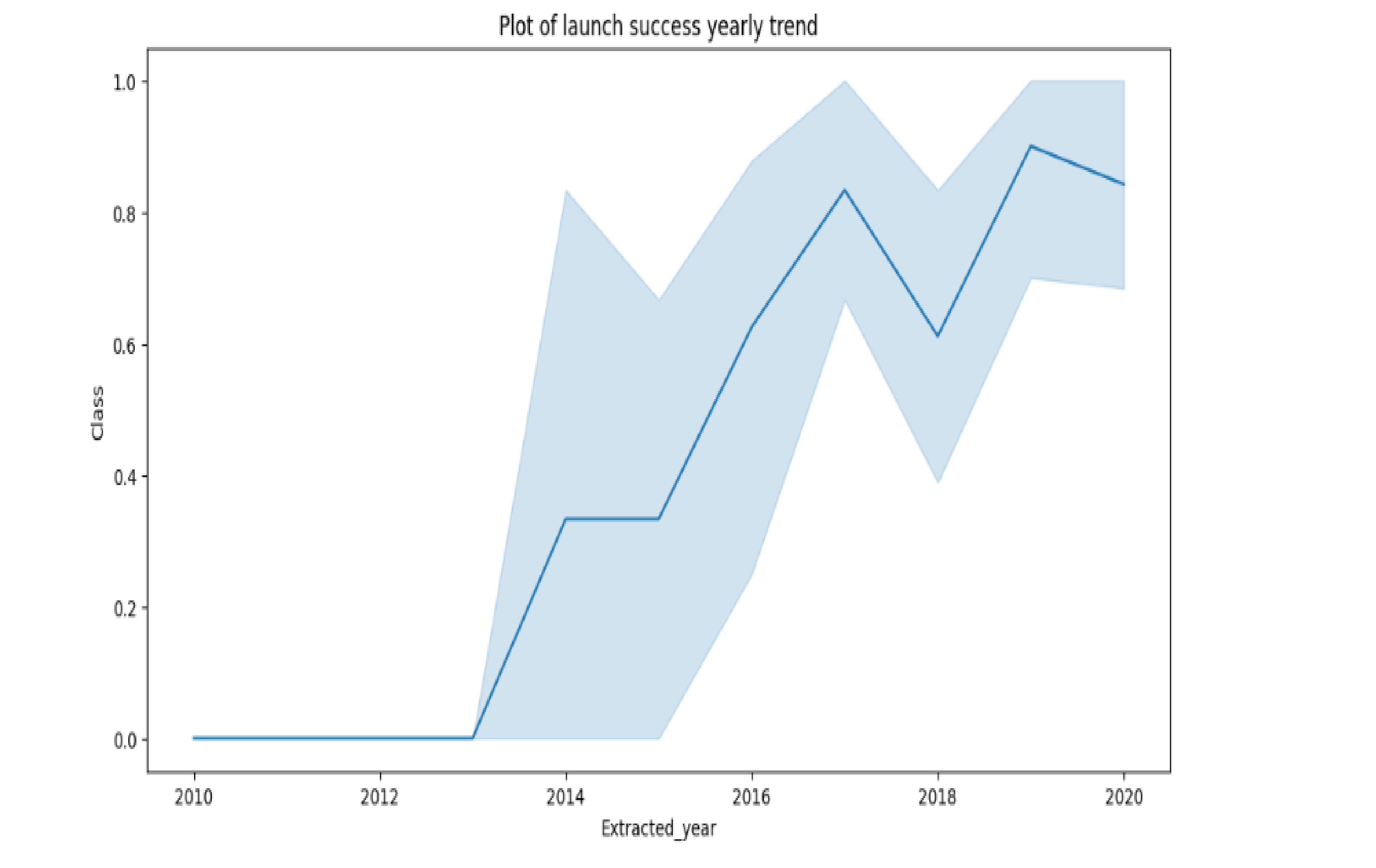
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO,
- LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can
- observe that success rate
- since 2013 kept on
- increasing till 2020.



All Launch Site Names

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

[6]:

	Launch Site
0	CCAFS LC-40
1	CCAFS SLC-40
2	KSC LC-39A
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- We used the query above to display 5 records where launch sites begin with
- `CCA`

[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

Total Payload Mass

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

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

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

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- We observed that the dates of the
- first successful landing outcome on
- ground pad was 22 nd December
- 2015

	firstsuccessfull_landing_date
0	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

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	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.

