IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Laud Ochei
08 November 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 for 62 million dollars; other providers charge up to 165 million dollars each; much of the savings is due to Space X's ability to reuse the first stage. As a result, if we can predict whether the first stage will land, we can estimate the cost of a launch. This data can be used if another company wants to compete with Space X for a rocket launch. The project's goal is to build a machine learning pipeline that can predict whether the first stage will successfully land.

- Problems you want to find answers
  - What factors influence whether the rocket lands successfully?
  - The interplay of various features that determines the likelihood of a successful landing.
  - What operational conditions are required to ensure a successful landing program?

Section 1

Methodology

Methodology [1]

Executive Summary

- Data collection methodology:
  - Data from SpaceX was obtained from two sources:
    - **SpaceX API:**

      https://api.spacexdata.com/v4/rockets/

    - **Web Scraping:**

      https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)

- Perform data wrangling
  - After summarising and analysing features, collected data was enriched by creating a landing outcome label based on outcome data.

Methodology [2]

• Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data collected up to this point was normalised, divided into training and test data sets, and evaluated by four different classification models, with the accuracy of each model evaluated using different parameter combinations.

Data Collection

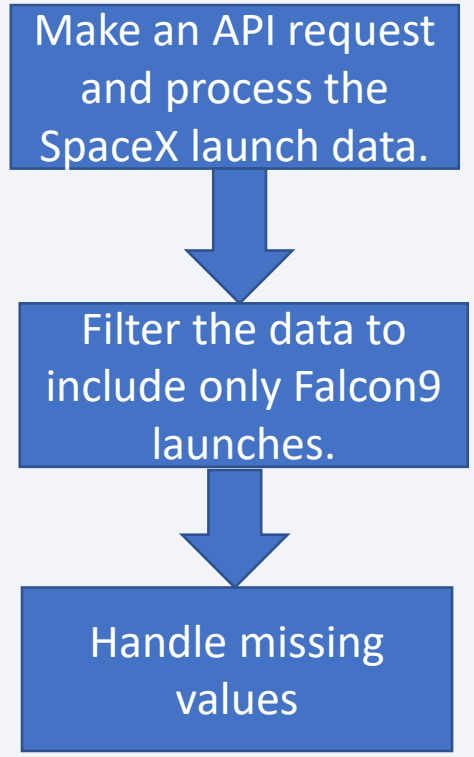
- Various methods were used to collect the data
  - Data was collected by sending a get request to the SpaceX API:

  https://api.spacexdata.com/v4/rockets/

  - Next, we decoded the response content as Json using the.json() function call and converted it to a pandas dataframe using the.json_normalize() function call.
  - We then cleaned the data, checked for missing values, and filled in the gaps as needed.
  - In addition, we used BeautifulSoup to scrape Wikipedia for Falcon 9 launch records.
  - https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches
  - The goal was to extract the launch records as an HTML table, parse it, and convert it to a pandas dataframe for future analysis.

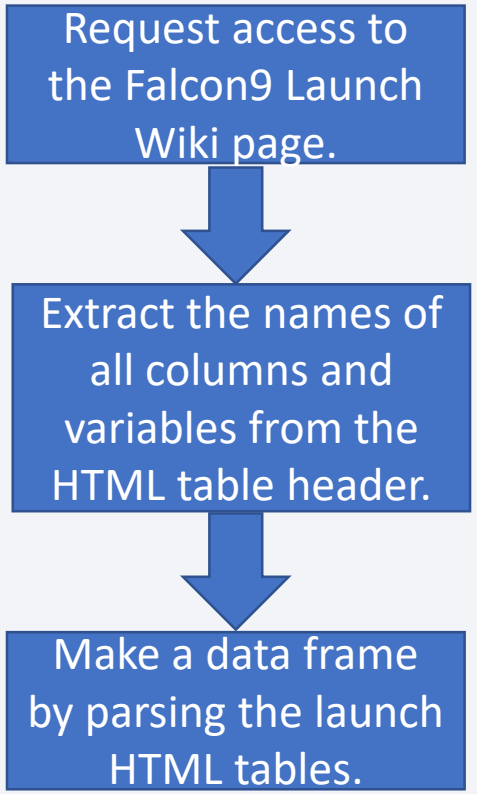
Data Collection – SpaceX API

- SpaceX provides a public API through which data may be downloaded and used.
- This API was used in accordance with the flowchart below, and data was then persisted.
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/data-collection-api.ipynb



Data Collection - Scraping

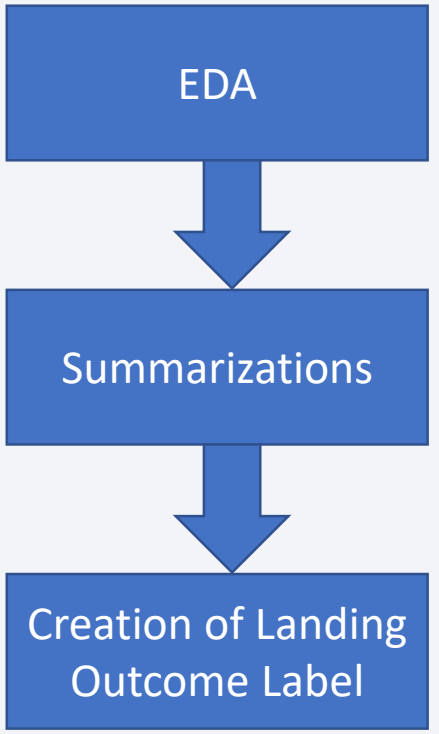
- Data from SpaceX launches is also available on Wikipedia.
- According to the flowchart, data is downloaded from Wikipedia and then persisted.
- The source code is available at:
- https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/webscraping.ipynb



Data Wrangling

- The dataset was first subjected to some exploratory data analysis (EDA).
- The number of launches per site, occurrences of each orbit, and mission outcomes per orbit type were then calculated.
- Finally, the Outcome column was used to generate the landing outcome label.

- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c01-What-is-Data%20Science/spacex-data-collection-api.ipynb

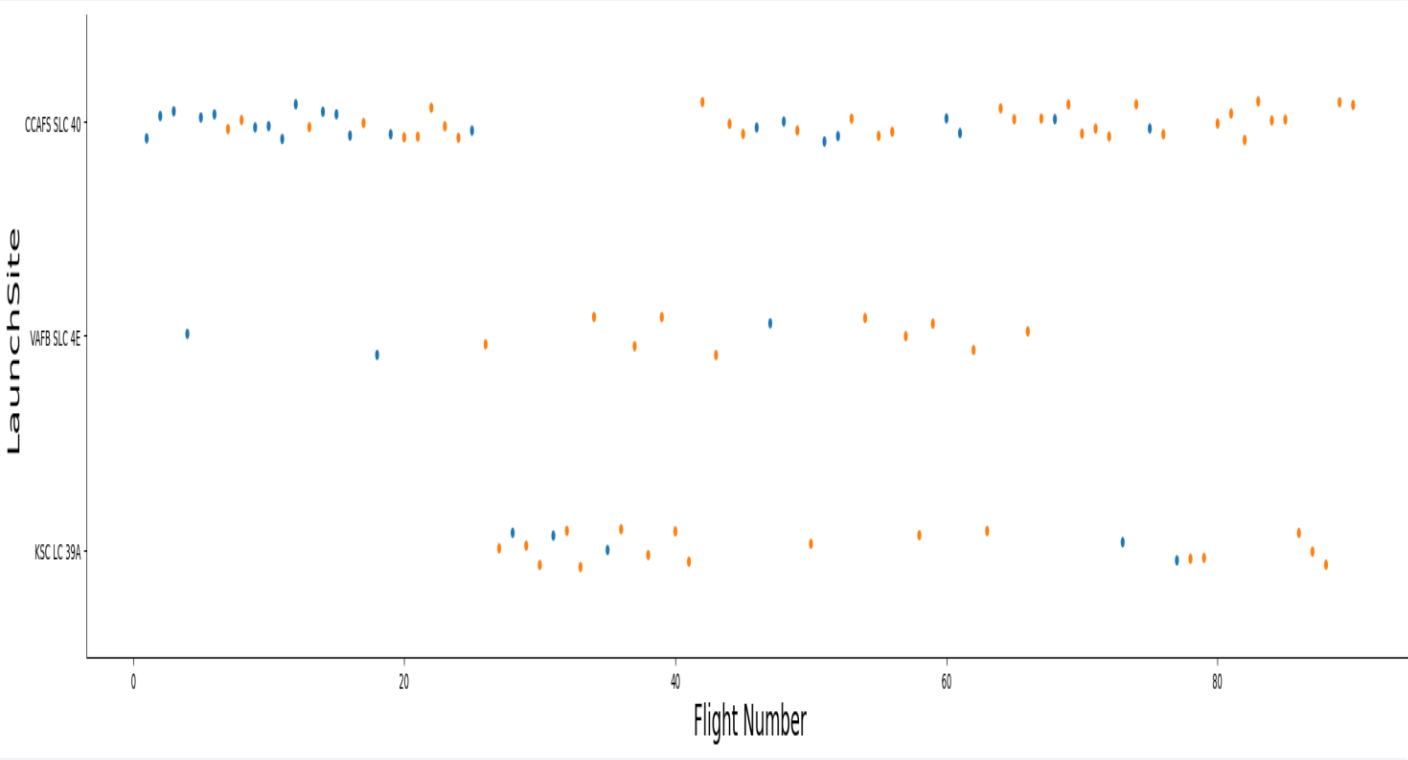


EDA with Data Visualization

- SQL queries were run to determine:
  - names of the space mission's unique launch sites
  - top 5 launch sites whose names start with the letter 'CCA‘
  - total payload mass carried by NASA-launched boosters (CRS)
  - average payload mass carried by the F9 v1.1 booster
  - date of the first successful landing outcome in a ground pad
  - names of successful drone ship boosters with payload masses ranging from 4000 to 6000 kg
  - total number of mission successes and failures
- SQL queries were run to determine:
  - names of the booster versions that carried the most payload mass
  - determine the number of failed landings in droneships, their booster versions, and launch site names in 2015.
  - Rank of landing outcomes (such as Failure (droneship) or Success (ground pad)) between 2010-06-04 and 2017-03-20.
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Scatterplots and bar plots were used to visualise the relationship between two features when exploring data:
  - Payload Mass X Flight Number, Launch Site X Flight Number, Payload Mass X Flight Number, Payload and Flight Number
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/eda-sql-coursera_sqllite.ipynb



