IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Meena Sawez
2/3/2023

Outline

Executive Summary

Summary of methodologies

- Data Collection through API and Web Scrapping
- Data Wrangling
- Exploratory Data Analysis with SQL and Data Visualization
- Interactive Map with Folium
- Dashboard with Plotly Dash
- Predictive Analysis using Machine Learning

Summary of all results

- Exploratory Data Analysis results
- Interactive Analytics
- Predictive Analysis

Introduction

Project background and context

- SpaceX 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 SpaceX 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 SpaceX for a rocket launch.

Problems you want to find answers

- What factors can determine if a rocket that has been launched can land successfully?
- What conditions need to be present to perform a successful landing of a launched rocket?

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One-hot encoding data fields for Machine Learning and cleaning of NULL values and irrelevant information
- 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 a get request to SpaceX API
  - The response was decoded as Json and turned into a Pandas Data Frame
  - The Data was then cleaned of any NULL and missing values
  - Data was also Scrapped from Wikipedia for Falcon 9 launches with Beautiful Soup

Data Collection – SpaceX API

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response.status_code
```

Out[10]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- We sent a request to the SpaceX API to collect the data and clean the requested data.
- The link to the notebook can be found here.

Data Collection - Scraping

- Scrapping the data from Wikipedia using Beautiful Soup to verify the correct table
- The link to the notebook can be found here.

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
data
```

Out[5]: <Response [200]>

Create a `BeautifulSoup` object from the HTML `response`

In [8]:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data.text)
```

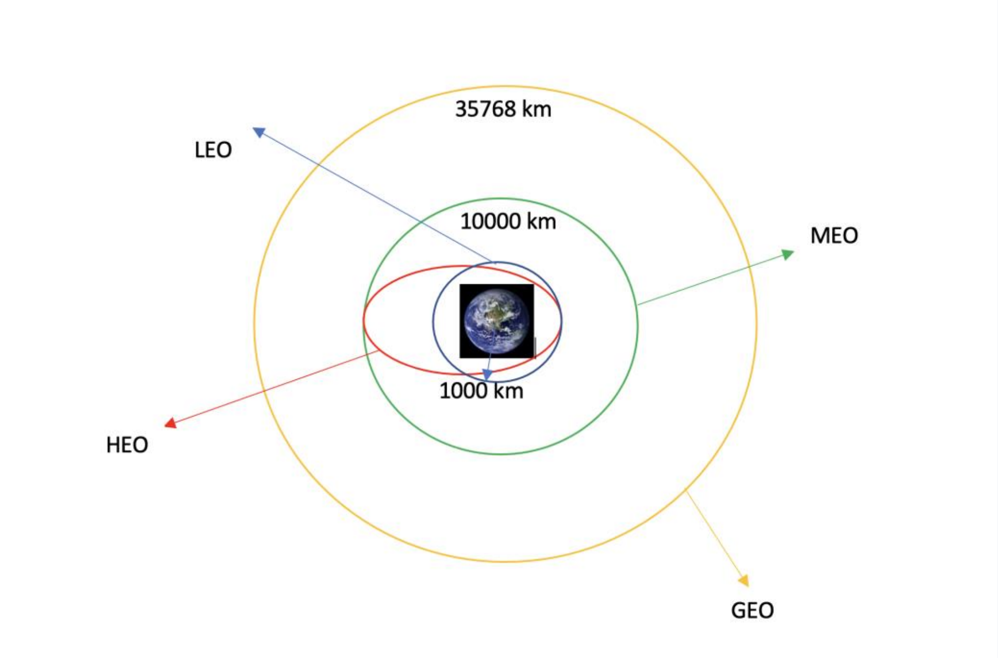
Print the page title to verify if the `BeautifulSoup` object was created properly

In [9]:

```
# Use soup.title attribute
soup.title
```

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

Data Wrangling



- Exploratory Data Analysis was applied to determine training labels
- Calculated the number of launches at each site, the number and occurrence of each orbit, and the landing outcomes were labeled based on successful landing or not
- The link to the notebook can be found here.

In [10]:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
liste = []

for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        liste.append(0)
    else:
        liste.append(1)

landing_class = liste
print(landing_class)
```