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	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.

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	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

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

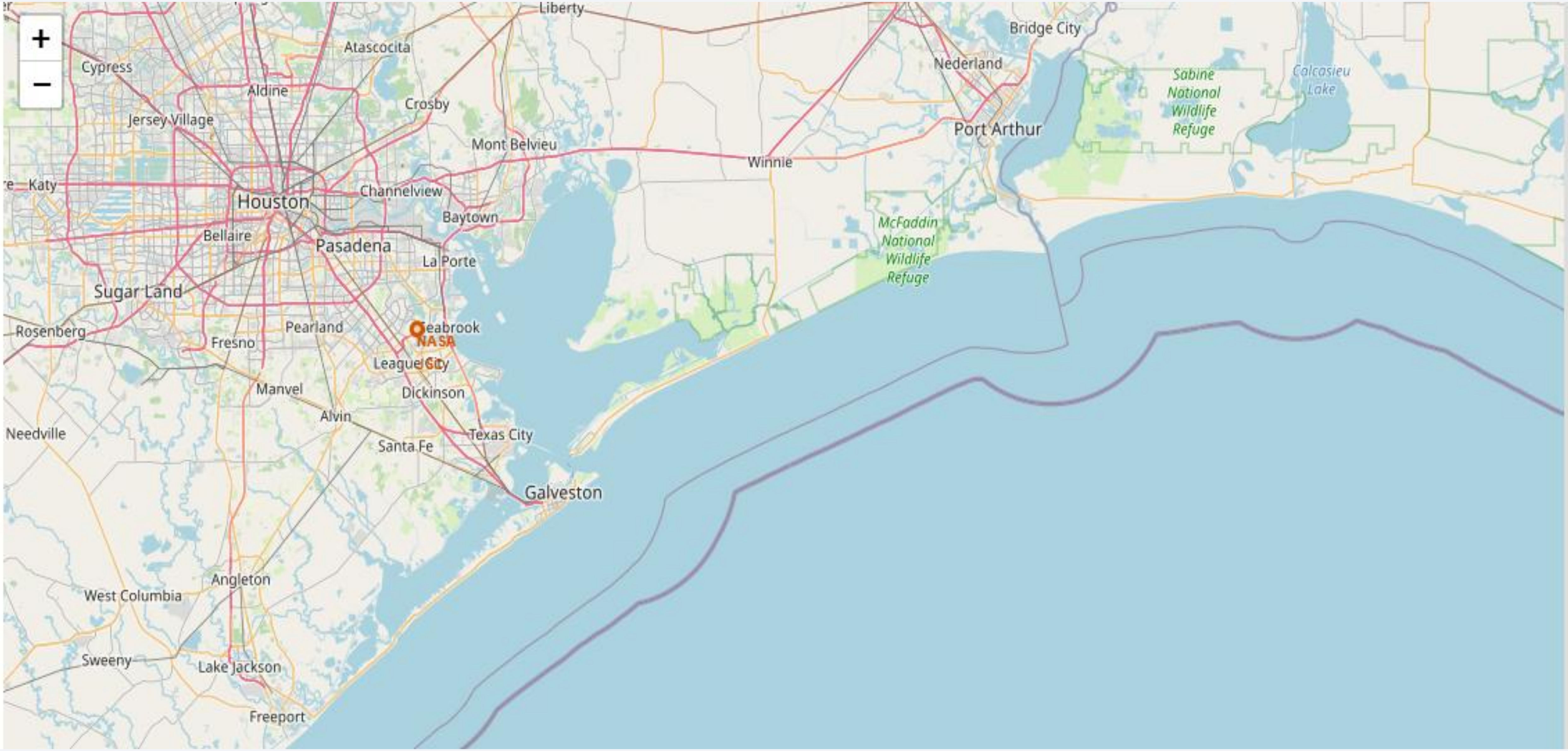
- We selected Landing outcomes
- and the COUNT of landing
- outcomes from the data and
- used the WHERE clause to filter
- for landing outcomes BETWEEN
- 2010-06-04 to 2010-03-20.
- We applied the GROUP BY
- clause to group the landing
- outcomes and the ORDER BY
- clause to order the grouped
- landing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

