IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oluwamayowa Akinmolayan
13th June, 2022.

Table of Contents

Executive Summary

- ✓Space Exploration Technologies Corp. is an American spacecraft manufacturer, space launch provider, and a satellite communication corporation headquartered in California.
- ✓It was founded in 2002 by Elon Musk with the goal of reducing space transportation costs.
- ✓SpaceX advertises its Falcon 9 rockets which is a cheaper alternative to other providers due to the fact that its first stage can be reused.
- ✓To determine the cost of a launch, we have to determine if the first stage would land .

Introduction

- The primary objective of this project is to create a machine learning pipeline that would predict if a rockets first stage would land successfully. This information can be used by a rival company that would launch a bid against Musk’s SpaceX.
- Problems you want to find answers to:
  - ❑ What factors determine if a rocket would land successfully?
  - ❑ What conditions need to be met in order to ensure a successful landing?
  - ❑ The interaction among features that combine to give a high success rate.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data for this project was collected from an API: specifically the SpaceX REST API.
- Perform data wrangling
    - The data collected was stored in a JSON format, then loaded into a pandas dataframe to enable data cleaning and exploratory data analysis (EDA).
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Splitting into training and test data. Machine learning models were built using Grid Search allowing evaluation of the best hyperparameters that would provide the highest accuracy.

Data Collection

- The data was collected as follows:
  - ❑ A get request was sent to the SPACEX API.
  - ❑ The response gotten was decoded as a JSON and turned into a pandas dataframe using .json_normalize()
  - ❑ Falcon9 rocket launches were selected into the dataframe and we then checked for missing values if there were any.
  - ❑ Conversely, web scraping of Falcon9 launch data from Wikipedia was also performed using python's BeautifulSoup package.
  - ❑ The data gotten was then parsed as a HTML file before it was loaded into a pandas dataframe.

Data Collection – SpaceX API

- The get requests to the SpaceX API to collect data, clean the data and allow data wrangling.
- For reference purposes, the GitHub URL to the notebook is included here:

Capstone.ipynb (github.com)

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

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

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

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

200
```

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

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

Data Collection - Scraping

- Web scraping of Falcon9 launch records with Beautiful Soup.

- The GitHub URL to the notebook is:

  [Data Collection with webscraping.ipynb (github.com)]()

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
x = requests.get(static_url)
print(x.status_code)
```
```
200
```