In [11]:

```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[11]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

In [6]:

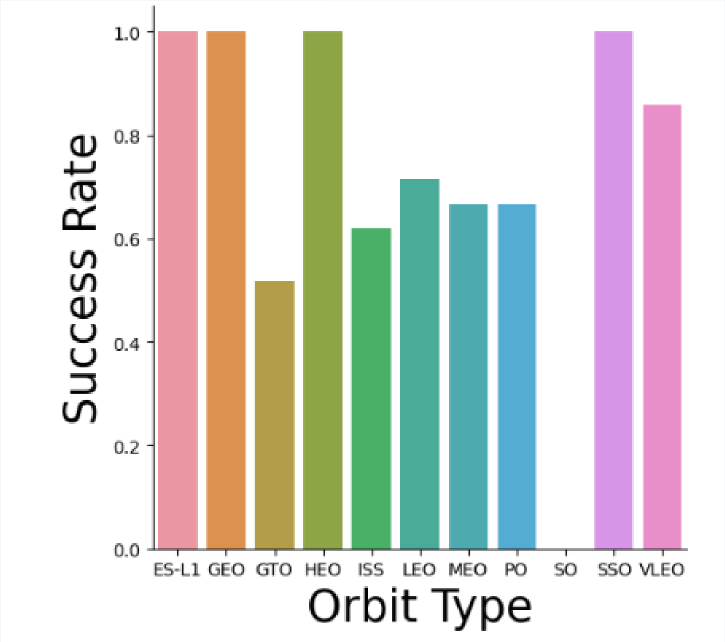
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

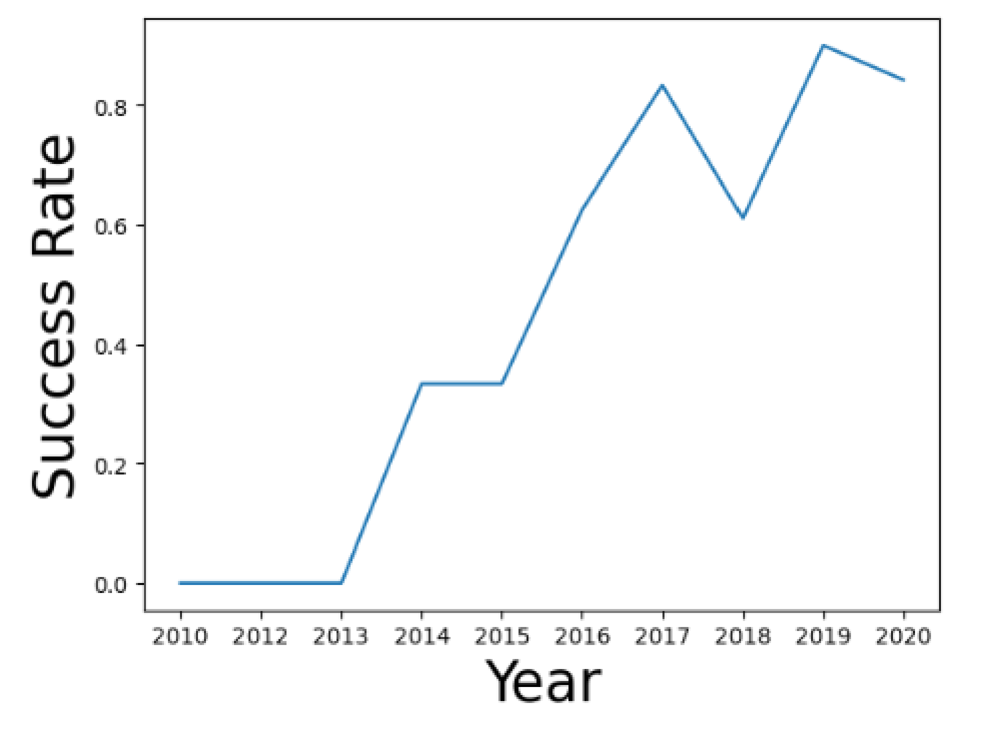
Out[6]:

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

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight numbers and their launch sites, payloads and launch sites, the success rates of each orbit, flight numbers and their orbit types, and the launch success yearly trend.
- The link to the notebook can be found here.





EDA with SQL

- Loaded the SpaceX dataset into PostgreSQL database without leaving Jupyter notebook.
- Applied EDA with SQL to get insight from data. Writing queries to figure out :
  - The names of unique launch sites in the Space Mission
  - The total payload mass carried by boosters launched by NASA
  - 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 ships, their booster versions, and launch site names.
- The link to the notebook can be found here.

Build an Interactive Map with Folium

- Marking all launch sites and adding map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the Folium Map.
- Assigning the feature launch outcomes to class 0 and 1 [Successes = 1 or Failures = 0].
