IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gires Folomo
30 January 2023

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.

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
- Predict if the first stage will land given the data

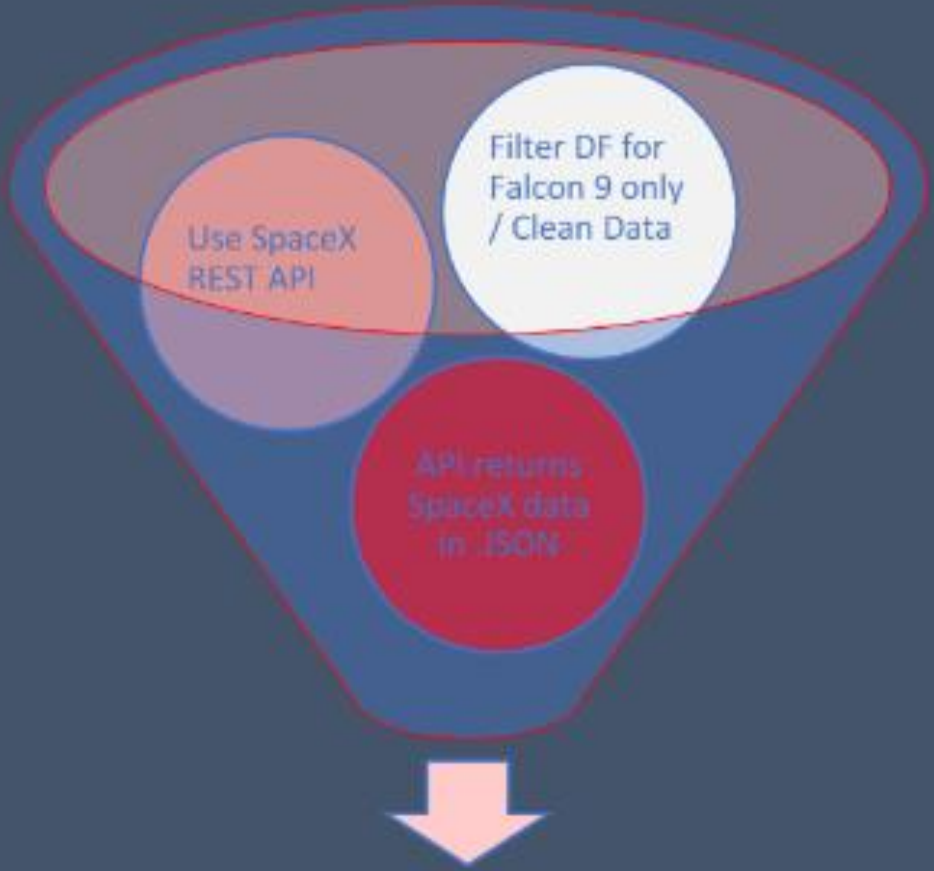
Section 1

Methodology

Methodology

- **Data collection methodology**
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- **Perform data wrangling**
  - Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Plot scatter and bar graphs to find relationships between variables and see patterns
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - Logistic Regression, SVM, Decision Tree and KNN

Data collection – SpaceX API



Normalize data into flat data file such as .csv

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

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

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

Task 2: Convert the json result into a data frame

```
data=pd.json_normalize(response.json())
```

Task 3: Constructing the dataset by combining the columns into a dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

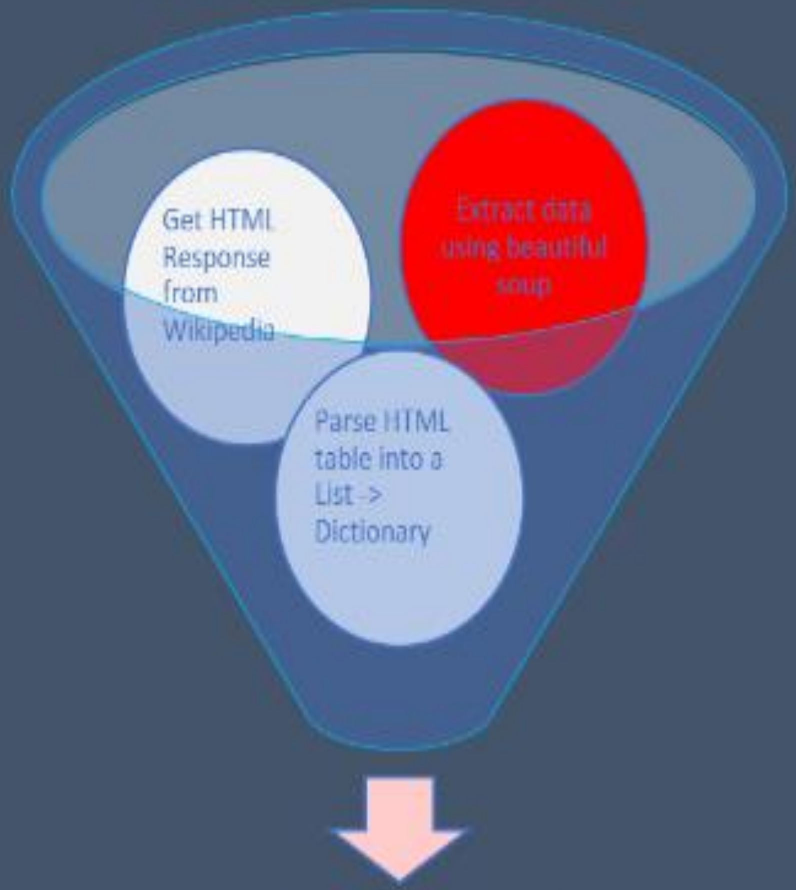
Task 4: Creating a Pandas data frame from the dictionary

```
data= pd.DataFrame(launch_dict)
```

Task 5: Exporting the data frame to csv

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data collection – Web Scrapping



Normalize data into flat data file such as .csv

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

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

r= requests.get(static_url)
data=r.text
```

Task 2: Create a BeautifulSoup object from the HTML `response`