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(x.content, "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>
```

Data Wrangling

Exploratory Data Analysis (EDA) was performed on the dataframe. Some of the steps taken were:

- ❑ Finding the percentage of missing values.
- ❑ Finding the unique launch sites, how many times they occurred in the dataset.
- ❑ Finding the occurrence of each orbit.
- ❑ Identifying the various possible landing outcomes in the dataset.

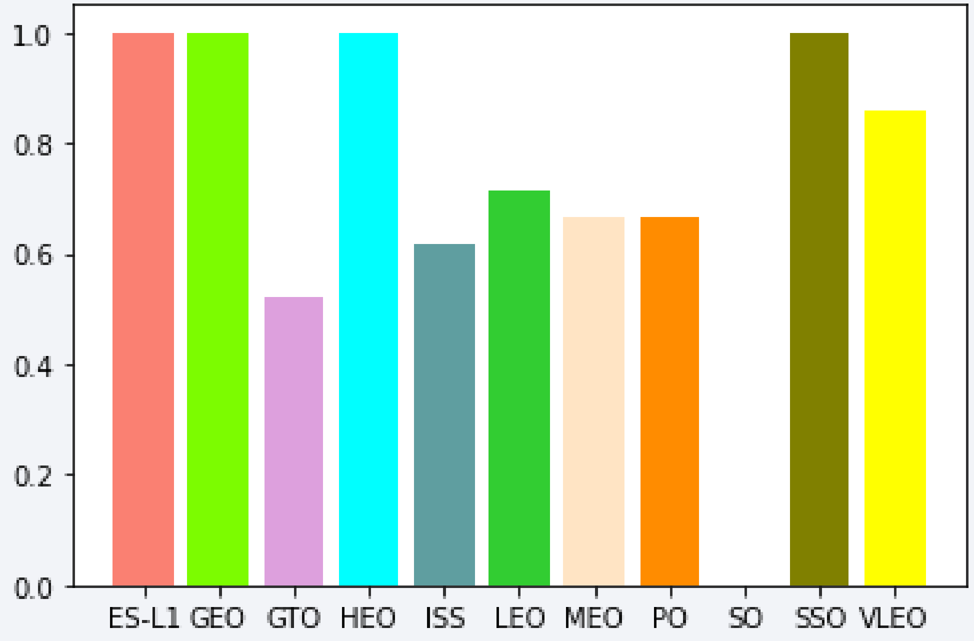
The GiHub of the notebook is: [Data Wrangling.ipynb (github.com)]()

EDA with Data Visualization

The data was explored by creating scatter plots to show the relationship between launch sites and flight number to verify if the launch site had any effects on the rocket payload.

A bar chart was created to find what orbits had the best chance of success.

- For reference purposes, the GitHub URL of the notebook: [EDA for Data Visulaization.ipynb (github.com)]()



EDA with SQL

- The data was loaded into a relational database management system and loaded into a table named SPACEXTBL.
- SQL queries were written to find insights into some of the following questions:
  - ❑To find the unique launch sites used in the dataset.
  - ❑The total payload mass carried by all boosters launched by NASA:
  - ❑To find the date when the first successful landing outcome in ground pad was achieved:
  - ❑To find the total number of successful and failed mission outcomes:

For reference purpose, the GitHub link to the notebook is : [EDA with SQL.ipynb (github.com)]

Build an Interactive Map with Folium

- All launch sites were marked and map objects like markers, circles, lines to mark either success or failure were added for each site.
- Color based markers were added to identify which launch sites had a relatively high success rate.
- The distance from launch sites to landmarks like coastlines, railways, highways were also calculated.
- For GitHub URL to the notebook is provided for external purposes: [Visual Analytics with Folium.ipynb (github.com)]()

Build a Dashboard with Plotly Dash

- An interactive dashboard was created with Plotly.
- Pie Charts showing total launches by certain sites can be shown in the dashboard.
- A scatter graph showing the relationship between Outcome and Payload Mass(Kg) for different booster versions.

Predictive Analysis (Classification)

- First thing that was done was to standardize the data by converting both the features and target variables to NumPy arrays that would be understood by the model. The data was then split into training and test data to evaluate how the model would perform on new data. Grid Search CV was used to build different models having different hyperparameters that allowed picking the model that performed best on test data based on the accuracy score.
- For reference purposes, the GitHub URL to the notebook is: Machine Learning Prediction.ipynb (github.com)

Results

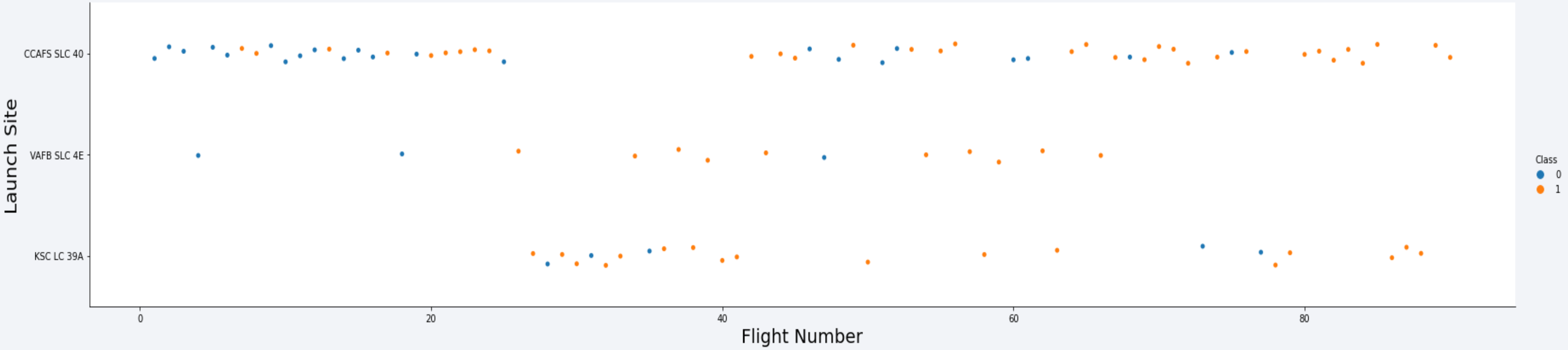
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

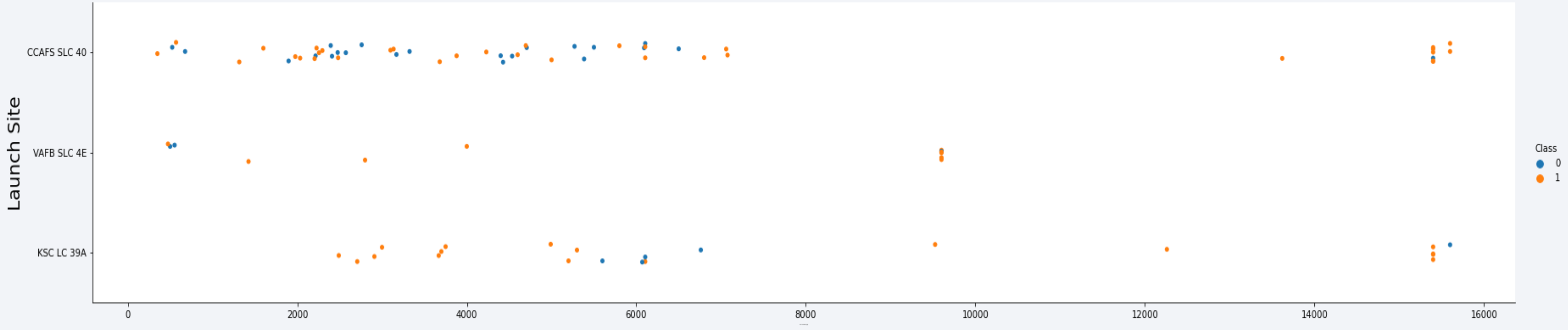
Flight Number vs. Launch Site

- The plot shows that higher flight number correlates with greater success rate.



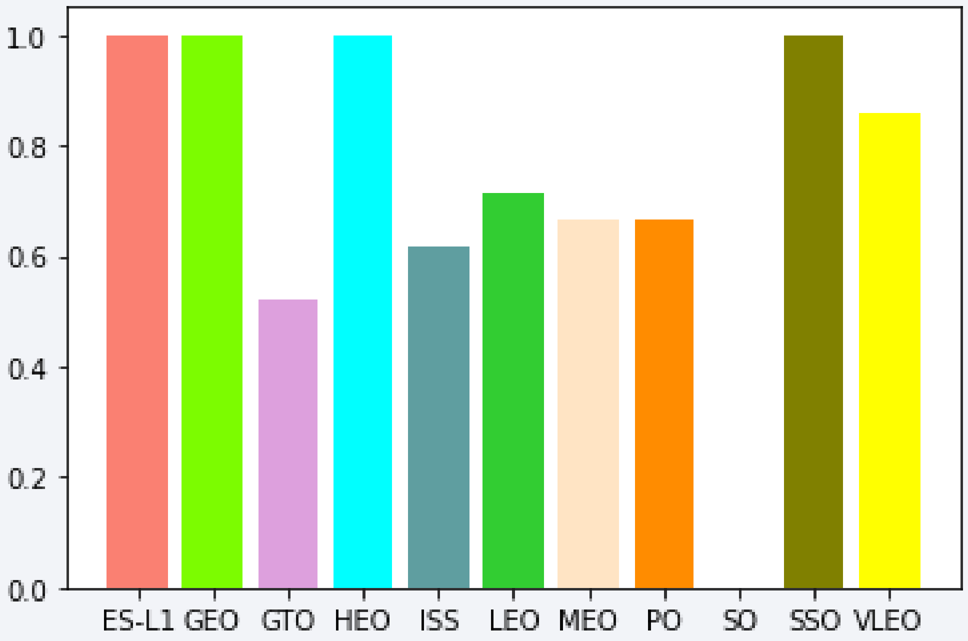
Payload vs. Launch Site

- The greater the payload mass, the greater the chances of a successful landing.



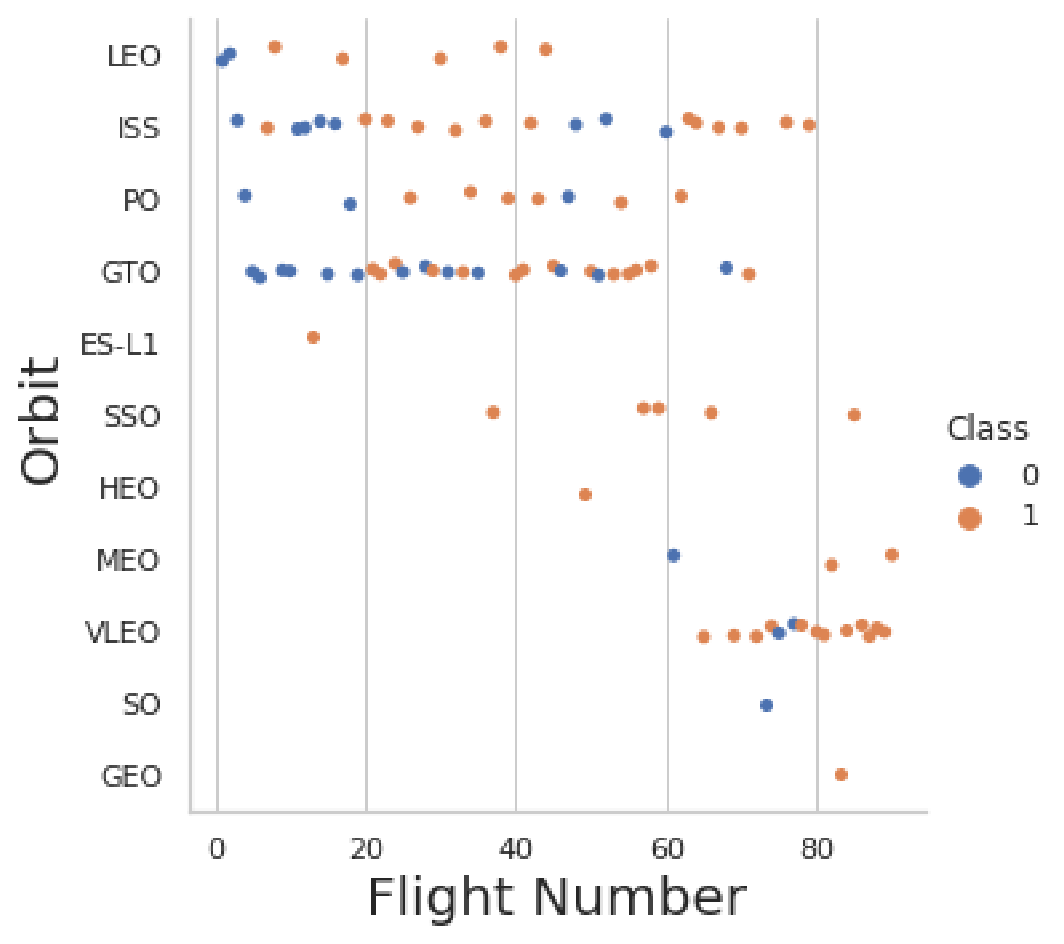
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO orbits have the highest success rates.



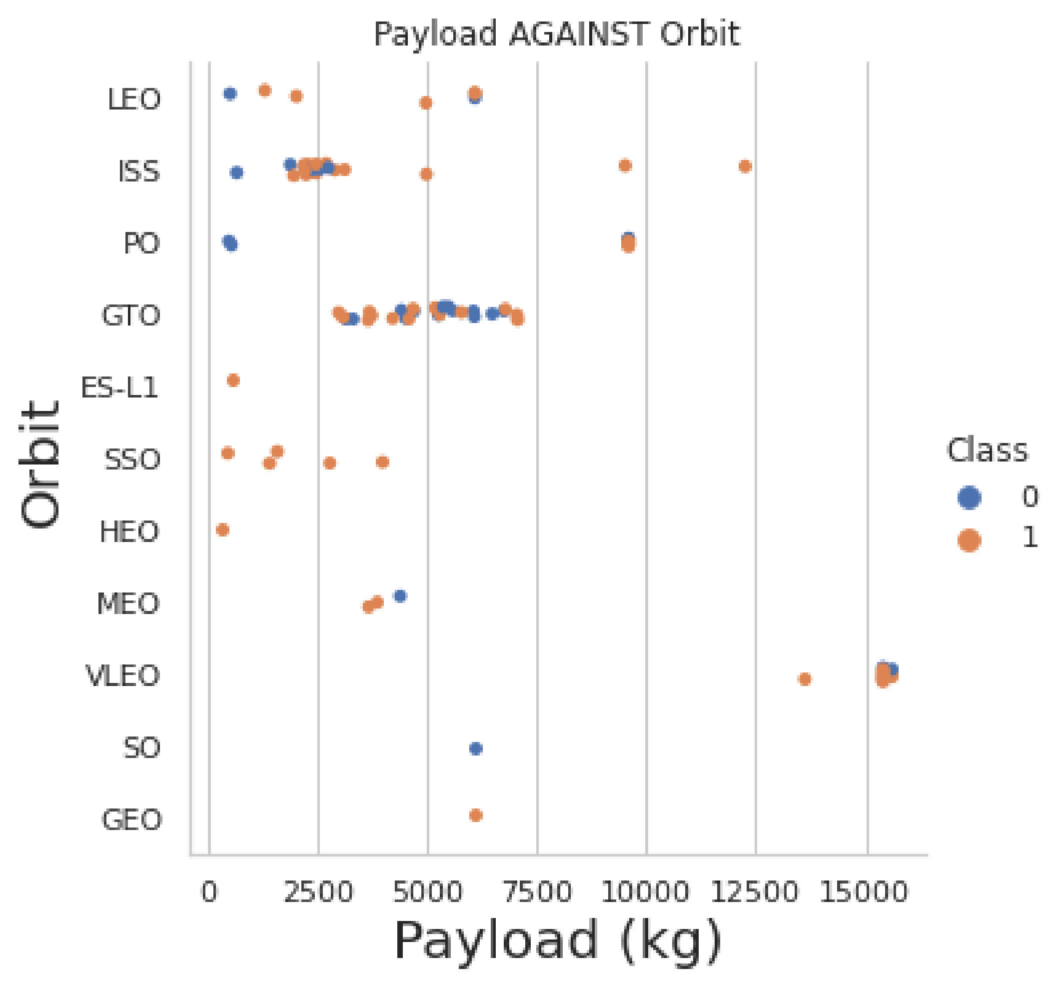
Flight Number vs. Orbit Type

- In the LEO orbit, greater flight numbers corresponds to more success, while in the GTO orbit, there is no correspondence between the success rate and flight number.



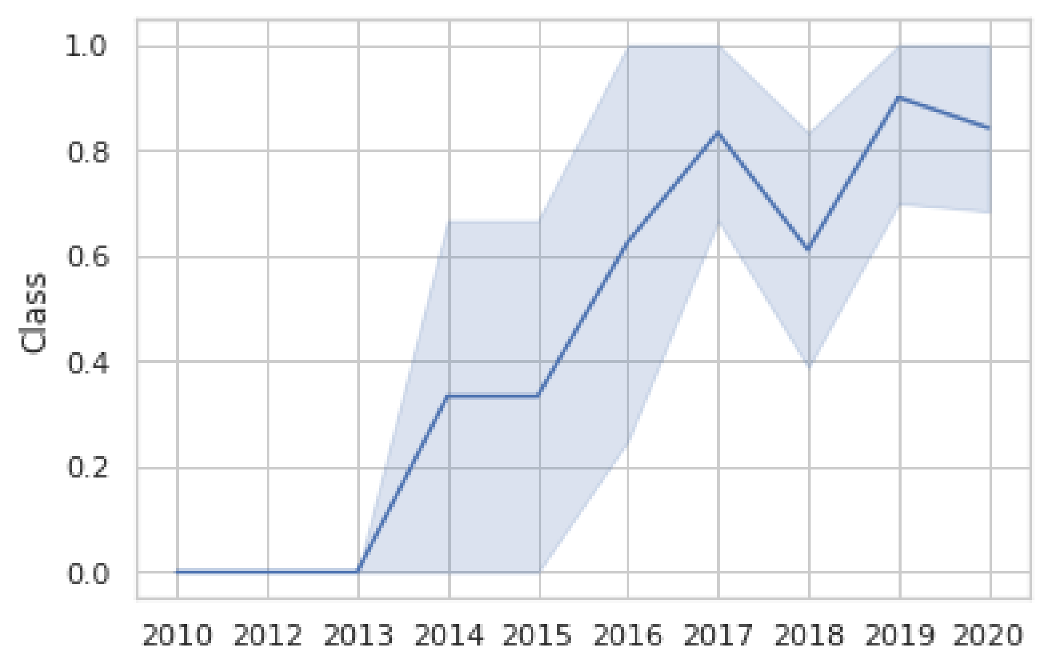
Payload vs. Orbit Type

- The plot shows that in the VLEO, ISS and PO orbits, heavier payloads mean high successrates.



Launch Success Yearly Trend

- There has been a sharp upward trend since 2013.



All Launch Site Names

- The DISTINCT keyword was used in the query to request for the names of unique launch sites from the database table.