- Using the color-labeled marker clusters, identifying which launch sites have relatively high success rates.
- Calculating the distances between launch sites to their proximities to answer the following questions:
  - Are launch sites near railways, highways and coastline?
  - Do launch sites keep certain distances away from cities?
- The link to the notebook can be found here.

Build a Dashboard with Plotly Dash

- Building an interactive dashboard with Plotly Dash
- Plotting pie Charts showing the total launches by certain sites
- Plotting Scatter graph showing the Relationship with Outcome and Payload Mass (kg) for the different booster versions.
- The link to the notebook can be found here.

Predictive Analysis (Classification)

- Loading the data using NumPy and Pandas, transforming the data, Split the Data into training ad testing.
- Building different machine learning models and tune different hyperparameters using GridSearchCV.
- Using accuracy as the metric for the model, improved the model using feature engineering and algorithm tuning.
- Finding the best performing classification model.
- The link to the notebook can be found here.

Results

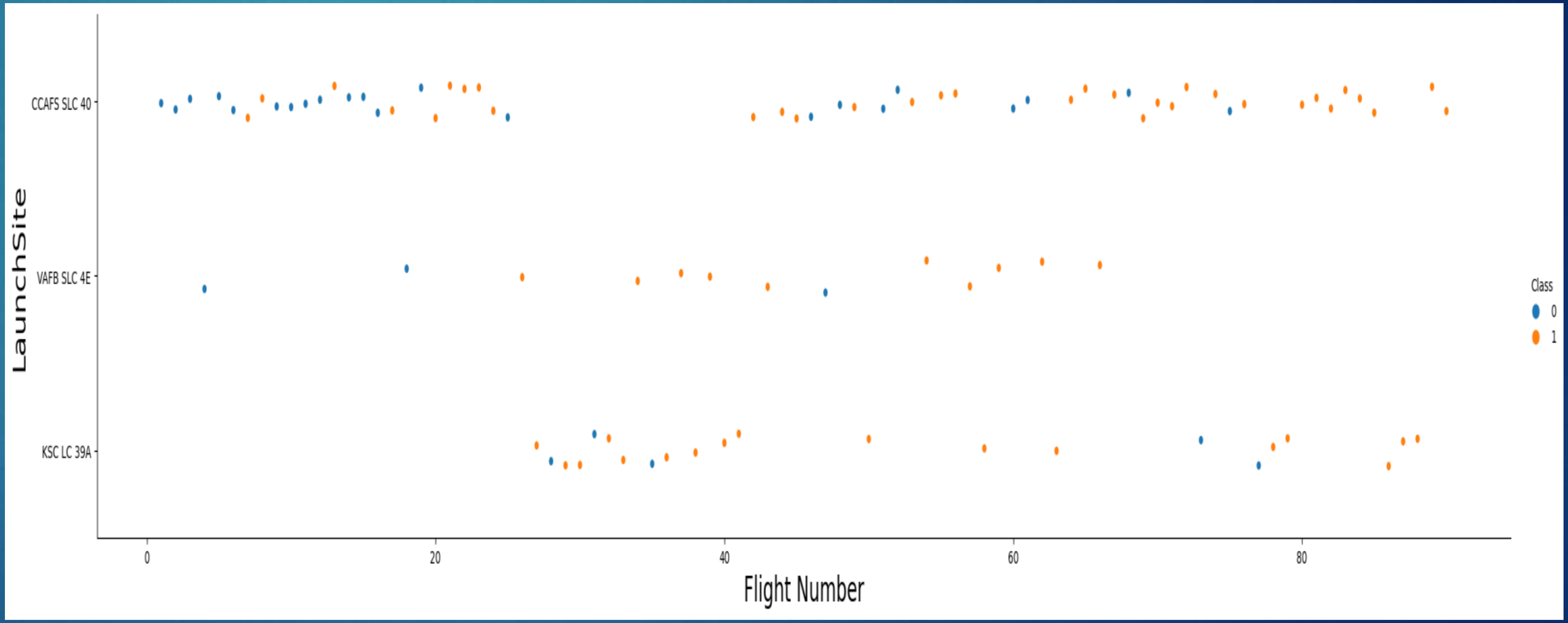
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

