IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nataliia Ocheretnia
January, 18
2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

The era of space travel has already arrived! Keeping up with the times, the Space U company was created. The company's founders have a noble and achievable goal - to make space travel accessible to everyone!

Summary of methodologies

- Data collection from API and Web scraping.
- Data wrangling
- EDA (Exploratory Data Analysis) with SQL, Pandas, Matplotlib.
- Visual Analytics with Folium and Plotly Dash.
- Predictive analysis

Summary of all results

- Choosing the best predictive modeling algorithm between Logistic regression, SVM, Decision tree and KNN classifiers.
- The best method for successful landing prediction

Introduction

The commercial space age is here, companies are making space travel affordable for everyone.
Space Y will complete the commercial space race.
The company can save millions in every launch! One reason Space Y can do this is our the rocket launches are relatively inexpensive. Falcon 9 rocket launches cost of 62 million dollars; other providers cost upwards of 165 million dollars each.

Our goal is to determine if the first stage of our rocket will land successfully using Data Science and Machine learning models.

Because, if we can determine if the first stage will land, we can determine the cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We gathered our data for predictive analysis from the SpaceX with REST API and Web scraping wiki pages
- Perform data wrangling
  - We used the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records, after that the Data was converted into a Pandas Dataframe for analysis and visualization.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

Data Collection

- Various methods were used in data collection process:
  - “Get” request to SpeceX API for collection row data
  - “.json()” function for decoding response content into pandas DF and ”.json_normalize()” for normalizations
  - Then data was cleaned and checked for missing values and supplemented where nesessary.
  - BeautafulSoup were used to perform web scraped pages from Wikipedia for Falcon 9 launch records
  - the objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for analysis.

Data Collection – SpaceX API

- The Get request to the SpeceX API were used to collect, clean data and made some basic data wrangling and formatting

- Data collection API notebook https://github.com/NataliiaOcheretnia/DataLearning/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

```
# Use json_normalize method to convert the json result into a dataframe

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

```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

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

```
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 - Scraping

- BeautifulSoup was used for web scraping Falcon 9 launch records.
- Table was parsed and converted into a pandas dataframe.

- The web scraping notebook https://github.com/NataliiaOcheretnia/DataLearning/blob/main/WebScraping_Review_Lab%20(2).ipynb

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

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

```
# 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
```

```
200
```

2. Create a BeautifulSoup object from the HTML response

```
# 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

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

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

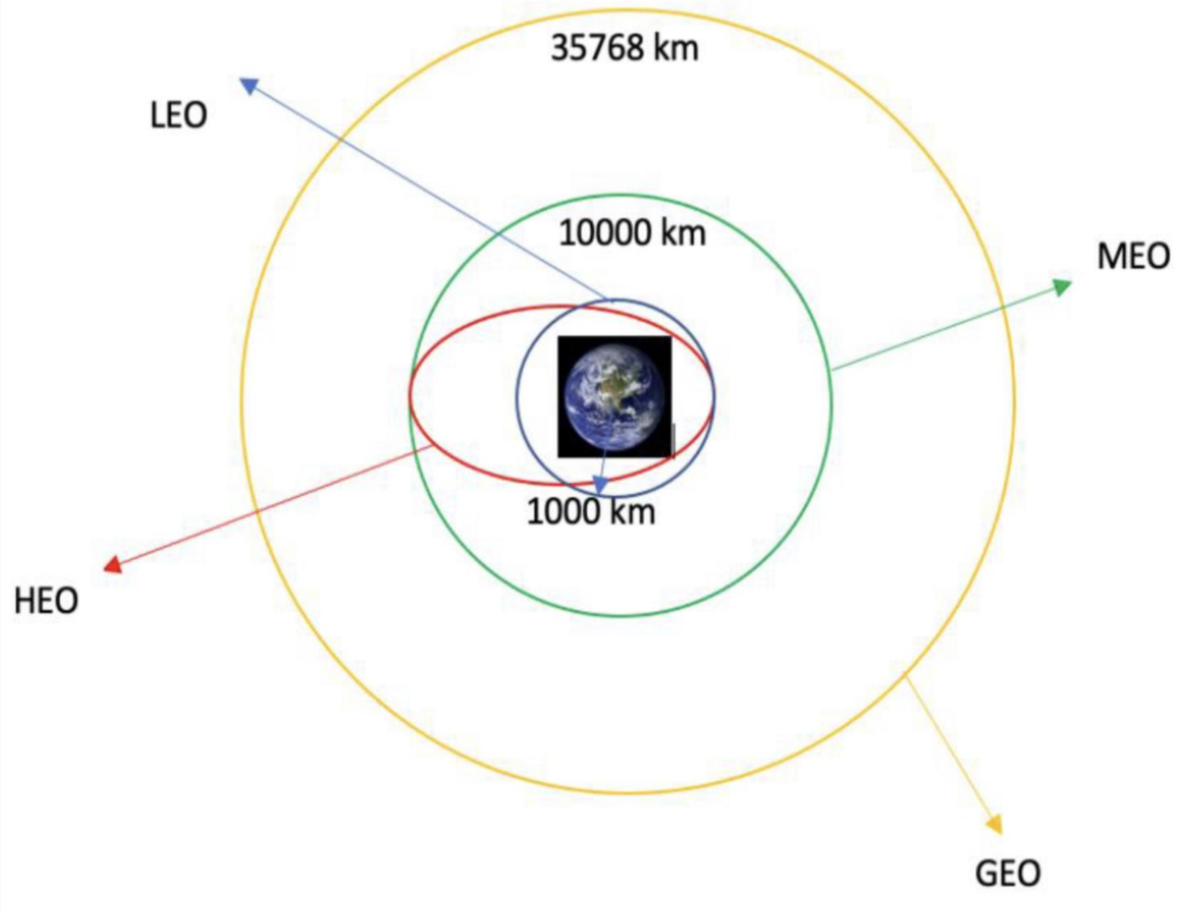
3. Extract all column names from the HTML table header

```
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
```

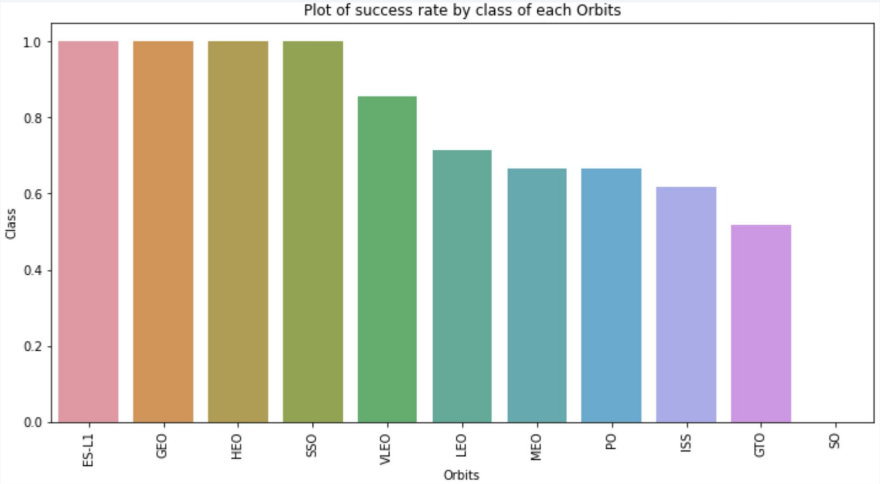
4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

