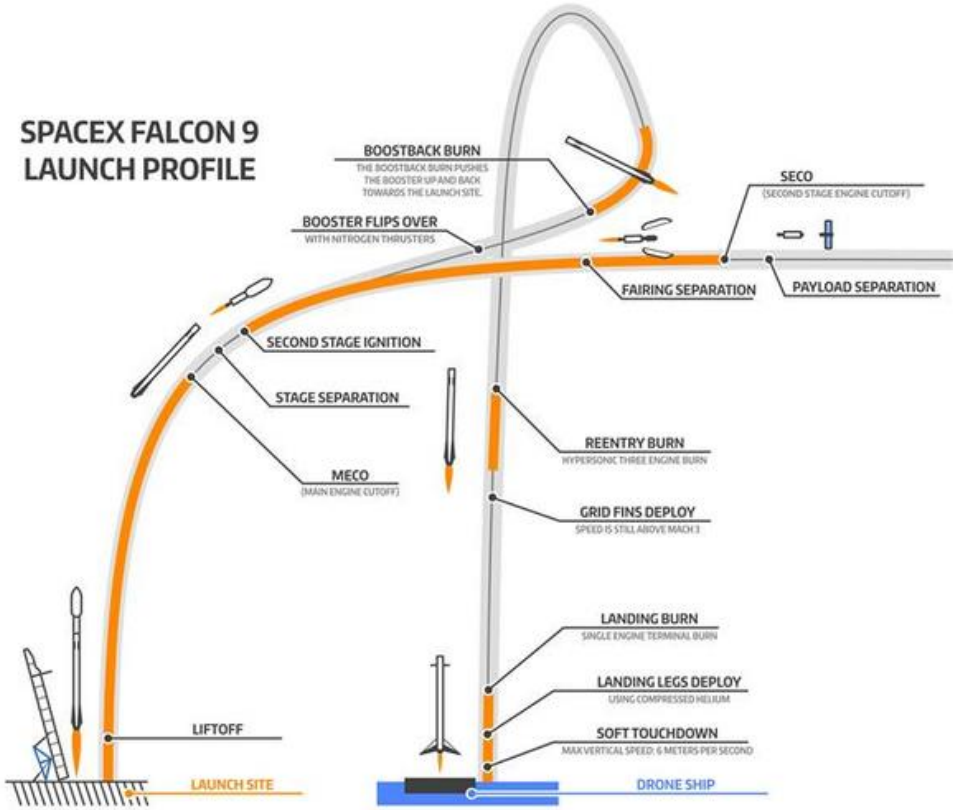
SpaceX Launches

Nursultan Bazarbay

12/01/2023

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix



EXECUTIVE SUMMARY

- 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.

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.

Space Y can against Space X for a rocket launch. This goal of our project is to create a machine learning pipeline to predict if the first stage will land successfully.

Questions to be answered:

1. What factors determine if the rocket will land successfully?
2. The interaction amongst various features that determine the success rate of a successful landing.
3. What operating conditions needs to be in place to ensure a successful landing program.

METHODOLOGY

1) Collect the data: SpaceX API and web scraping from Wikipedia.
2) Process the data: one-hot encoding to create categorical features
3) Perform exploratory data analysis (EDA) using visualization and SQL
4) Perform interactive visual analytics using Folium and Plotly Dash
5) Perform predictive analysis using classification models - build, tune, evaluate classification models

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.

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection – Web Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

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

Print the page title to verify if the `BeautifulSoup` object was created properly

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

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

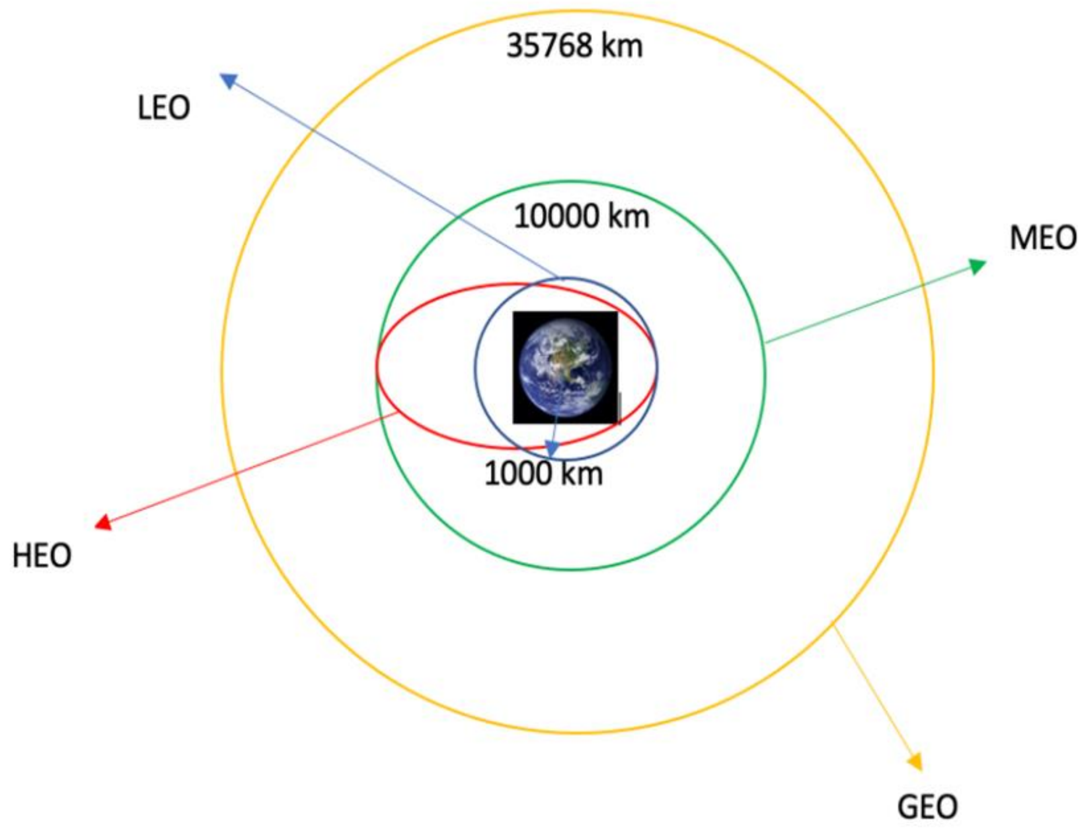
3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Processing (Wrangling)

