IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Potjamarn Arpornratn
10/17/2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- This report is created to predict if SpaceX Falcon 9 First stage will land successfully. This information will be used to determine the cost of the launch for bidding against SpaceX.
- We want to predict whether Falcon 9 First Stage would land successfully so that we can determine the cost of the launch. If Falcon 9 First Stage could be reused, then the cost will be cheaper.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was extracted from multiple sources: SpaceX API, wikipedia, and database.
- Perform data wrangling
  - Data wrangling will be performed after the data was collected. We will examine the data: remove NaN, clean, and reformat the data and
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Skilearn models (LogisticRegression, SVC, DecisionTree, and KNeighborsClassifier) were used to model the data. GridSearchCV is used for hyperparameter tuning. Then, score() method were used to score each model and performed predictions.

Data Collection

The data were collected from several sources: SpaceX REST API endpoints, past rocket launch data from SpaceX, and webscraping on wikipedia page.

You need to present your data collection process use key phrases and flowcharts

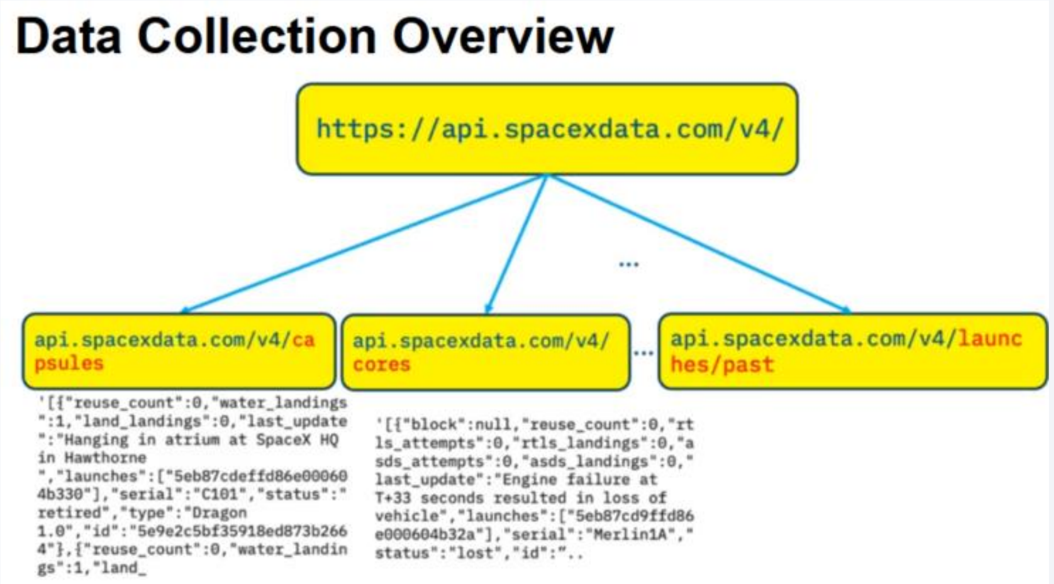
Data Collection – SpaceX API

- First set of data was collected using SpaceX REST calls on the past endpoint.

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

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

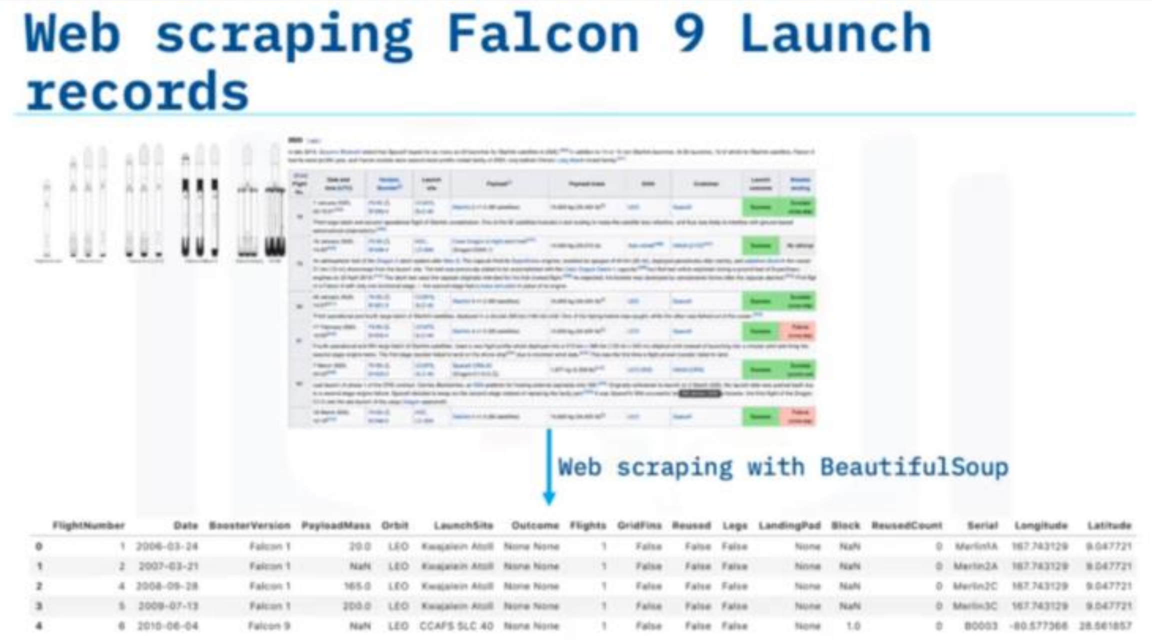
- GitHub URL of the completed SpaceX API calls notebook: [See jupyter-labs-spacex-data-collection-api.ipynb]



Data Collection - Scraping

- Web scraping process started from fetching the data then parsed response data using BeautifulSoup

- See "Jupyter Notebook on GitHub URL" (click here) of the completed web scraping notebook, as an external reference and peer-review purpose




```python
# use requests.get() method with the provided static_url and headers
# assign the response to a object
response = requests.get(static_url, headers=headers)
response.status_code
```