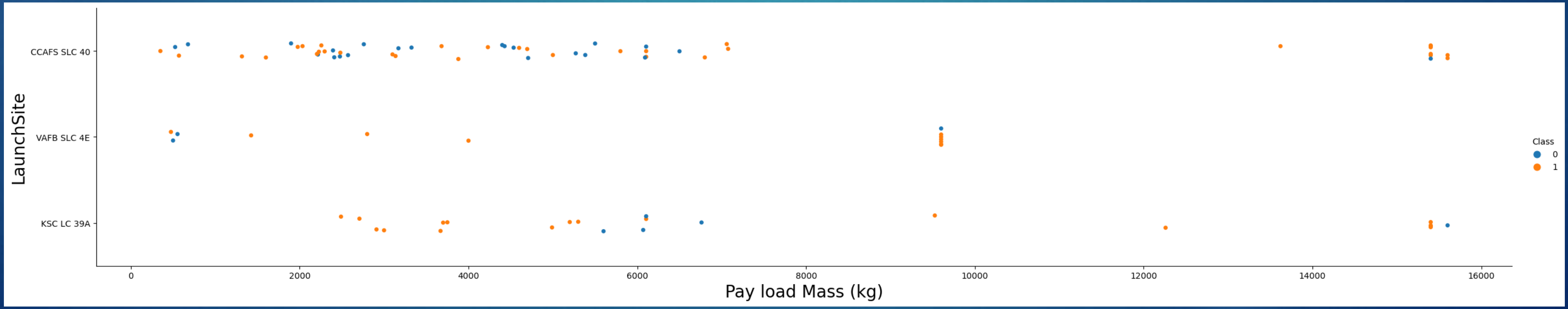
Flight Number vs. Launch Site

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



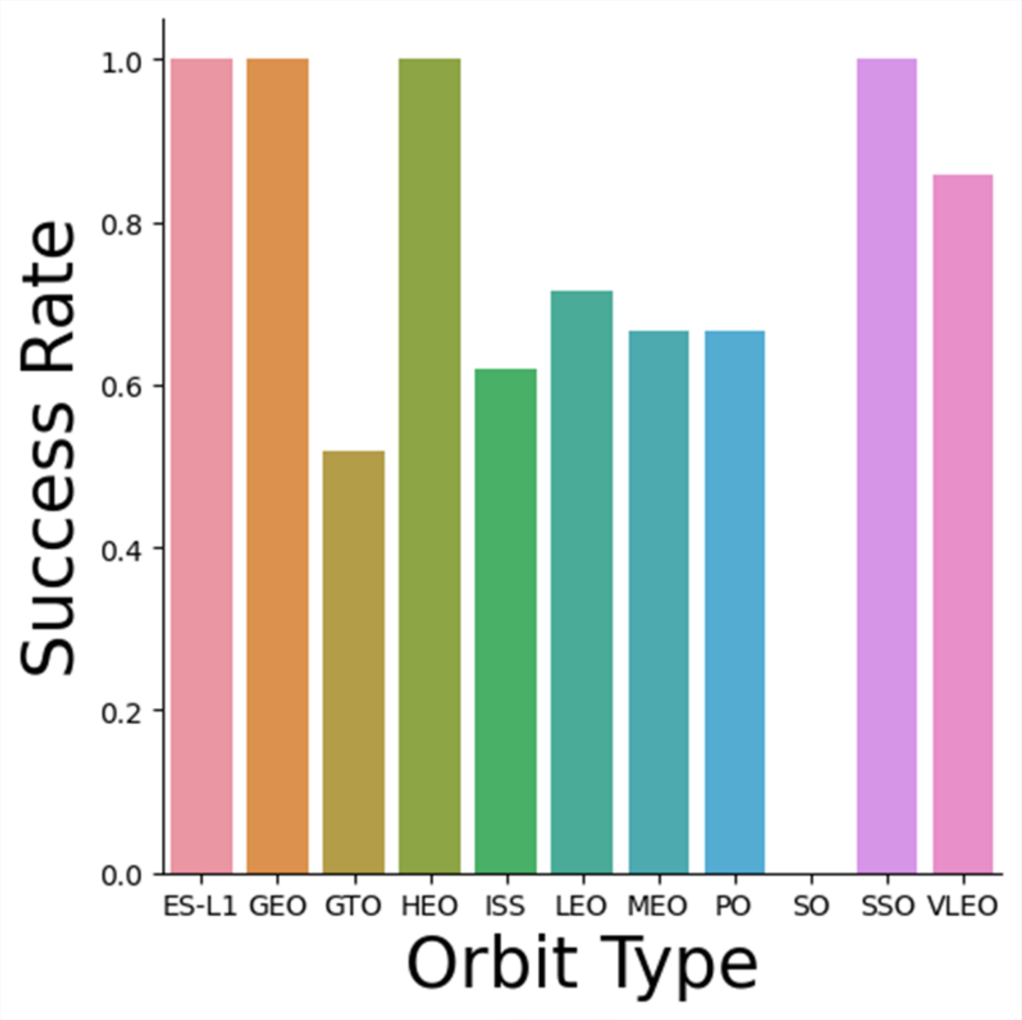
Payload vs. Launch Site

- The greater the payload, the high the success rate for the rocket
- In this case, CCAPS SLC 40 has a greater success rate with a high payload



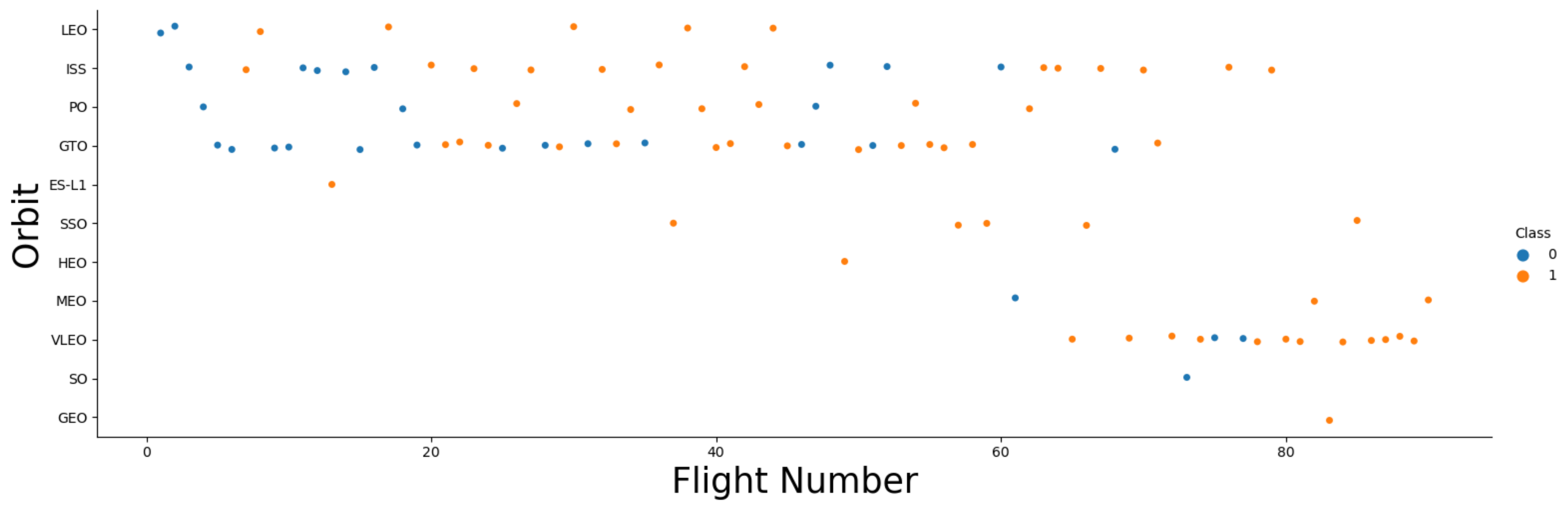
Success Rate vs. Orbit Type

- There are certain orbit types that have a high success rate such as SSO, HEO, GEO, and ES-L1



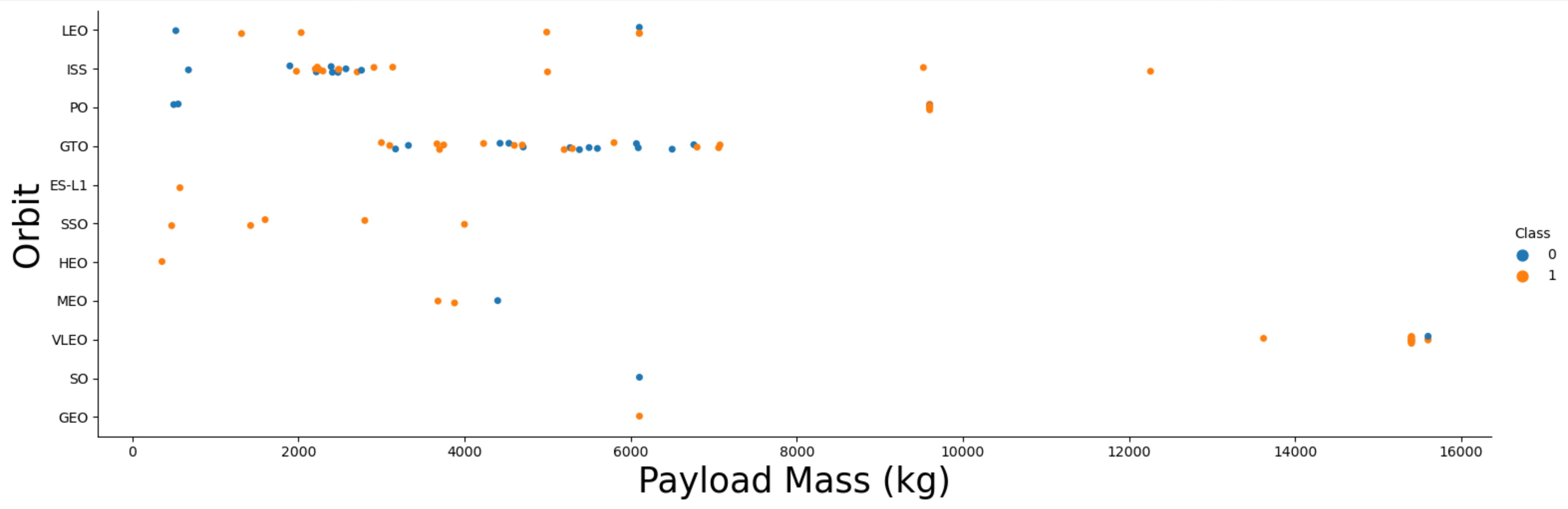
Flight Number vs. Orbit Type

- There is no clear relationship between the flight number and the type of Orbit



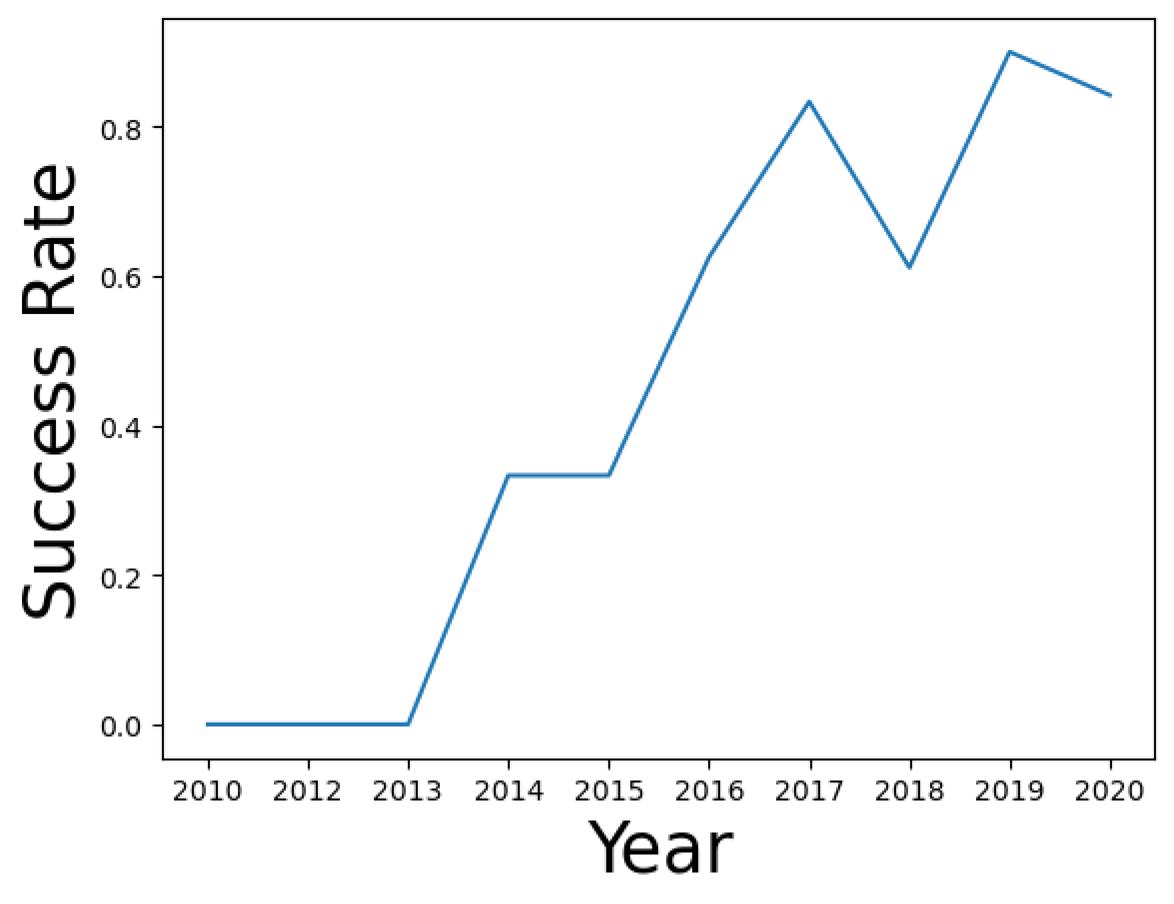
Payload vs. Orbit Type

- The heavier the payload, the more successful landings are for PO, LEO, and ISS orbits.



Launch Success Yearly Trend

- The success rate of launches per year has steadily increased until 2020



All Launch Site Names

- The different launch sites from SpaceX data

In [10]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[10]:

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

Launch Site Names Begin with 'CCA'

- 5 records of launch sites beginning with CCA

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