```
In [6]: %%sql
        SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL

 * ibm_db_sa://zx197772:***@2d46b6b4-cbf6-40eb-bbc
Done.
Out[6]:
```

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

Launch Site Names Begin with 'CCA'

In [7]:
```
%%sql
SELECT * FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE '%CCA%'
LIMIT 5
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[7]:

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

- The records of 5 sites having characters ‘CCA’ was chosen from the dataset. The ‘LIMIT’ keyword was used in the query to restrict the results to a specified number.

Total Payload Mass

- The total payload carried by boosters launched by NASA is 48213.

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

In [8]:

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

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[8]:

1
48213

Average Payload Mass by F9 v1.1

- The average payload carried by F9 v 1.1 boosters is : 2928

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

In [9]:
```
%%sql
SELECT AVG (PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[9]:

1
2928

First Successful Ground Landing Date

- The first successful ground pad mission was achieved on: 22-12-2015. The ‘MIN’ keyword was added to the query to find the earliest date that satisfies the condition.

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

Hint:Use min function

In [10]:
```
%%sql
SELECT MIN (DATE) FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE '%Success%'
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[10]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The booster versions that have successfully landed on drone ships and had payloads between the range of 4000 to 5999 kg are:

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

In [11]:

```
%%sql
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000) AND LANDING__OUTCOME = 'Success (drone ship)'
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[11]:

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failed missions are:

List the total number of successful and failure mission outcomes

In [12]:

```
%%sql
SELECT MISSION_OUTCOME, COUNT (MISSION_OUTCOME) FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[12]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass. This was achieved by using a sub-query in the 'WHERE' condition.