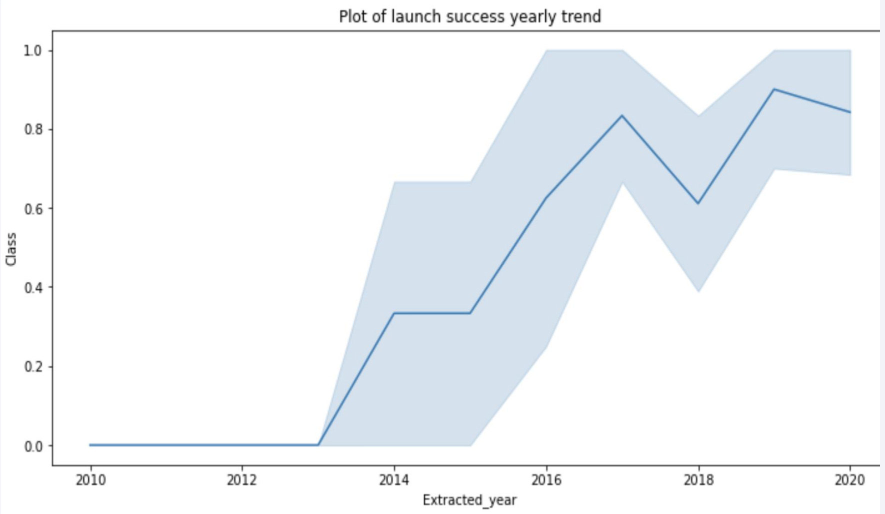
Data Wrangling

- 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.
- The Web scraping notebook https://github.com/NataliiaOcheretnia/DataLearning/blob/main/WebScraping_Review_Lab%20(2).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.

- EDA https://github.com/NataliiaOcheretnia/DataLearning/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- The SpaceX dataset was load into a PostgreSQL database without leaving the jupyter notebook.
- EDA was applied 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.

The link

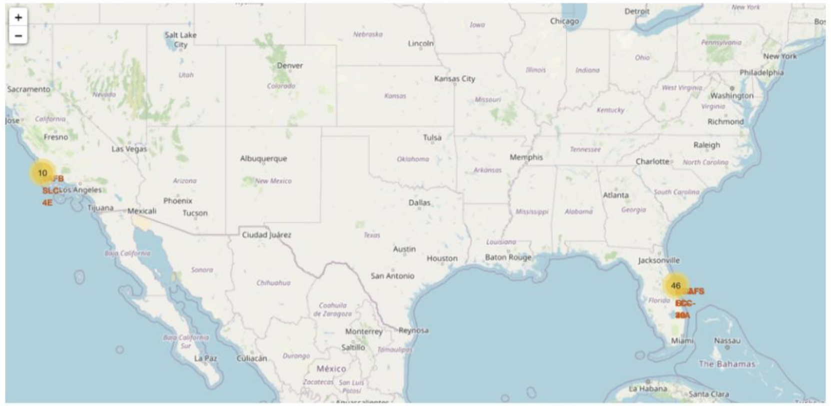
Build an Interactive Map with Folium

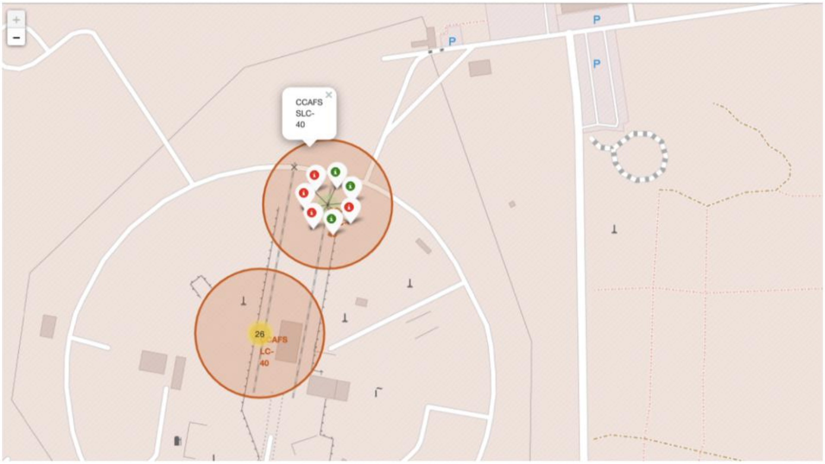
- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled 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 launchsites near railways,highways and coastlines?

  - Do launch sites keep certain distance away from cities?

  [Launch sites location with Folium]

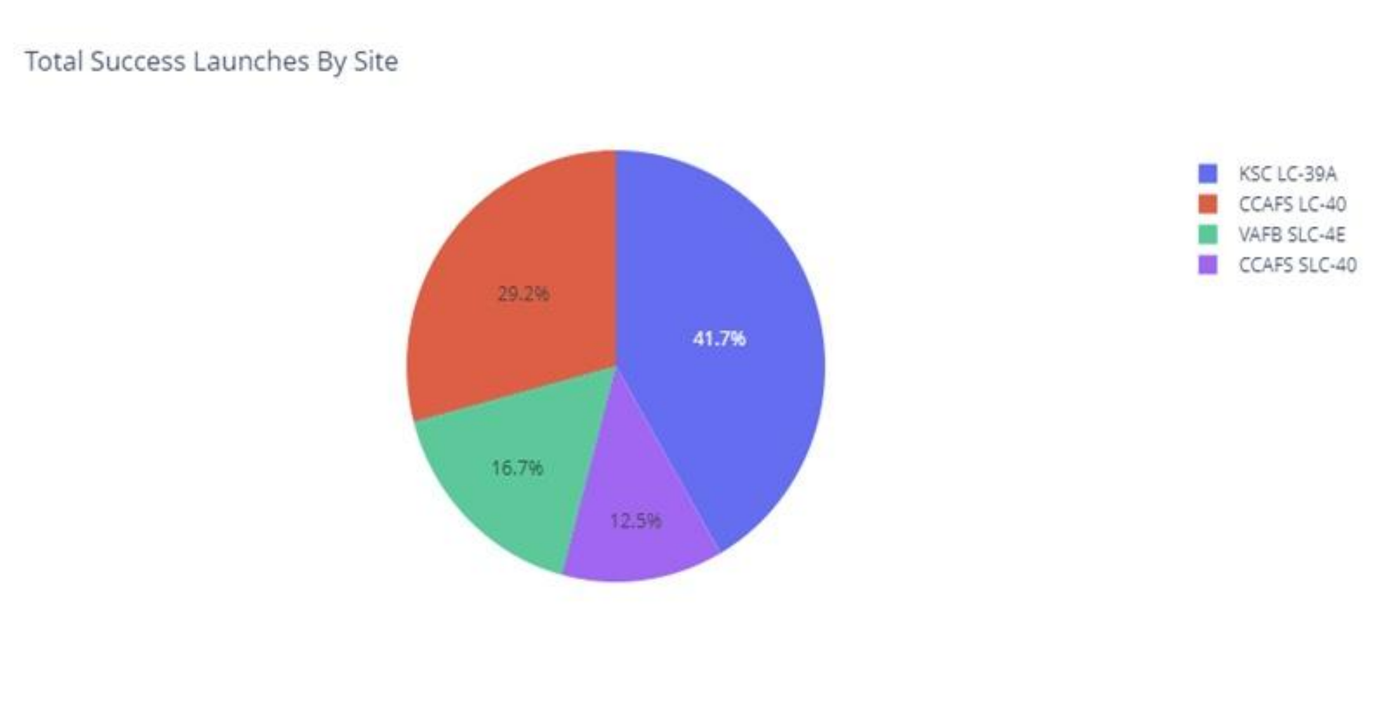




Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash. This dashboard application contains input components such as dropdown list and a range slider to interact with a pie chart and a scatter point chart.