```
%sql select launch_site from SPACEXTBL where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- 48,213 Kg is the total payload mass launched by NASA (CRS)

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3
Done.
```

1
48213

Average Payload Mass by F9 v1.1

- 2534 Kg is the average payload of a F9 v1.1 booster

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE '%F9 v1.1%'
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.datab
Done.
```

1
2534

First Successful Ground Landing Date

- December 22, 2015 is when the first successful landing on a ground pad happened.

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success (ground pad)'
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.data
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Booster versions of Successful Drone Ships

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success (drone ship)' AND (PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000)
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/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

- How many total number of launches were Successful or Failures

List the total number of successful and failure mission outcomes

```
: %sql SELECT COUNT(*)  AS Success FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%'
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

12]:

success
100

```
: %sql SELECT COUNT(*)  AS Failure FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Failure%'
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

13]:

failure

Boosters Carried Maximum Payload

- Booster versions that have carried the highest amount of payload

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3245
Done.
```

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

2015 Launch Records

- Landing outcomes that were failures in 2015 as well as their Booster Version and launch site

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, DATE FROM SPACEXTBL WHERE (DATE>'2015-01-01' AND DATE <'2016-01-01') AND LANDING__OUTCOME
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- The following outcomes display the landing outcomes of each launch between June 4, 2010 and March 20, 2017 and Ranks them based how many of the landing outcomes were attempted.

Rank 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

```
%sql SELECT LANDING__OUTCOME, COUNT(*) as TOTALS, RANK() OVER (ORDER BY COUNT(*) DESC) as Total_Rank FROM SPACEXTBL WHERE (DATE>'2010-06-04' AN
```

```
 * ibm_db_sa://zmz12664:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

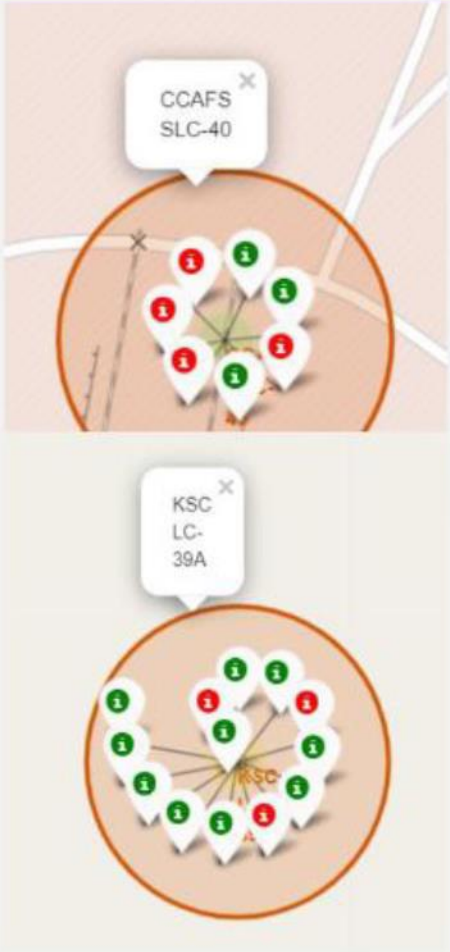
landing__outcome	totals	total_rank
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	1	7
Precluded (drone ship)	1	7