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

In [13]:

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

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[13]:

booster_version
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

- The boosters and launch sites with failed drone pad landings in 2015 are:

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

In [14]:
```
%%sql
SELECT BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME FROM SPACEXTBL
WHERE DATE LIKE '%2015%' AND LANDING__OUTCOME = 'Failure (drone ship)'
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[14]:

booster_version	launch_site	landing__outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- The 'LIKE' keyword was used to specify the year we needed i.e 2015; 'AND' keyword was used to add a second condition to be satsisfied- in this case, a drone ship failure.

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.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (groun

In [15]:
```
%%sql
SELECT LANDING__OUTCOME, COUNT (LANDING__OUTCOME) as COUNT FROM SPACEXTBL
WHERE (DATE between '2010-06-04' AND '2017-03-20')
GROUP BY LANDING__OUTCOME
ORDER BY COUNT DESC
```

```
 * ibm_db_sa://zxl97772:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l0
Done.
```

Out[15]:

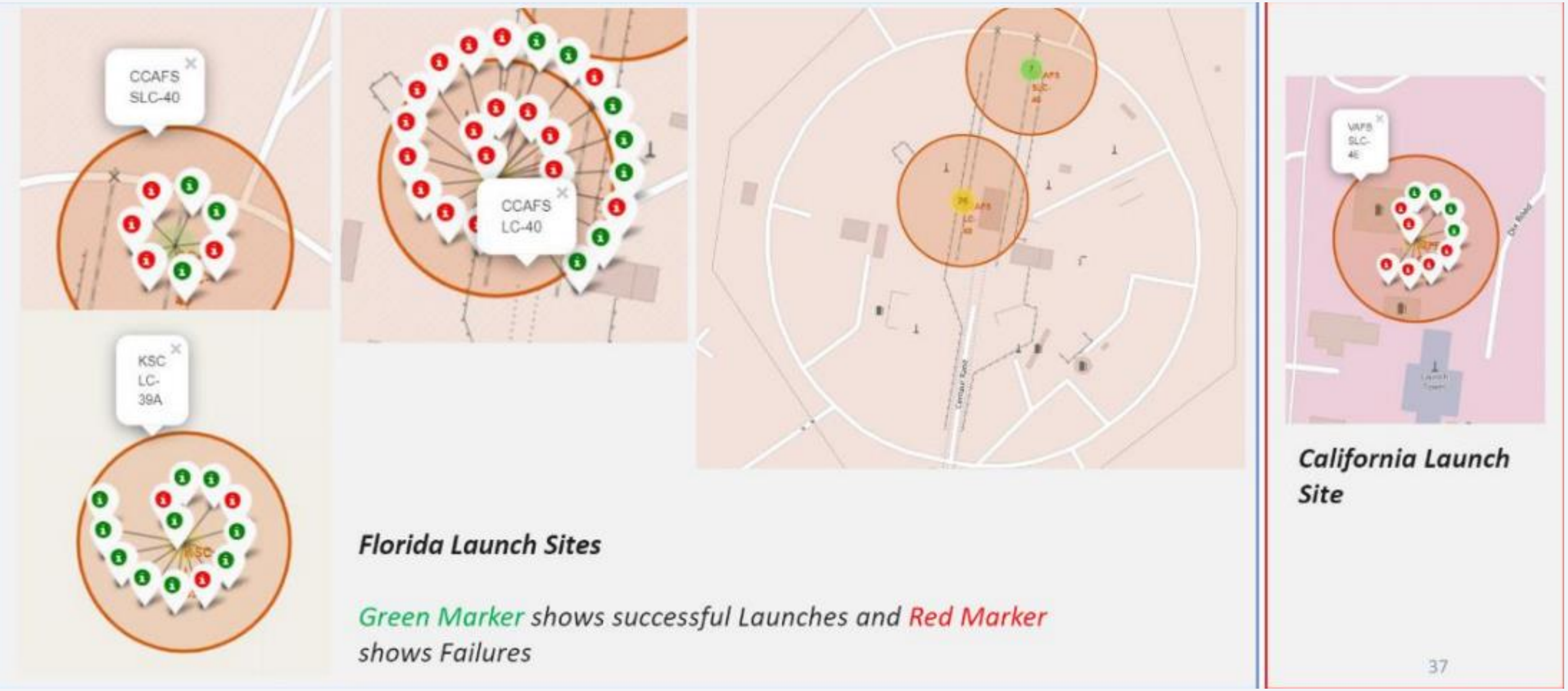
landing_outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Markers of Global Launch Sites

Markers showing Launch Sites with color labels



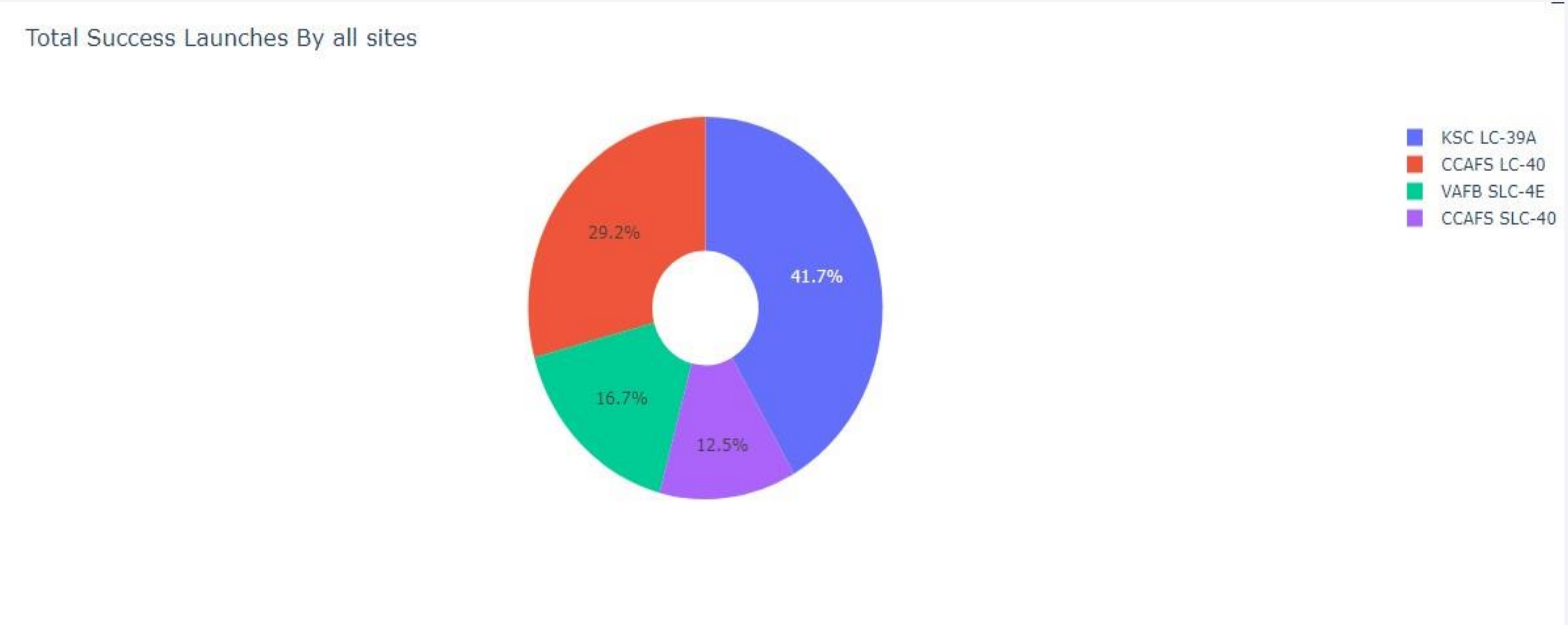
Launch Sites distance to landmarks

- 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

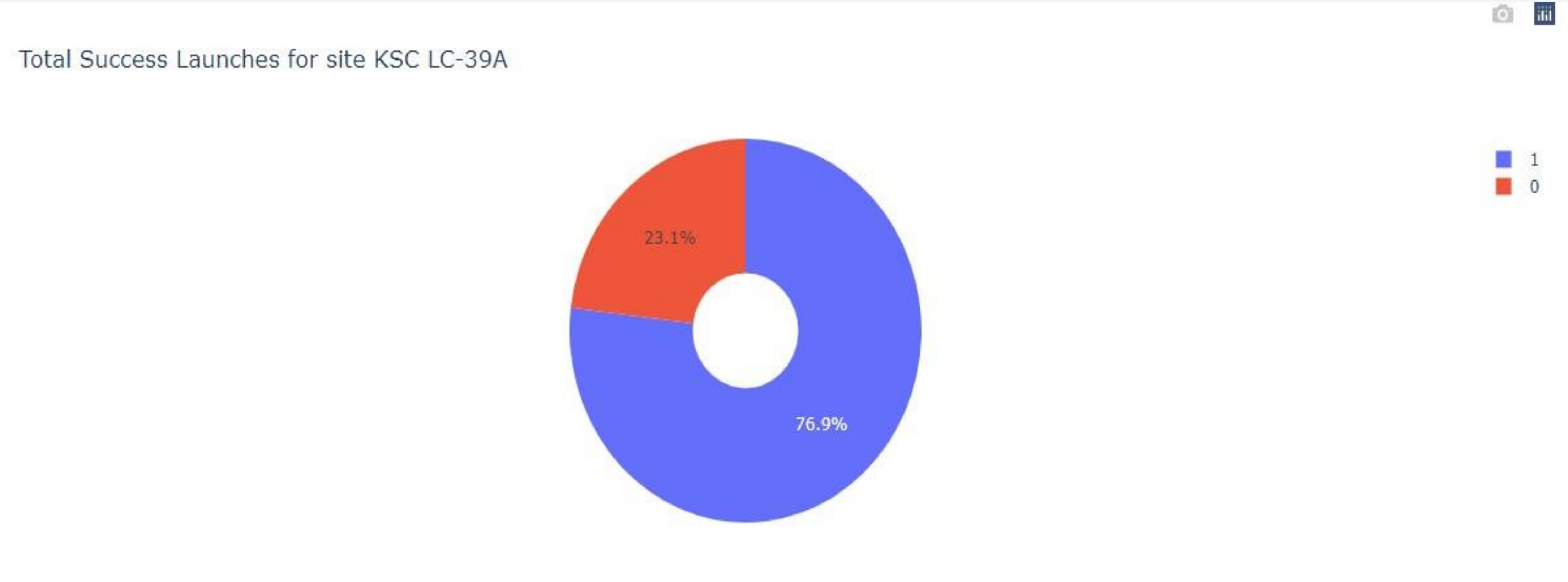
Build a Dashboard with Plotly Dash

Pie Chart of Launch Success Count



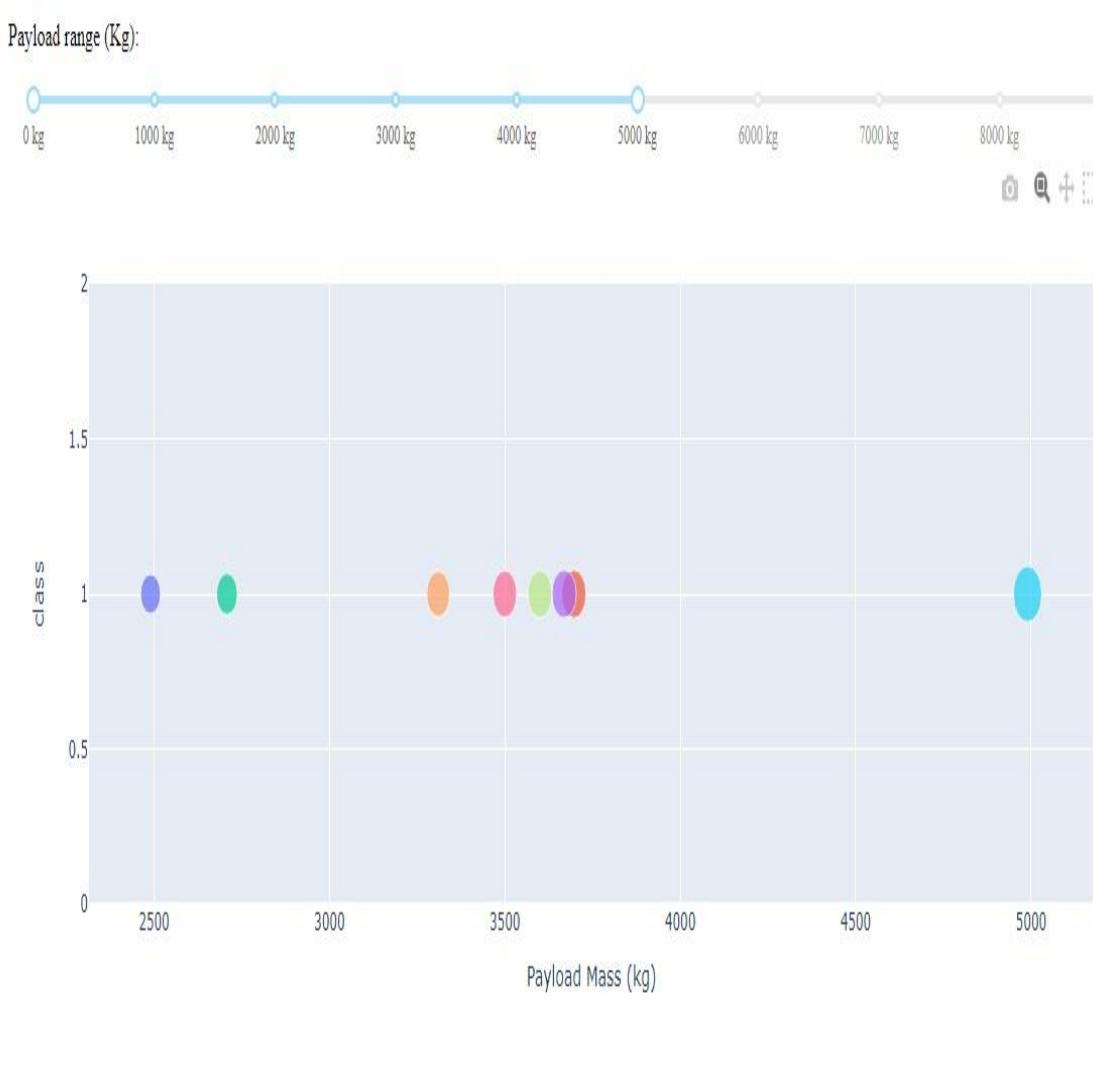
- KSC LC-39A had the most success among all the launch sites.

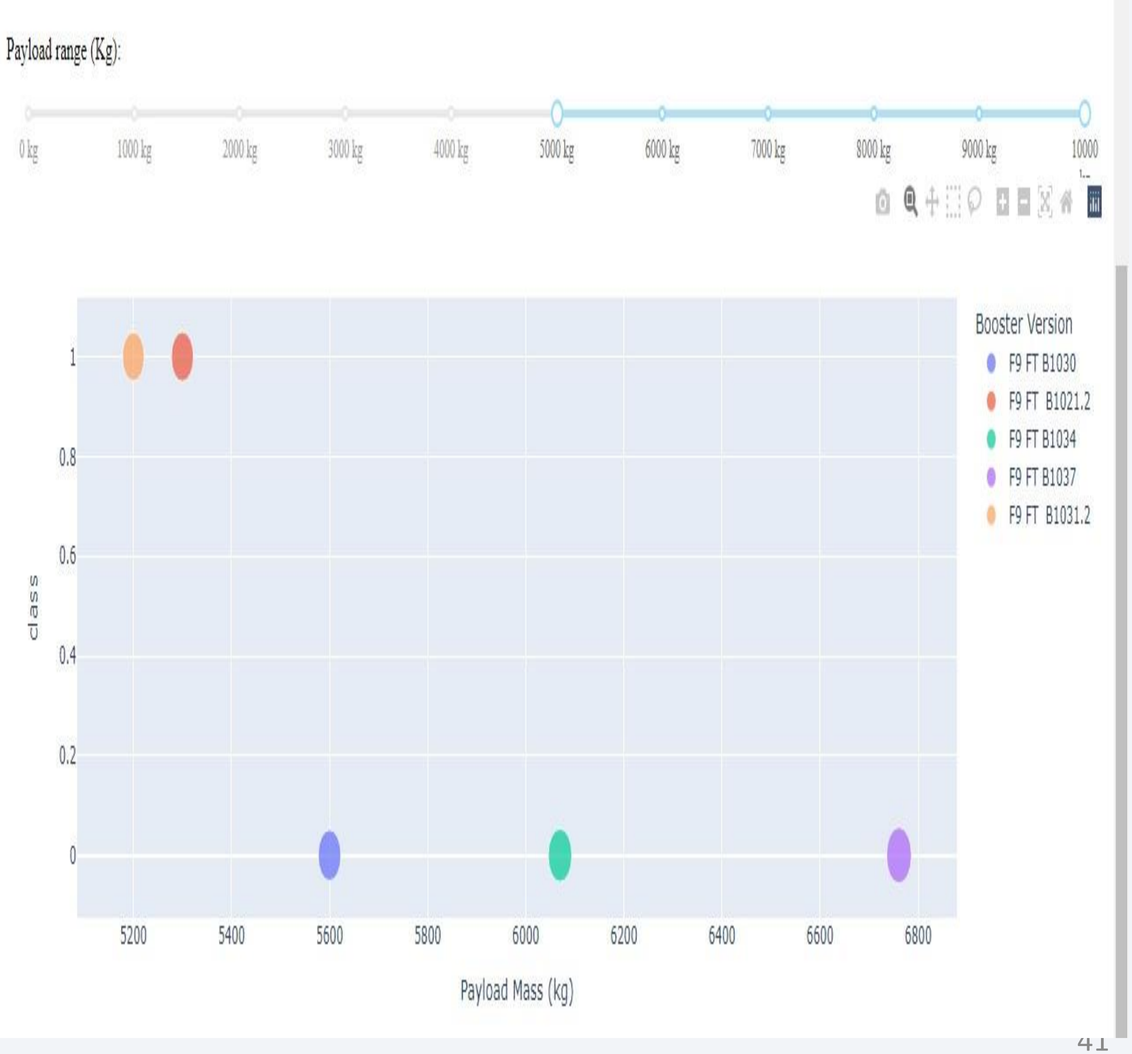
Pie Chart of the KSC LC-39A launch site



- This launch site has a 76.9% success rate and a 23.1% failure rate.