```
soup = BeautifulSoup(data,"html.parser")
```

Task 3: Find the tables

```
html_tables = soup.find_all('table')
```

Task 4: Creating a dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Task 5: Exporting the data frame to csv

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

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.

- GitHub link:

https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/b2cd5016ede39d41c6bffc6cc6adc4e3c938bcf7/Final_Assignment%202.ipynb

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 Wrangling

1. Introduction

- Exploratory Data Analysis (EDA) was performed to find some patterns in the data and determine what would be the label for training supervised models.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

- Those outcomes were converted into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.
- GitHub link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/EDA_1.ipynb

Data Wrangling

2. Process

TASK 1:
Calculate the number of launches on each site

TASK 2:
Calculate the number and occurrence of each orbit

TASK 3:
Calculate the number and occurrence of mission outcome per orbit type

TASK 4:
Create a landing outcome label from Outcome column

TASK 5:
Exporting the dataset to csv

EDA with Data Visualization

Three types of graphs were used namely; **scatter plot, bar graph and line plot.**

a) Scatter plot: to observe and show relationships between two numeric variables (FlightNumber vs. PayloadMass, Flight Number vs. Launch Site, Payload vs. Launch Site, FlightNumber vs. Orbit type, Payload vs. Orbit type,)

b) Bar graph: to visualize the relationship between success rate of each orbit type