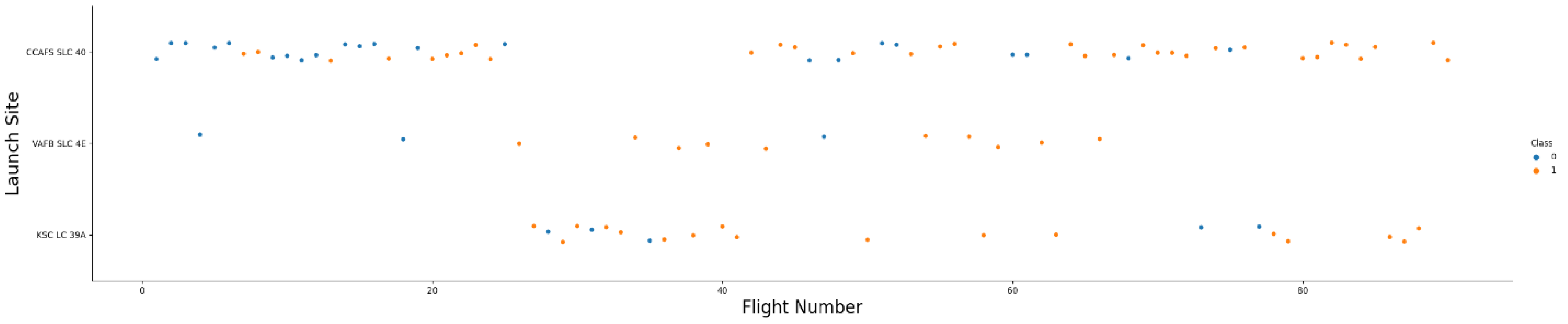
- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.



EDA with visualization results

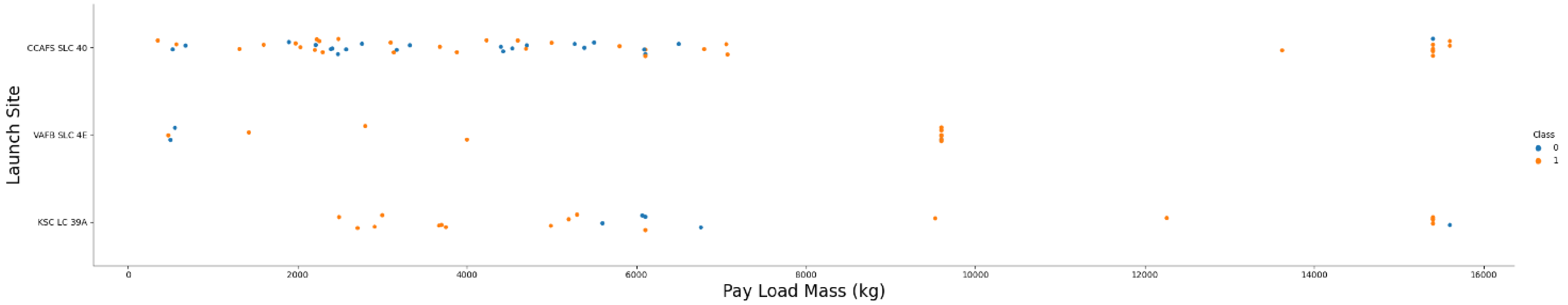
EDA: Flight Number vs. Launch Site

- SpaceX mainly uses CCAFS SLC-40 (except for the period between flights 27-41)
- Success rate for first 20 flights is quite low and it increases later