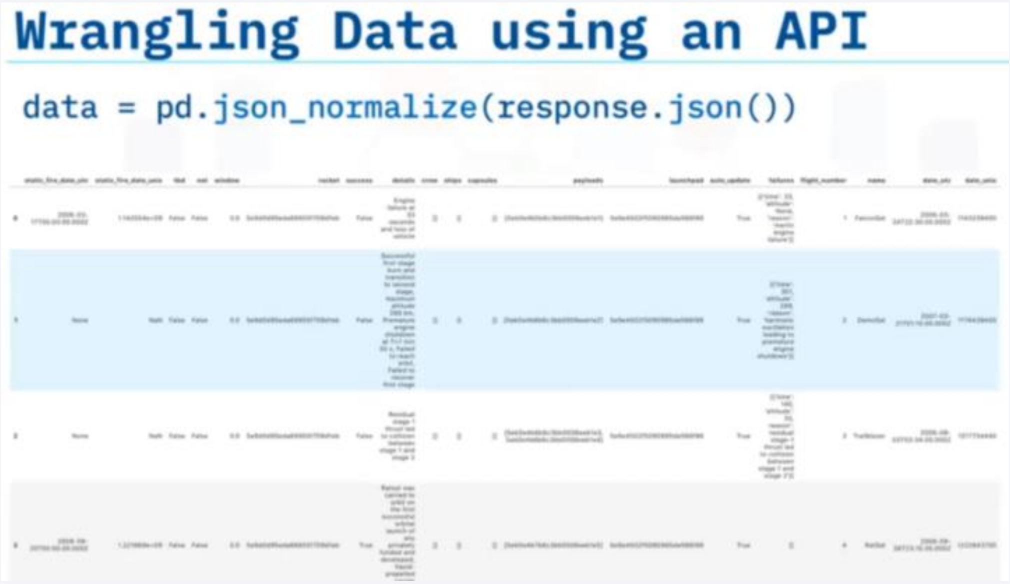
```
200
```

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
beautiful_soup = BeautifulSoup(response.text, 'html.parser')
```

Data Wrangling

The data was cleaned, NaN (if exists) was removed. Numerical and Categorical columns were identified and processed. Landing outcome label were created and assigned to Class column.

Here is GitHub URLof completed data wrangling related notebooks, as an external reference and peer-review purpose.



EDA with Data Visualization

Scatter charts were created for visualization of: FlightNumber vs. PayloadMass, Flight Number and Launch Site, Payload Mass and Launch Site, success rate of each orbit, FlightNumber and Orbit, Payload Mass and Orbit, and launch success yearly trend. After EDA, Feature Engineering was performed on the insights.

[See GitHub URL]() of completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

First, blank rows were removed from the table, then the following SQL queries were performed:

- Fetch names of the unique launch sites in the space mission
- Fetch records where launch sites begin with the string 'CCA'
- Fetch total payload mass carried by boosters launched by NASA (CRS)
- Fetch average payload mass carried by booster version F9 v1.1
- Fetch the date when the first successful landing outcome in ground pad was achieved.
- Fetch a list of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Fetch total number of successful and failure mission outcomes
- Fetch all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- 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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

See GitHub URL of completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

All launch sites along with success/failed rate were marked on the map using MarkerCluster, circles, markers, and MousePosition. Then proximity lines were added to the map using PolyLine objects to show the distance between coastline point and launch site.

Folium.Circle is used to add a highlighted circle area with a text label on a specific coordinate. Markers were created and added to the map to show all launch outcomes. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)

Explain why you added those objects

See GitHub URL (here) of completed interactive map with Folium map, as an external reference and peer-review purpose. Note: use this link to view the interactive notebook: https://nbviewer.org/

Build a Dashboard with Plotly Dash

Interactive Dashboard was built using Plotly Dash with the following functionality:

→ Launch Site drop-down (with pre-populated launch sites) and Callback functions to render pie chart for selected option. Pie chart displays successful launches by Site (if All Site option is selected) or success launches per selected site.

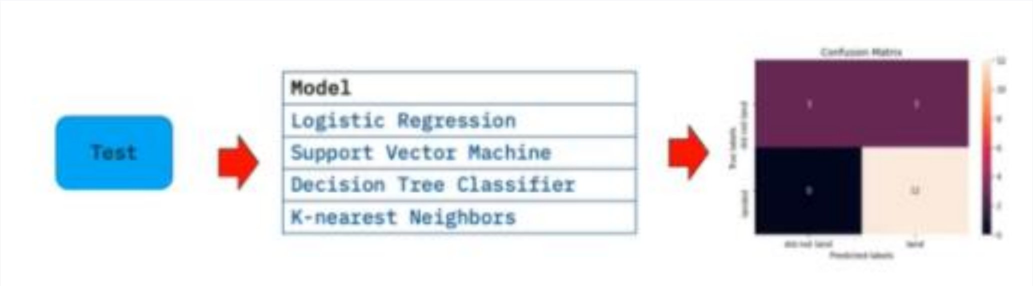
→ Range Slider for selected Payload

→ Callback function to render Scatter plot for selected launch site displays successful mission related to selected Slider Payload.

See GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

Machine learning pipeline was built using different models such as: Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. The models were evaluated, then hyperparameters tuned using GridSearchCV. Accuracy score and confusion matrix were generated for each model.



** Test score and accuracy (model.best_score_) score were captured for comparison.

See GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

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 scatter plots on the right, you can see that as the flight number increase, the success rate also increase. CCAFS SLC-40 has more success rate than other launch sites when flight number > 80

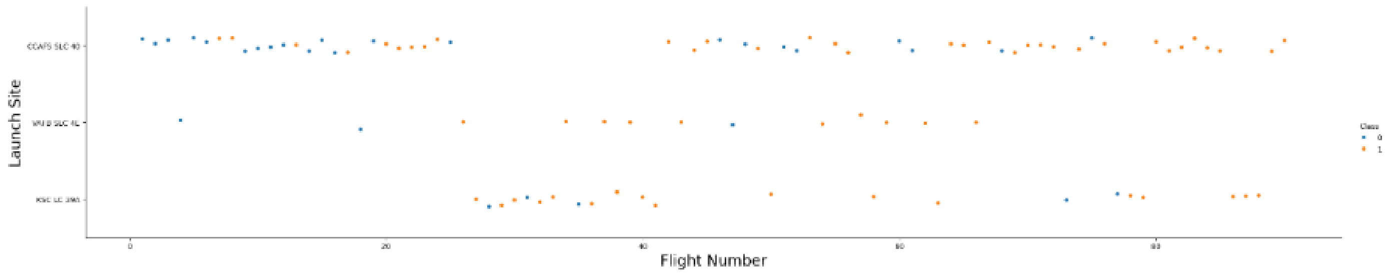
VAFB SLC-4E site has around 70 flights

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

In [5]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class va
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

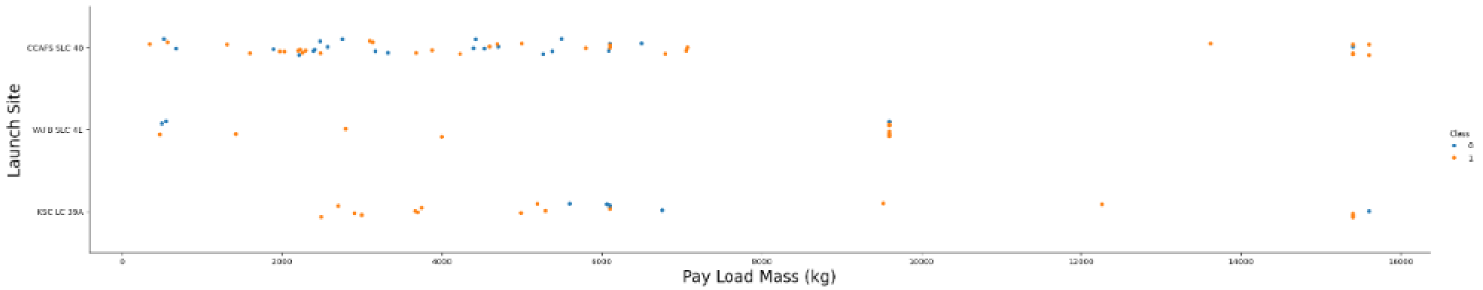
- Notice that CCAFS SLC-40 and KSC LC-39A launch sites has success rate with heavy PayloadMass (around 16,000 kgs)
- VAFB SLC-4E has no launch for heavy PayloadMass > 10,000 kgs

TASK 2: Visualize the relationship between Payload Mass and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

[6]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

- ToDo: Add observation

TASK 3: Visualize the relationship between success rate of each orbit type

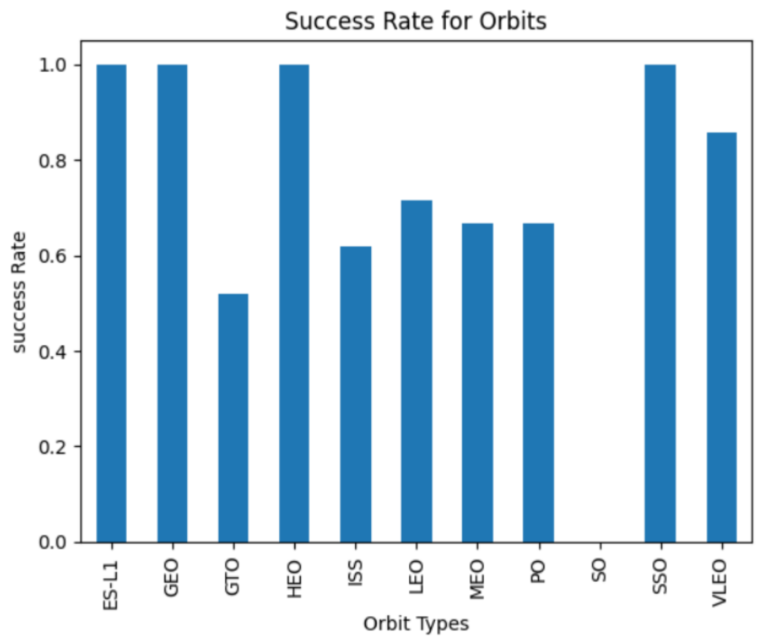
Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