- 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 [Inteactive Dashboard with ploty Dash]()



Predictive Analysis (Classification)

Summary of the model development process used to predict if the first stage will land successful.

- 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 for Logistic Regression, SVM, Decision Tree and KNN.
- 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.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

[Machine Learning Prediction notebook]

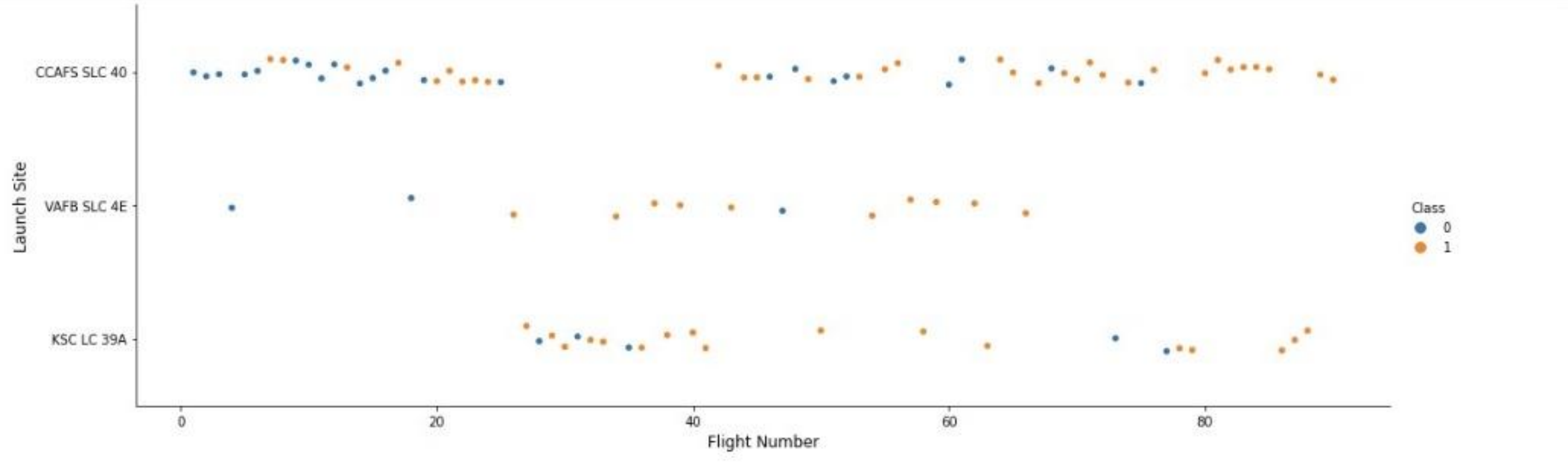
Results

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

Section 2

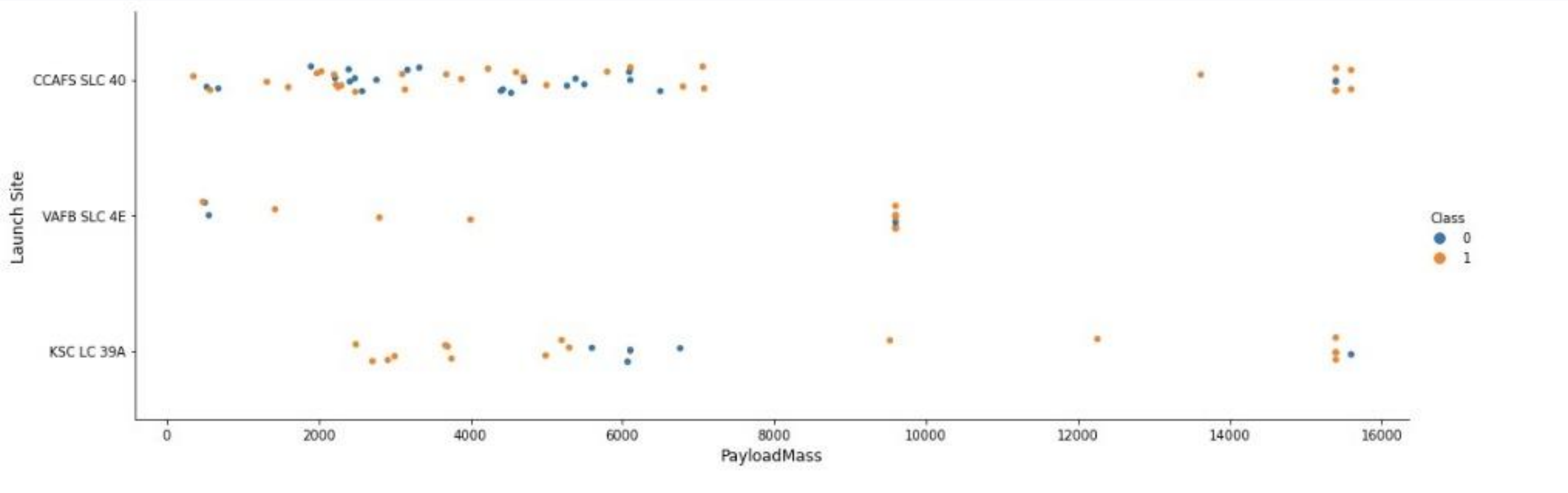
Insights drawn from EDA

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. Especially for CCAFS SLC 40, where majority launches are concentrated.
- VAFB SLC 4E and KSC LC 39A has higher successful rate which represents one third of the total launches.