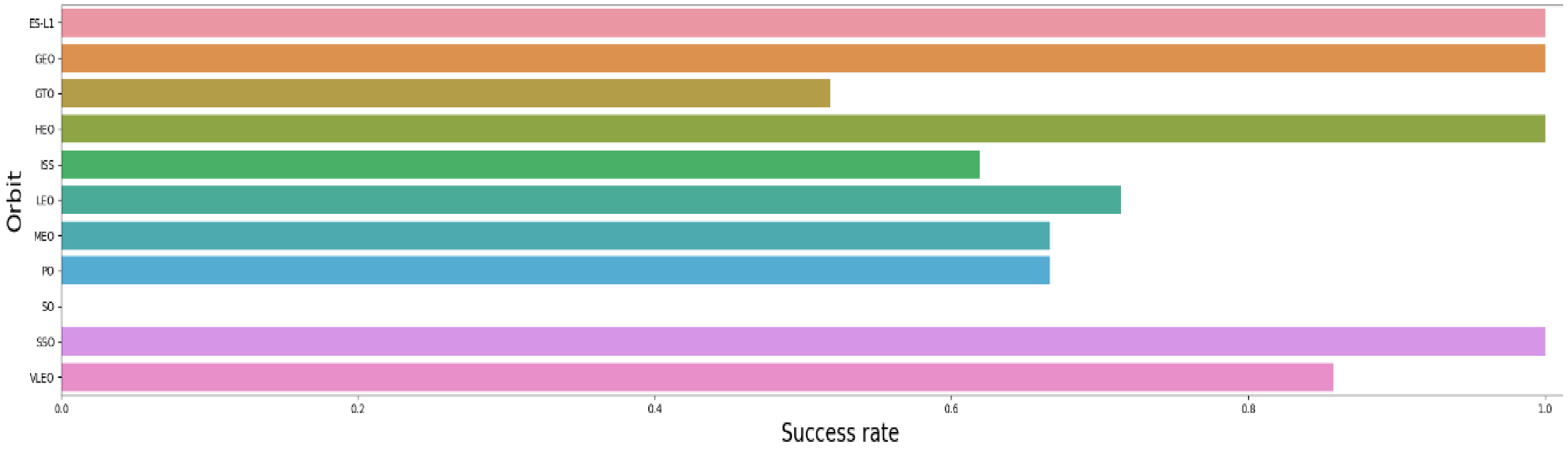
EDA: Payload vs. Launch Site

- Most payloads are under 8000kg
- VAFB-SLC launch site does not have rockets launched for heavypayload mass(greater than 10000kg)



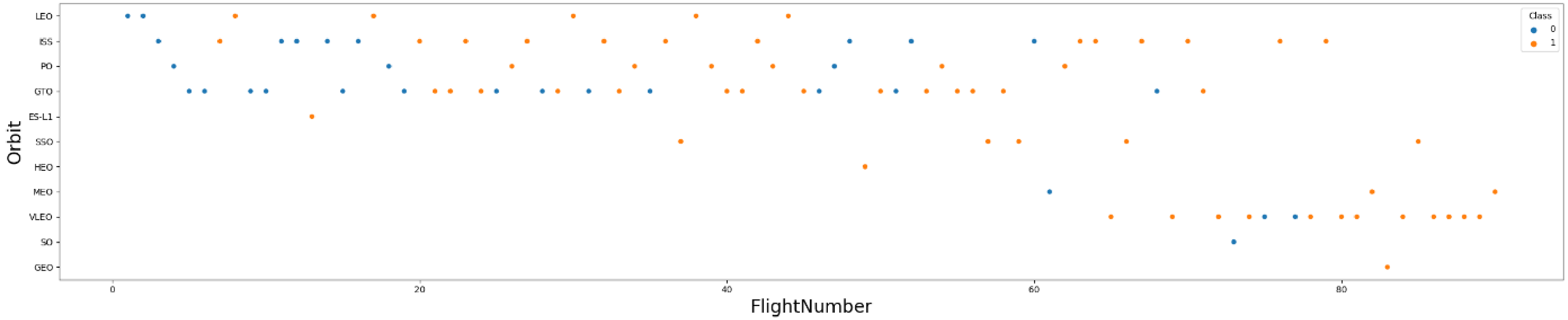
EDA: Orbit vs. Success rate

- ES-L1, GEO, HEO, SSO orbit have 100% success rates
- VLEO has second highest success rate with ~85%



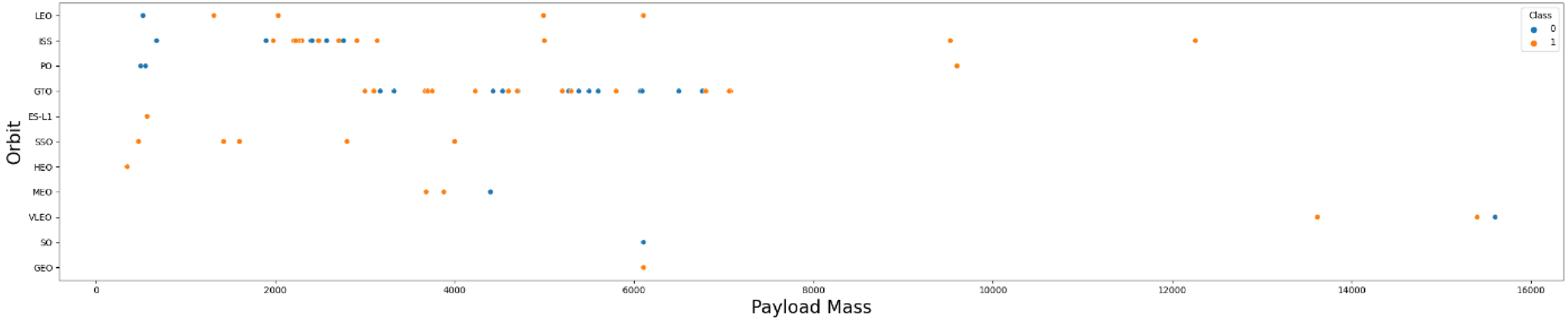
EDA: Flight number vs. Orbit Type

- LEO orbit Success appears related to the number of flights;
- No relationship between flight number when in GTO orbit.
- Initially the rockets were launched mostly to LEO, ISS, PO, GTO
- Only 1 launch to GEO orbit