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

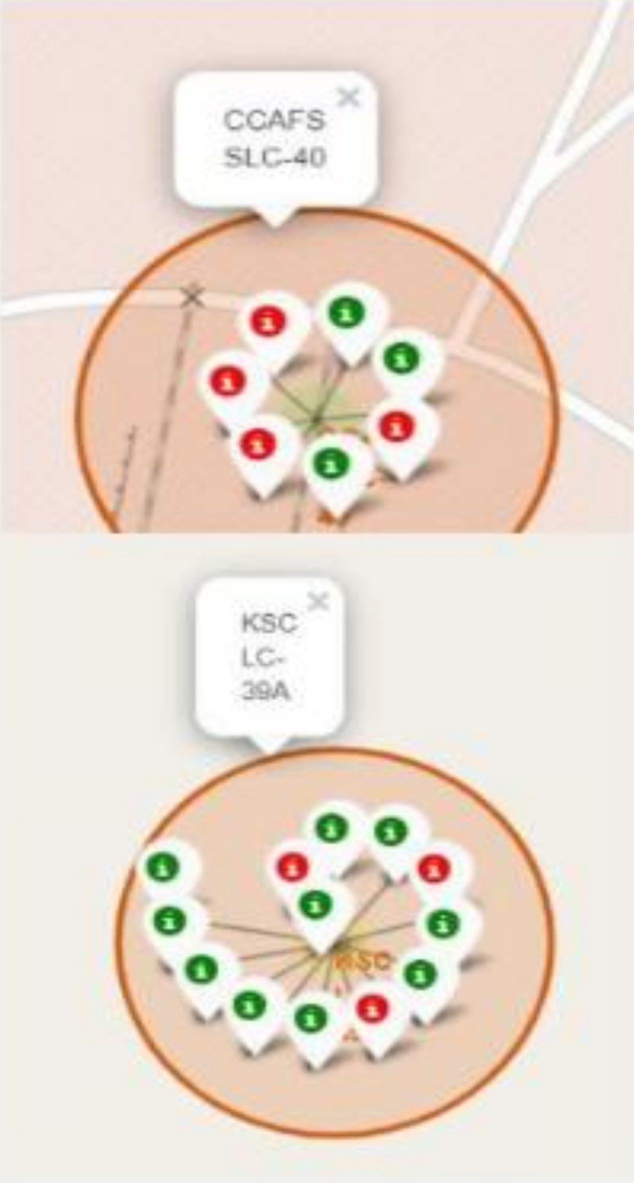
Section 3

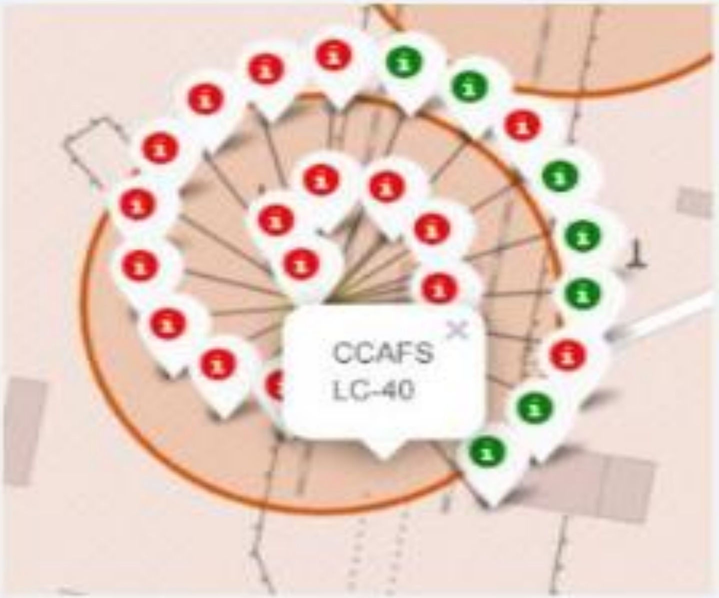
Launch Sites Proximities Analysis

All launch sites global map markers