```
# HINT use groupby method on Orbit column and get the mean of Class column
mean_per_orbit = df.groupby('Orbit')['Class'].mean()

mean_per_orbit.plot(kind='bar')
plt.title('Success Rate for Orbits')
plt.xlabel('Orbit Types')
plt.ylabel('success Rate')
```

```
Text(0, 0.5, 'success Rate')
```



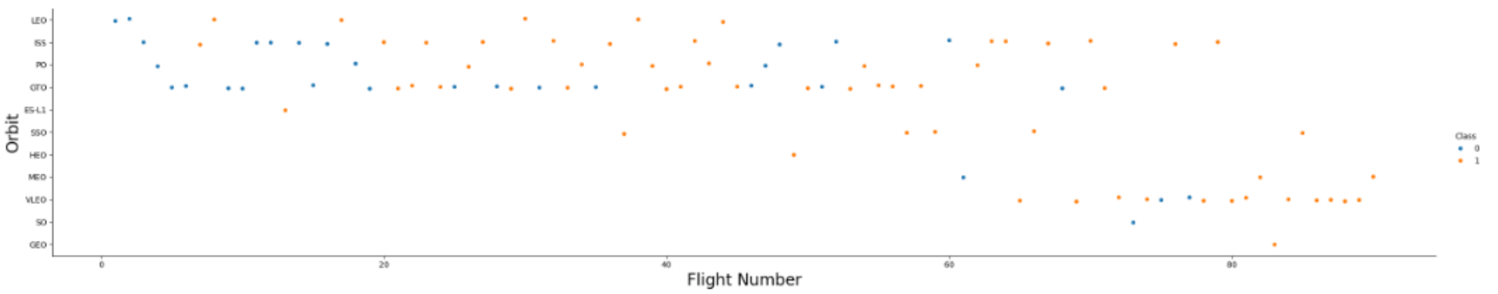
Flight Number vs. Orbit Type

- ToDo: Add observation

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class valu
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

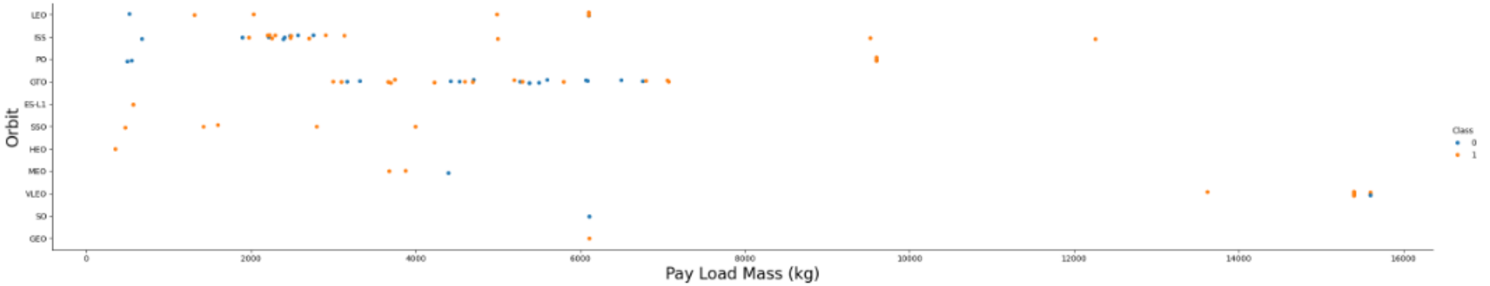
- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

TASK 5: Visualize the relationship between Payload Mass and Orbit type

Similarly, we can plot the Payload Mass vs. Orbit scatter point charts to reveal the relationship between Payload Mass and Orbit type

```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class valu
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



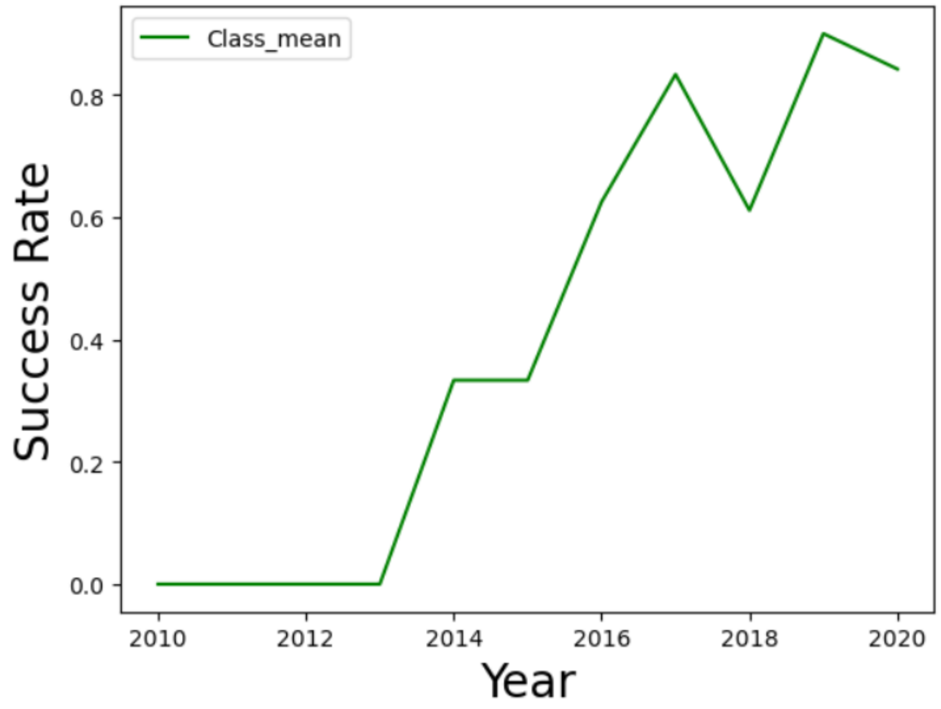
Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
yearly_success = df.groupby('Date', as_index=False)['Class'].mean()
yearly_success['Date'] = yearly_success['Date'].astype(int)
yearly_success.columns = ['Date', 'Class_mean']
yearly_success

yearly_success.plot(x='Date', y='Class_mean', color='green')
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

Find the names of the unique launch sites

ToDo: Add short explanation

```
%sql select distinct Launch_Site from SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

ToDo: Add short explanation here

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
```

```
 * sqlite:///my_data1.db
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer in ('NASA (CRS)');
```

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

sum(PAYLOAD_MASS__KG_)
45596

Todo: Add short explanation here

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version in ('F9 v1.1')
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

ToDo: Add short explanation here

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad

```
%sql select MIN(Date) from SPACEXTABLE where Landing_Outcome in ('Success (ground pad)')
```

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

MIN(Date)
2015-12-22

ToDo: Add short explanation here

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Query: %sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ > 4000 ar
```

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

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

ToDo: Add short explanation here

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

```
%sql select substr(Mission_Outcome, 1, 7) as Mission_Outcome,  count(*) from SPACEXTABLE group by 1;
```

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

Mission_Outcome	count(*)
Failure	1
Success	100

ToDo: Add short explanation here

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass

Query: %sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ in (select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)

ToDo: Add short explanation here

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

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