Build an Interactive Map with Folium

- Folium Maps were used with markers
- Circles, lines, and marker clusters
- Markers represent points, such as launch sites
- Circles represent highlighted areas around specific coordinates, such as NASA Johnson Space Centre
- Marker clusters represent groups of events in each coordinate, such as launches at a launch site
- Lines represent distances between two coordinates.
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Plotly dash was used to create an interactive dashboard.
- Pie charts were created to display each site's total launches.
- Scatter graphs for each booster version were created to display the relationship between the outcome and payload mass (kg).
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/spacex_dash_app.py

Predictive Analysis (Classification)

- The data was loaded with numpy and pandas, transformed it, and divided it into training and testing sets.
- Using GridSearchCV, various machine learning models were created and various hyperparameters were tuned.
- The accuracy of the model's metrics was used and it was improved through feature engineering and algorithm tuning.
- The most effective classification model was discovered
- The source code is available at: https://github.com/laudcharlesochei/data-science-notes/blob/master/c10-Applied-Data-Science-Capstone/space-x-machine-learning-prediction.ipynb

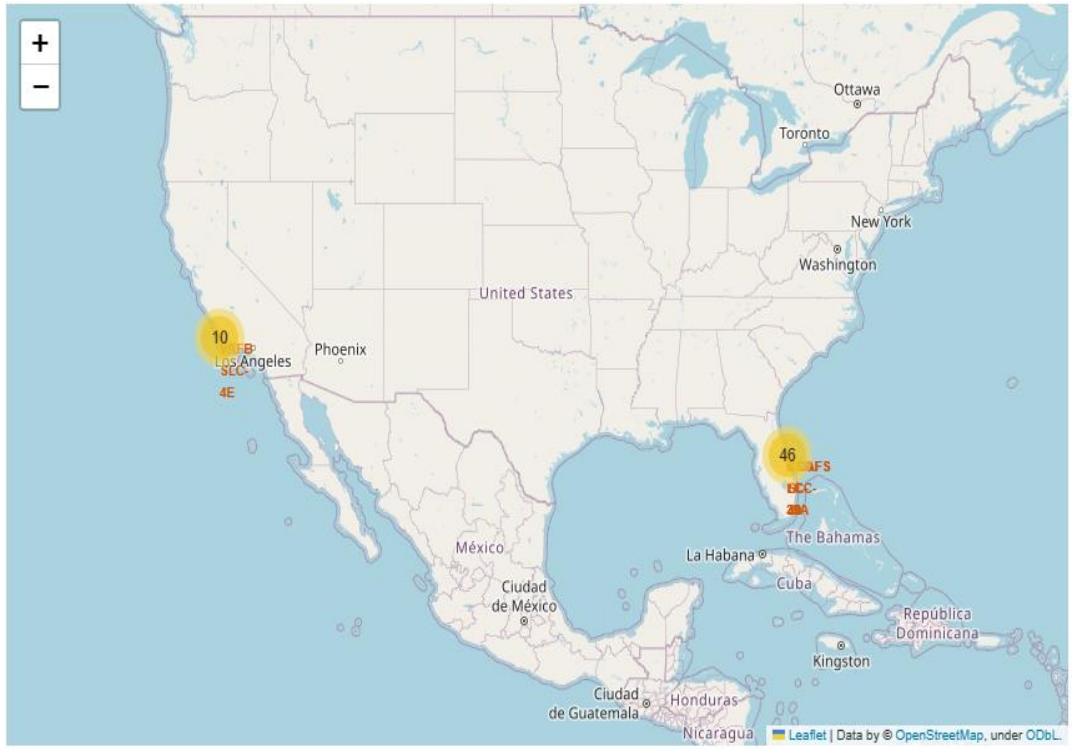
Results[1]

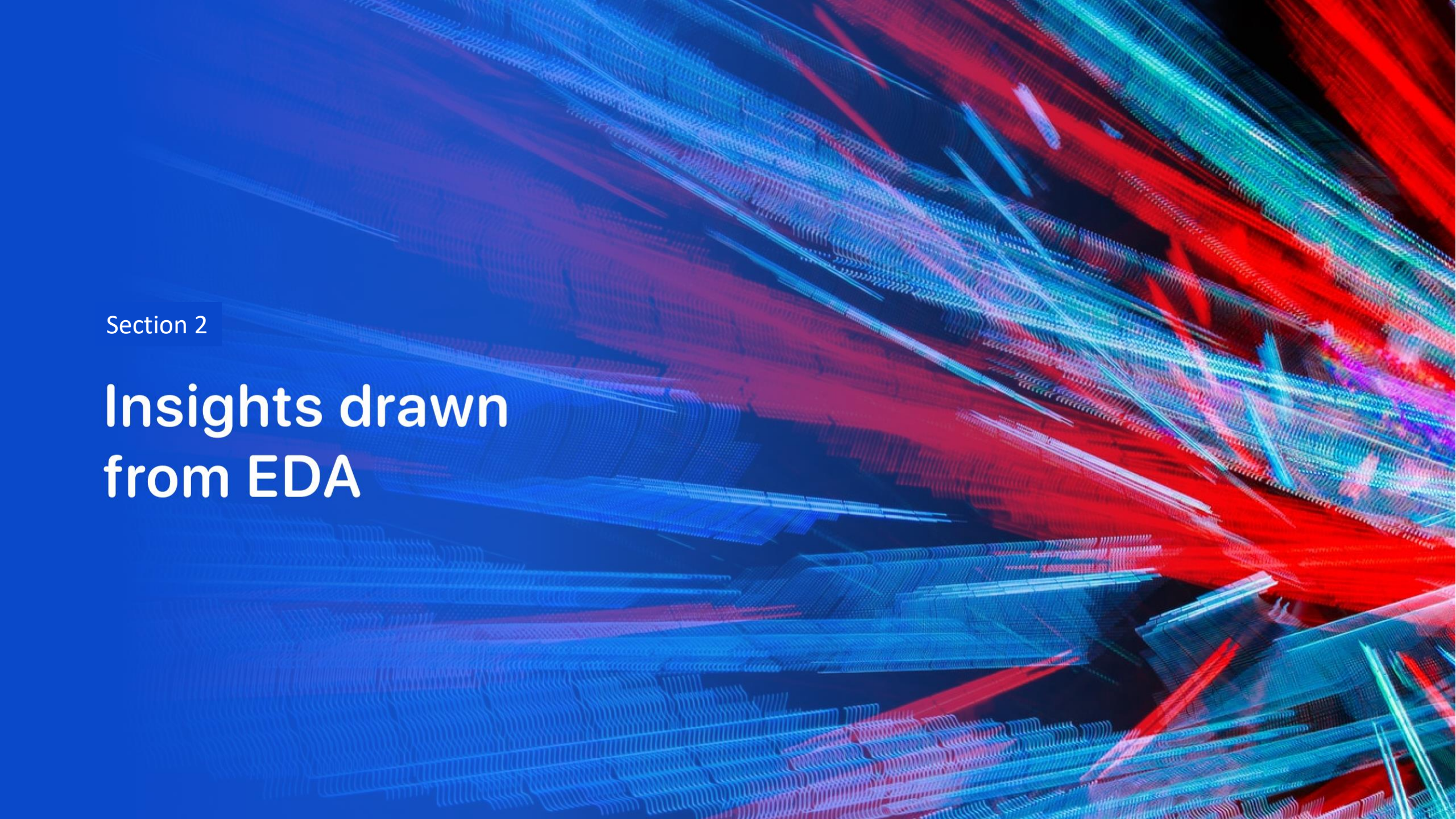
- Exploratory data analysis results:
  - Space X launches from four different locations
  - First launches were done by Space X and NASA.
  - Average payload of the F9 v1.1 booster is 2,928 kg; The first successful landing occurred in 2015, five years after the first launch.
  - Many Falcon 9 booster versions were successful at landing in drone ships with payloads greater than the average
  - Nearly all mission outcomes were successful.
  - In 2015, two booster versions failed to land on drone ships: F9 v1.1 B1012 and F9 v1.1 B1015.
  - As time passed, the number of successful landings increased.

Results[2]

- Using interactive analytics, it was possible to determine that launch sites used to be in safe locations, such as near the sea, and had a good logistic infrastructure surrounding them.