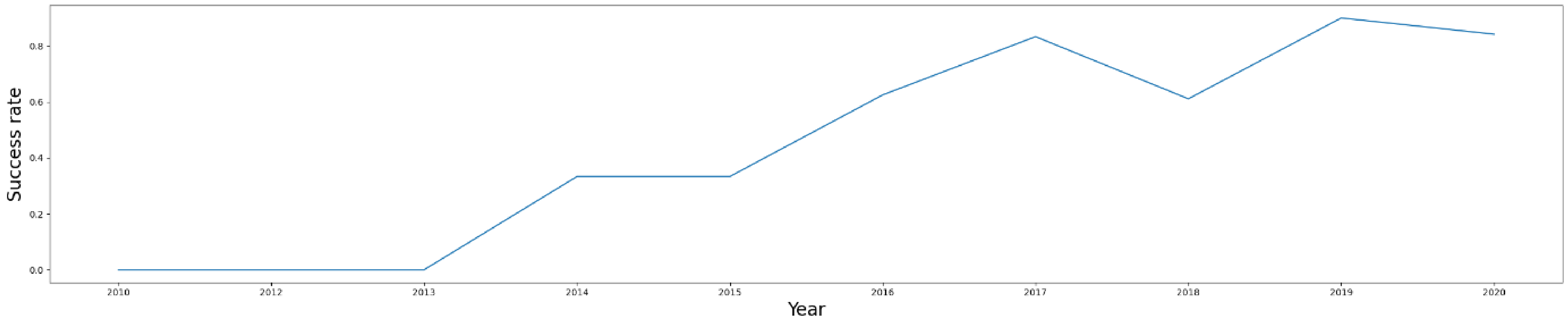
EDA: Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



EDA: Payload vs. Orbit Type

- For the first 3 years success rate was 0.
- Since 2013 it kept increasing till 2020.



EDA with SQL results slides

EDA w/ SQL: Task 1

- Display the names of the unique launch sites in the space mission

```
[15]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL

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

[15]:

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

EDA w/ SQL: Task 2

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

```
[16]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE "CCA%" LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[16]:

Date	Time(UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

EDA w/ SQL: Task 3

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

```
[17]: %sql SELECT CUSTOMER, SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'  GROUP BY CUSTOMER
```

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

[17]:

Customer	SUM(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

EDA w/ SQL: Task 4

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

```
[18]: %sql SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_version = 'F9 v1.1'  GROUP BY Booster_version

 * sqlite:///my_data1.db

Done.
```

[18]:

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

EDA w/ SQL: Task 5

- List the date when the first successful landing outcome in ground pad was achieved.

```
[20]: %sql SELECT MIN(DATE), Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'

 * sqlite:///my_data1.db

Done.

[20]:
```

MIN(DATE)	Landing_Outcome
01-05-2017	Success (ground pad)

EDA w/ SQL: Task 6

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[24]: %sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[24]:

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

EDA w/ SQL: Task 7

- List the total number of successful and failure mission outcomes

```
[25]: %sql SELECT Mission_Outcome, count(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome
 * sqlite:///my_data1.db
Done.
[25]: ,,,,,,,,,,,,,,,,,,,,,,,,,
```

Mission_Outcome	count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

EDA w/ SQL: Task 8

- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[30]: %sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
 * sqlite:///my_data1.db
Done.
[30]:
```

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

EDA w/ SQL: Task 9

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- *Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.*

```
[40]: %sql SELECT substr(Date, 4, 2), LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE substr(Date,7,4) = '2015' AND LANDING_OUTCOME = 'Failure (drone ship)'
```

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

Done.
```

[40]:

substr(Date, 4, 2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

EDA w/ SQL: Task 10

- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
[68]: %sql SELECT LANDING_OUTCOME, count(LANDING_OUTCOME) FROM SPACEXTBL  WHERE LANDING_OUTCOME = 'Success' c
```

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

[68]: ',,,,,,,,,,,,,,,,,,,,,'

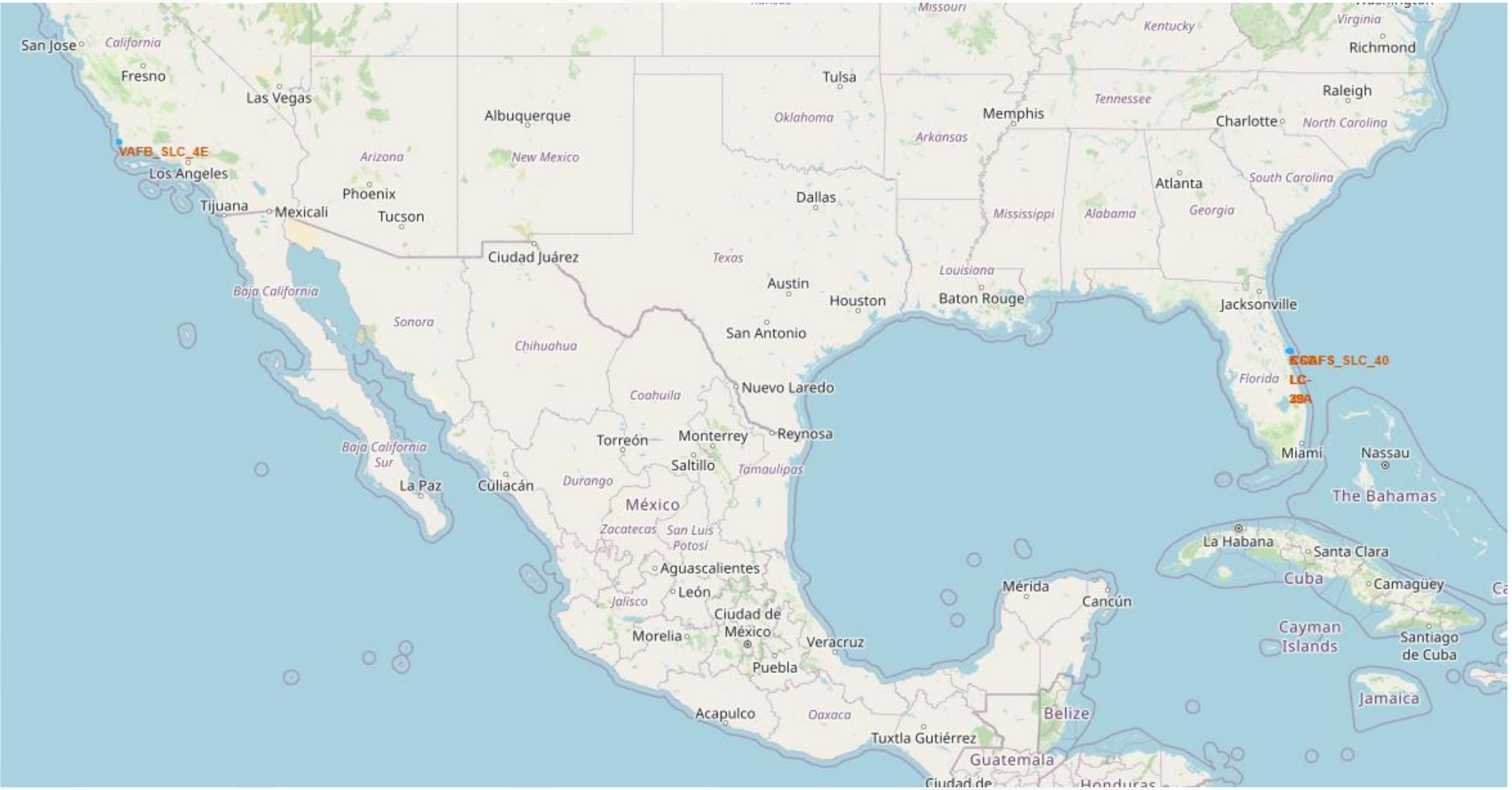
Landing_Outcome	count(LANDING_OUTCOME)
Success	38
Success (drone ship)	14
Success (ground pad)	6

Interactive maps with Folium
Proximity Analysis