```
%sql select substr(Date, 6, 2) as Month, Booster_Version, Launch_Site from SPACEXTABLE \
    where Landing_Outcome = 'Failure (drone ship)' and Date in (select Date from SPACEXTABLE where substr(Date,0,5)='2015')
```

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Present your query result with a short explanation here

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

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

Query: %sql select Landing_Outcome, count(*) from SPACEXTABLE group by Landing_Outcome having Date between '2010-06-04' and '2017-03-20' order by 2 desc

ToDo: Add short explanation here

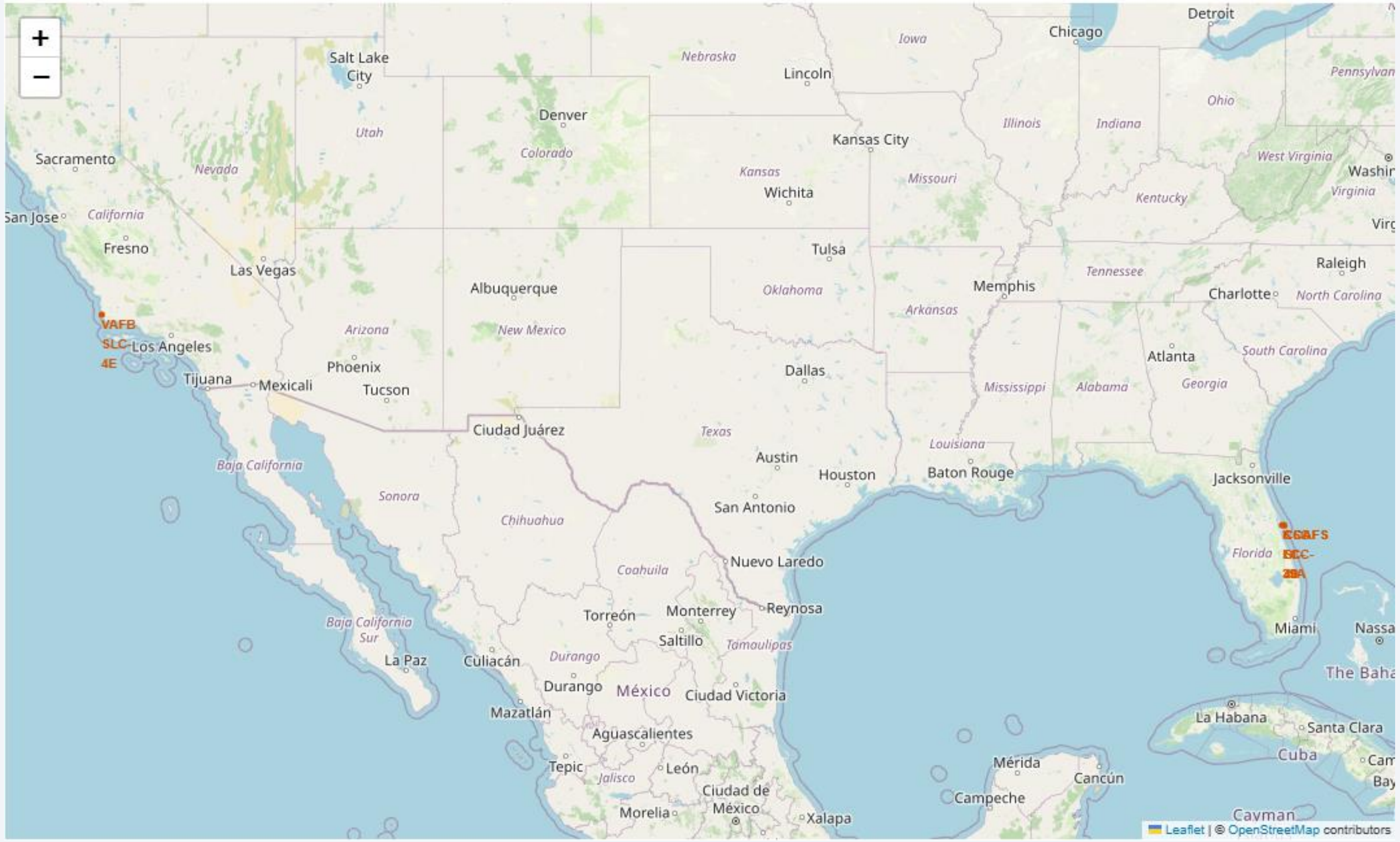
Landing_Outcome	count(*)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Initial SpaceX Launch Sites Map

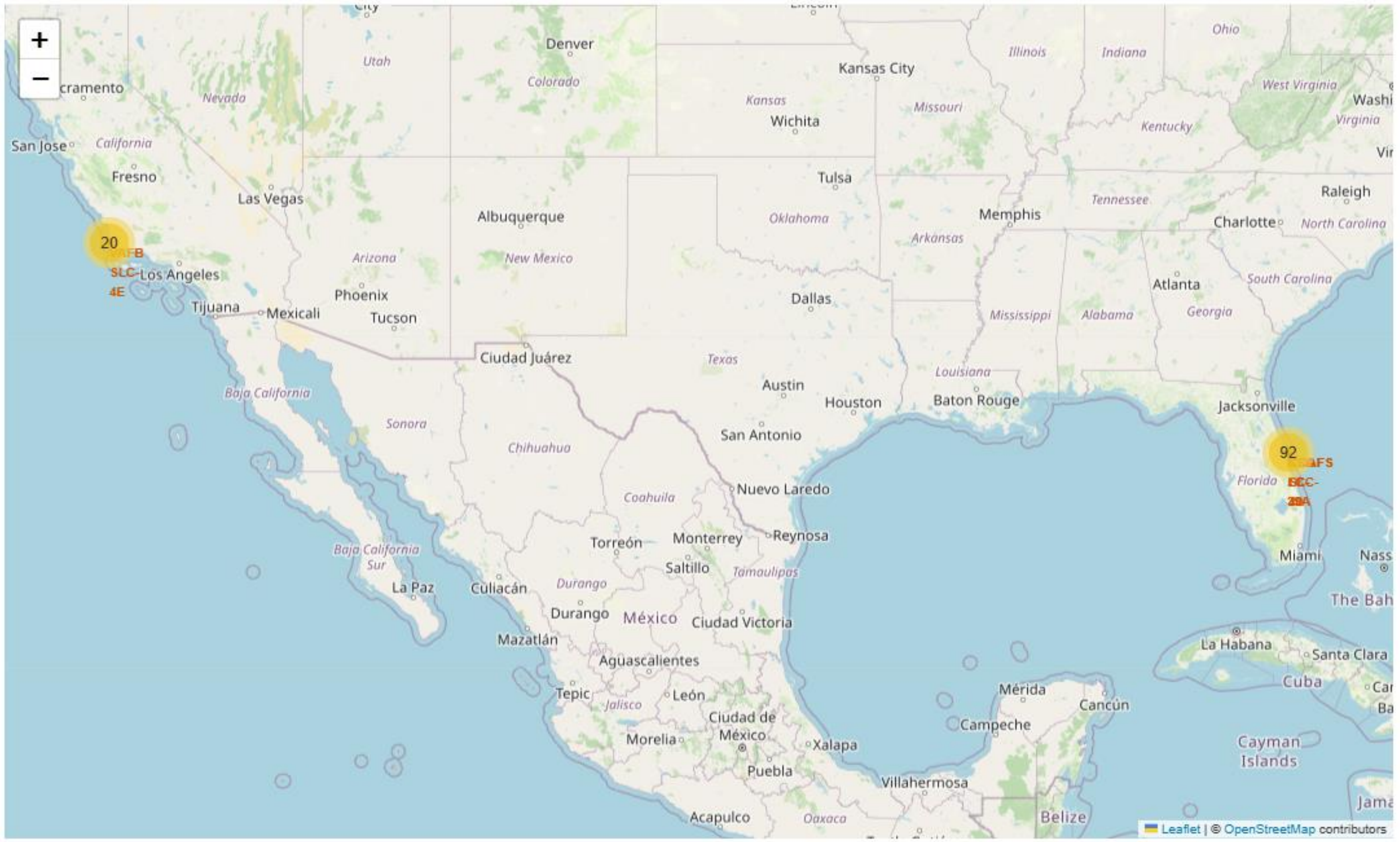
Explain the important elements and findings on the screenshot



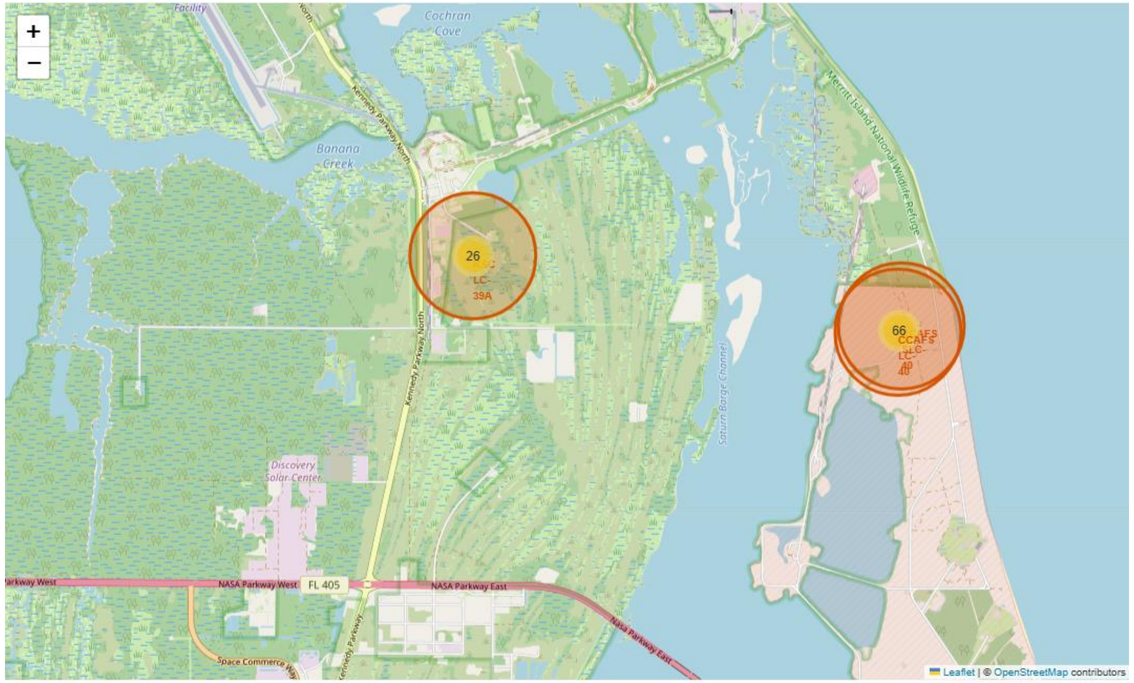
Map with success/failed launches marked

Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

Explain the important elements and findings on the screenshot

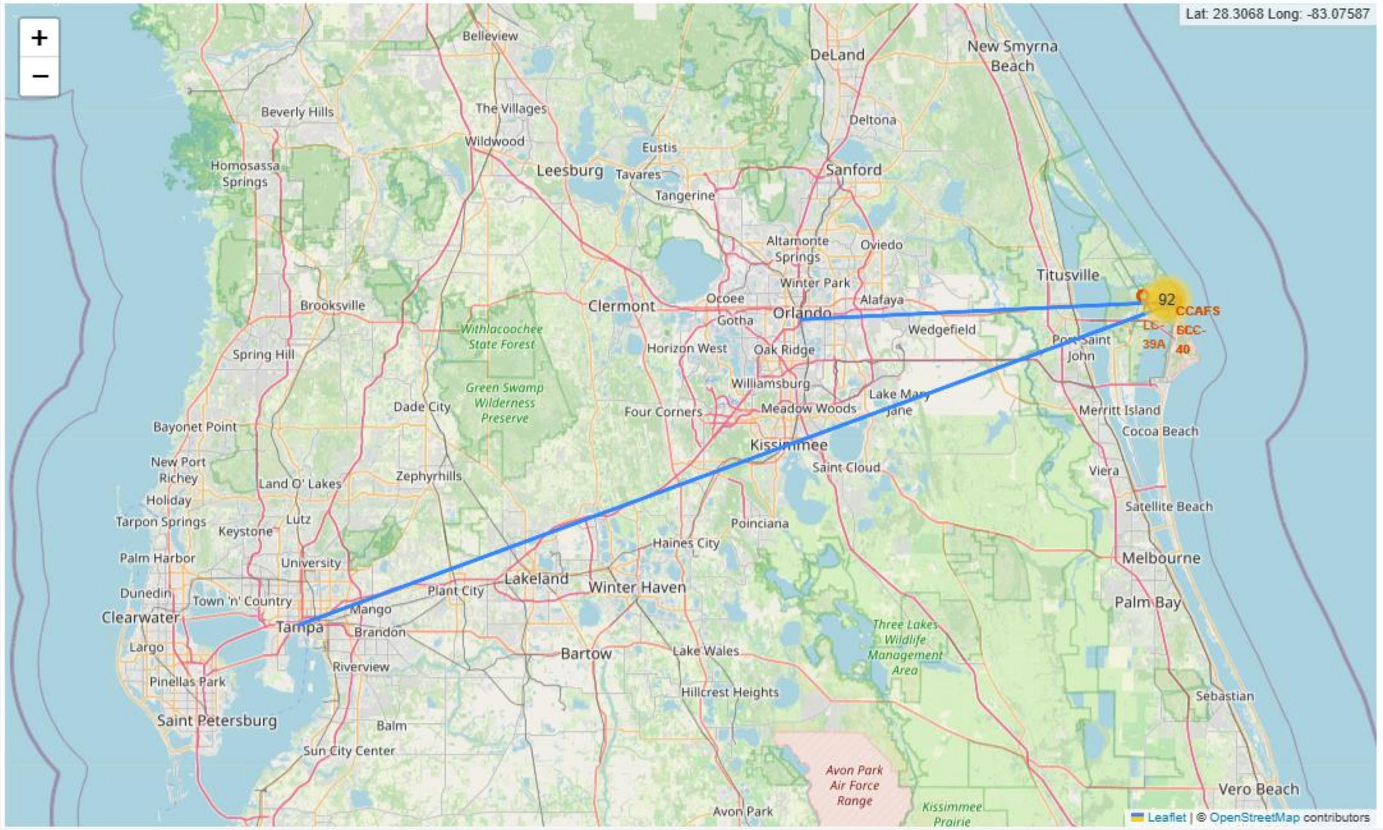


Map with success/failed launches marked (cont.)



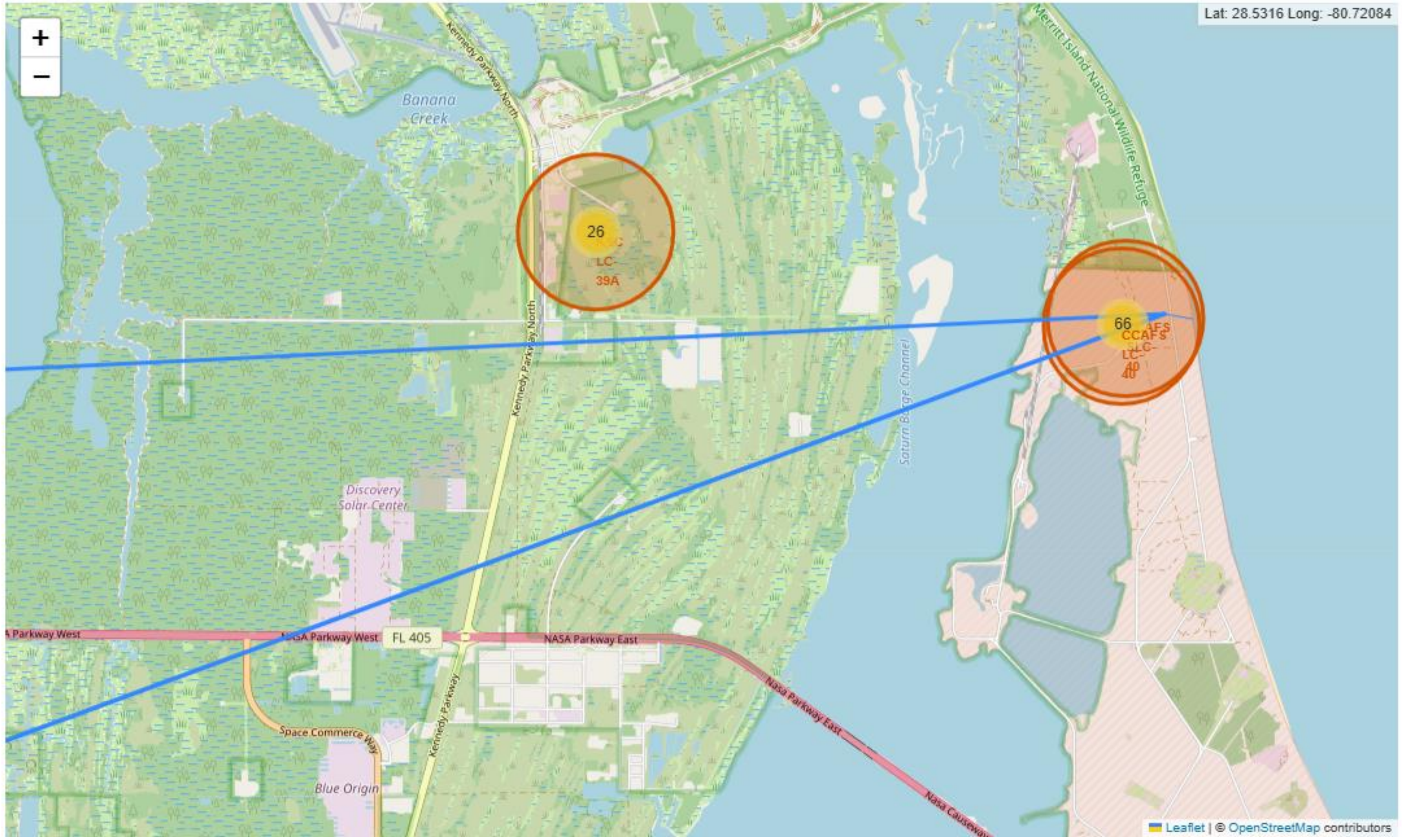
Calculate the distances between a launch site to its proximities