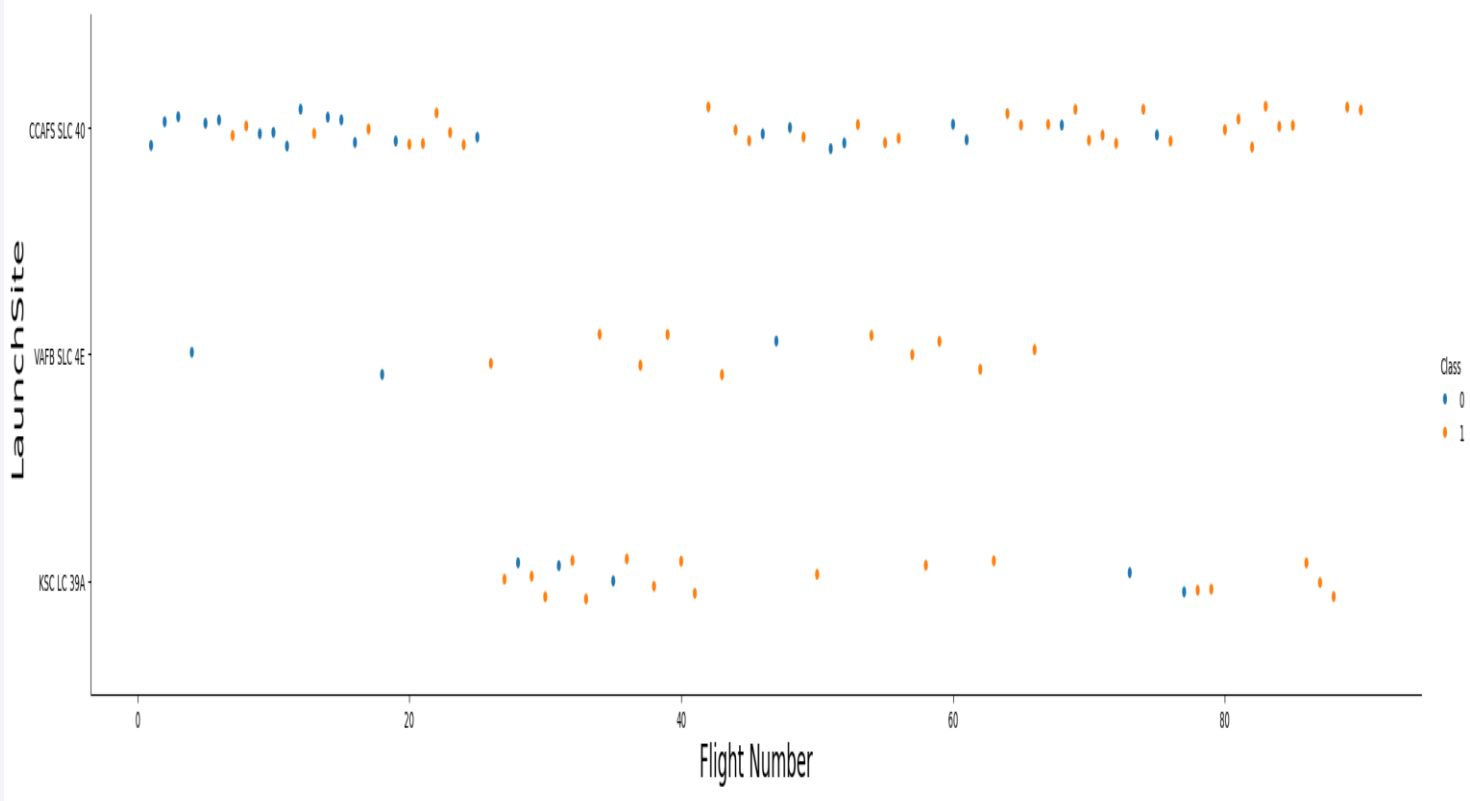
- The majority of launches take place at east-coast launch sites.




Section 2
Insights drawn from EDA

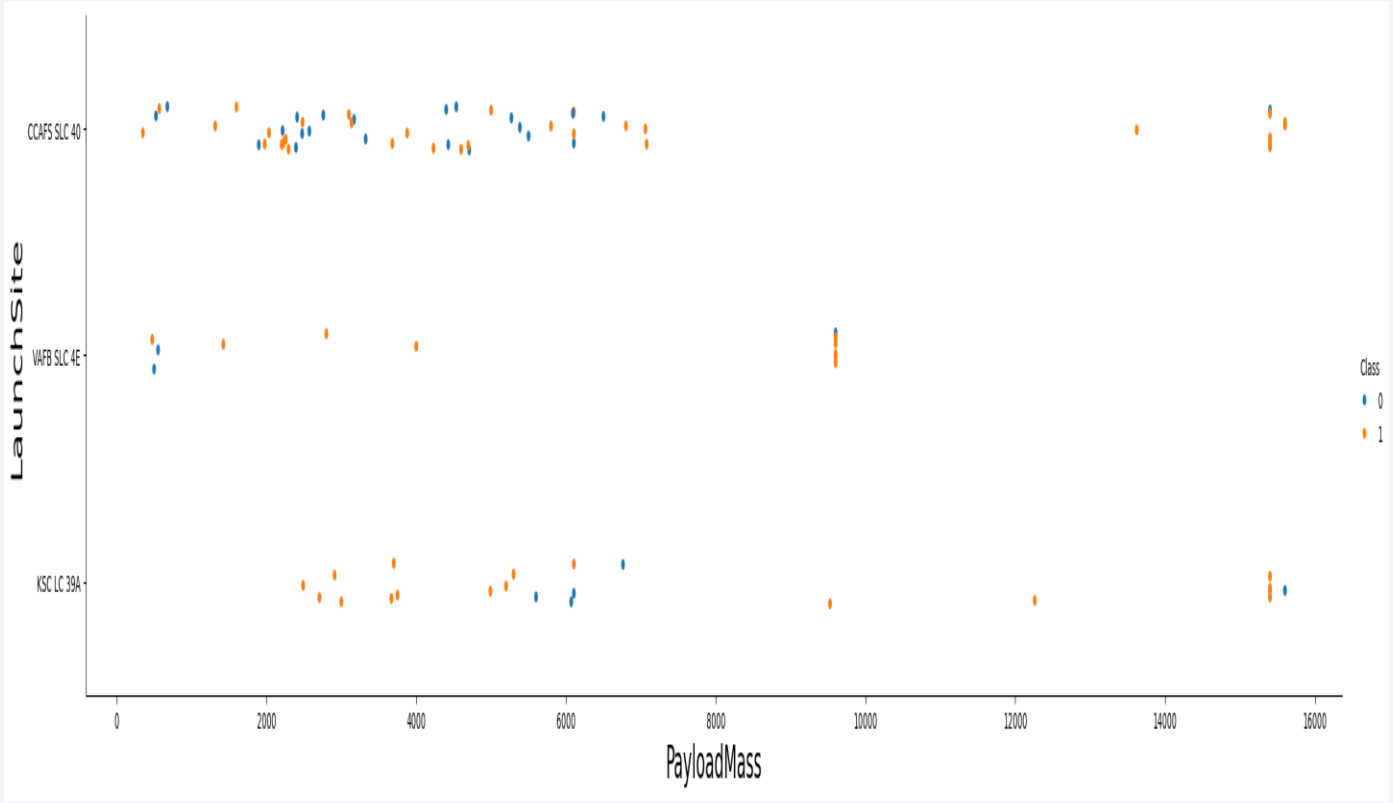
Flight Number vs. Launch Site

- It was discovered from the plot that the greater the number of flights at a launch site, the higher the success rate at that launch site.