Markers showing launch sites with color labels

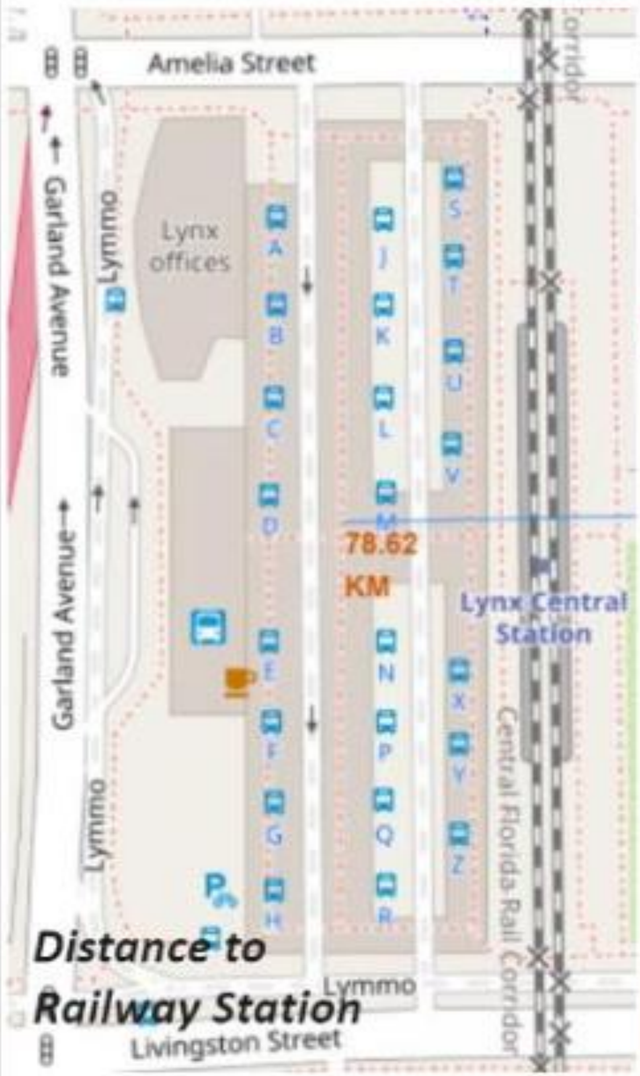


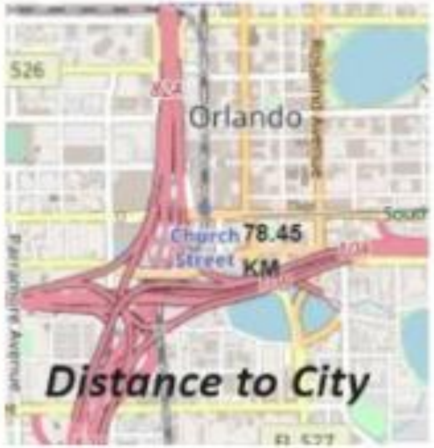


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