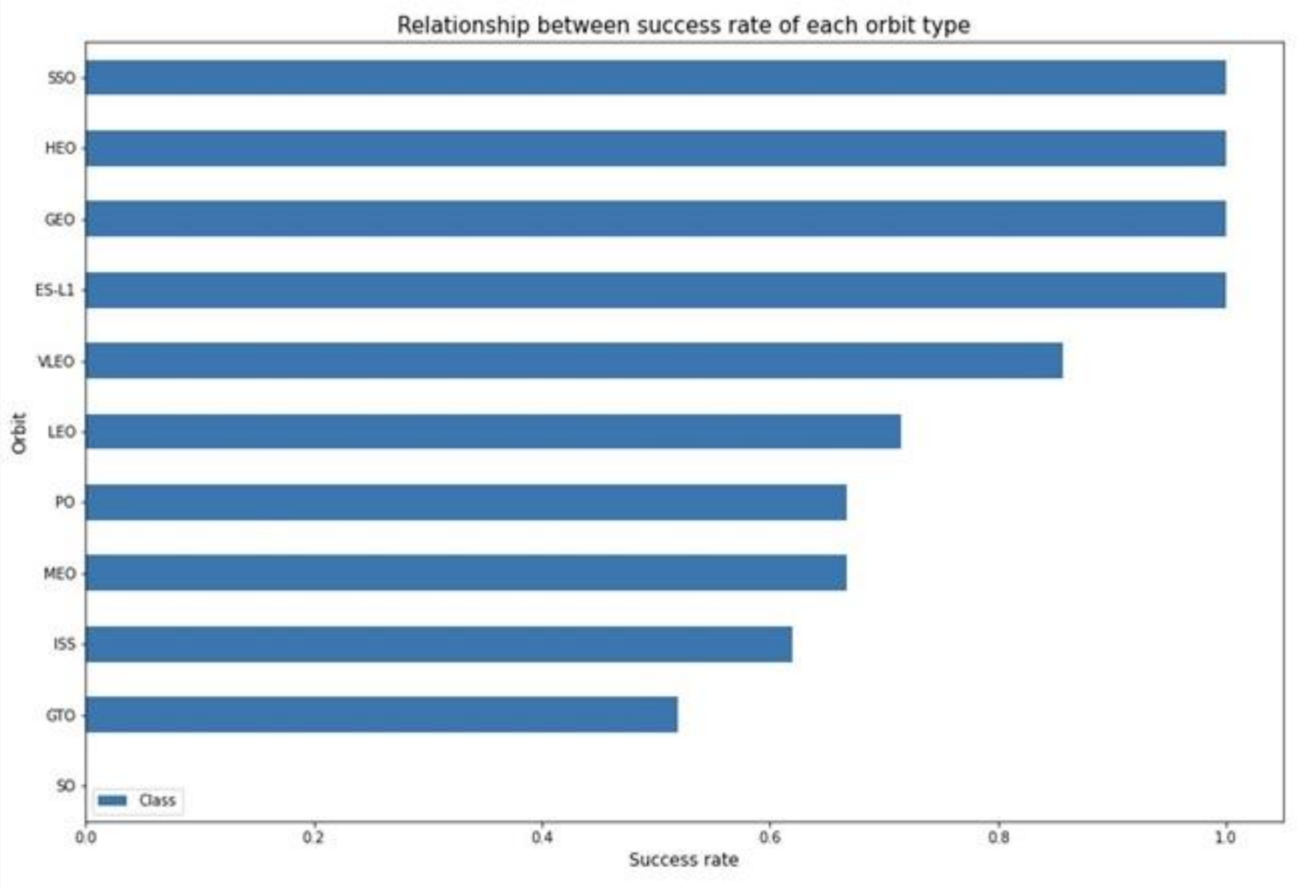
Payload vs. Launch Site



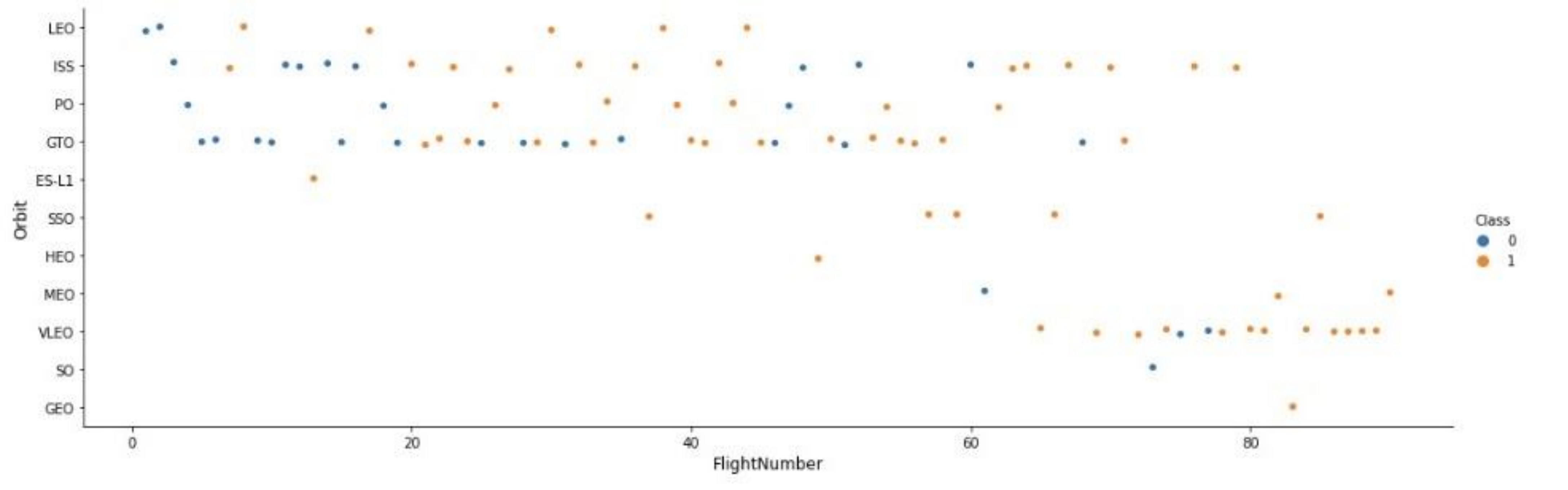
- CCAFS SLC 40 has launched rockets less than 7K kg and greater than 13K kg, and we can see that the success rate for the rocket much higher with the greater payload mass.
- In VAFB SLC 4E launch site there are no rockets launched for payload mass greater than 10K kilo.
- In KSC LC 39A launch site there are no launched lower than 2,5K kg.

Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The bar chart must be interpreted with the number of launches per orbit type.

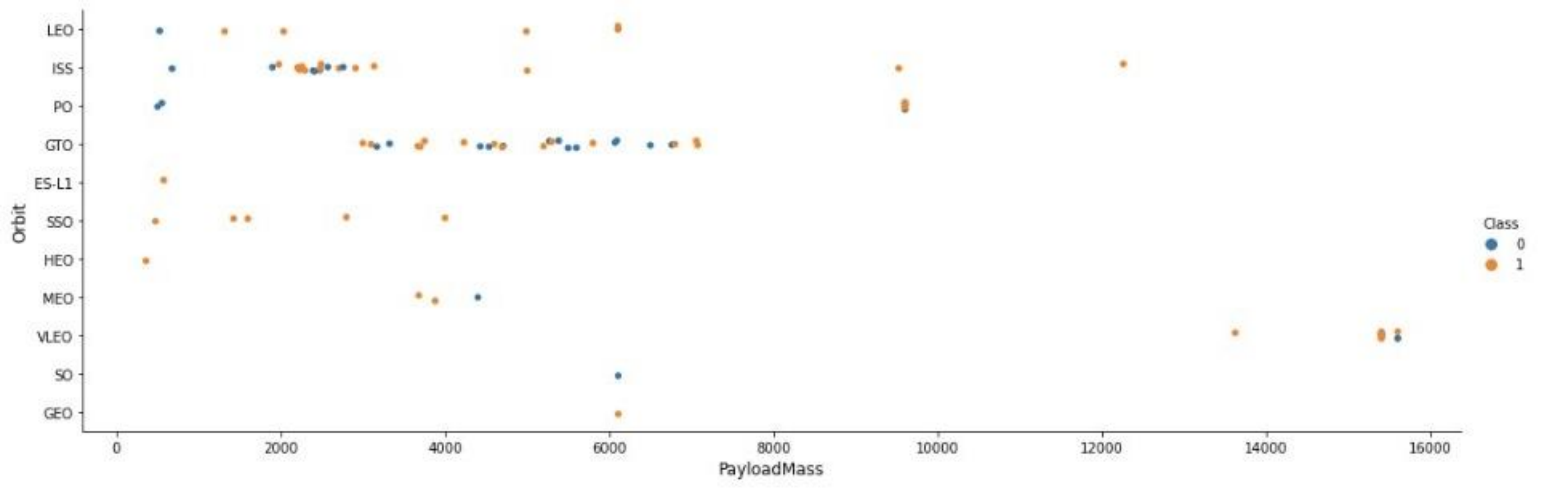


Flight Number vs. Orbit Type



- The plot 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.
- As expected, there are more failures at the beginning of series of launches but, after the first 40 launches the ratio improves by reducing the 50%of unsuccessful landings.