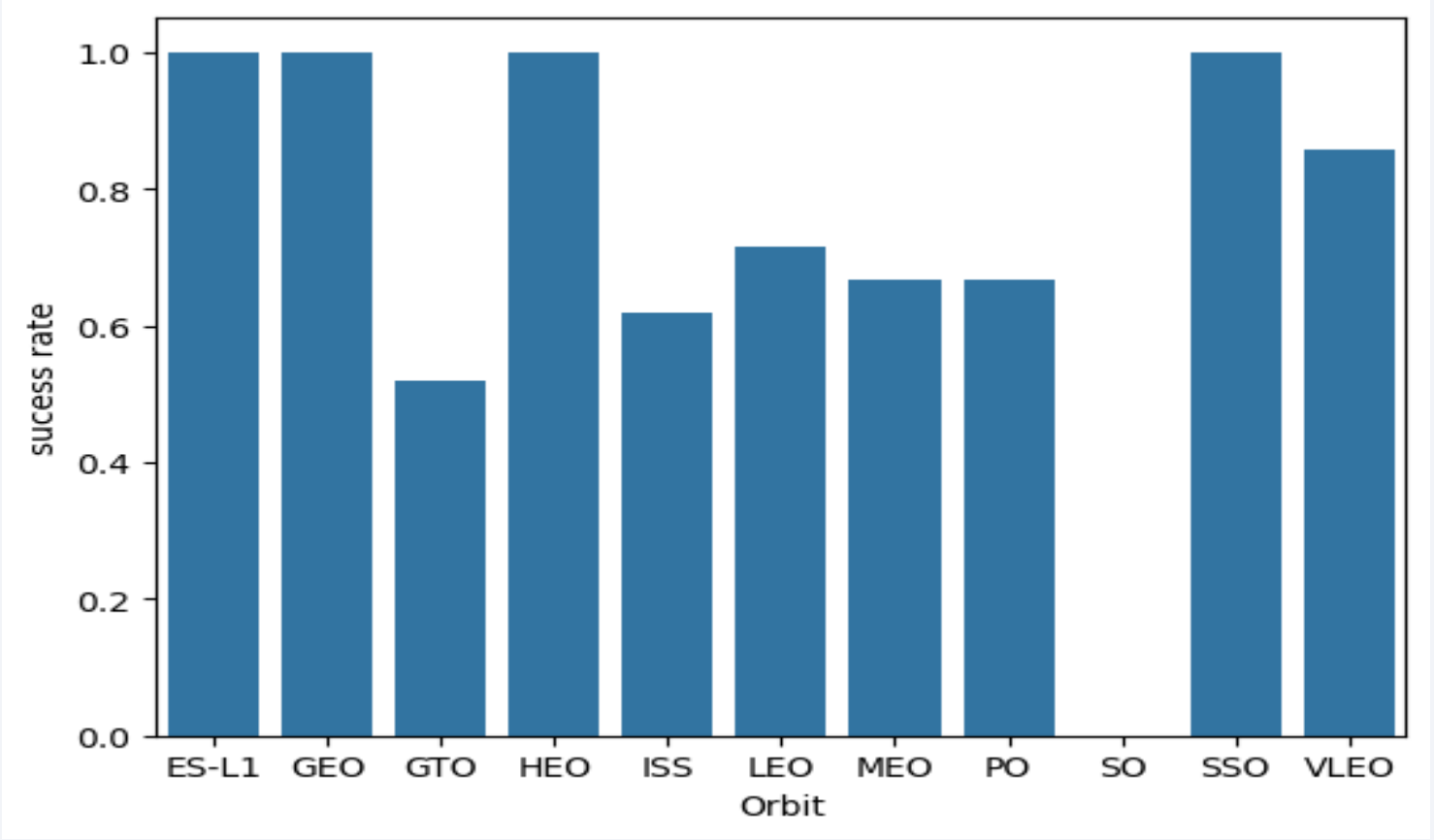
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40, the greater the rocket's success rate.



Success Rate vs. Orbit Type

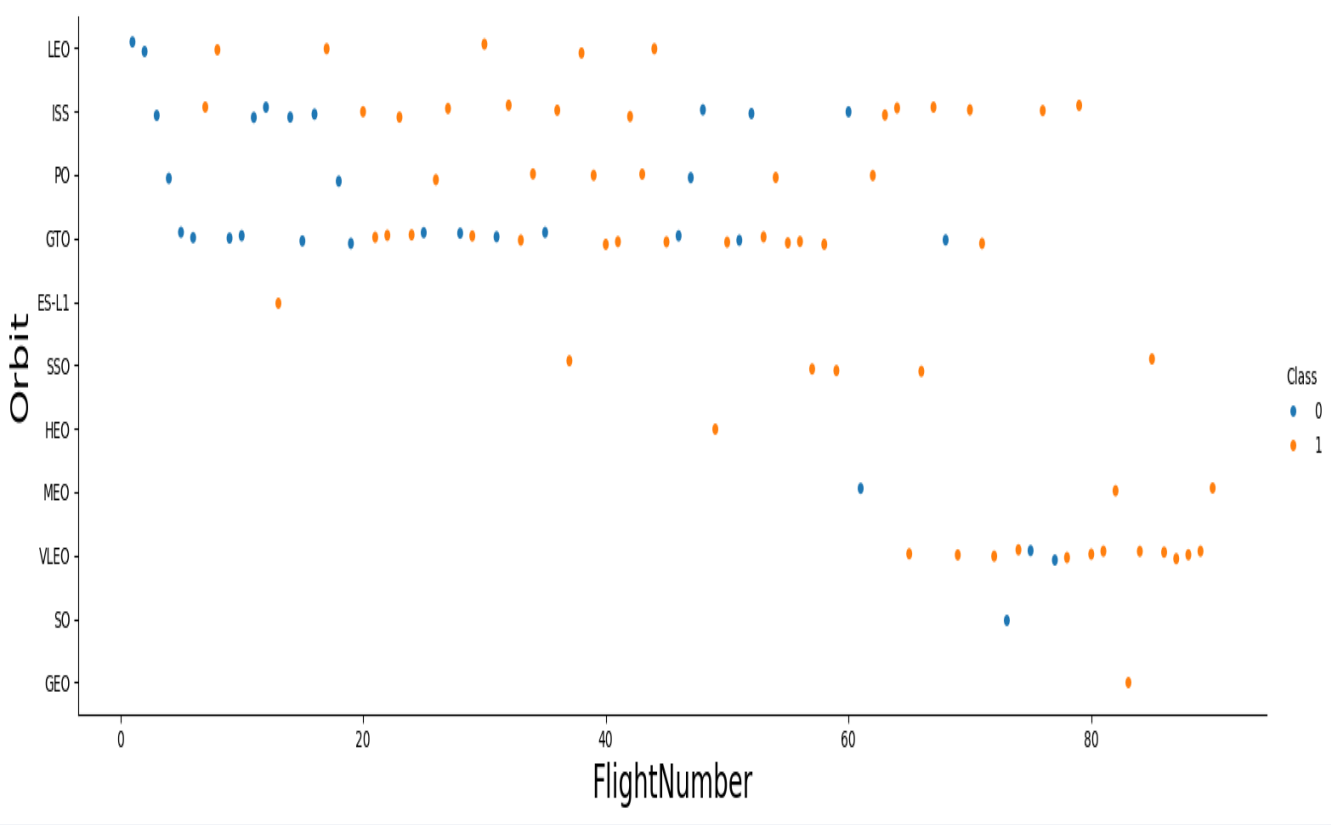
- The plot shows that ES-L1, GEO, HEO, SSO and VLEO had the most success rate.



Flight Number vs. Orbit Type

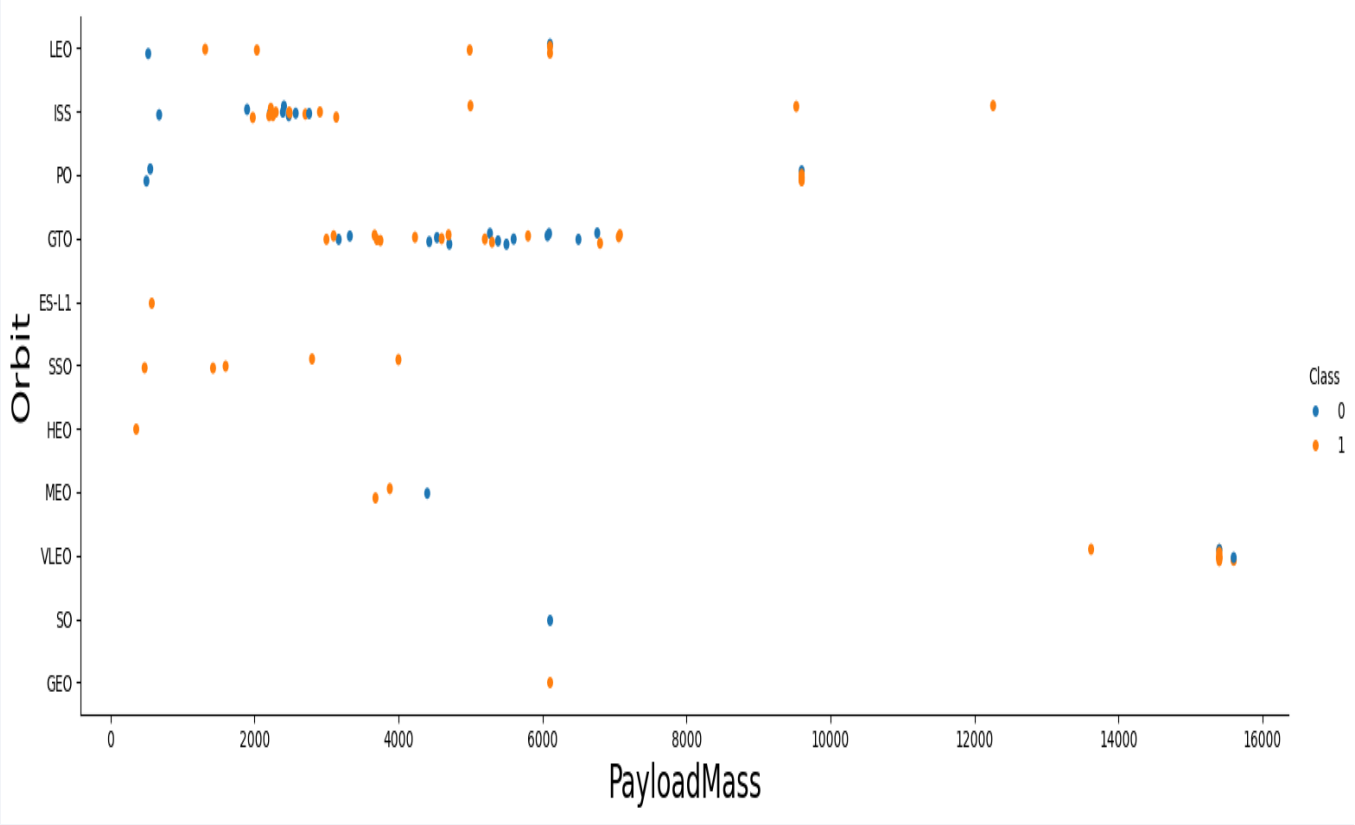
- The graph shows the relationship between Flight Number and Orbit Type.