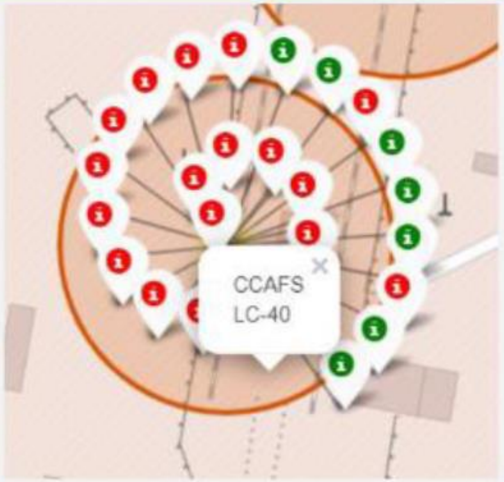
Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels





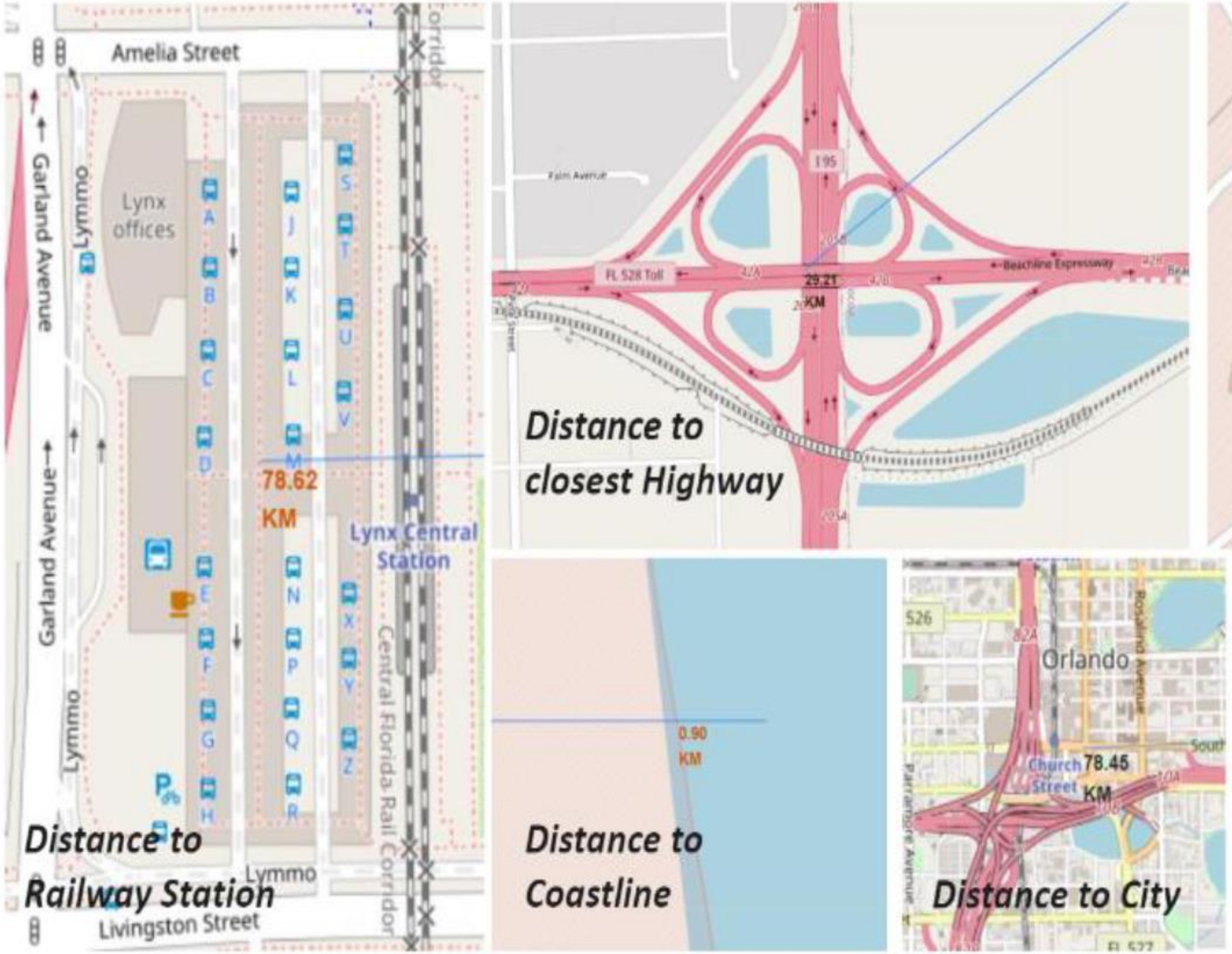
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