Scatter plot of Payload Mass against Launch Outcome for different load categories



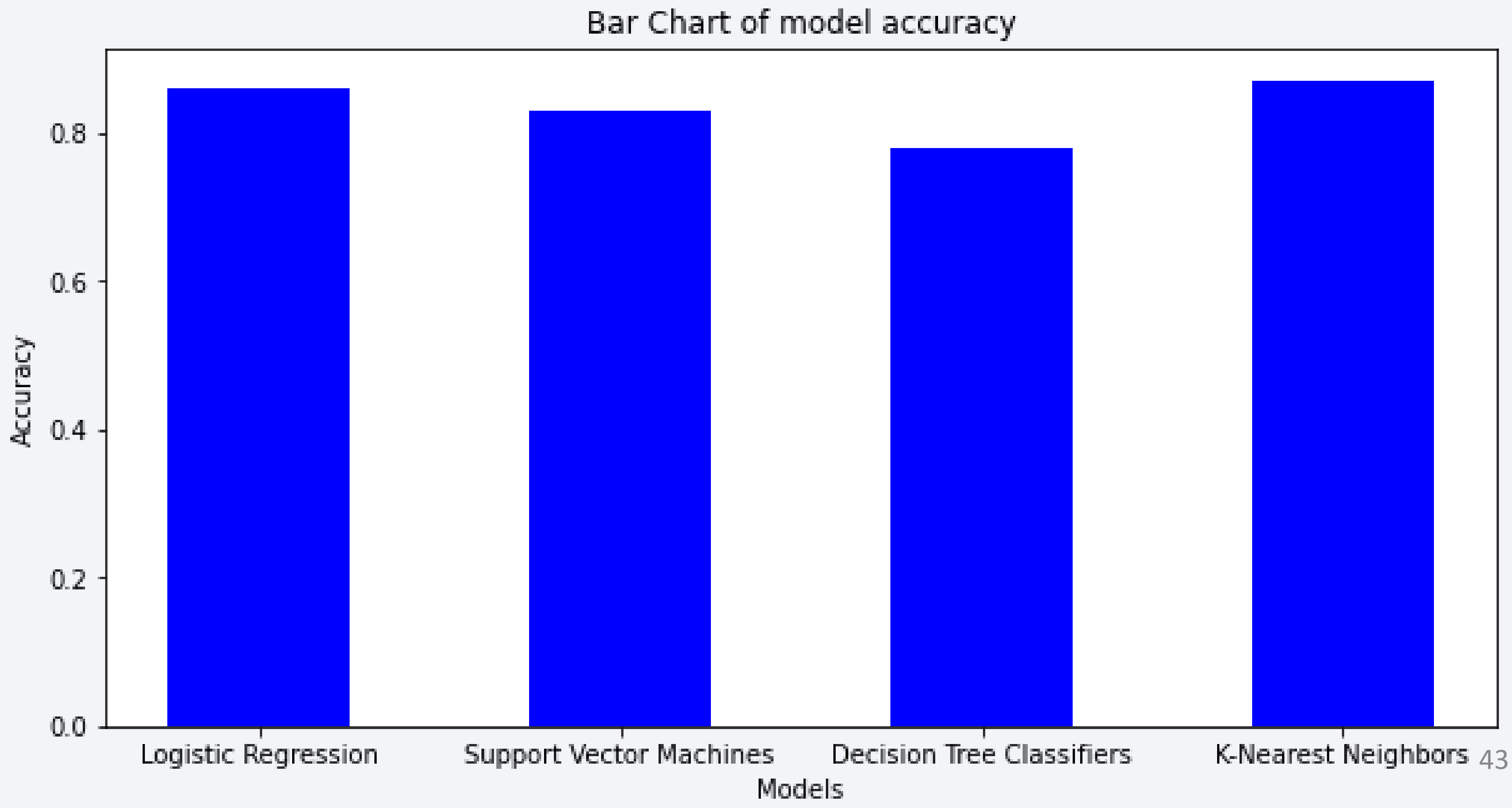


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The plot shows that the model with the highest accuracy is K-Nearest Neighbors.



Confusion Matrix

- The confusion matrix of the KNN model shows that model can distinguish between various classes. The main issue if the False Positive predictions i.e. the number of unsuccessful landings that were predicted as successful.

Conclusions

- The following conclusions can be drawn from the presentation:
  - ❑ Success rate tends to increase as the flight number increases.
  - ❑ The ES-L1, GEO, HEO and SSO orbits have the highest success rates.
  - ❑ Success rate has steadily increased since 2013.
  - ❑ KSC LC-39A launch site recorded the most success.
  - ❑ The KNN classifier was the best model for predicting whether the first stage would land successfully or not.

Thank you!