Launch Site distance to landmarks



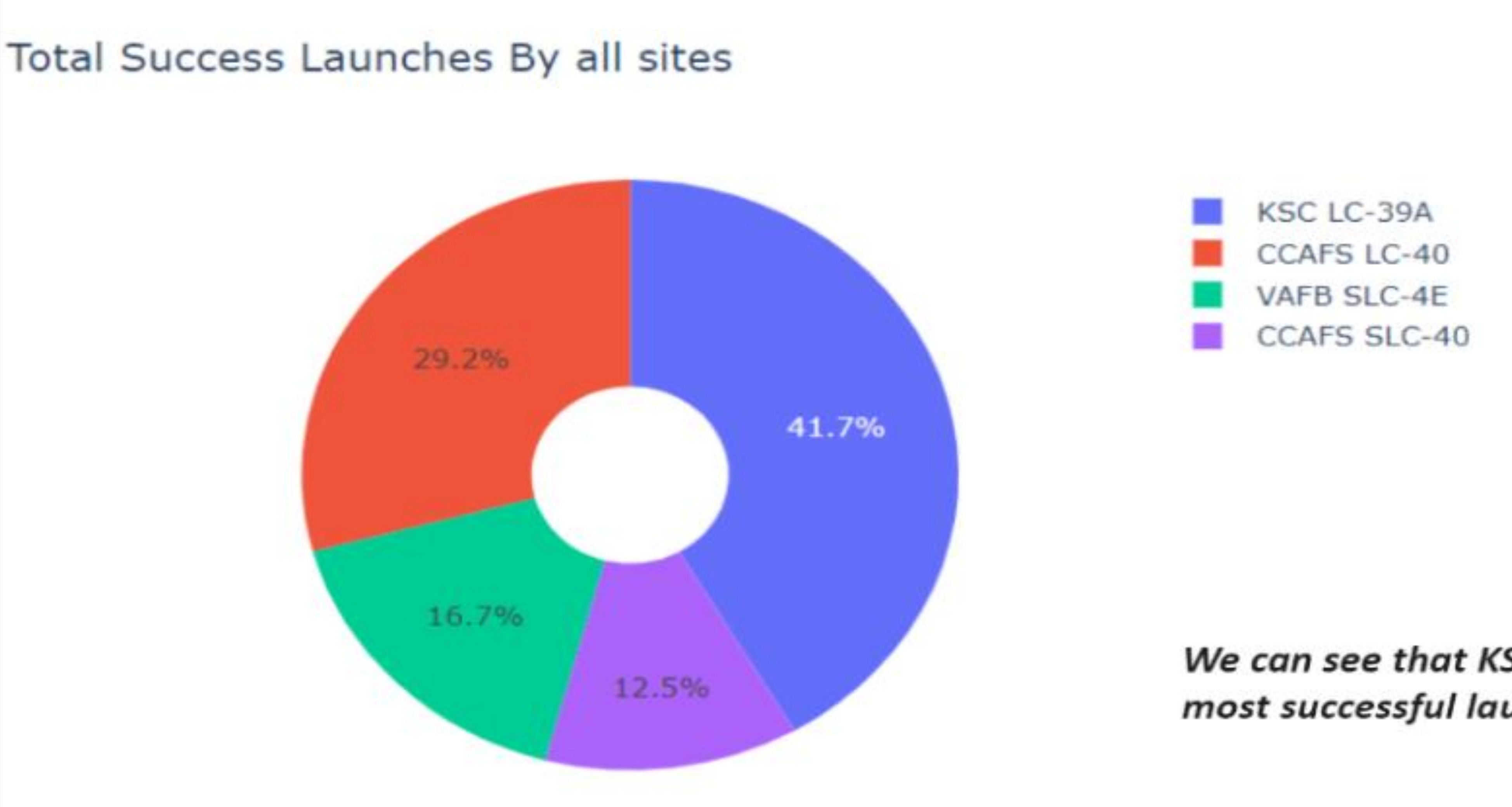


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

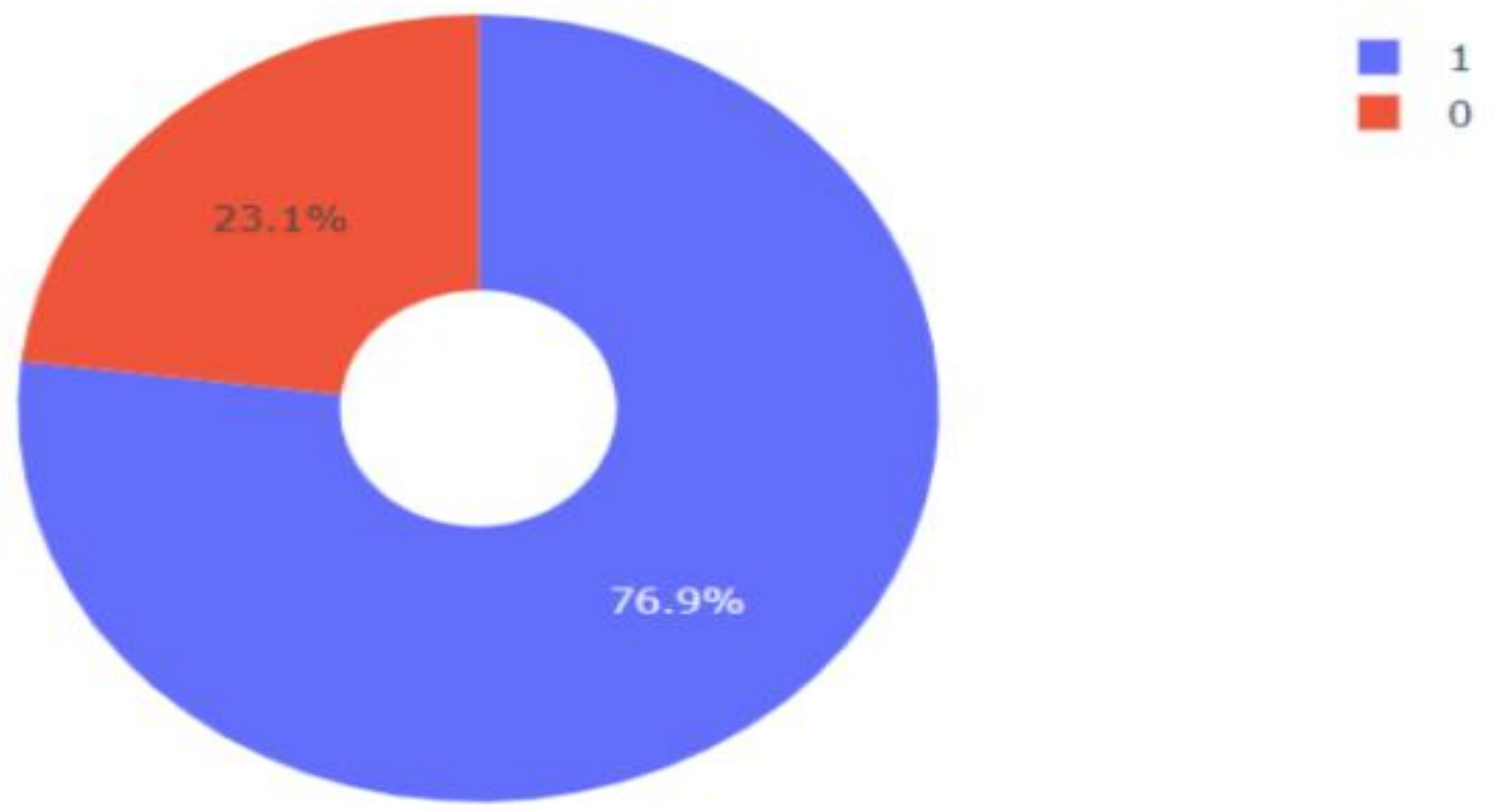
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