c) Line plot: to visualize the success rates over years

GitHub Link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/EDA%20Dataviz.ipynb

EDA with SQL

The SQL extension was loaded and to establish a connection with the database then SQL queries written to solve the below tasks:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery
- 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
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

GitHub link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

It is possible that launch success rate may depend on the location and proximities of a launch site. The interactive map was built to:

a) Mark all launch sites on a map using coordinates with the aid of **Circle markers**

b) Mark the success (green circle) and failed (red) launches for each site with the aid of **Mark clusters**

c) Measure the distances between a launch site to its proximities with the aid of **Lines**

GitHub link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/Launch%20Sites%20Locations.ipynb

Build a Dashboard with Plotly Dash

A Plotly Dash application was built for users to perform interactive visual analytics on SpaceX launch data in real-time.

Two types of plots/graphs were used:

a) **Pie chart**: to visualize launch success counts by site

b) **Scatter chart**: to observe the correlation between payload and mission outcomes by site

GitHub link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/spacex_dash_app.py

Predictive Analysis (Classification)

<u>**Object**</u>: to create a machine learning pipeline to predict if the first stage will land.

Model Building

- Loading dataset into NumPy and Pandas
- Data transformation
- Split data in train/test sets
- Choose the type of algorithm
- Create the GridSearchCV object and fit to find the best parameters
- Fit datasets in GridSearchCV objects and train the dataset

Model Evaluation

- Calculate the accuracy using the method score
- Get tuned hyperparameters for each algorithm
- Plot confusion matrix

Predictive Analysis (Classification)

Model improvement

- Feature engineering
- Parameters tuned

Finding the best performing model

- The best model is the one with the highest accuracy score on test data

GitHub link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/SpaceX_Machine%20Learning_Prediction.ipynb

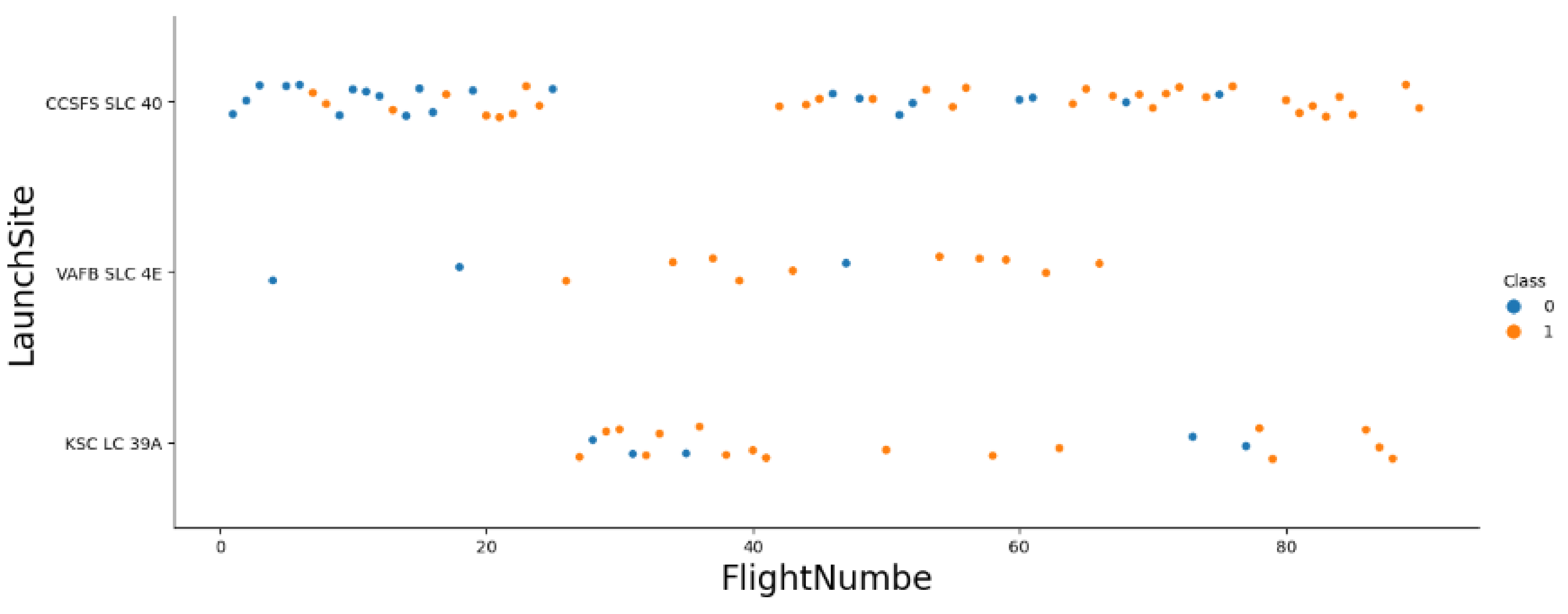
Results

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

Section 2

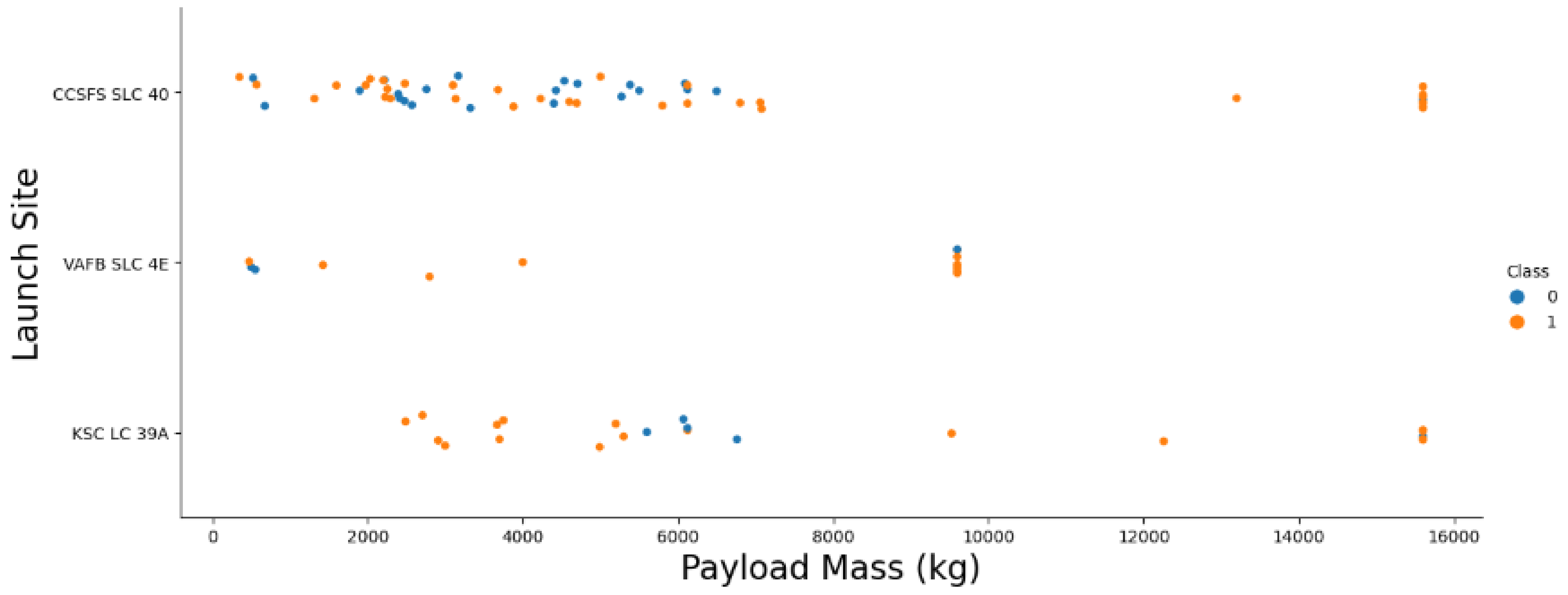
Insights drawn from EDA

Flight Number vs. Launch Site



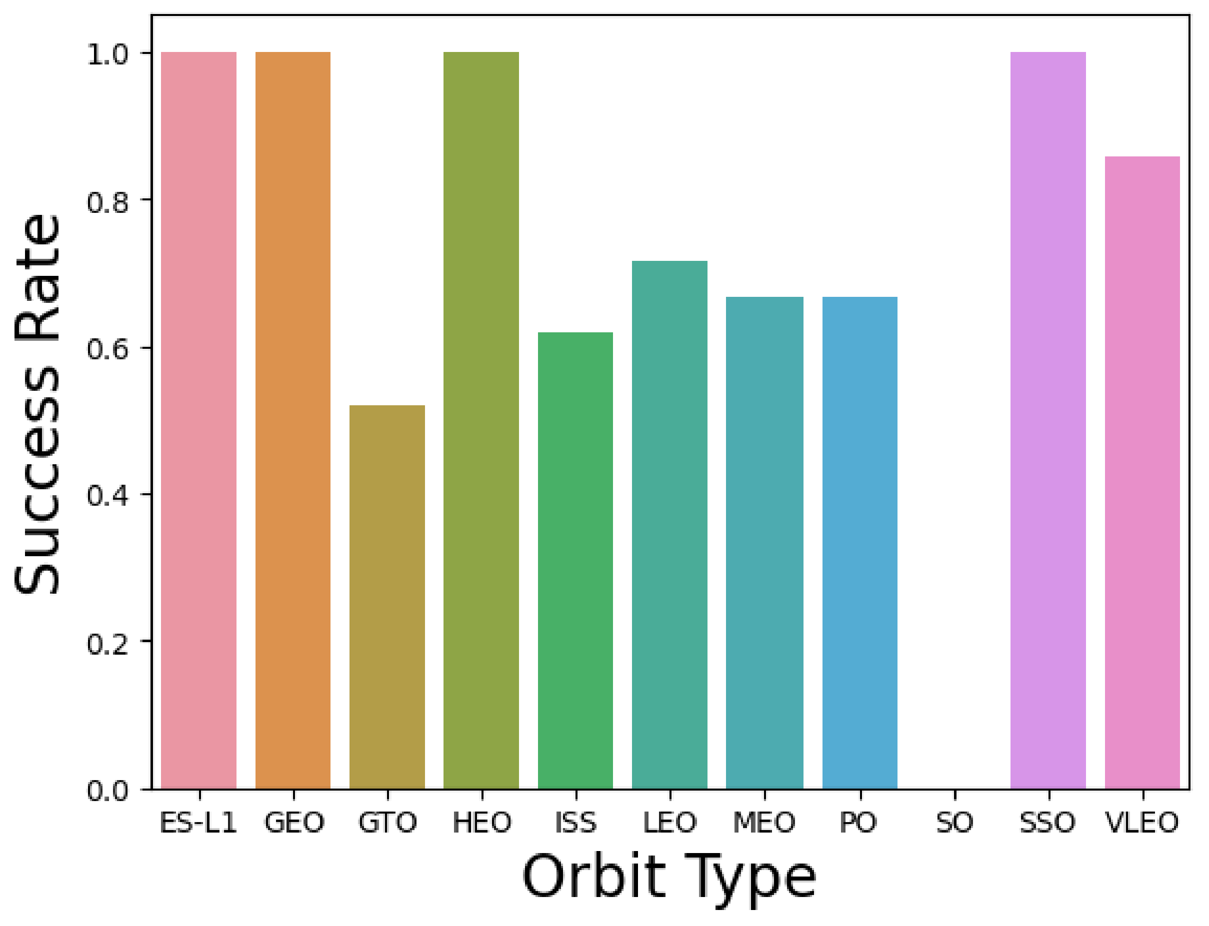
- Success rate increases with flight number for all launch sites.

Payload vs. Launch Site



- Success rate increases with payload mass for all launch sites.

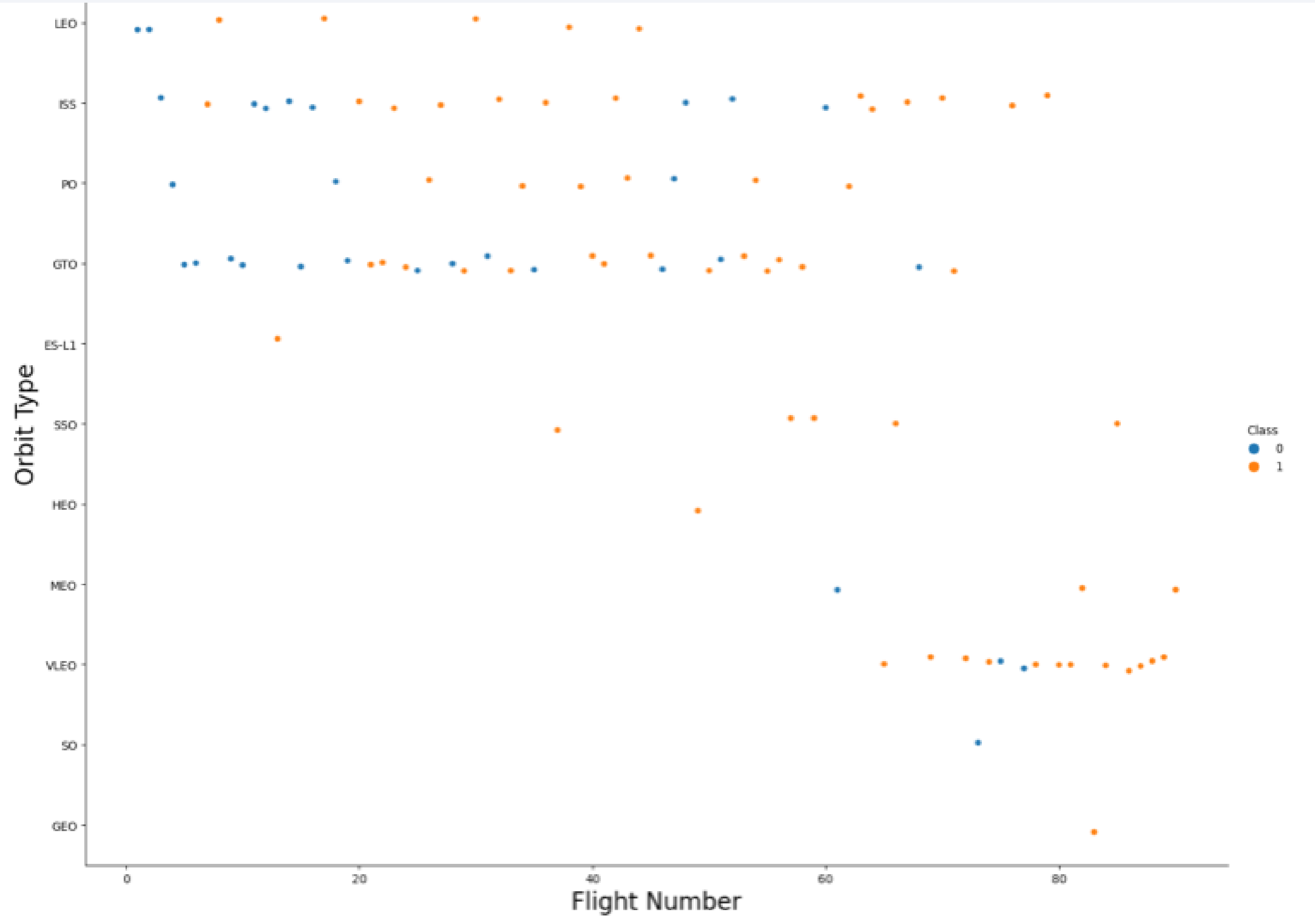
Success Rate vs. Orbit Type



- The best success rate (100%) is recorded with orbits ES-L1, GEO, HEO, SSO

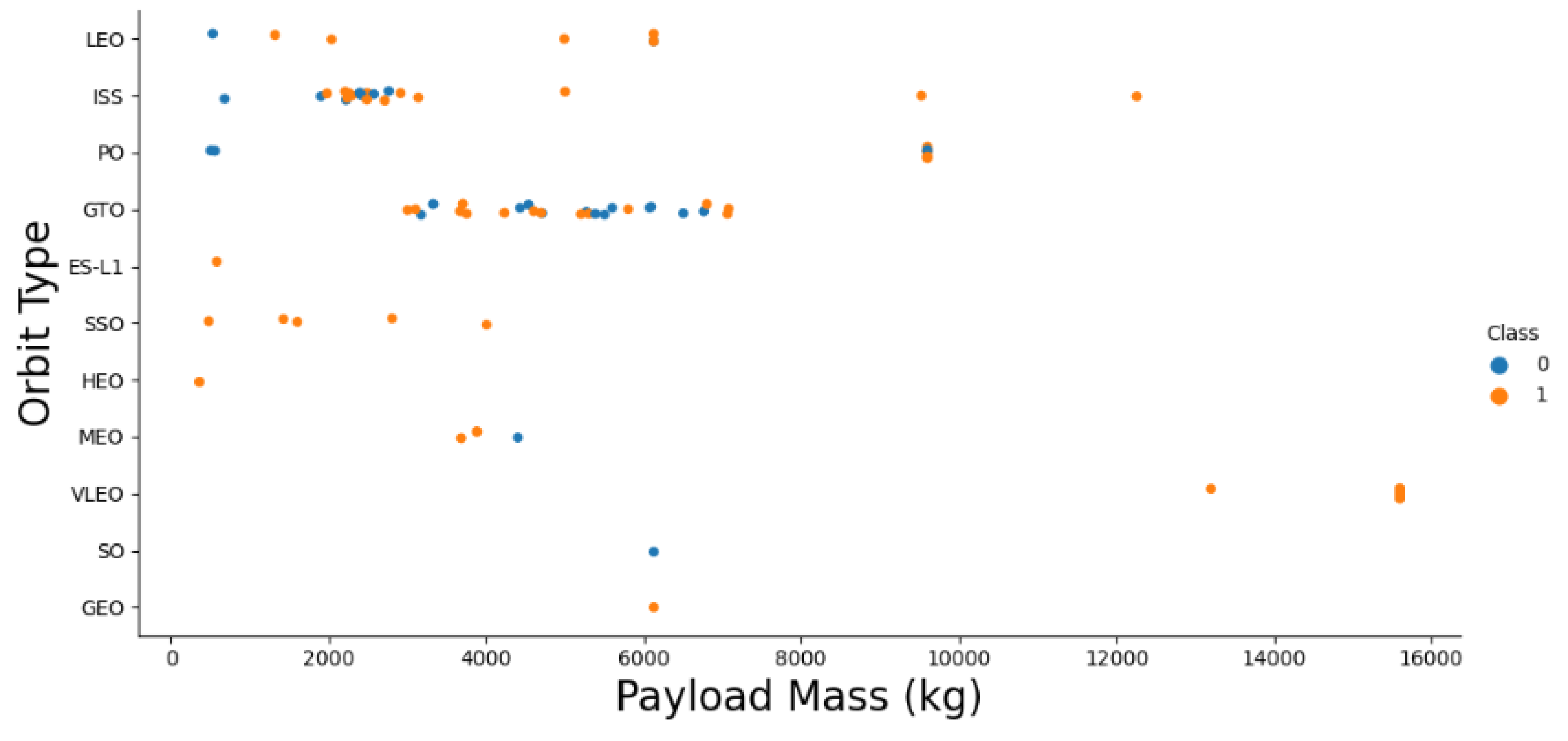
✓ **Orbit type has a significant effect on success rate**.

Flight Number vs. Orbit Type



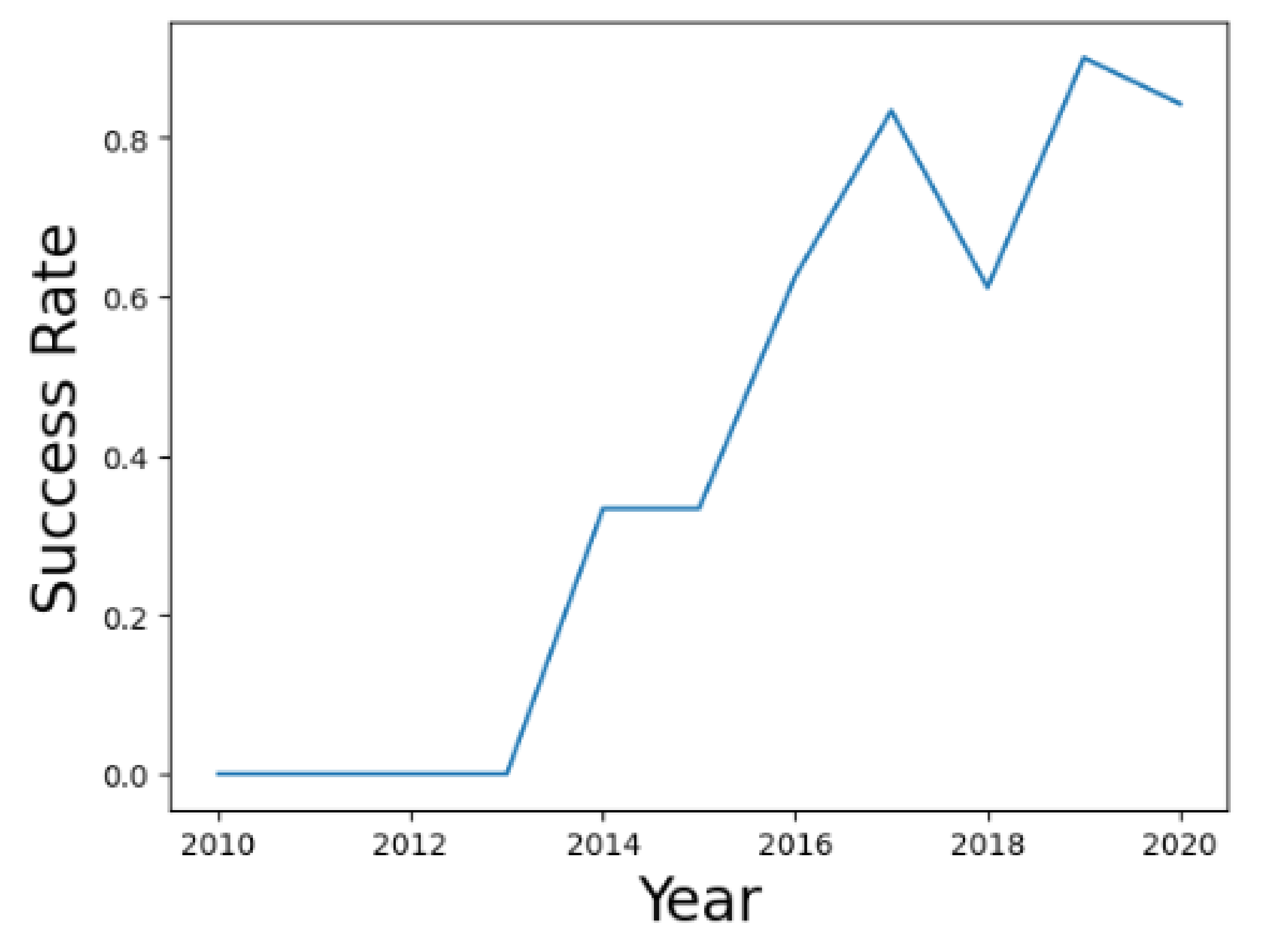
- Only orbits LEO and SSO clearly show that success rate increases with flight number

Payload vs. Orbit Type



- Success rate increases with payload mass (heaviness) only for orbits LEO, ISS, SSO AND VLEO.

Launch Success Yearly Trend



- Success rate is seen to be increasing from 2013 to 2020.

All Launch Site Names