Explain the important elements and findings on the screenshot



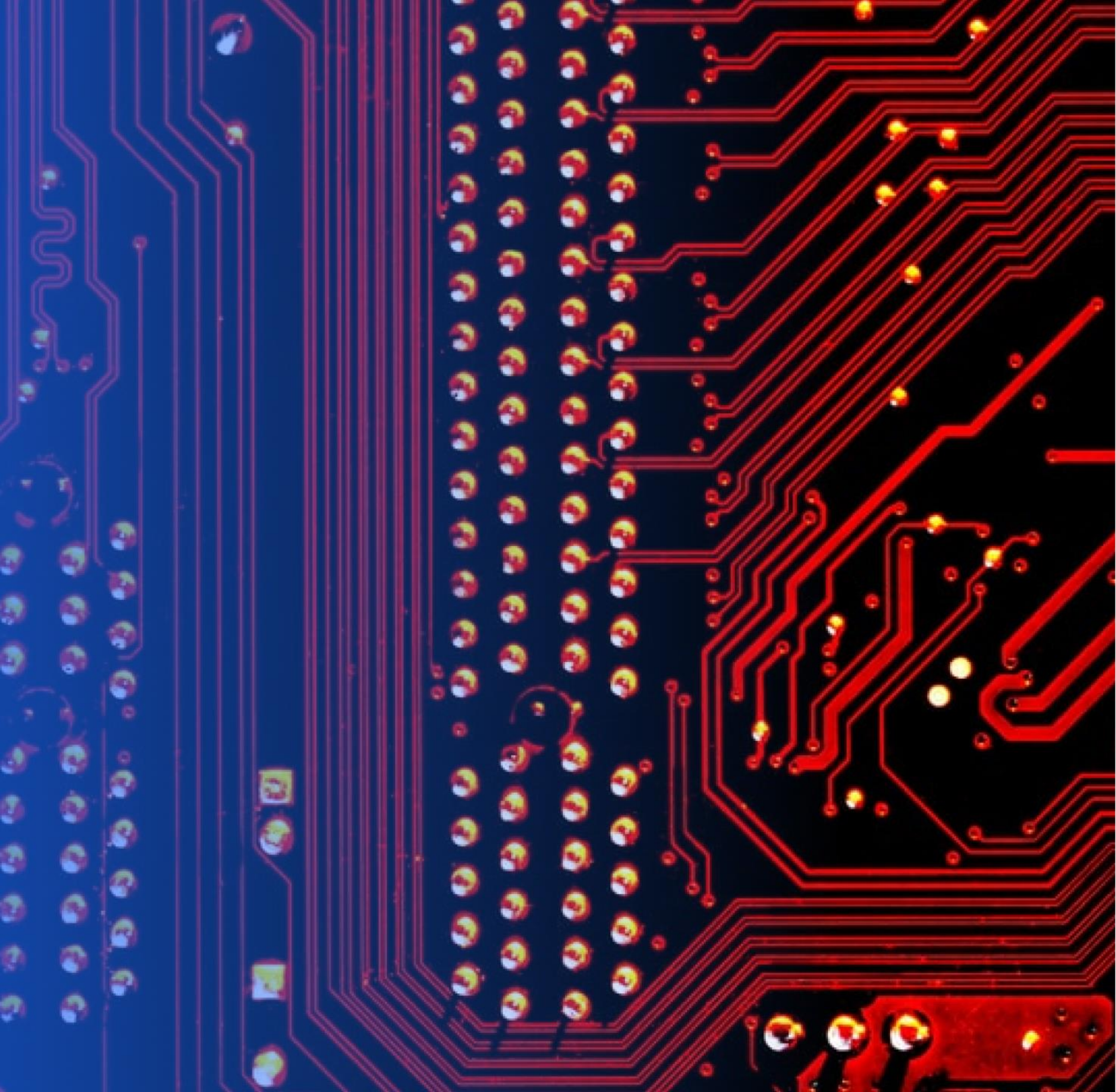
Note: Additional screenshot on the next slide

Calculate the distances between a launch site to its proximities (cont.)



Section 4

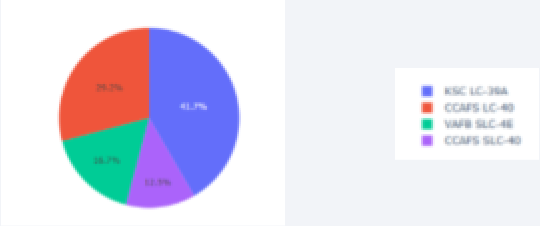
Build a Dashboard with Plotly Dash

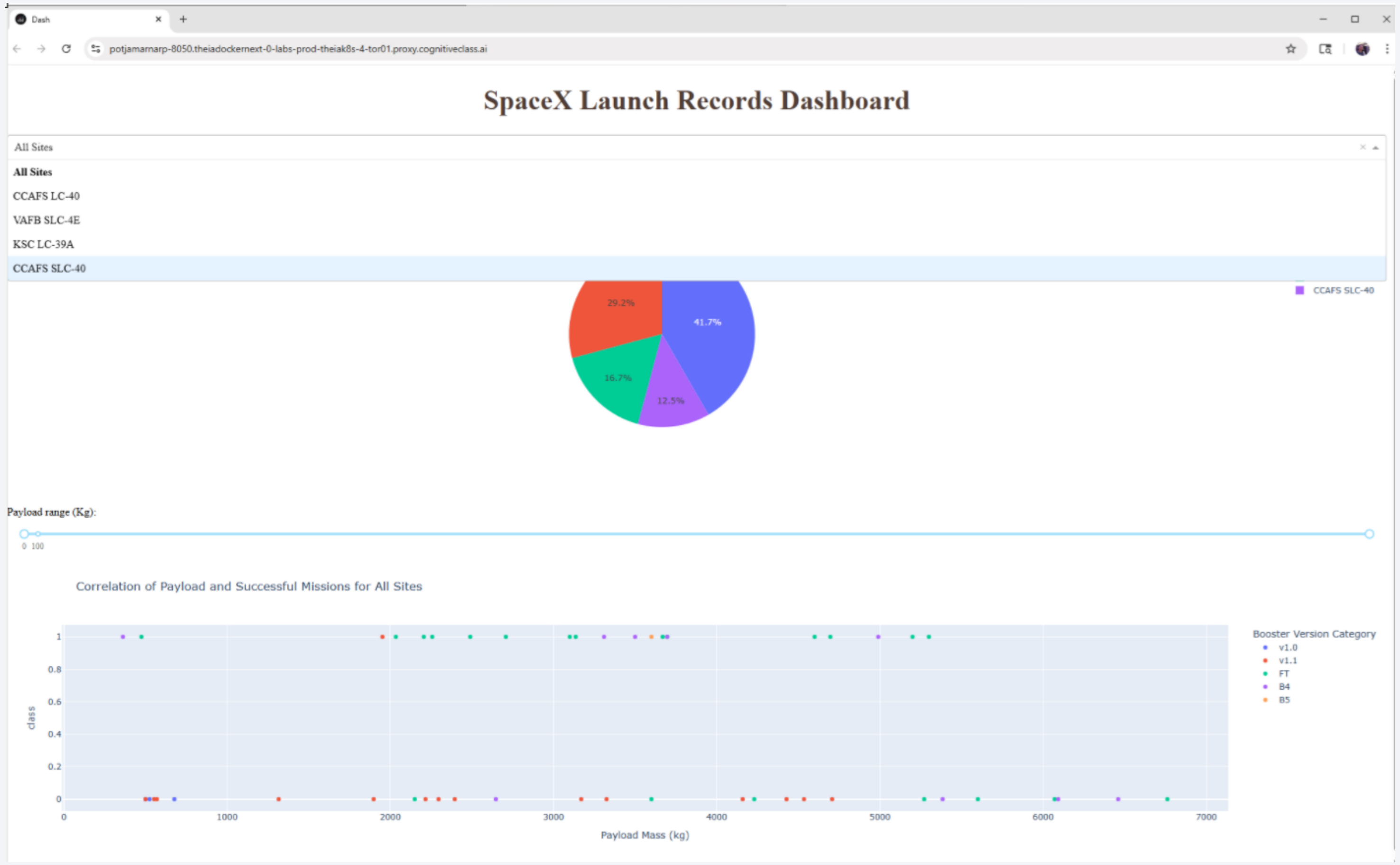


Dashboard with Site Drop-down

Screenshot of launch success count for all sites, in a pie chart

Explain the important elements and findings on the screenshot





Launch site with highest launch success ratio

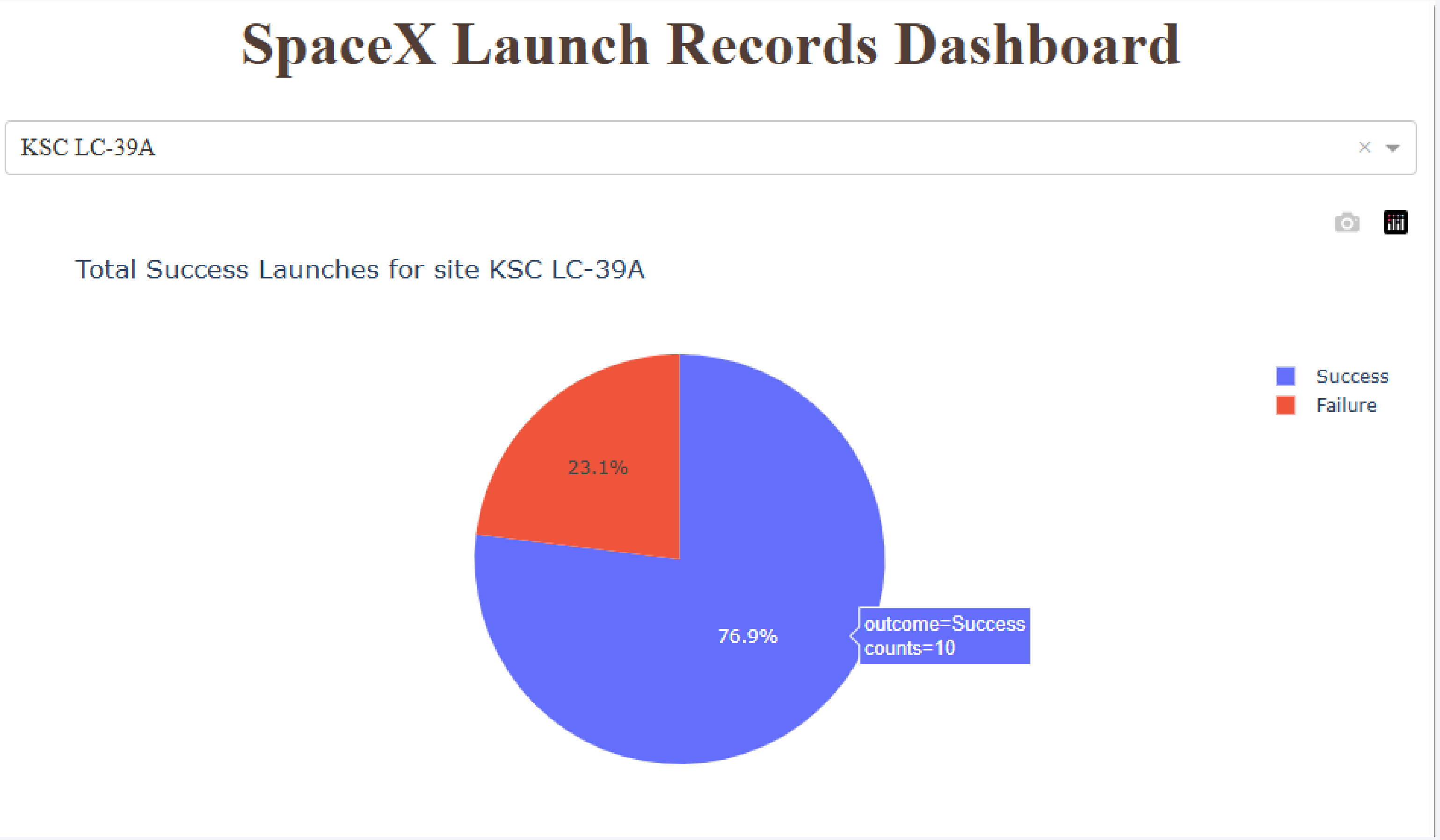