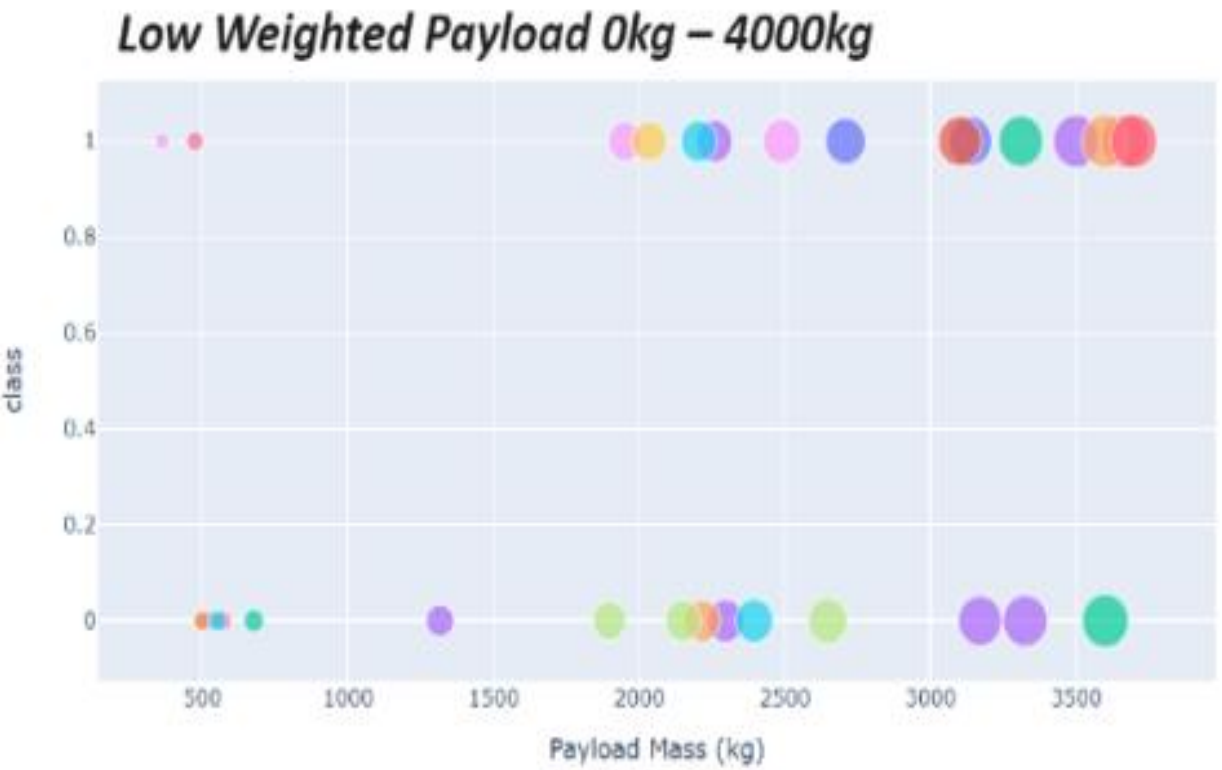
We can see that KSC LC-39A had the most successful launches from all the sites