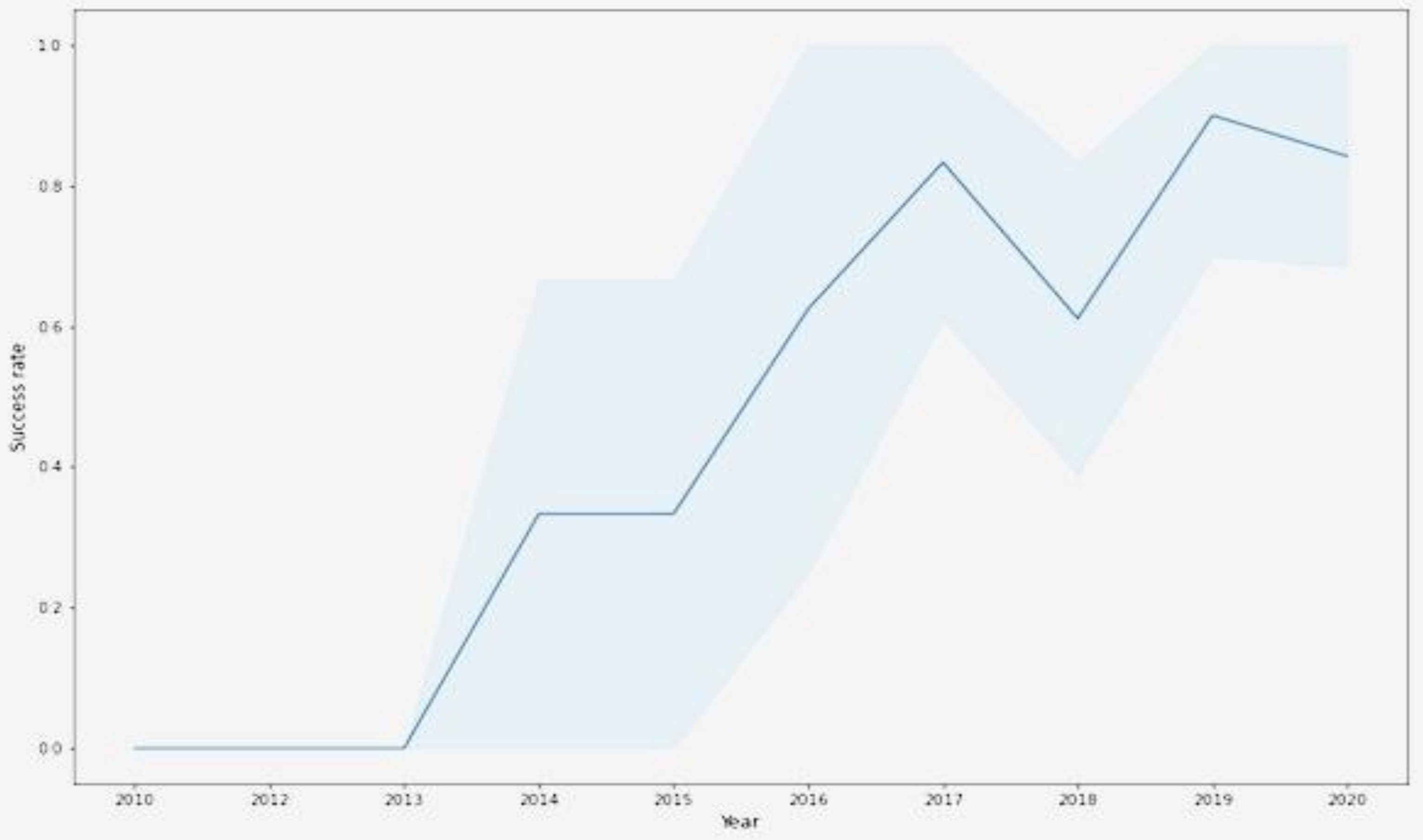
Payload vs. Orbit Type



- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.
- Exists visible limit of Payload around 7K kg, and less than 10 launches exceed that limit.
- With heavy payloads the successful landing rate are more for Polar, LEO and ISS.

Launch Success Yearly Trend

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



All Launch Site Names

- This slide shows 4 unique launch sites in the space mission

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

```
%sql SELECT DISTINCT(launch_site) FROM SPACEXTBL;

 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa3
Done.
```

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

Launch Site Names Begin with 'CCA'

- We used the query SELECT with clarifying parameters WHERE, LIKE and LIMIT to show 5 records with “CCA”

```
%sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	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

```
%sql SELECT SUM(payload_mass__kg_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE customer='NASA (CRS)';
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdor
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 using AVG() function

```
%sql SELECT AVG(payload_mass__kg_) AS AVG_PAYLOAD_MASS FROM SPACEXTBL WHERE booster_version='F9 v1.1';
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdom
Done.
```

avg_payload_mass
2928

First Successful Ground Landing Date

To find the date when the first successful landing outcome was achieved we use the MIN function

```
%sql SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL WHERE landing_outcome LIKE '%Failure (drone ship)%' 
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/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

Successful Drone Ship Landing with Payload between 4000 and 6000

The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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, payload_mass__kg_, landing_outcome FROM SPACEXTBL \
      WHERE landing_outcome='Success (drone ship)' AND (payload_mass__kg_ BETWEEN 4000 AND 6000) ;
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdc
Done.
```

booster_version	payload_mass__kg_	landing_outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes we uses the combination of COUNT function and GROUP BY statement in our query

```
%sql SELECT mission_outcome, COUNT(mission_outcome) AS TOTAL FROM SPACEXTBL GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdom
Done.
```

mission_outcome	total
Failure (in flight)	1
Success	99
Success (payload status unclear)	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), (SELECT MAX(payload_mass__kg_) AS "maximum_payload_mass" FROM SPACEXTBL) FROM SPACEXTBL
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