Piechart for the launch site KSC LC-39A with highest launch success ratio

When compare success VS failure rate per site, KSC LC-39A has 76.9% success rate while the rest has much less success rate:

CCAFS LC-40 has 26.9% success rate

CCAFS SLC-40 has 42.9% success rate

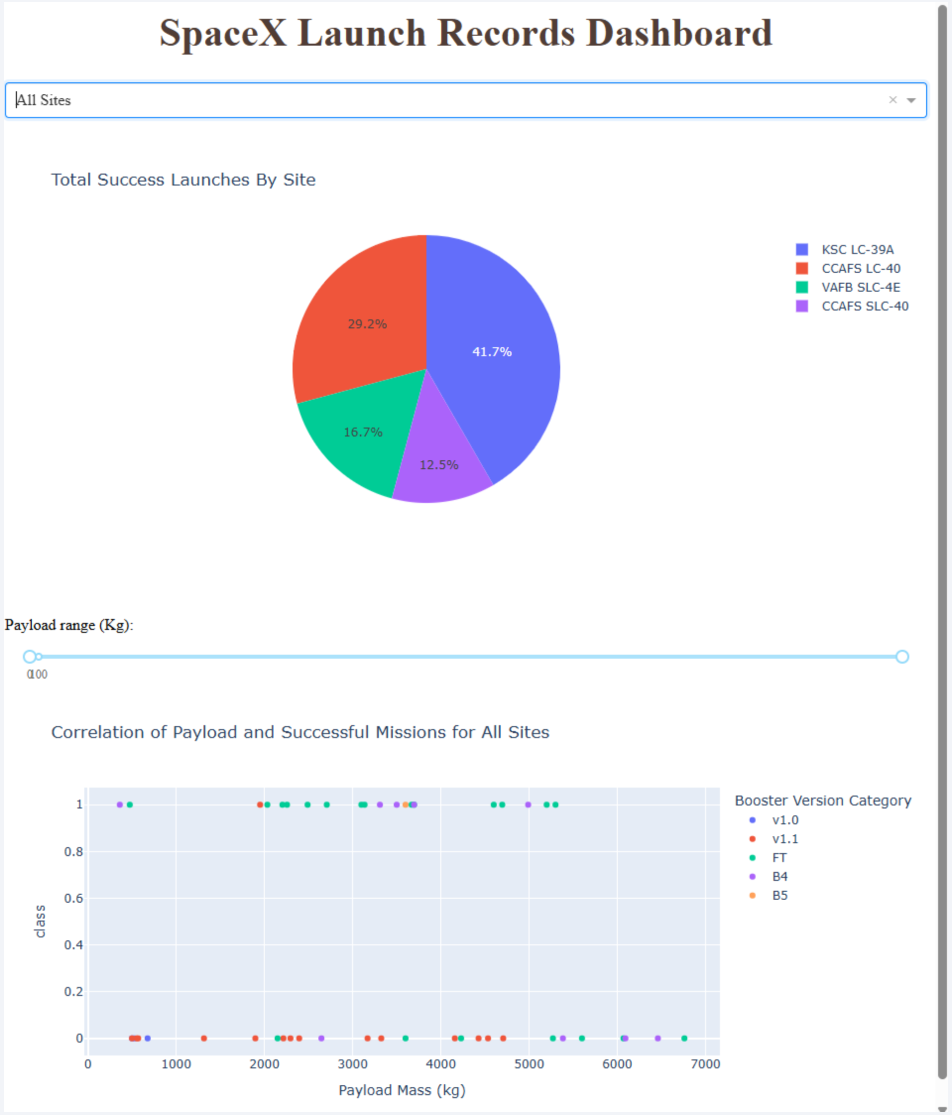
VAFB SLC-4E has 40% success rate



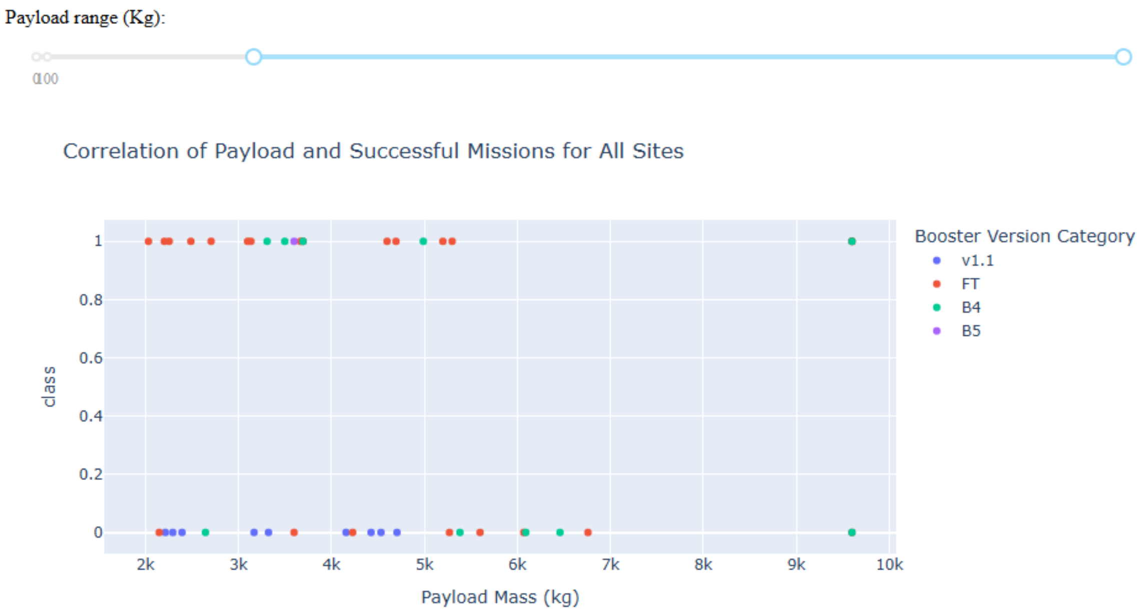
Payload VS Launch Outcome for Sites

Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider (see next slide for additional screenshots)

Notice that payload range(s) between 2000 - 5500 Kgs has the highest launch success rate. And payload ranges between *6000 – 9600 Kgs* has *lowest* launch success rate