- In the LEO orbit, success is related to the number of flights, whereas there is no relationship between flight number and orbit in the GTO orbit.



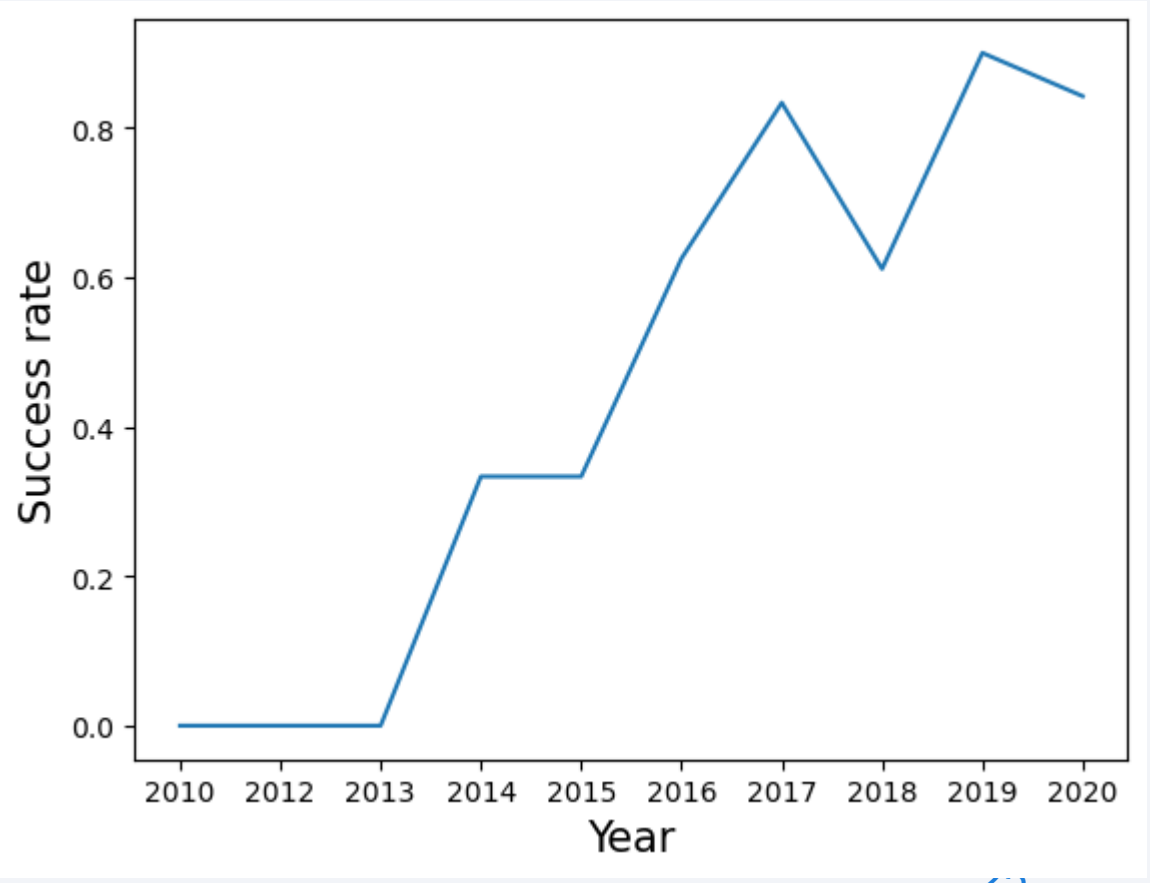
Payload vs. Orbit Type

- We can see that more successful landings with large payloads occur in PO, LEO, and ISS orbits.



Launch Success Yearly Trend

- The plot shows that the success rate has been increasing since 2013 and will continue to do so until 2020.



All Launch Site Names

- The DISTINCT keyword WAS USED to display only unique SpaceX launch sites.

Task 1

Display the names of the unique launch sites in the space mission

In [35]:
```
%sql select distinct Launch_Site from SPACEXTBL
```

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

Out[35]:

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

Launch Site Names Begin with 'CCA'

- The query shown in the figure was used to display 5 records where launch sites begin with `CCA

```
In [36]: %sql select * from SPACEXTBL where Launch_Site like 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

Out[36]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	L
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	
12/8/2010	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	
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- The query in the figure was used to calculate the total payload carried by NASA boosters to be 45596.

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

```
In [37]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

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

Out[37]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was calculated to be 2928.4 kg.

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

```
In [39]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like 'NASA%'
 * sqlite:///my_data1.db
Done.
Out[39]: sum(PAYLOAD_MASS__KG_)
                          99980
```

```
In [41]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1%'
 * sqlite:///my_data1.db
Done.
Out[41]: avg(PAYLOAD_MASS__KG_)
                  2534.6666666666665
```

First Successful Ground Landing Date

- It was discovered that the first successful landing outcome on the ground pad occurred on December 22, 2015.

```
[50]: %sql SELECT min(Date), "Landing_Outcome" FROM SPACEXTBL WHERE "Landing_Outcome" = "Success (ground pad)";

 * sqlite:///my_data1.db
Done.

[50]: min(Date)  Landing_Outcome
      2015-12-22 Success (ground pad)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- The WHERE clause was used to filter for boosters that successfully landed on the drone ship, and the AND condition was used to determine successful landings with payload masses greater than 4000 but less than 6000.

```
[38]: %%sql

SELECT Booster_Version, PAYLOAD_MASS__KG_, "Landing_Outcome" FROM SPACEXTBL
WHERE "Landing_Outcome" = "Success (drone ship)"
and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

[38]:

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

- A wildcard character like '%' was used to filter the total number of successful and failed mission outcomes from the MissionOutcome column.

- The result shows that the total number of successful mission outcomes is 100 and the total number of failed mission outcomes is 1

In [48]:
```
%%sql

select count(*) from SPACEXTBL
where "Mission_Outcome" like "Success%"
```

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

Out[48]:

count(*)
100

In [47]:
```
%%sql

select count(*) from SPACEXTBL
where "Mission_Outcome" like "Failure%"
```

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

Out[47]:

count(*)
1

Boosters Carried Maximum Payload

- Using a subquery in the WHERE clause and the MAX() function, we determined which booster carried the most payload.

```
[26]: %%sql

select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

[26]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- For the year 2015, a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions was used to filter for failed landing outcomes in drone ships, booster versions, and launch site names.

```
[36]: %%sql

SELECT Booster_Version, Launch_Site, "Landing_Outcome" FROM SPACEXTBL
WHERE "Landing_Outcome" = "Failure (drone ship)"
and Date Between '2015-01-01' AND '2015-12-31'

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

[36]:

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)

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

- Landing outcomes and the COUNT of landing outcomes were selected from the data, and the WHERE clause was used to filter for landing outcomes BETWEEN 2010-06-04 and 2010-03-20.

- The GROUP BY clause was used to group the landing outcomes, and the ORDER BY clause was used to order the grouped landing outcomes in descending order.

```
[70]: %%sql

SELECT "Landing_Outcome", COUNT("Landing_Outcome") from SPACEXTBL
where Date Between '2010-06-04' AND '2017-03-20'
group by "Landing_Outcome"
order by COUNT("Landing_Outcome") desc
```

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

[70]:

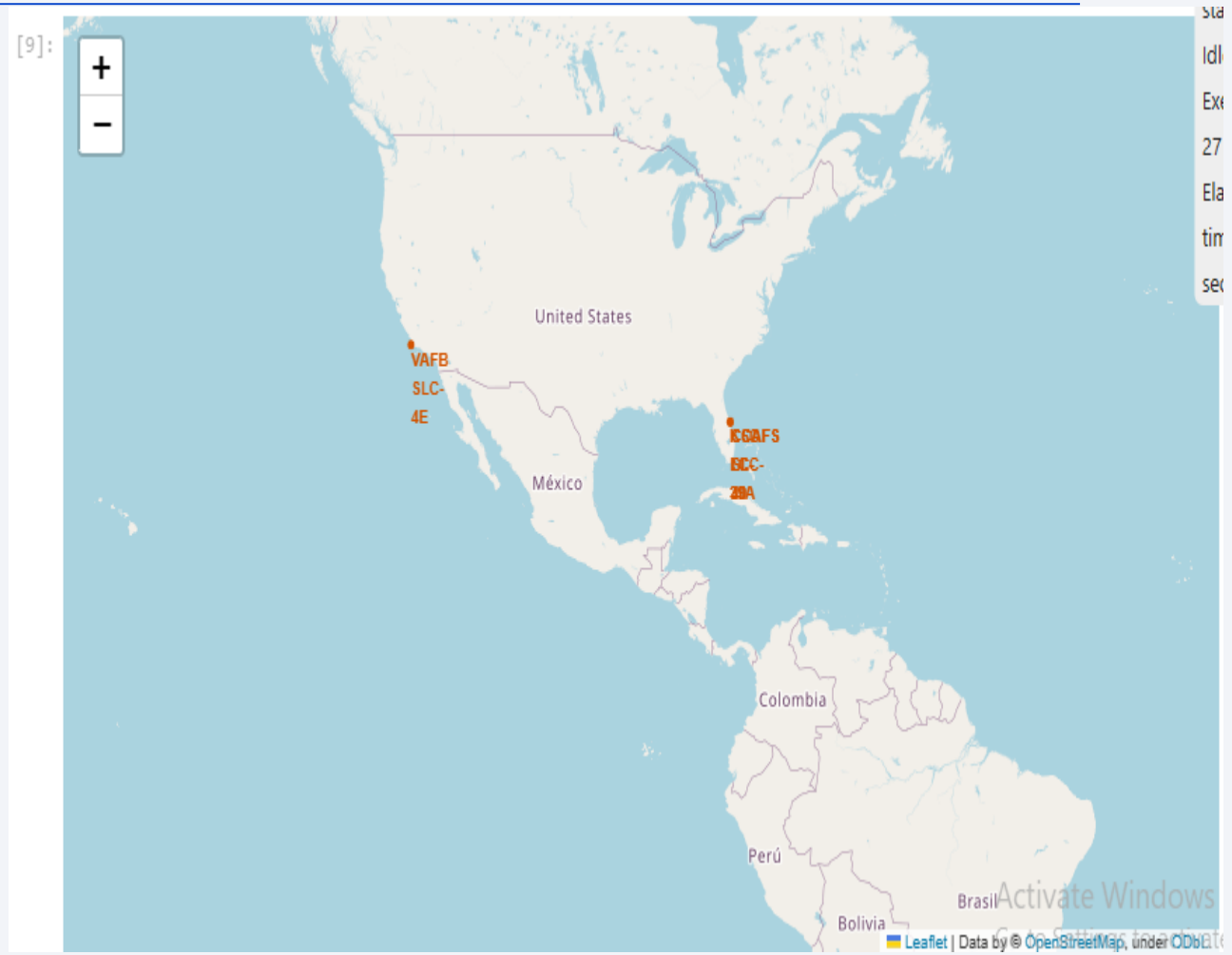
Landing_Outcome	COUNT("Landing_Outcome")
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

- The SpaceX launch sites are mostly in the coast – Florida and California



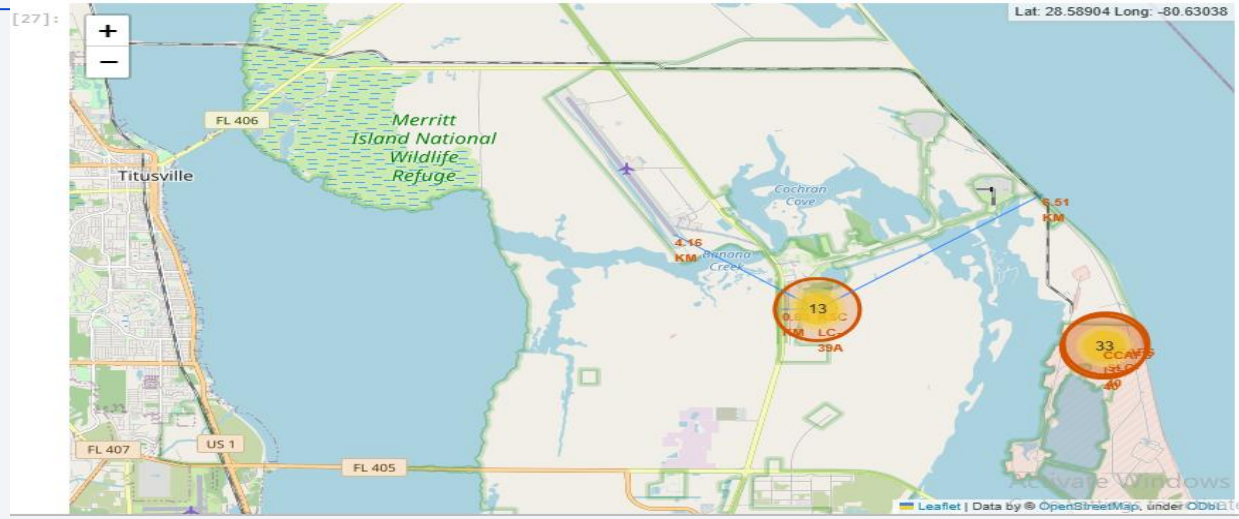
Markers showing launch sites with colour labels

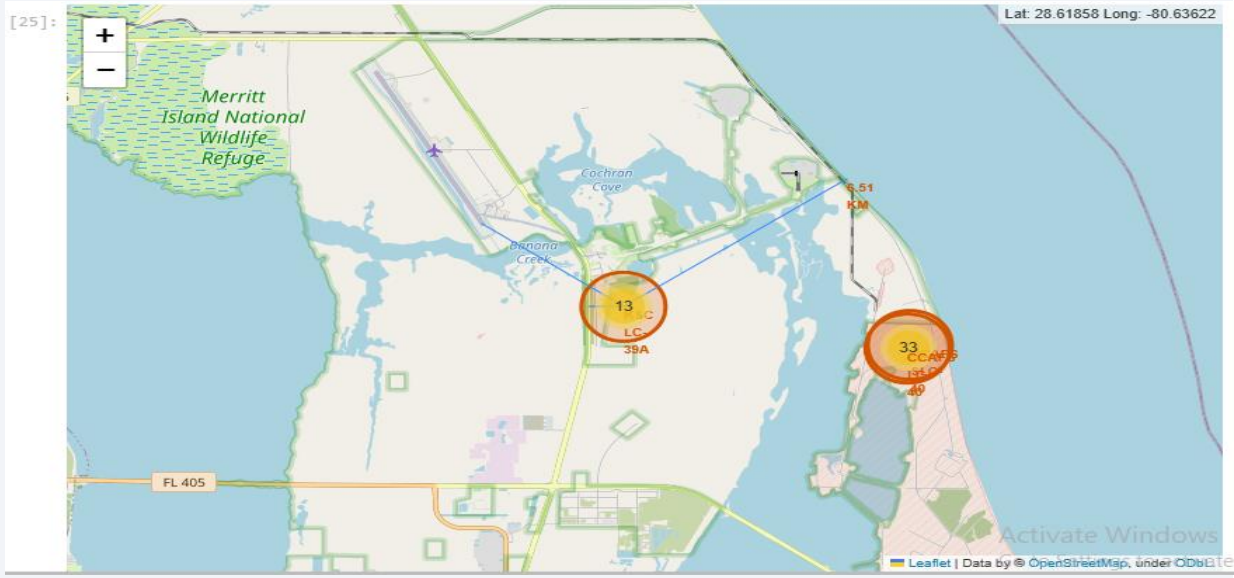
- The figure shows the Florida and California launch sites.
- The Greens markers show successful launch ad Red show unsuccessful launch.

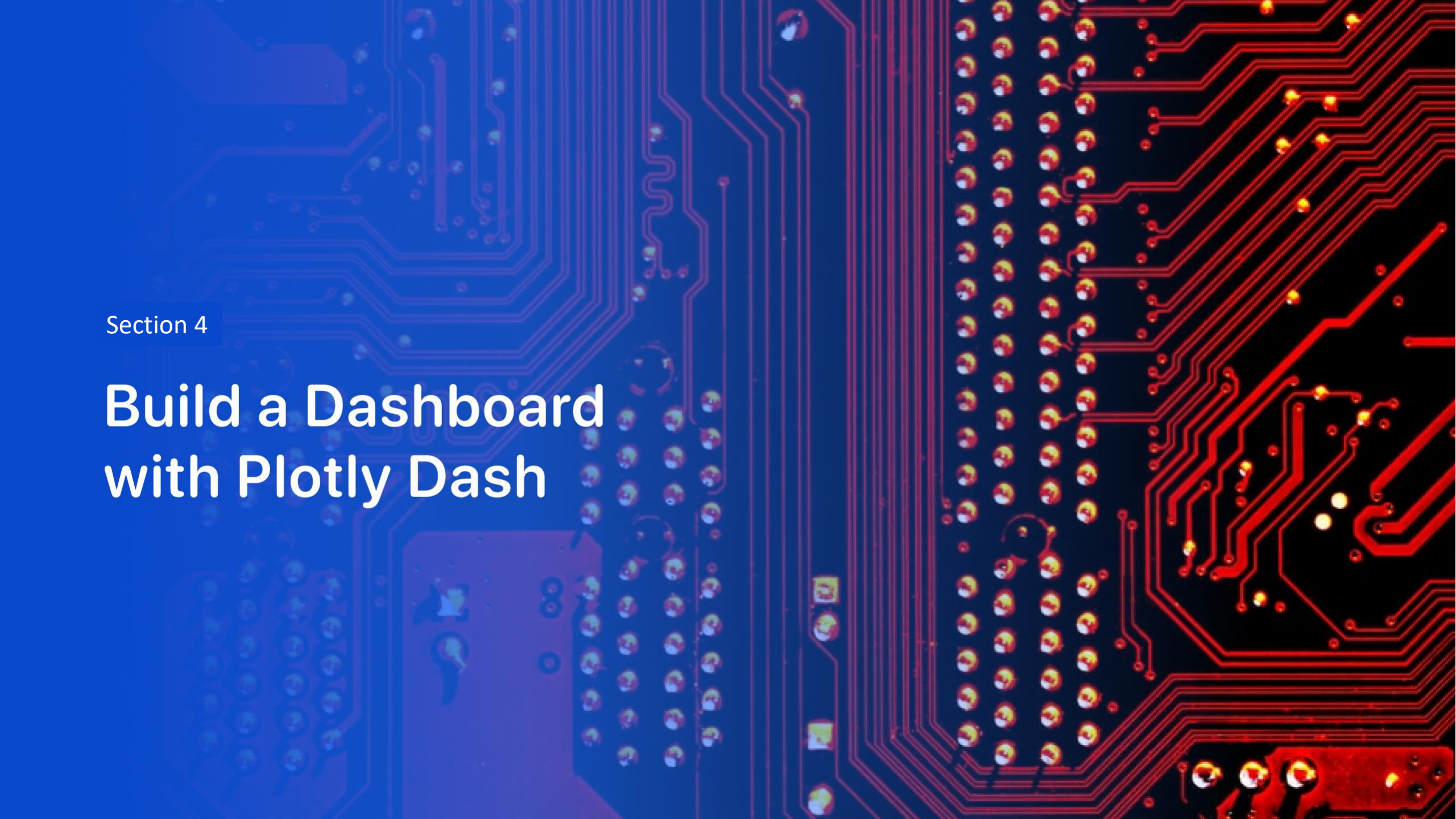
Launch Site distance to landmarks

- The figure shows the closet distance to several landmarks such as:
  - Highways
  - Railways
  - Airports
  - Coastline
  - City

For example, you can see that the launch site (labelled 33 to the right) is close to the railway.