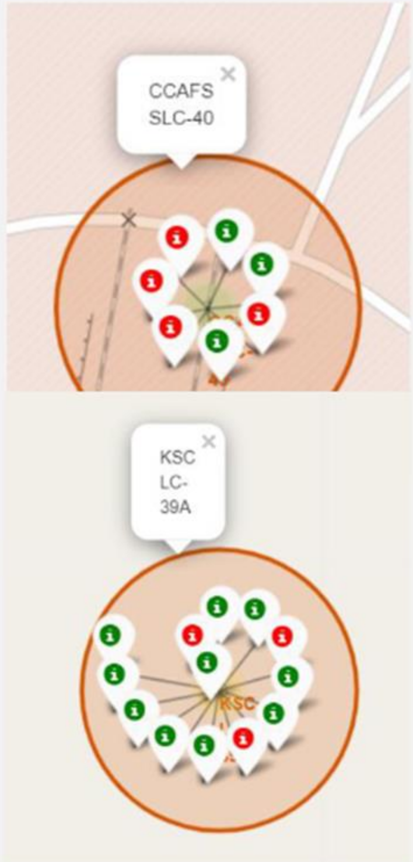
Launch Sites Location Analysis – Task 1

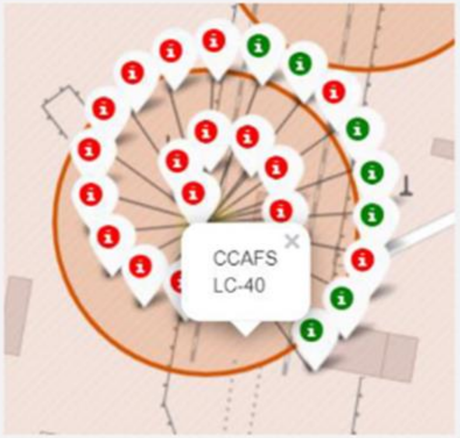
- All launch sites are in relative proximity to the Equator line.

- All launch sites are in very close proximity to the coast



Launch Sites Location Analysis – Task 2



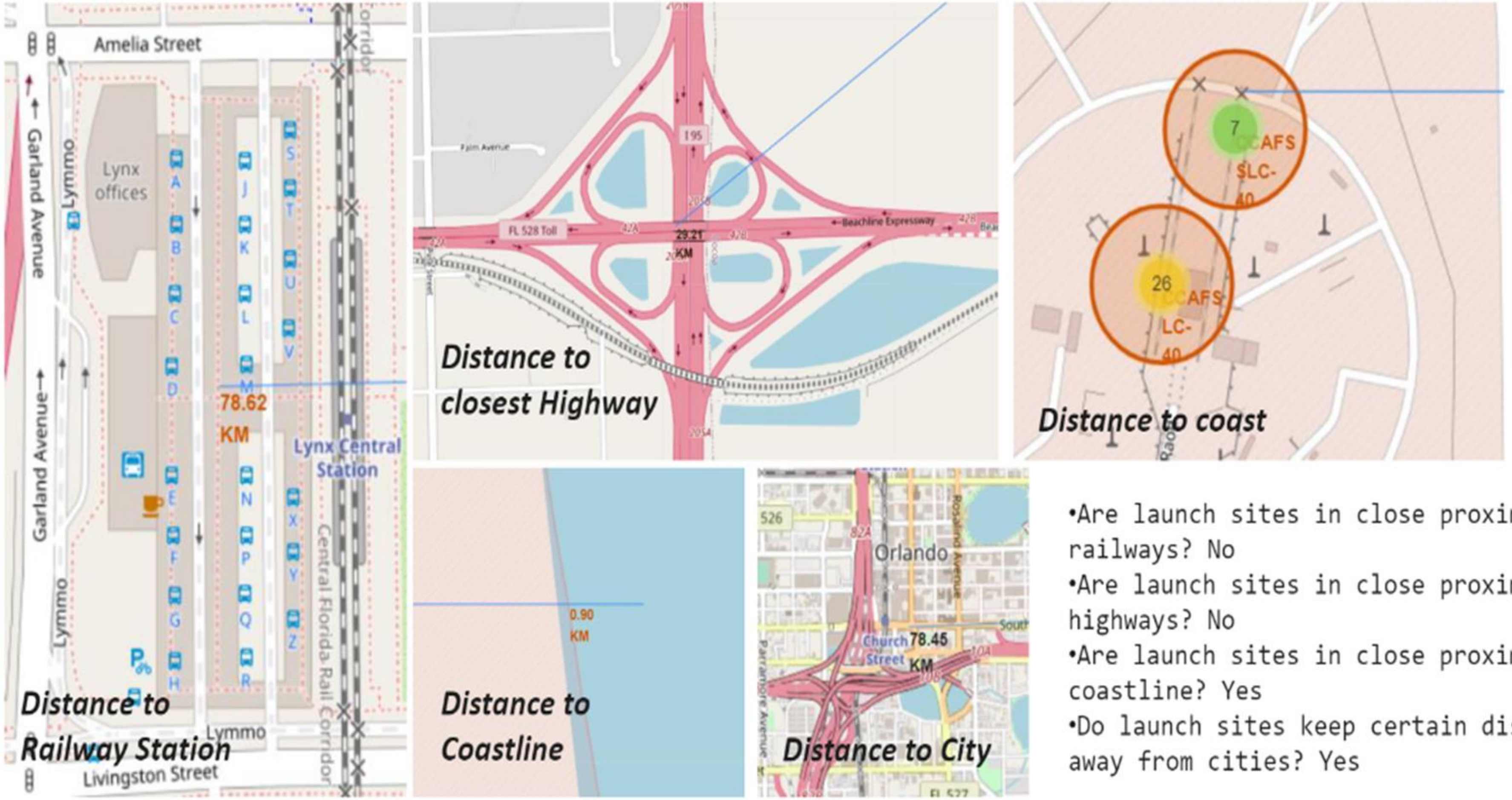


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

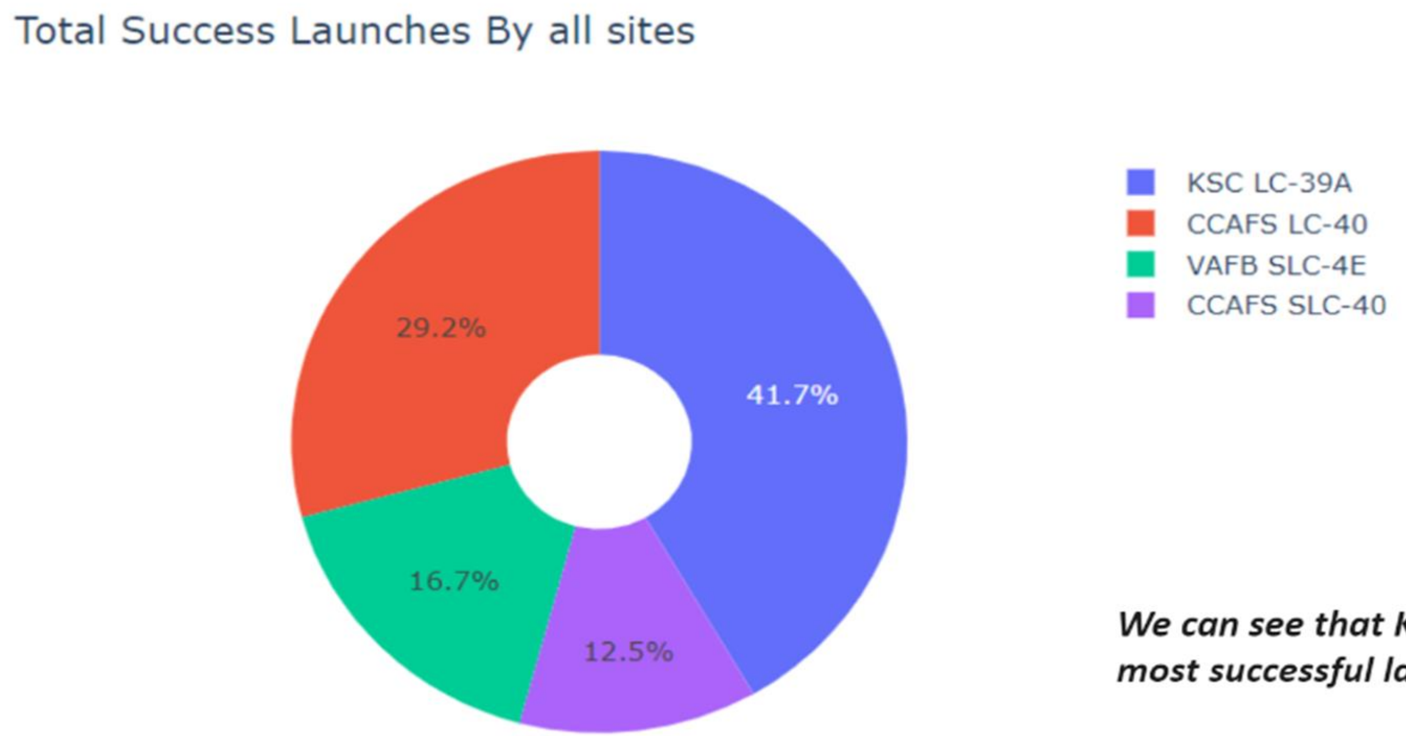
Launch Sites Location Analysis – Task 3



- 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