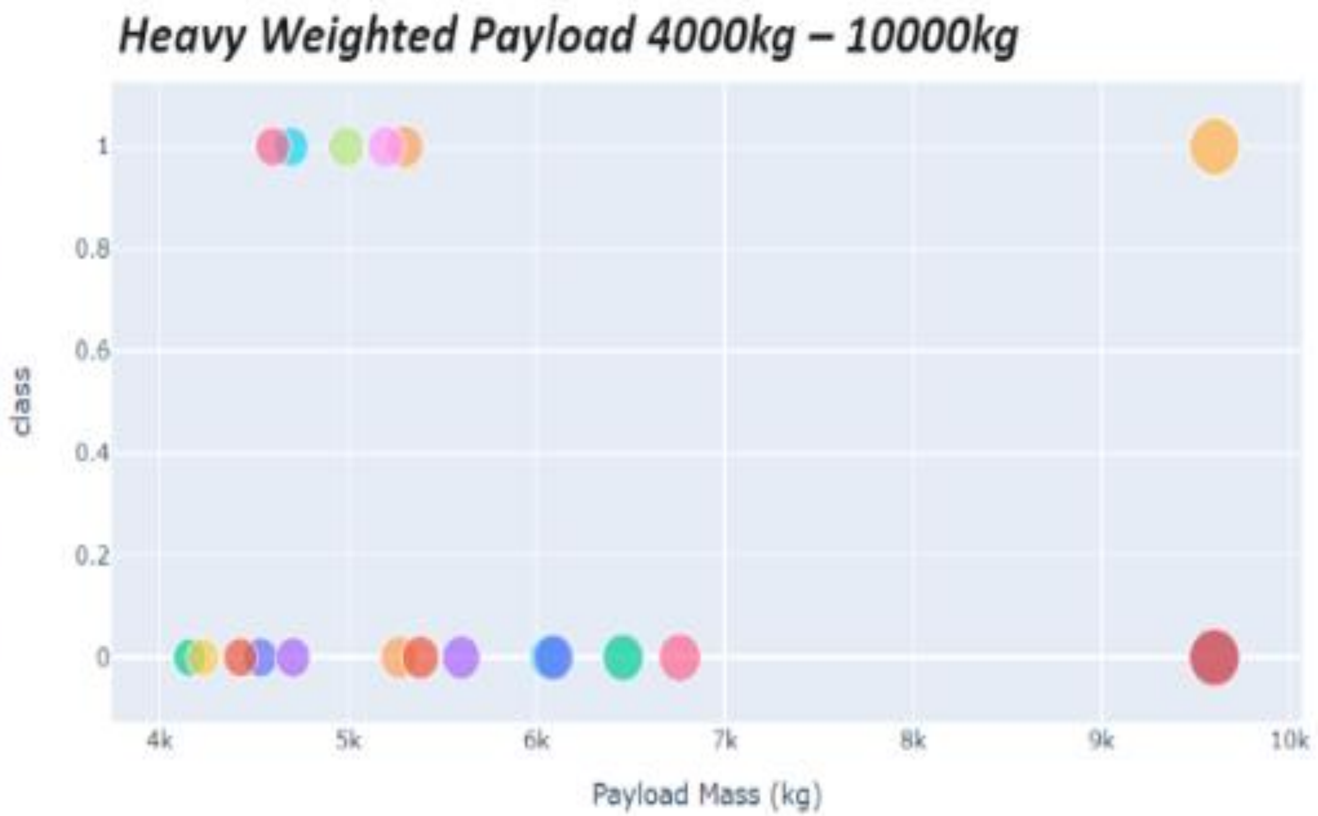
Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