booster_version	maximum_payload_mass
F9 B4 B1039.2	15600
F9 B4 B1040.2	15600
F9 B4 B1041.2	15600
F9 B4 B1043.2	15600
F9 B4 B1039.1	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

```
%sql SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL WHERE landing_outcome LIKE '%Failure (drone ship)%' A
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/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

- 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, COUNT(landing_outcome) AS "total" FROM SPACEXTBL WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20')
```

```
 * ibm_db_sa://ycy00214:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

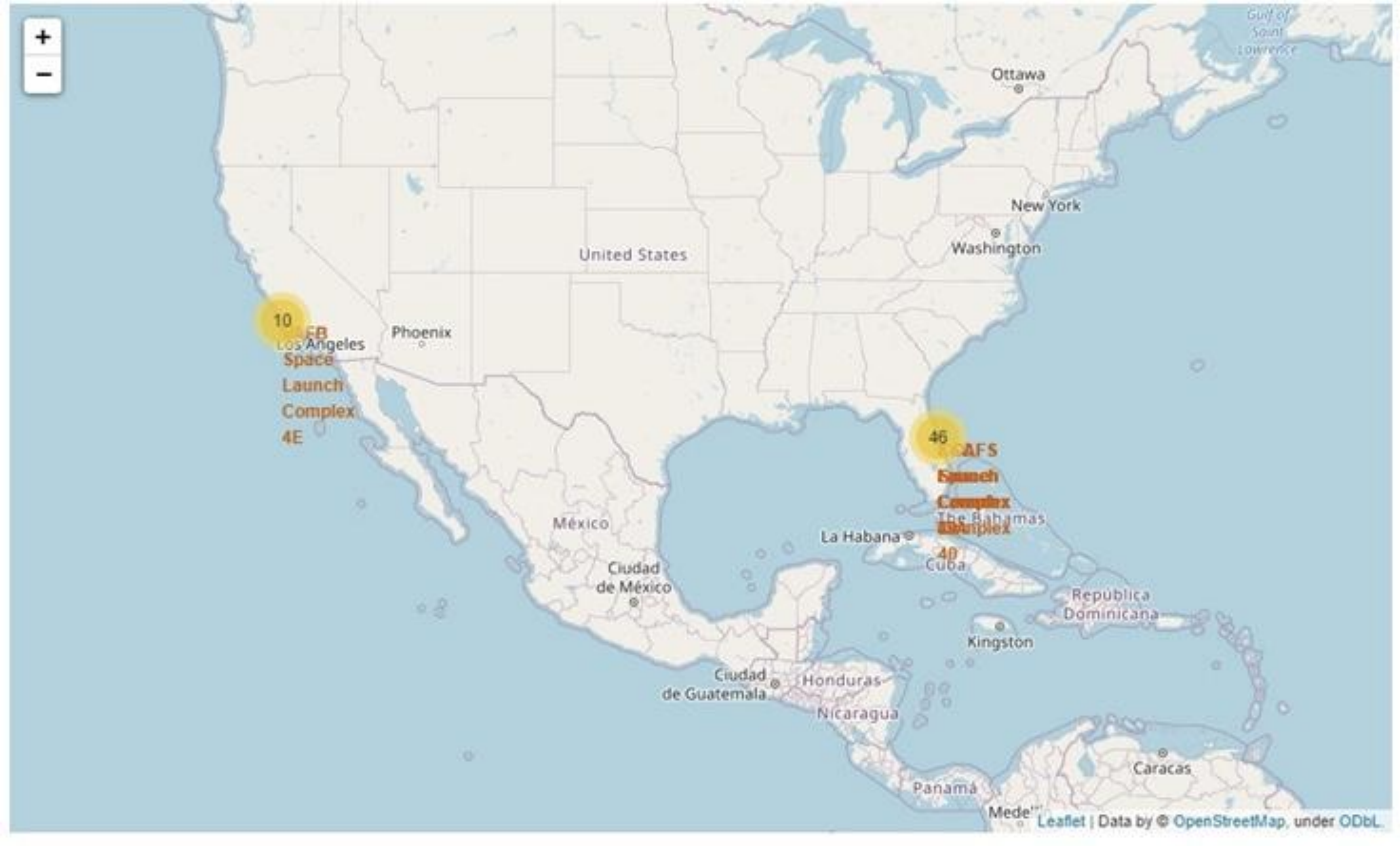