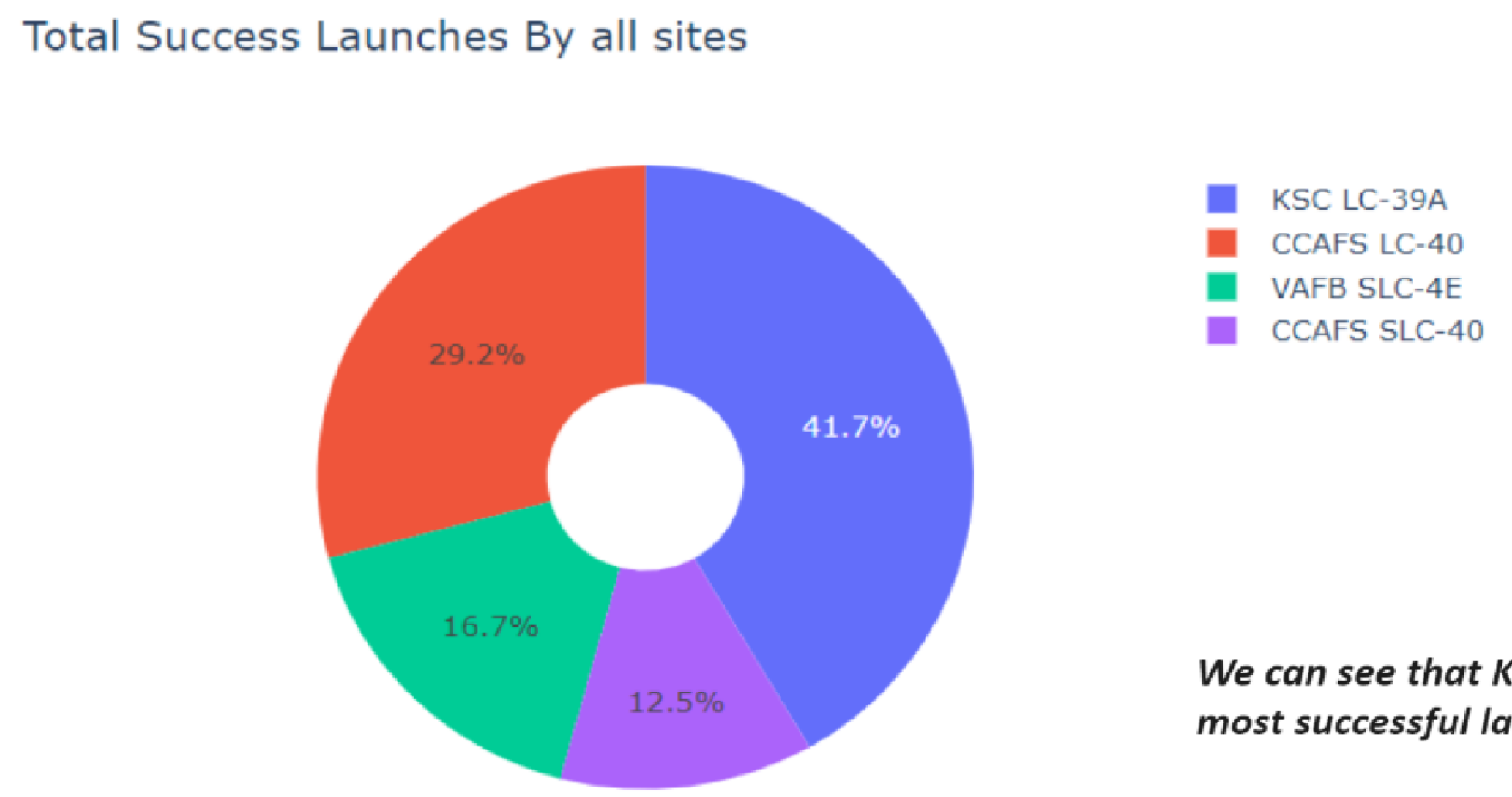


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

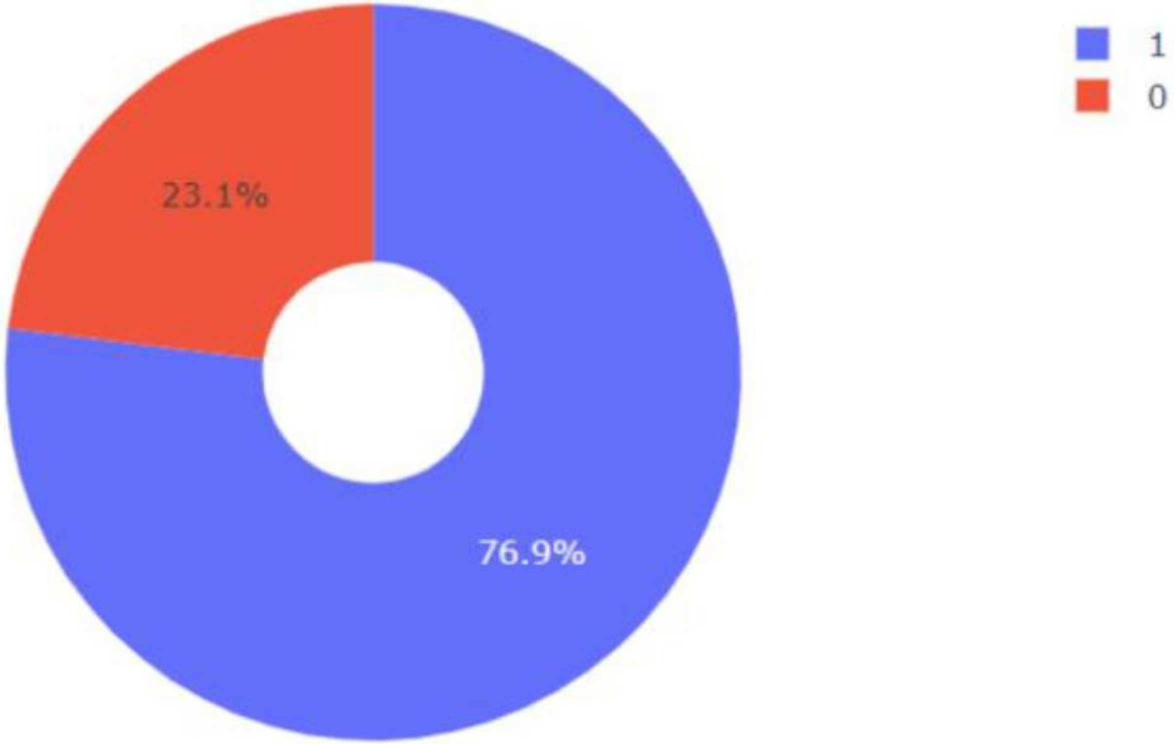
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



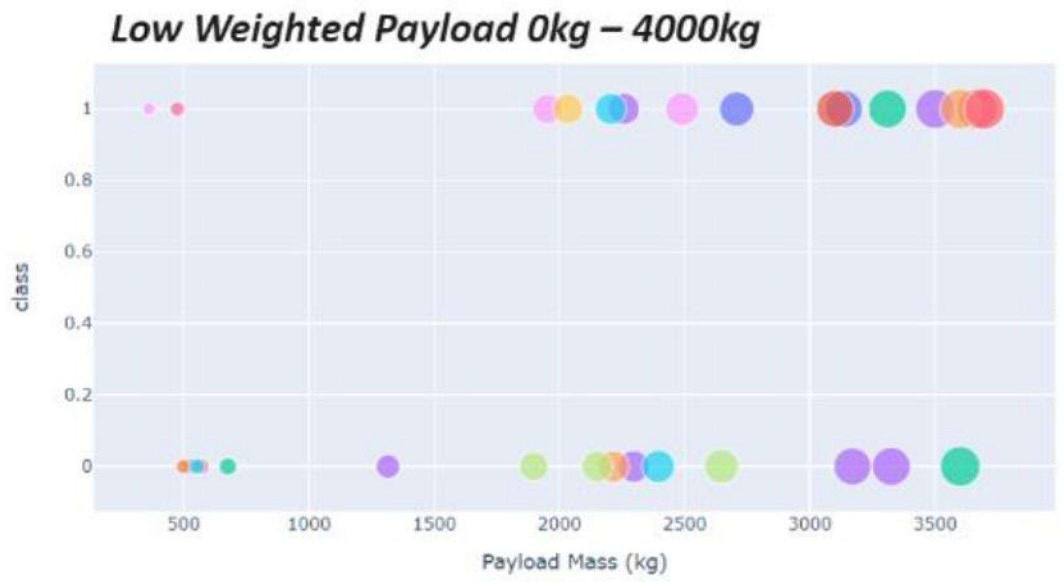
We can see that KSC LC-39A had the most successful launches from all the sites