The decision tree classifier is the model with

the highest classification accuracy

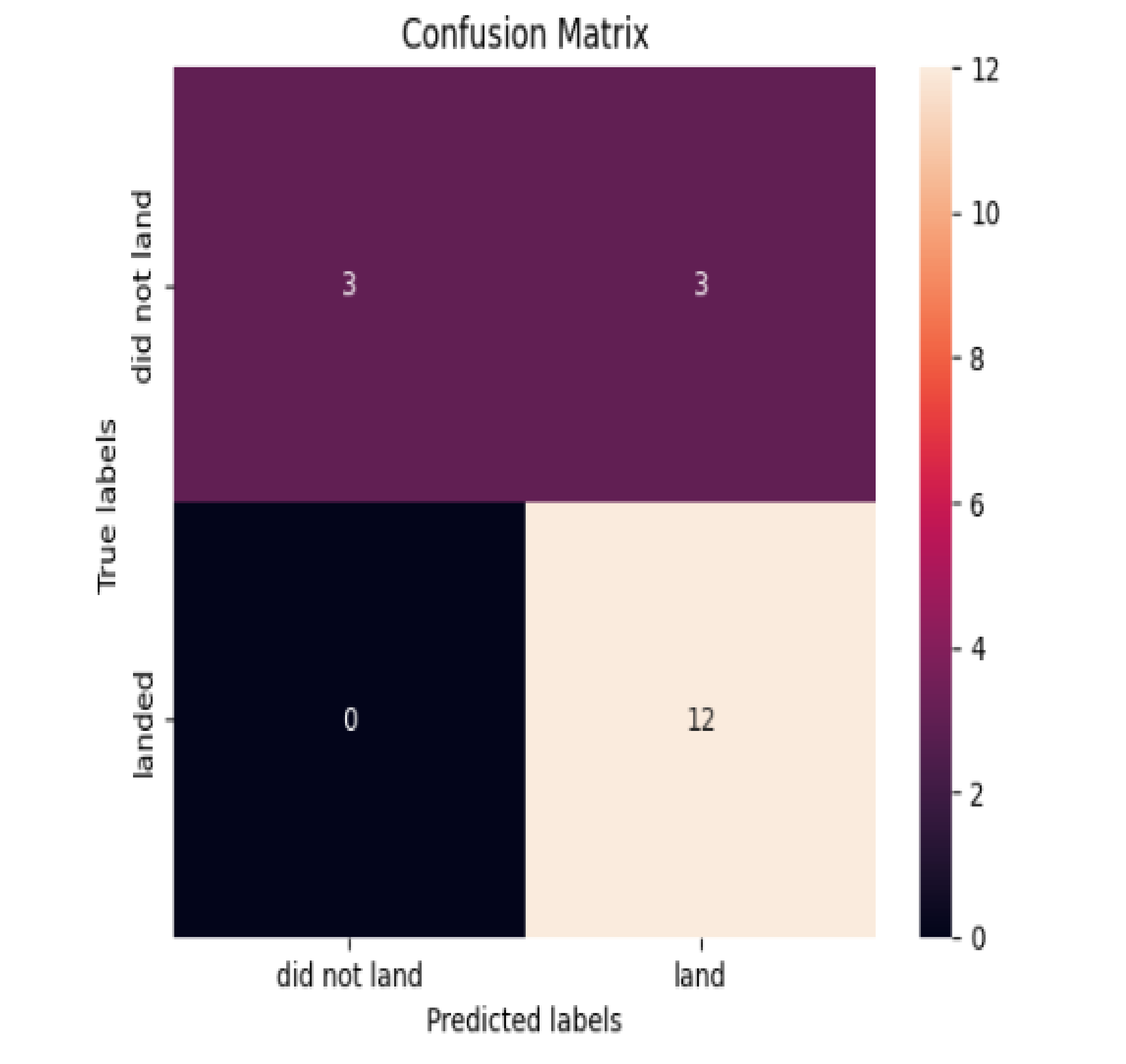
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
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'}
```

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 larger the flight amount at a launch site, the greater the success rate at a launch site.

Launch success rate started to increase in 2013 till 2020.

Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

KSC LC-39A had the most successful launches of any sites.

The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!