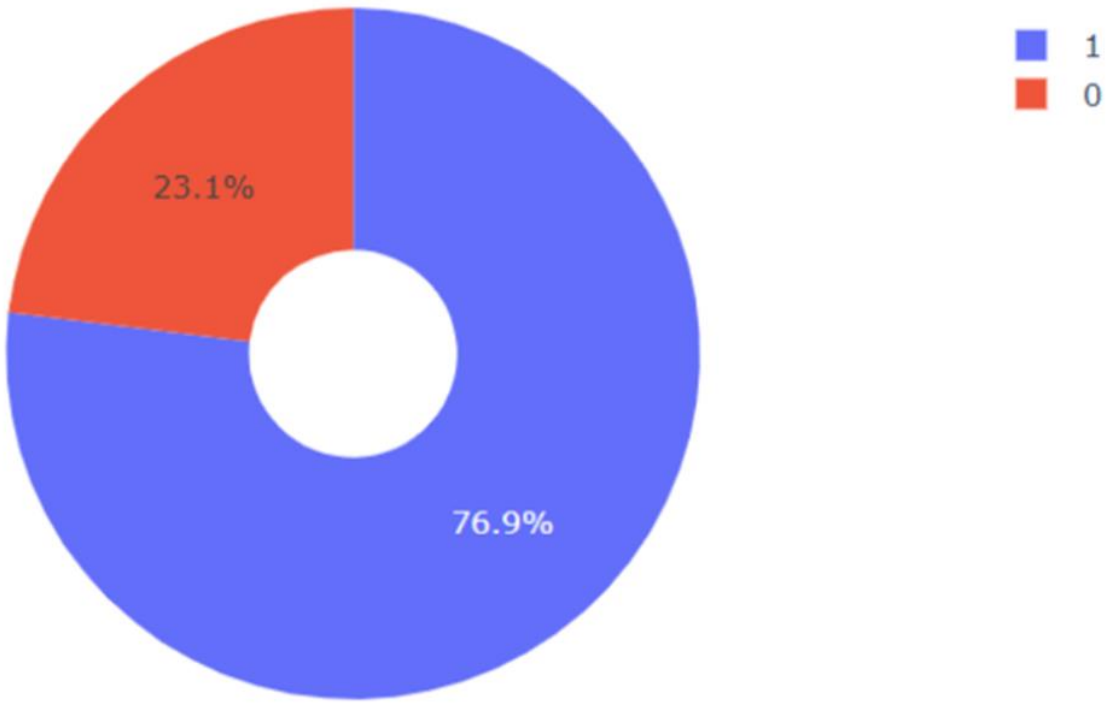
Interactive Dashboard

Pie chart showing the success percentage achieved by each launch site



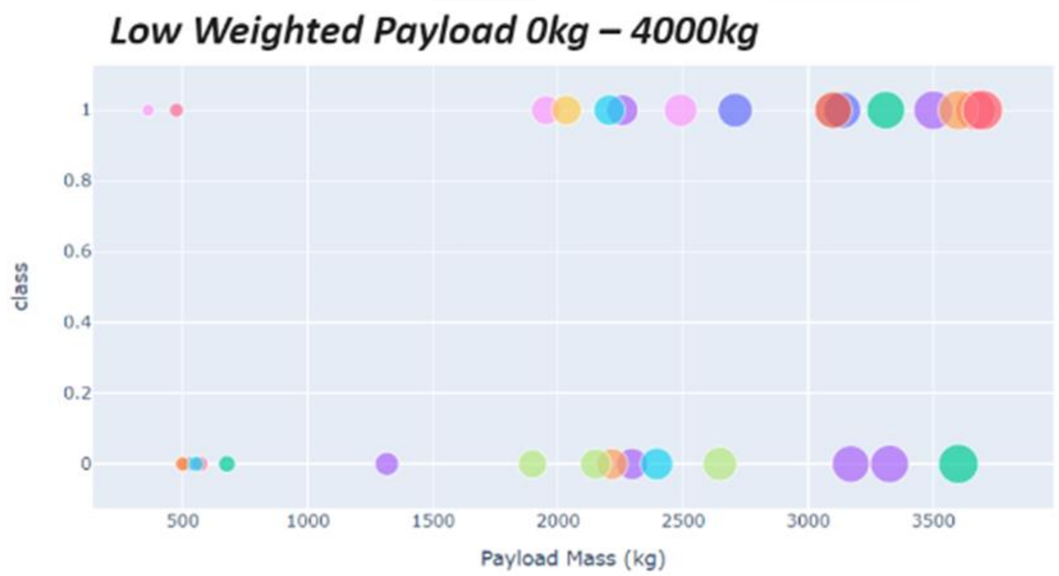
We can see that KSC LC-39A had the most successful launches from all the sites

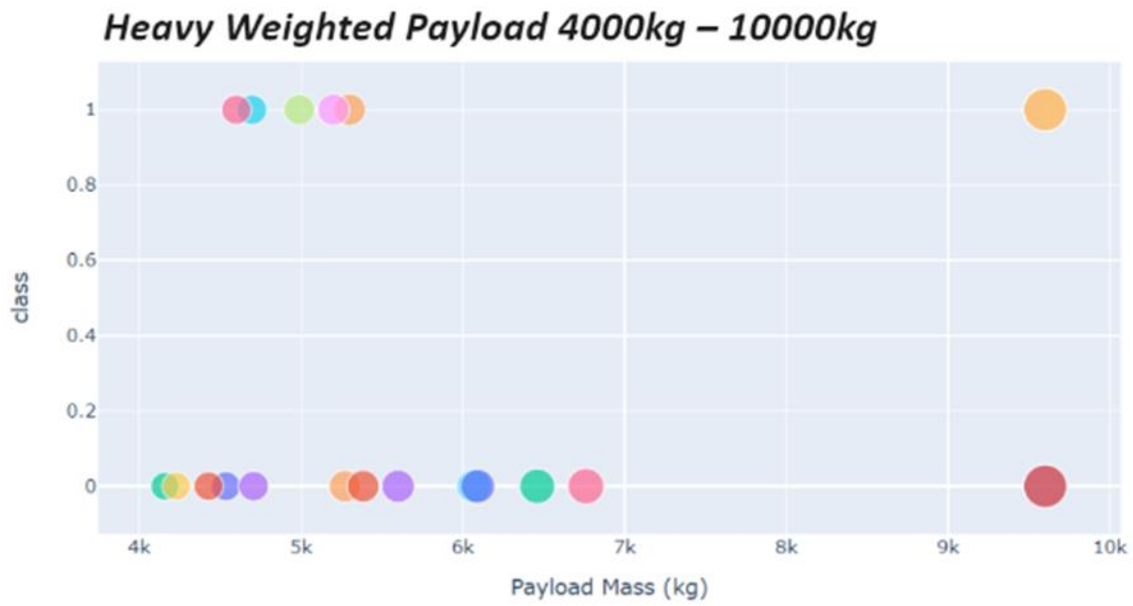
Pie chart showing the Launch site with the highest launch success ratio



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





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

Predictive Analysis

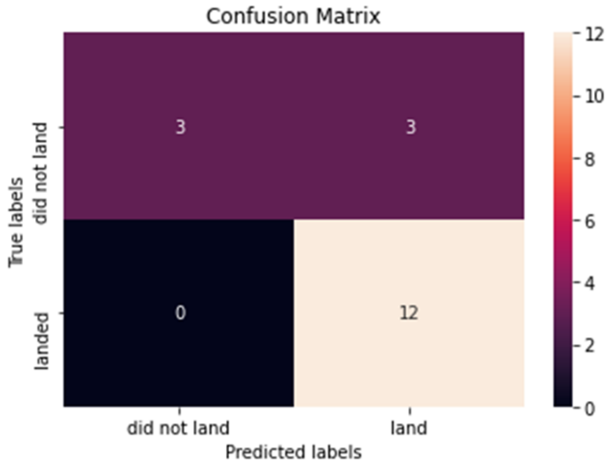
Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

Find the method performs best:

```
[32]: print("Decision Tree accuracy :",tree_cv.best_score_)
print("KNN accuracy:",knn_cv.best_score_)
print("SVM accuracy :",svm_cv.best_score_)
print("Logistic regression accuracy :",logreg_cv.best_score_)
```

```
Decision Tree accuracy : 0.8857142857142856
KNN accuracy: 0.8482142857142858
SVM accuracy : 0.8482142857142856
Logistic regression accuracy : 0.8464285714285713
```



Conclusions

- 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.