```
%sql SELECT DISTINCT LAUNCH_SITE FROM spacextbl;
```

Query Explanation:

Display names of all unique site launches in the “LAUNCH_SITE” column from the “spacextbl” table

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT* FROM spacextbl WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Query Explanation:

Display only 5 launch site names beginning with ‘CCA’ from spacextbl in the LAUNCH_SITE column.

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM spacextbl WHERE CUSTOMER = 'NASA (CRS)';
```

Query Explanation:

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

1
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM spacextbl WHERE BOOSTER_VERSION = 'F9 v1.1';
```

1
2928

Query Explanation:

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

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) FROM spacextbl WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

2015-12-22

Query Explanation:

Display the date when the first successful landing outcome in ground pad was achieved

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT distinct Booster_Version FROM spacextbl WHERE LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

Query Explanation:

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

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

Total Number of Successful and Failure Mission Outcomes

```
q = pd.read_sql("select substr(Mission_Outcome,1,7) as Mission_Outcom
e, count(*) from spacexdata  group by 1", conn)
q
```

Query Explanation:

Display the total number of successful and failure mission outcomes

	Mission_Outcome	count(*)
0	Failure	1
1	Success	100

Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl);
```

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

Query Explanation:

Display the names of the booster_versions which have carried the maximum payload mass

2015 Launch Records

```
%sql select distinct Landing__Outcome, Booster_Version, Launch_Site from spacextbl where Landing__Outcome='Failure (drone ship)';
```

Query Explanation:

Display the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 FT B1020	CCAFS LC-40
Failure (drone ship)	F9 FT B1024	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E

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

```
%sql select Landing__Outcome, count(*) from spacextbl where Date between '04-06-2011' and '20-03-2017' group by Landing__Outcome order by 2 desc;
```

Query Explanation:

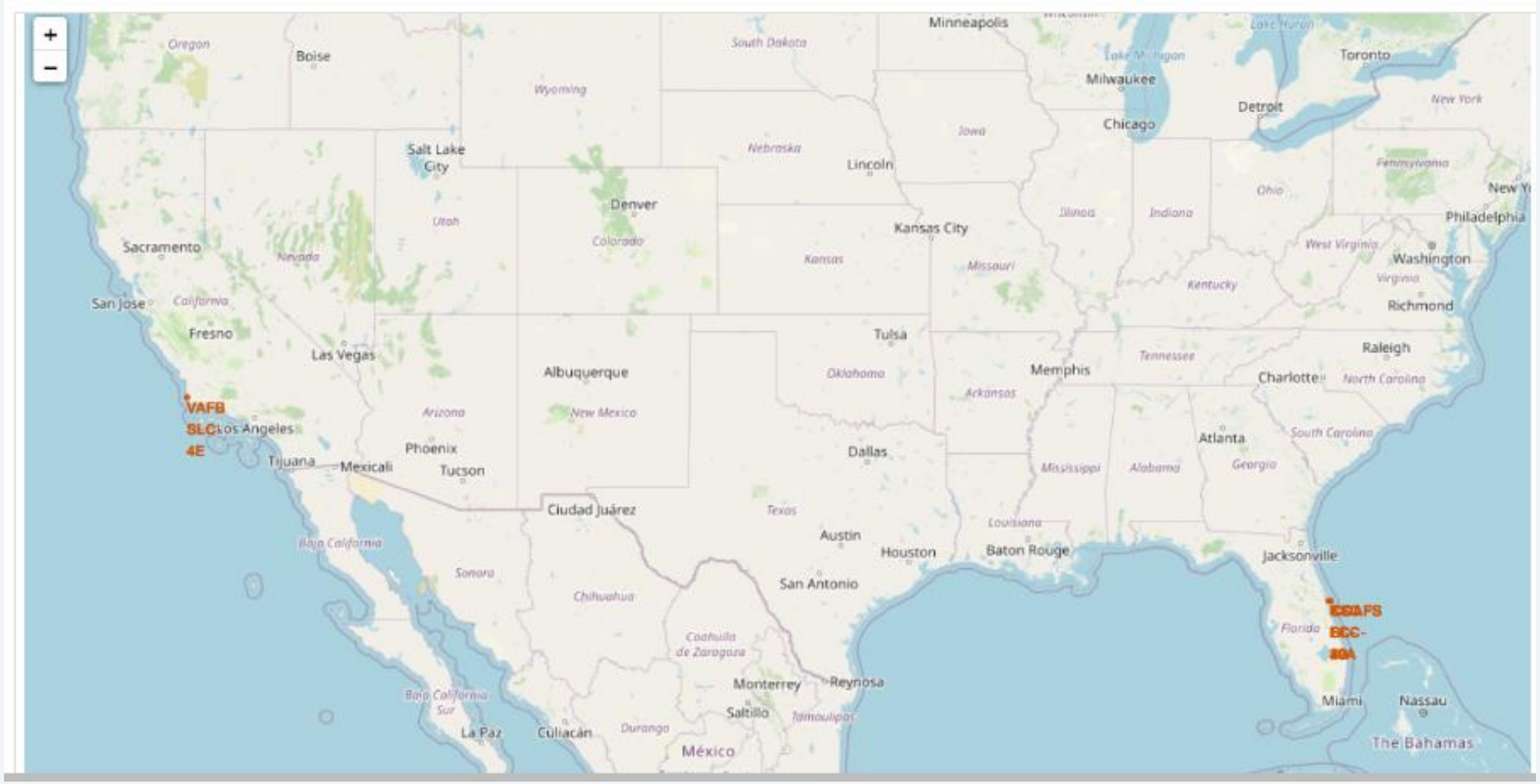
Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Landing__Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