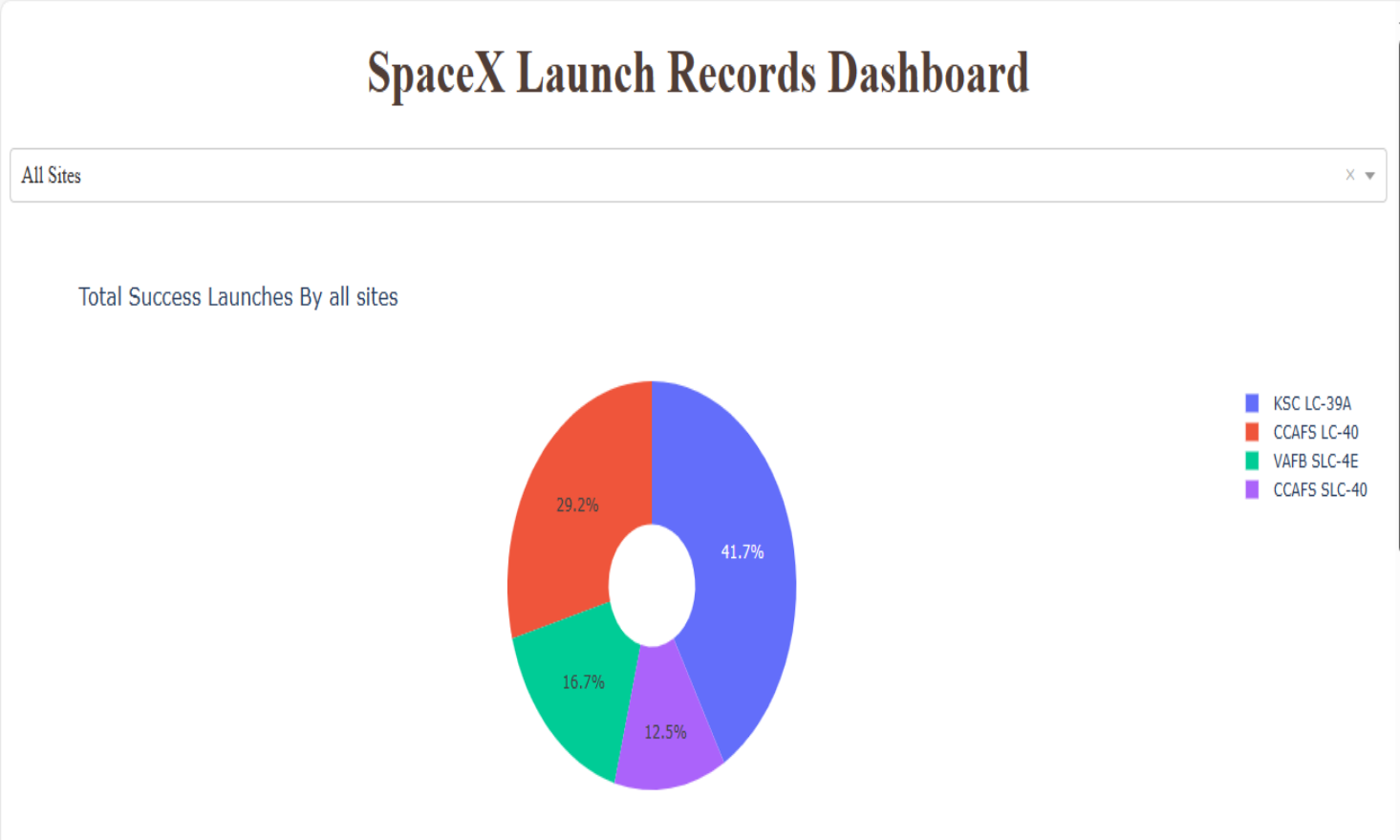





Section 4
Build a Dashboard with Plotly Dash

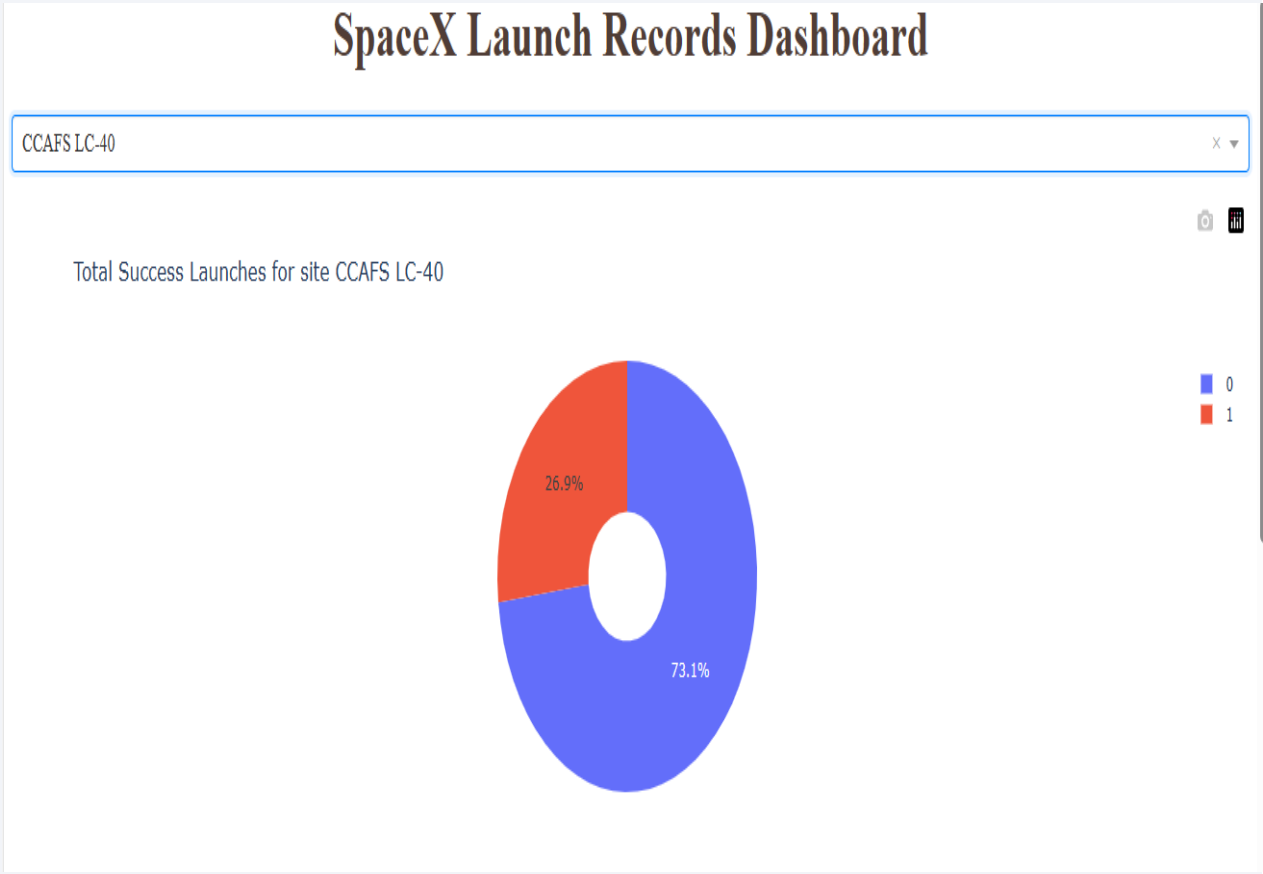
Success percentage achieved by each launch site

- The pie chart shows the success percentage achieved by each launch site.
- It can be seen that KSC LC-39A had the most successful launchers from all the sites