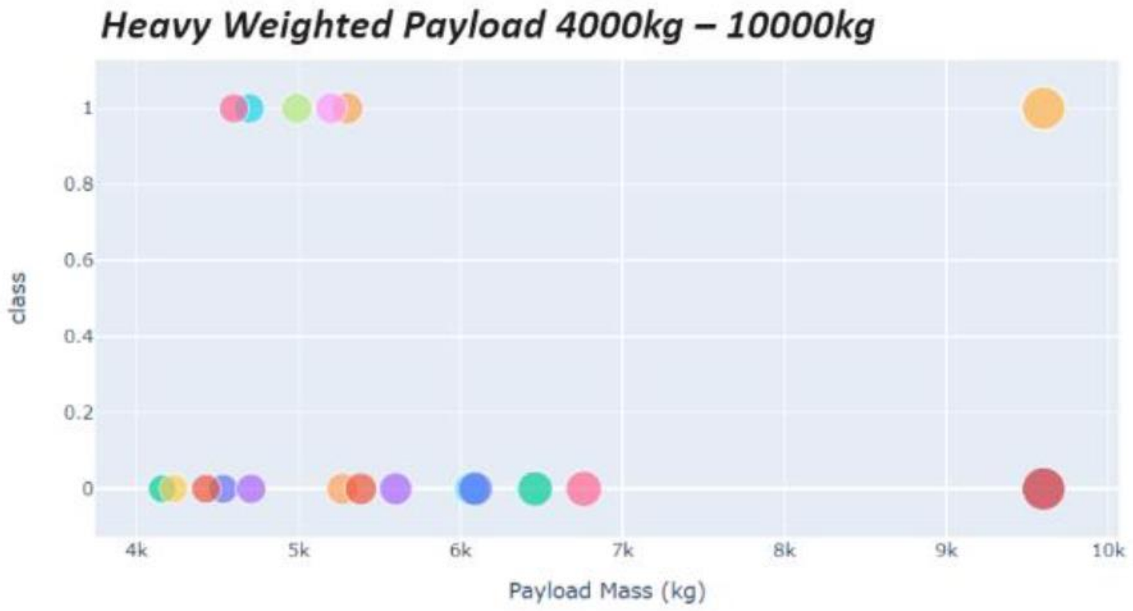
Pie chart showing the Launch site with the highest launch success ratio



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

Find the method performs best:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree Decision':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 Parameters :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Parameters :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Parameters :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree Decision with a score of 0.8892857142857145
```

Confusion Matrix

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

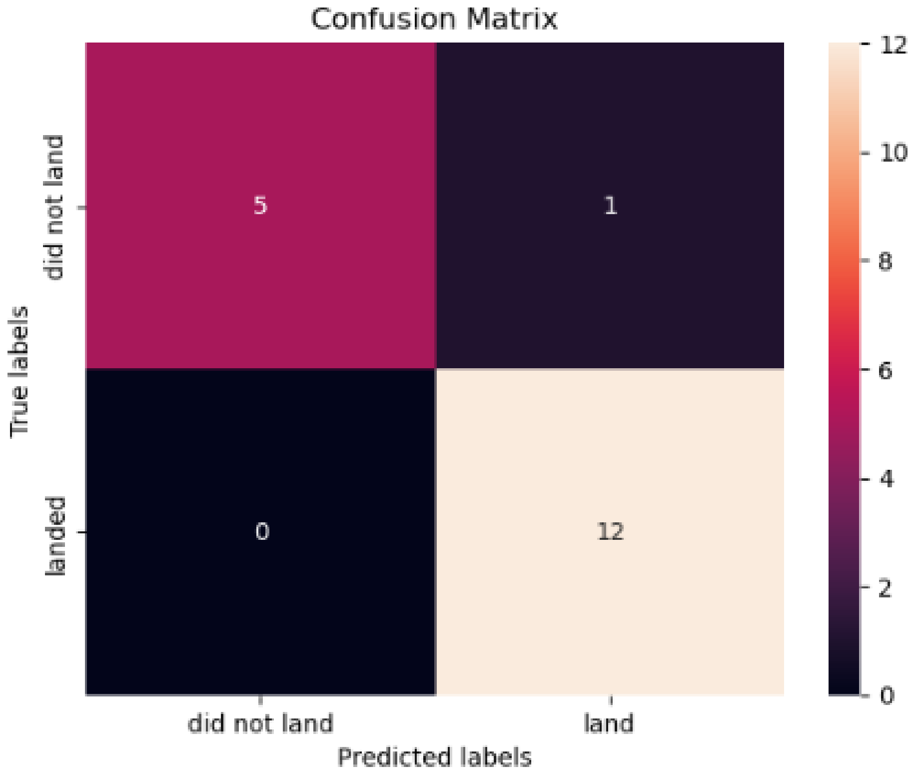
Calculate the accuracy of tree_cv on the test data using the method `score` :

```
print('Accuracy = ', tree_cv.score(X_test, Y_test))
```

```
Accuracy =  0.9444444444444444
```

We can plot the confusion matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Given all the information we can conclude:
  - The larger the payload at a launch site, the greater the success rate at a launch site.
  - Launch success rate started to increase from 2013 until 2020
  - Orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates
  - KSC LC-39A had the largest number of successful launches of any site
  - The decider tree classifier is the best machine learning algorithm for this task

Thank you!