landing_outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

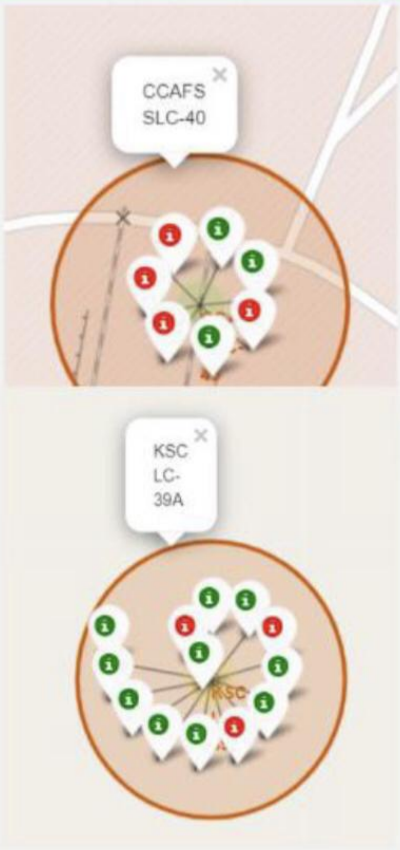
All launch sites global map markers

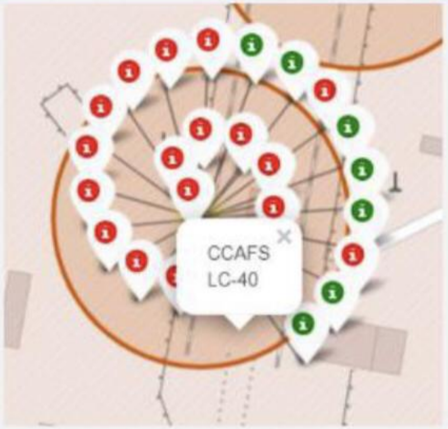
We can see that the SpaseX launch sites are are in the USA coasts Florida and California



Success/Failed Launches for each site

Markers are shoving launches sites with color labels





Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

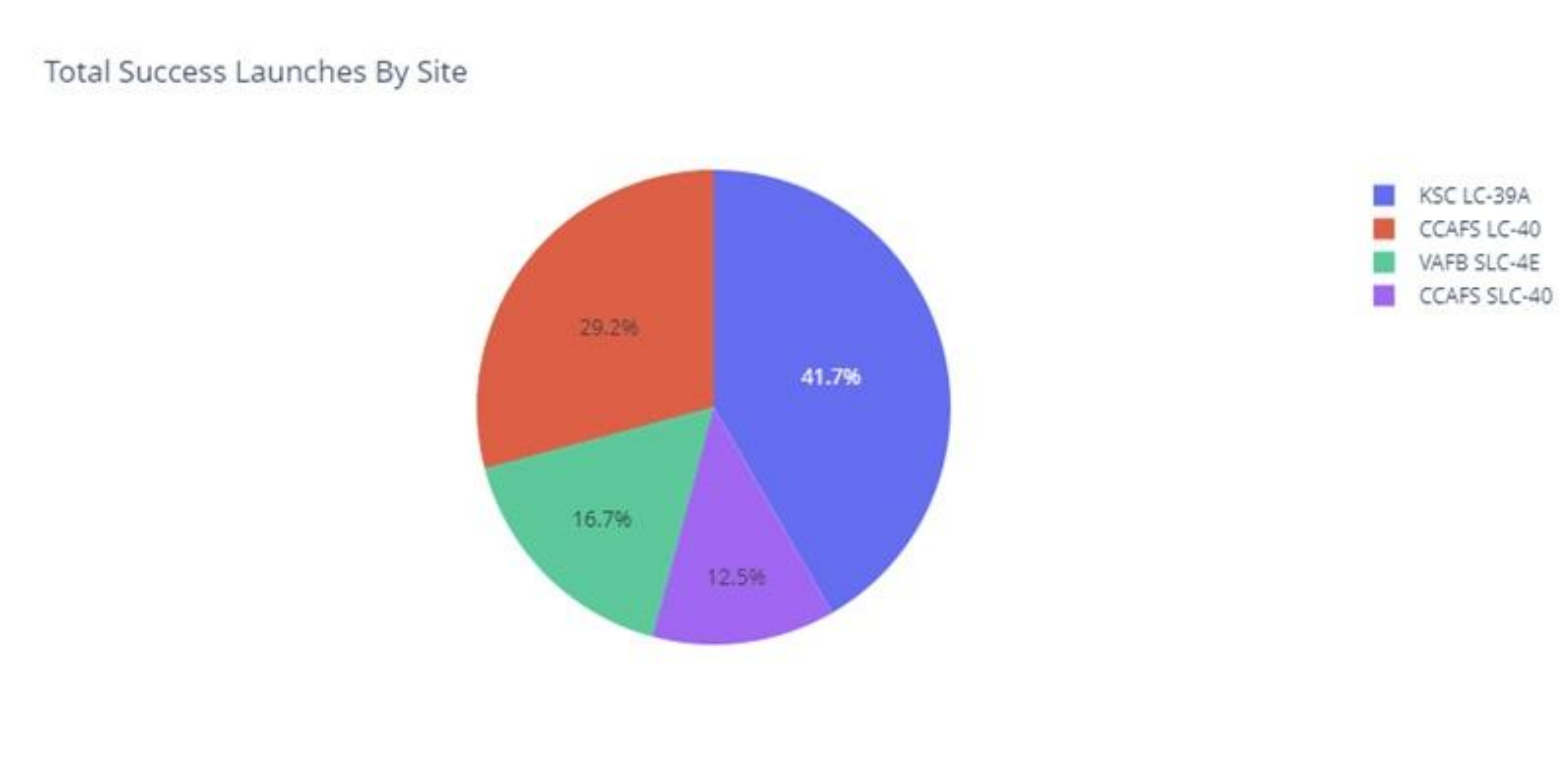
Launch sites distance to landmarks

•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

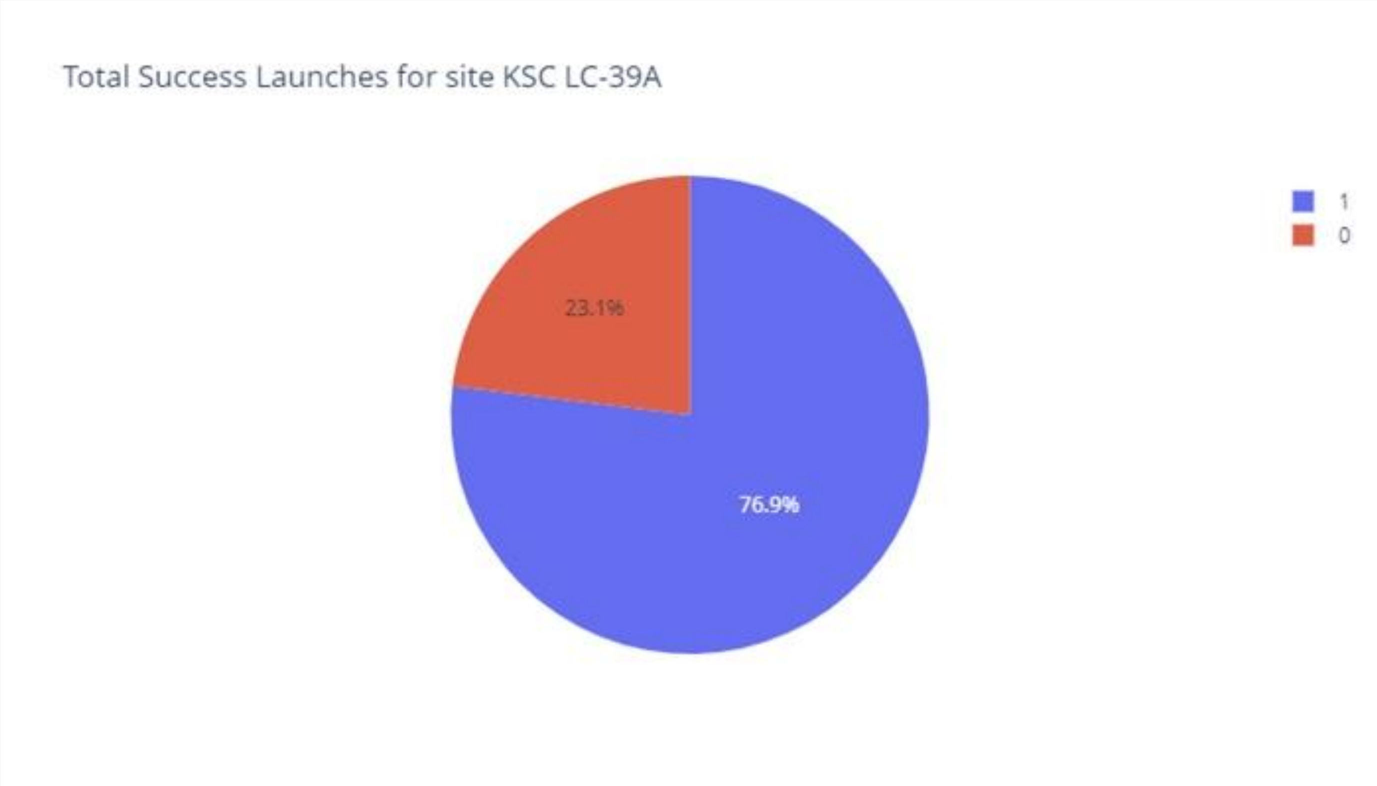
Build a Dashboard with Plotly Dash

The pie chart shows the total success percentage achieved by each launch site

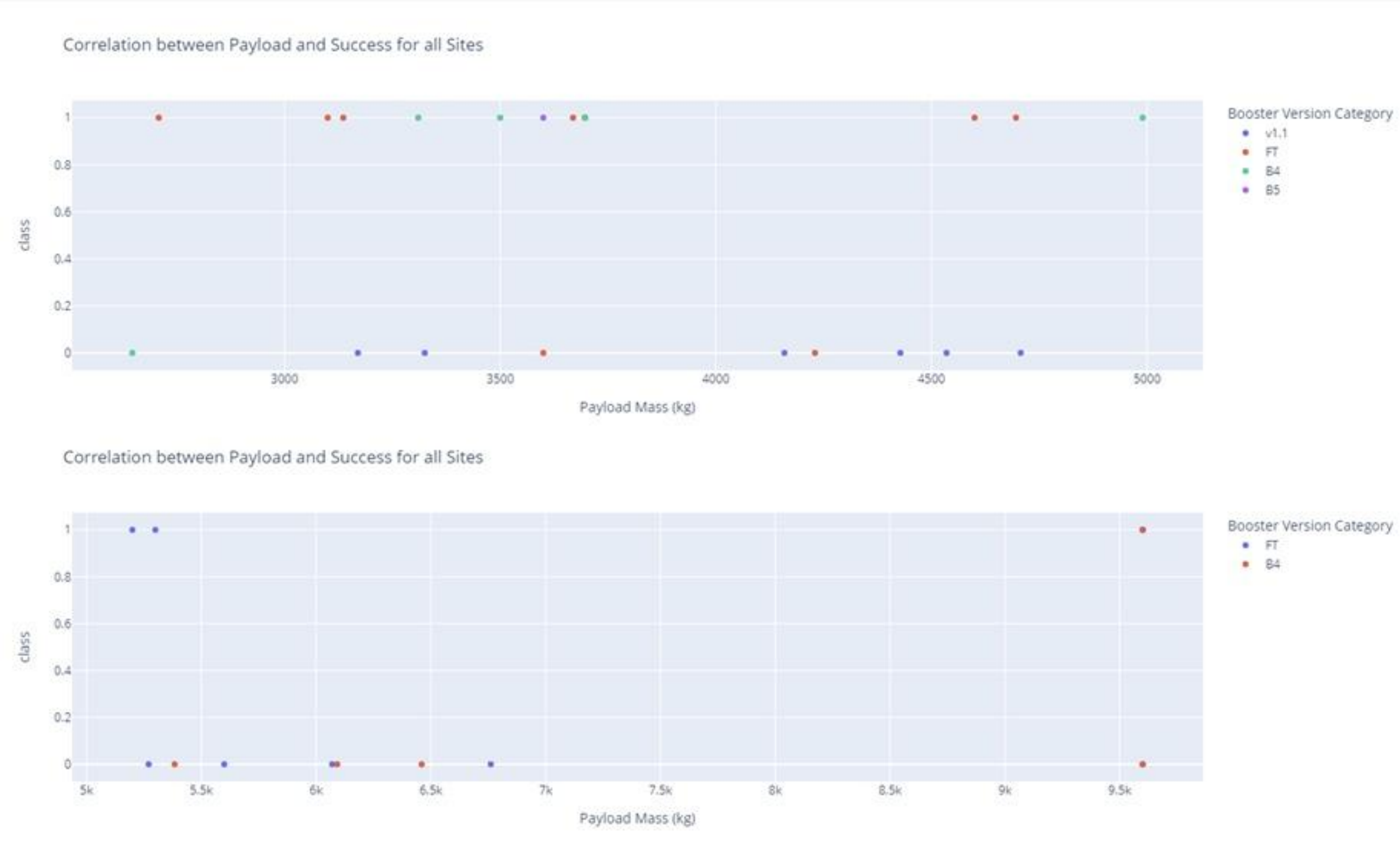