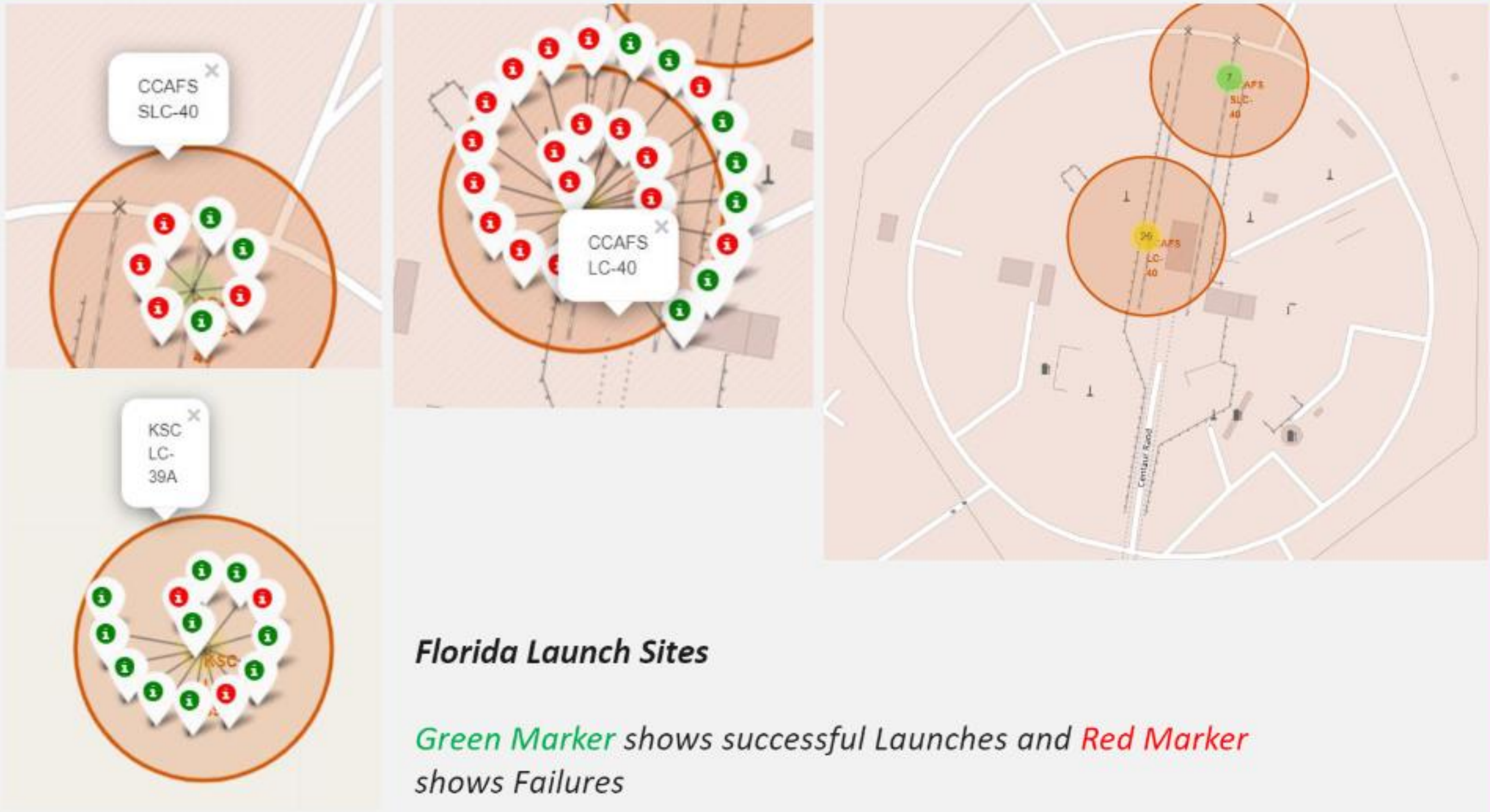
<Folium Map Screenshot 1>



Findings:

All launch sites are in very close proximity to the coast

<Folium Map Screenshot 2>



Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

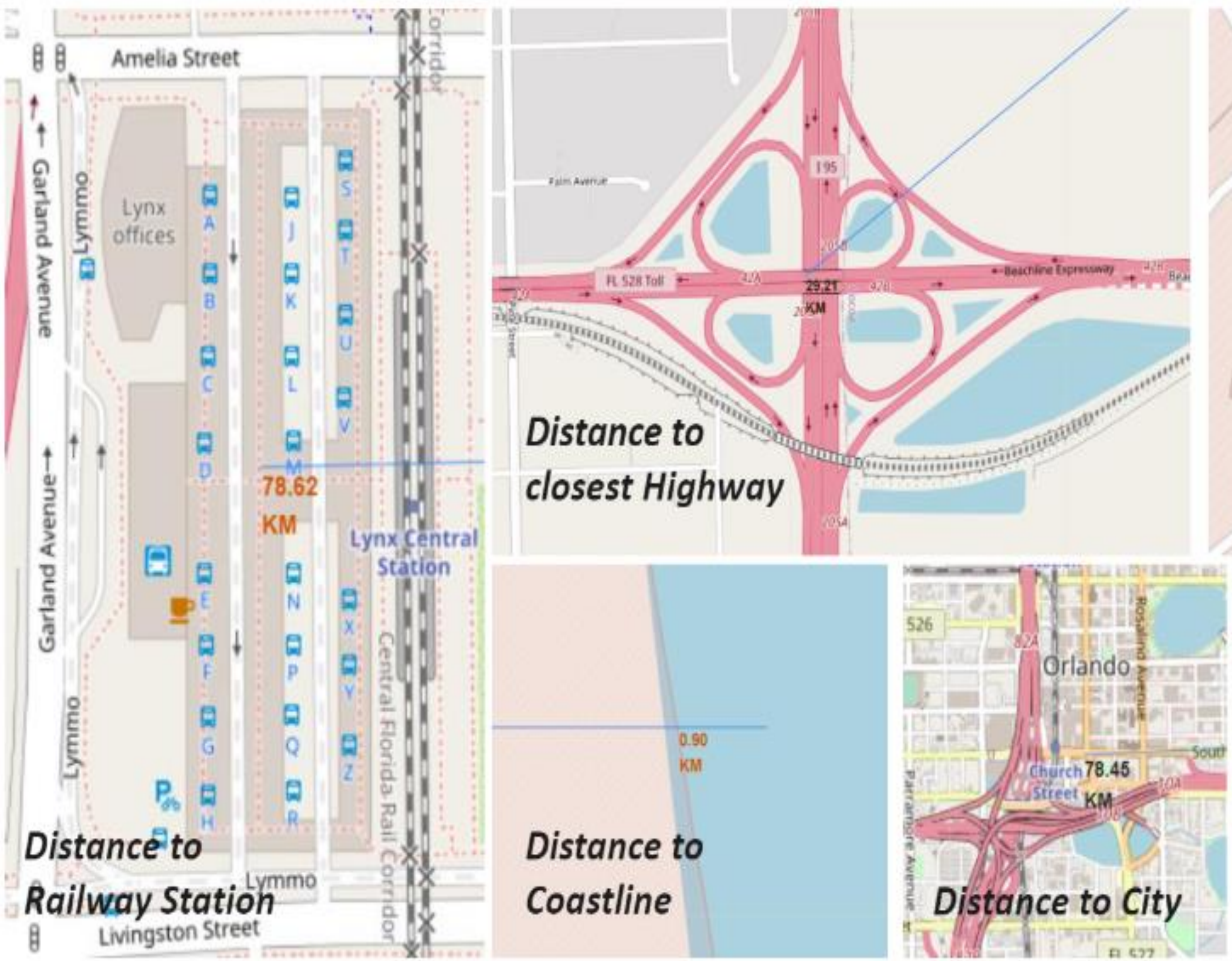
California Launch Site

37

Findings:

- Green: success
- Red: failure

<Folium Map Screenshot 3>

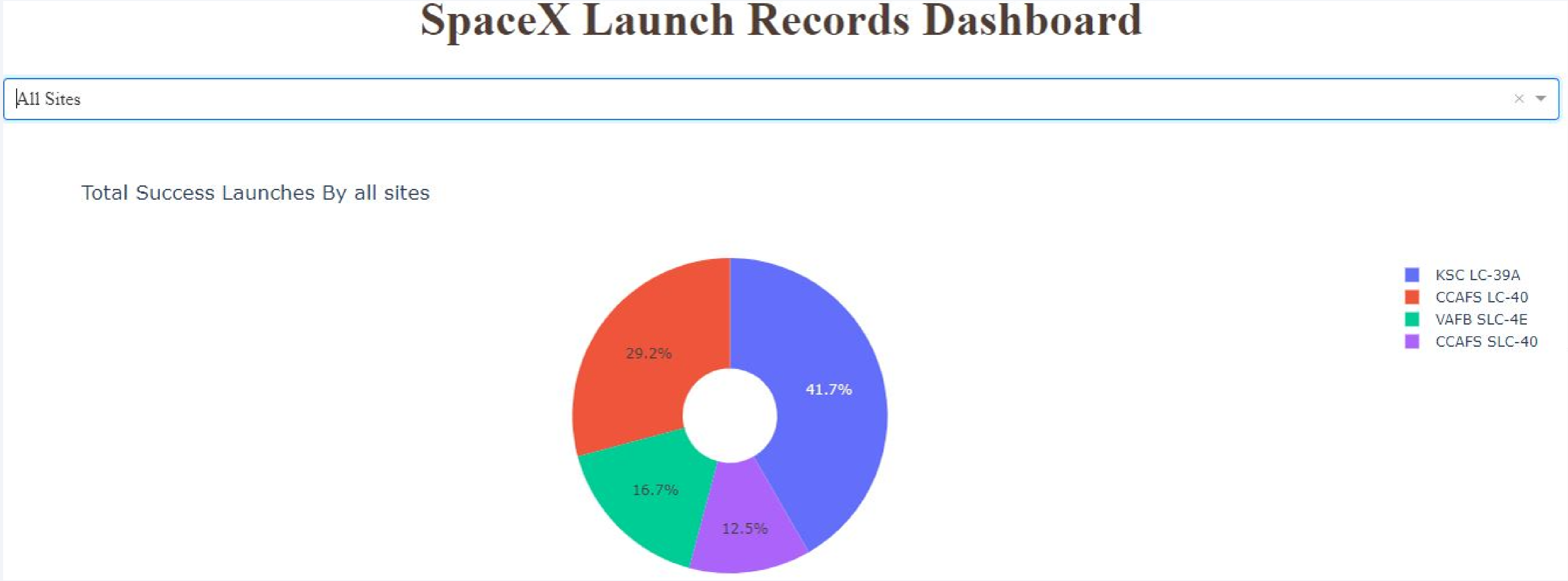


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

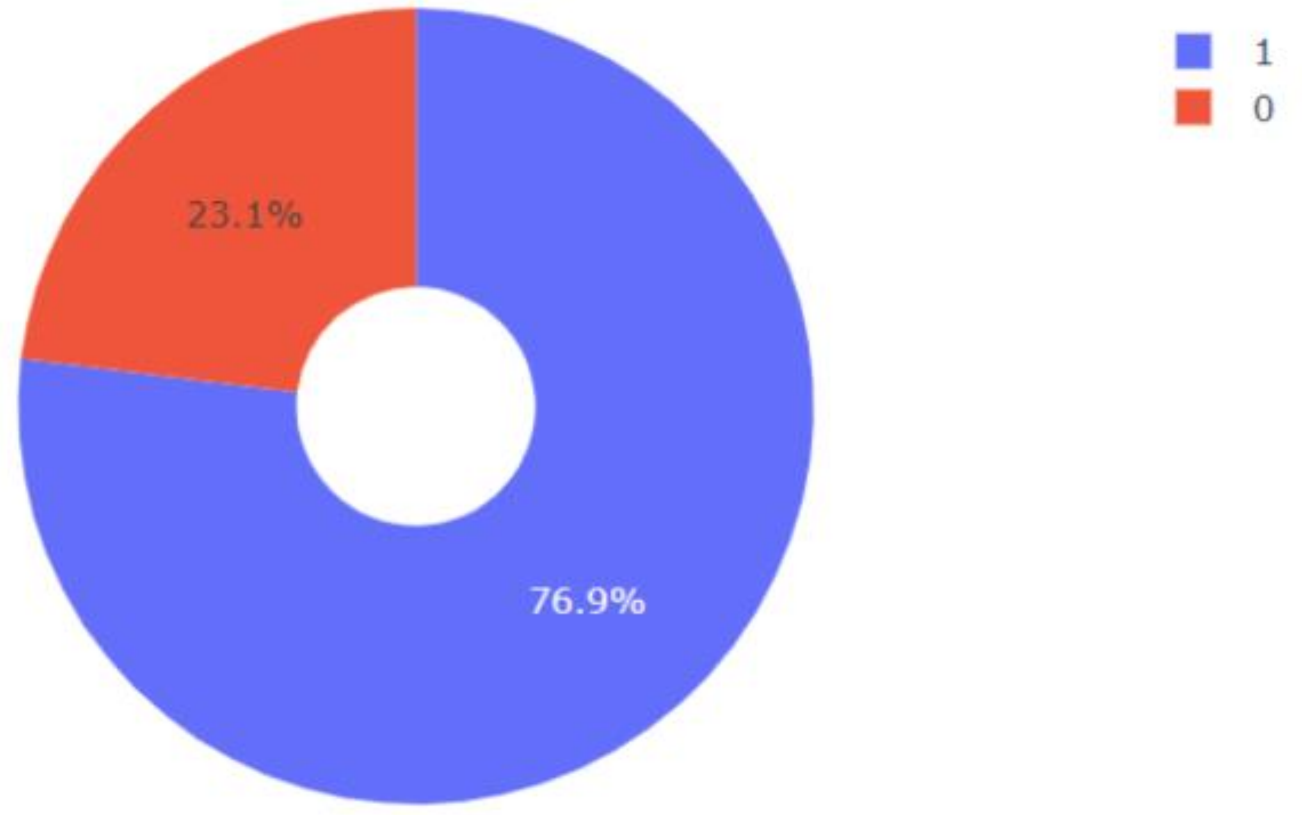
Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