Launch site with the highest launch success ratio

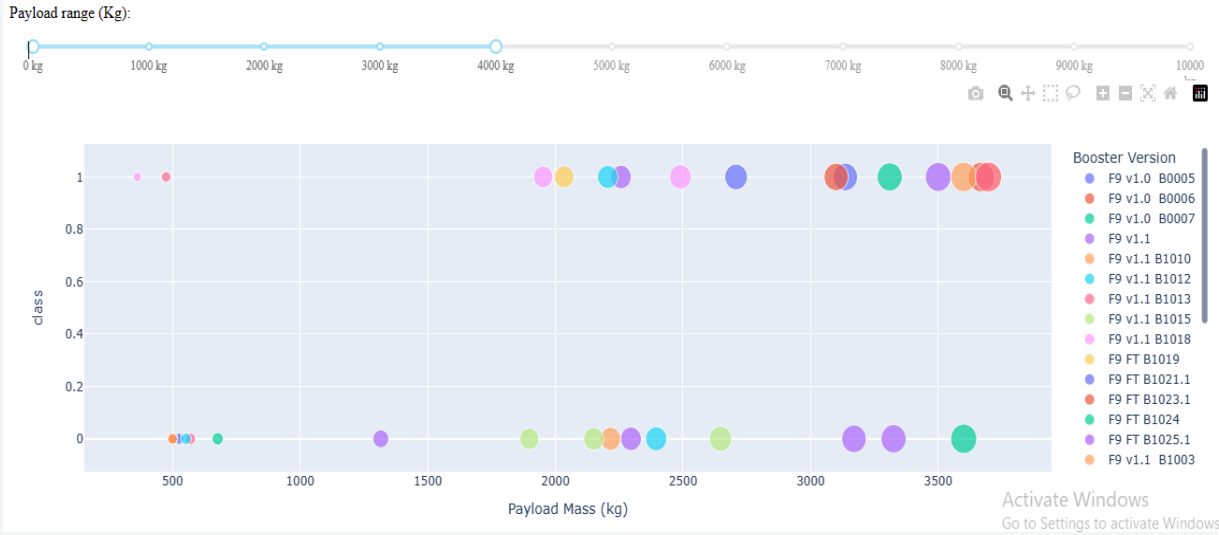
- The Pie chart shows the launch sites with the highest launch success ratio .
- KSC LC-39A activated a 76.9% success rate while getting a 23.1% failure rate.

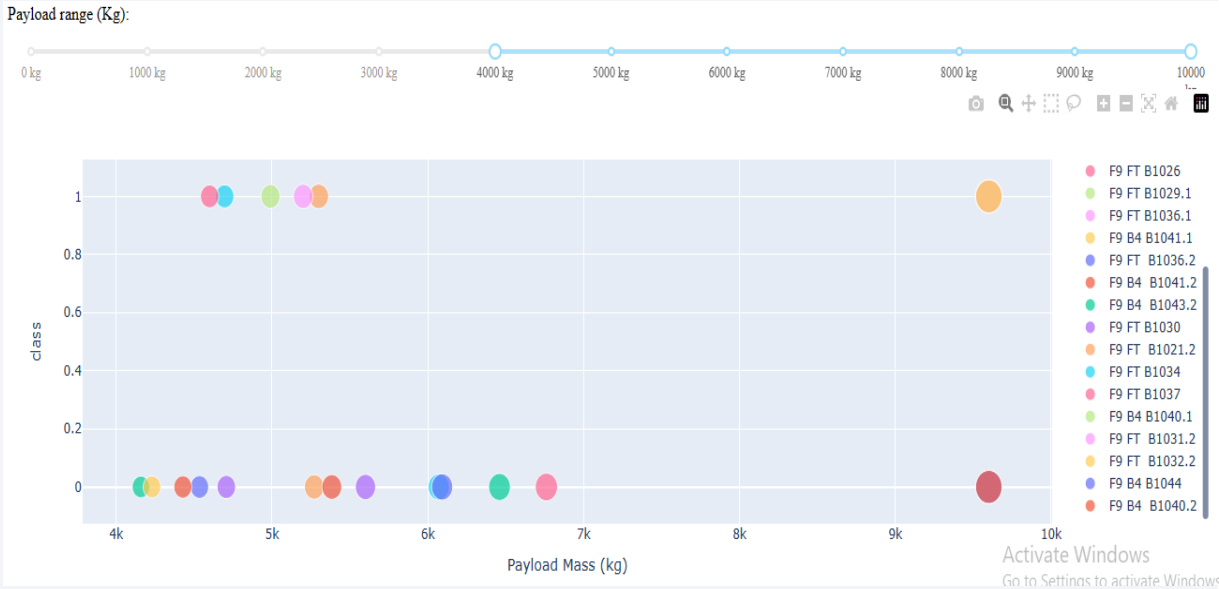


Payload vs Launch Outcome for all sites

- The scatter plot of payload versus Launch Outcome for all the sites with different payloads selected in the range slider.

- It can be seen that 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 a better classification accuracy.

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

In [84]:
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

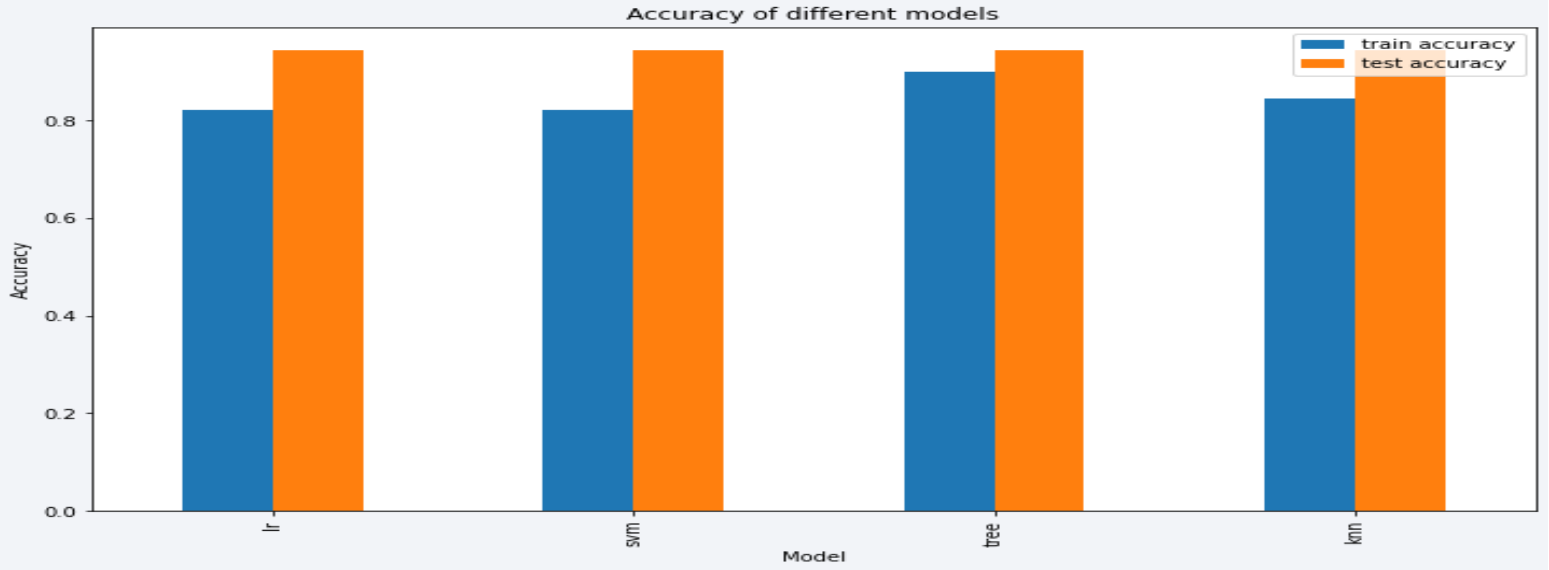
In [85]:
```
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X,Y)
```

Out[85]:
```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

In [86]:
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'min_sample
s_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}
accuracy : 0.9
```


- The bar chart shows that the accuracy of the decision tree for the training and test data is better than the other methods.



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

- The classifier predicted successful and unsuccessful landing, that true positives and true negatives.

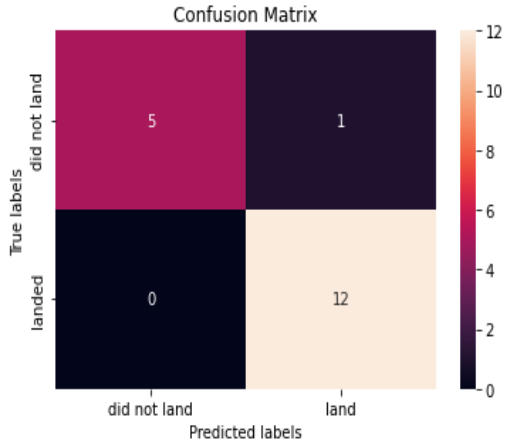
Calculate the accuracy of tree_cv on the test data using the method `score` :

In [87]:
```
tree_score = tree_cv.score(X_test, Y_test)
tree_score
```

Out[87]: 0.9444444444444444

We can plot the confusion matrix

In [88]:
```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- We can draw the following conclusion:
  - A launch site's success rate increases with the number of flights conducted there.
  - The launch success rate increased from 2013 to 2020.
  - Orbits with the highest success rate were ES-L1, GEO, HEO, SSO, and VLEO.
  - Out of all the sites, KSC LC-39A had the most successful launches.
  - For this task, the optimal machine learning algorithm is the decision tree classifier.

Appendix

- The source code for the SpaceX project is available at:
- https://github.com/laudcharlesochei/data-science-notes/tree/master/c10-Applied-Data-Science-Capstone

Thank you!