Pie chart shows the Launch site with the highest launch site success ratio

KSC LC-39A achieved 76,9% success rate while getting a 23,1% failure rate



Payload vs. Launch Outcome

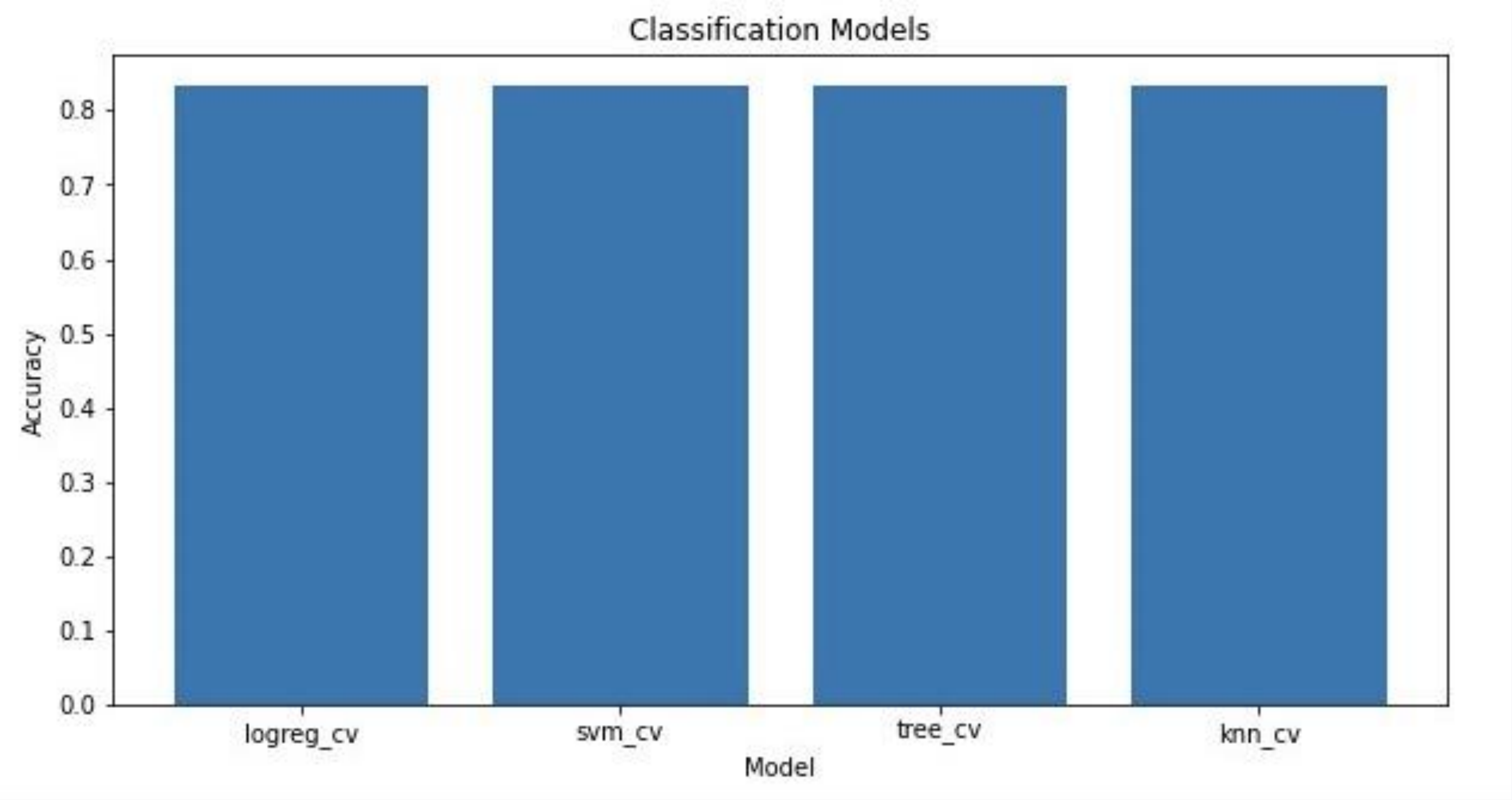


Section 5

Predictive Analysis (Classification)

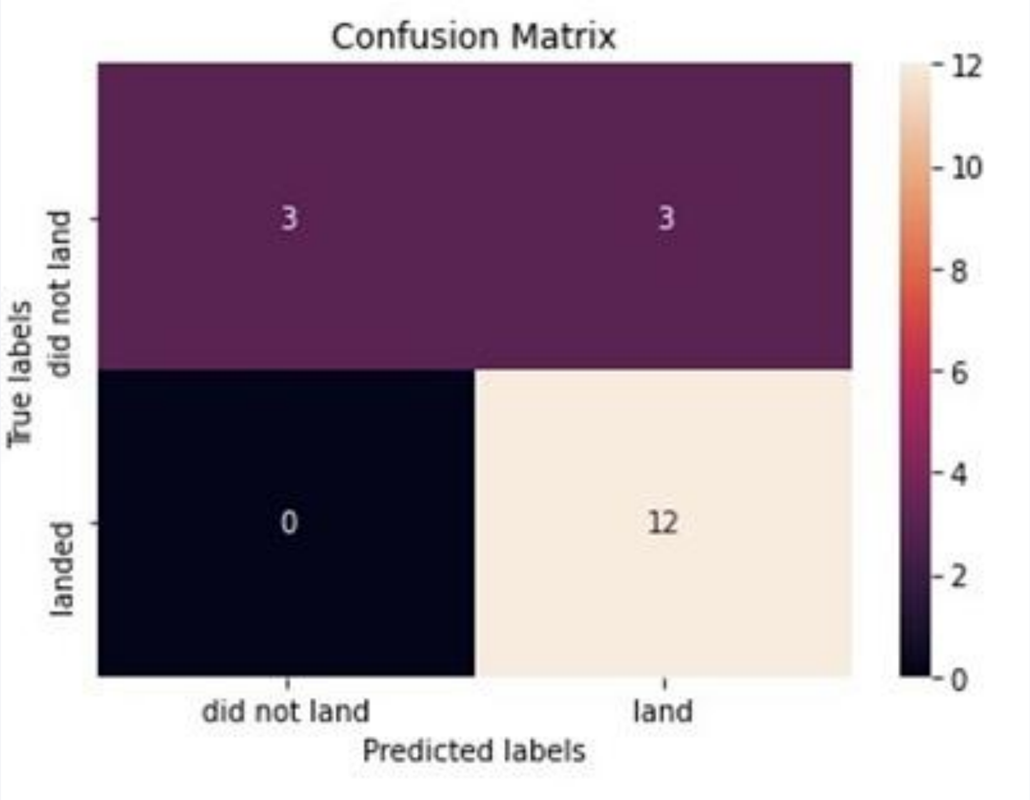
Classification Accuracy

As we can observe from the bar chart, all out models has similar accuracy.



Confusion Matrix

The confusion matrix for our classifiers 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.
- We can predict if the first stage of our competitor will land and determine the coast of launch by using any chosen machine learning model: KNN, Decision Tree, SVM or Log.regression.

Thank you!