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

<Dashboard Screenshot 3>

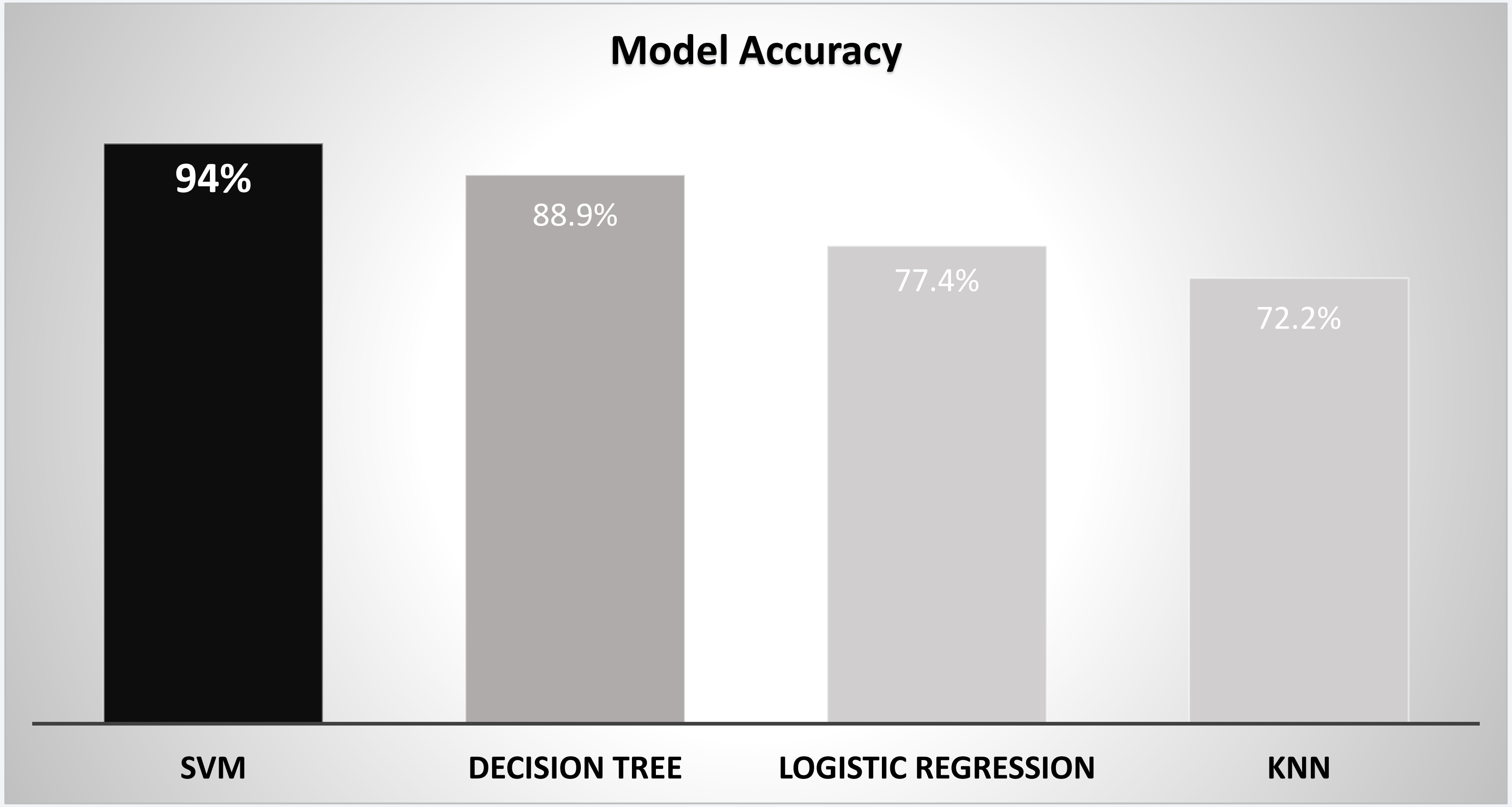
Findings:

- Payloads in range 2900 to 7000 kg have been the most successful but this depends on the booster version.

Section 5

Predictive Analysis (Classification)

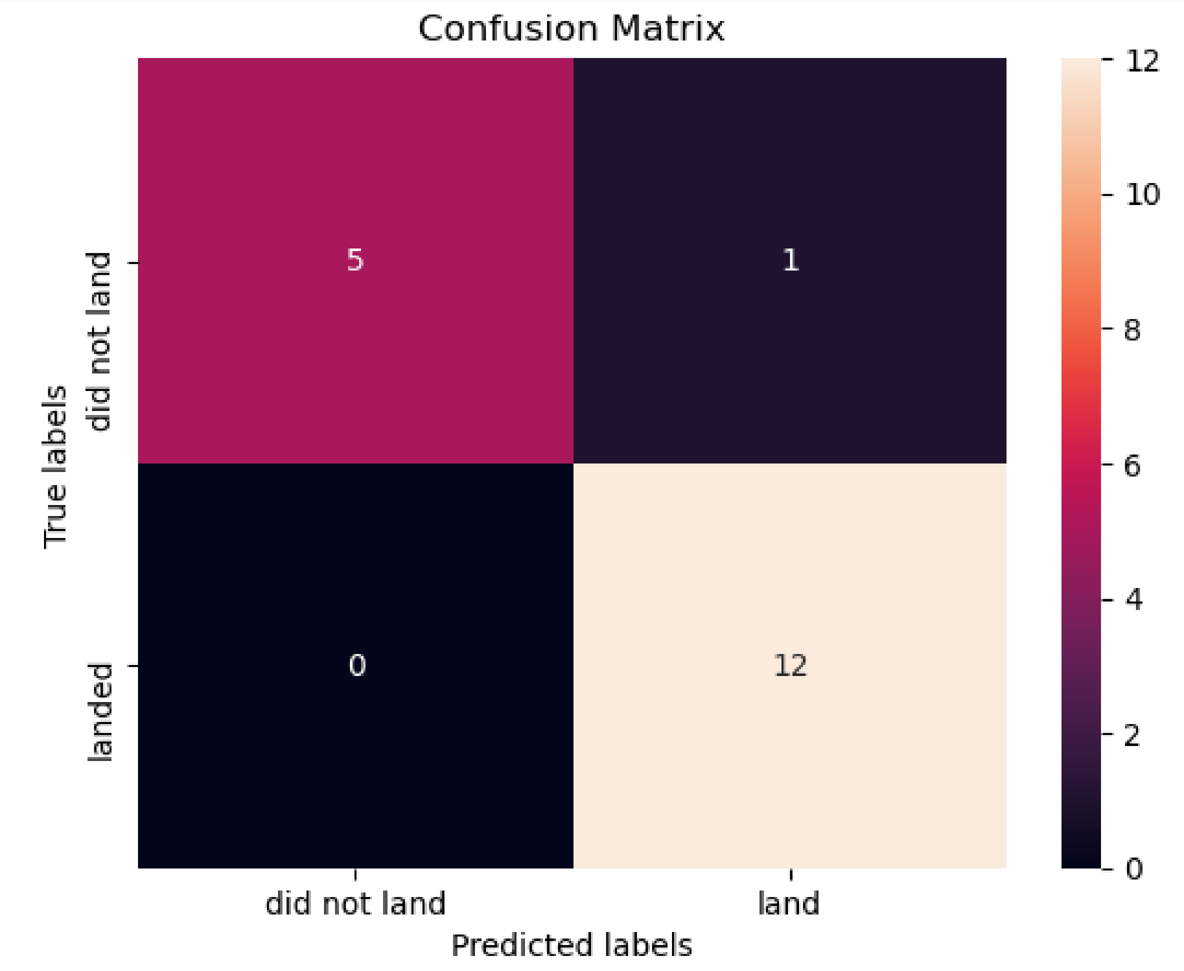
Classification Accuracy



Findings:

The **SVM** model had the highest classification accuracy of **94%**.

Confusion Matrix



SVM Confusion matrix:

- Land: 12 times correctly predicted out of 12.
- Did not land: 5 times correctly predicted out of 5
- Wrong prediction(s): 1

The model correctly predicted 17 times out 18.

Conclusions

In conclusion:

- The larger the flight amount 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, and SSO had 100% success rate while VLEO had over 80%.
- KSC LC-39A had the most successful launches of all sites.
- The SVM classifier was the best machine learning algorithm with 94% accuracy.

Appendix

- Dataset 1 link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/dataset_part_1.csv
- Dataset 2 link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/dataset_part_2.csv
- Dataset 3 link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/dataset_part_3.csv
- Web-scraped data link: https://github.com/GFolomo/-IBM-Data-Science-Capstone-SpaceX/blob/7884dfccab105ad28e1412ae7398af628c51ae66/spacex_web_scraped.csv

Thank you!