Payload VS Launch Outcome for Sites (cont.)



Section 5

Predictive Analysis (Classification)

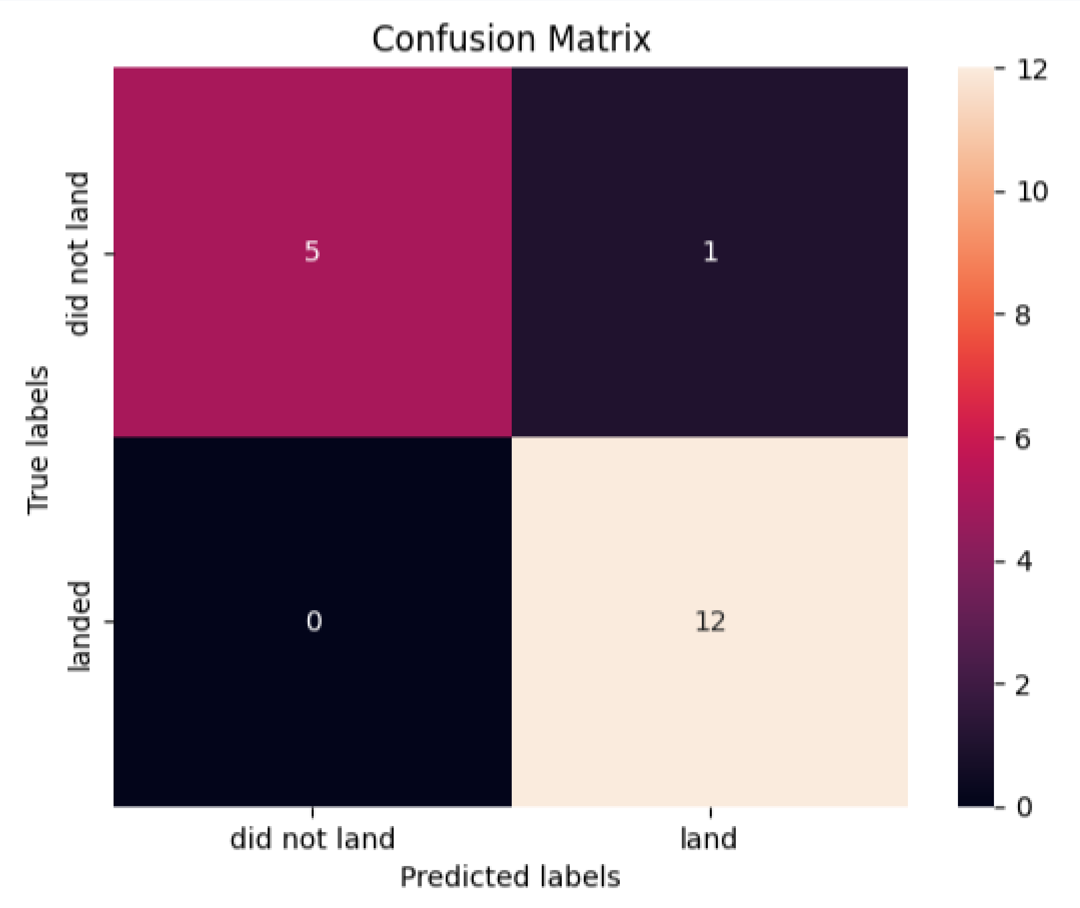
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

Confusion Matrix

- Judging by the confusion matrix, the best performing model are as follows:
- KNN with Test score 0.94
- LogisticRegression with Test score 0.94
- SVM with Test score 0.94
- Note that Confusion Matrix for all the above 3 models look very similar



Conclusions

Point 1

Point 2

Point 3

Point 4

...

Appendix

Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!