

Winning Space Race with Data Science

Muztoba Rabbani
01 August 2023



Outline

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

Executive Summary

- Summary of methodologies
 - Space X Data Collection using SpaceX API
 - SpaceX Data Collection with Web Scraping
 - SpaceX Data Wrangling
 - SpaceX Exploratory Data Analysis using SQL
 - SpaceX EDA DataViz Using Python modules: Pandas, numpy, Matplotlib
 - SpaceX Launch Sites Analysis with folium interactive visual analytics and plotly dash
 - SpaceX Machine Learning Landing Prediction
- Summary of all results
 - EDA restulsts
 - Interactive visual analytics and dashboards
 - Predictive analysis

Introduction

- **Project background and context**

SpaceX promotes Falcon 9 rocket launches on its website at a competitive price of 62 million dollars, significantly lower than other providers, whose costs can soar beyond 165 million dollars per launch. A major factor contributing to these substantial savings is SpaceX's groundbreaking ability to reuse the first stage of their rockets. By predicting whether the first stage will successfully land, one can ascertain the overall cost of a launch. This valuable insight becomes crucial if another company wishes to compete with SpaceX in bidding for rocket launch opportunities.

- **Problems you want to find answers**

The capstone's objective is to predict the successful landing of the Falcon 9 first stage using SpaceX's advertised launch data, providing valuable insights for competitive bidding in the rocket launch market.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Describe how data was collected
- Perform data wrangling
 - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.

- The initial data collection involved utilizing SpaceX's RESTful API, where a series of helper functions were defined to extract launch information using identification numbers. This process included making GET requests to the SpaceX API and parsing the JSON responses to ensure consistent data. The retrieved data was then converted into a Pandas data frame for further analysis.
- To complement the API data, web scraping techniques were employed to gather Falcon 9 historical launch records from a specific Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches." The BeautifulSoup and request libraries were utilized to extract the relevant HTML table records containing launch details. Subsequently, the extracted data was parsed and transformed into a Pandas data frame.
- By combining the data obtained from both the SpaceX API and the web scraping of the Wikipedia page, a comprehensive dataset was created, ready for further exploration and predictive modeling in the capstone project.

Data Collection – SpaceX API

- Data collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a Json result which was then converted into a Pandas data frame
- Add the GitHub URL of the completed SpaceX API calls notebook (<https://github.com/Muztoba/Capstone-assignment/blob/main/1.jupyter-labs-spacex-data-collection-api.ipynbs>)

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

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clo
```

We should see that the request was successfull with the 200 status response code

```
[10]: response.status_code
```

```
[10]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Using the dataframe `data` print the first 5 rows

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

Place your flowchart of web scraping here

Data Wrangling

- After obtaining and creating a Pandas DF from the collected data, data was filtered using the BoosterVersion column to only keep the Falcon 9 launches, then dealt with the missing data values in the LandingPad and PayloadMass columns. For the PayloadMass , missing data values were replaced using mean value of column.
- Also performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- [Here](#) is the GitHub URL of the completed data wrangling related notebooks.

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome` , create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome` ; otherwise, it's one. Then assign it to the variable `landing_class` :

```
[23]: # landing_class = 0 if bad_outcome  
# landing_class = 1 otherwise  
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)  
df['Class'].value_counts()
```

```
[23]: 1    60  
0    30  
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

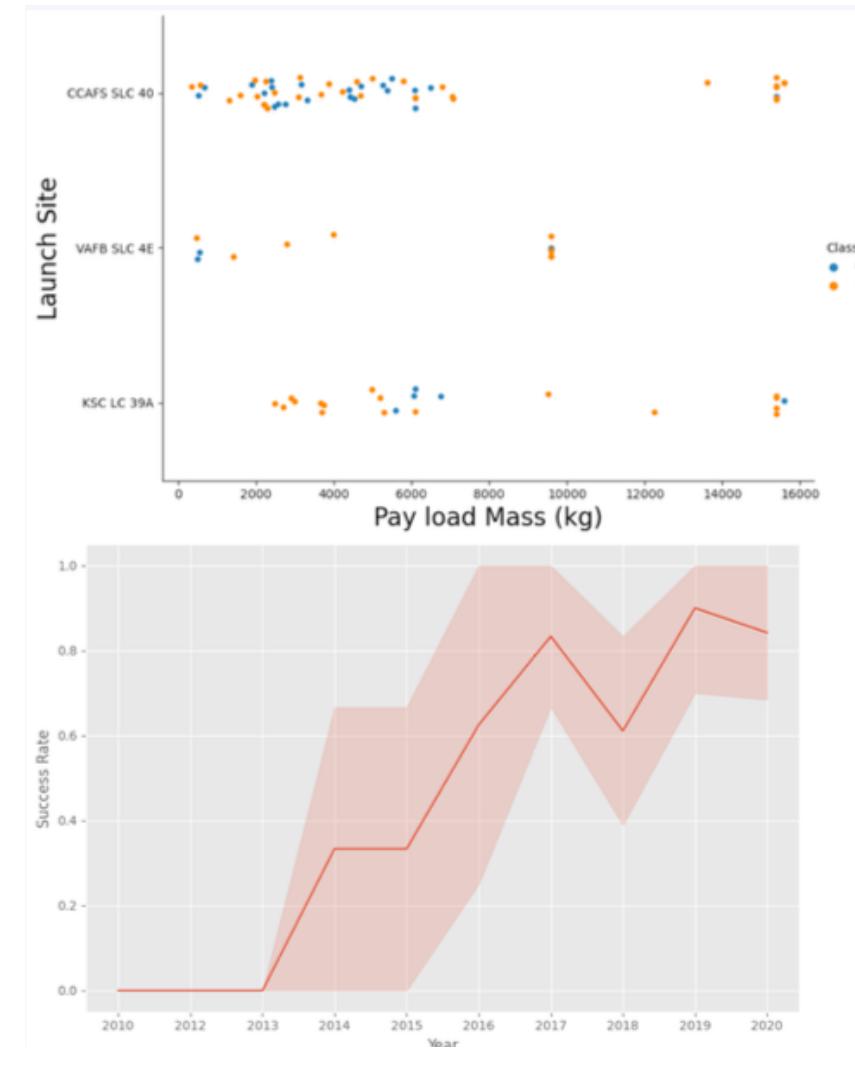
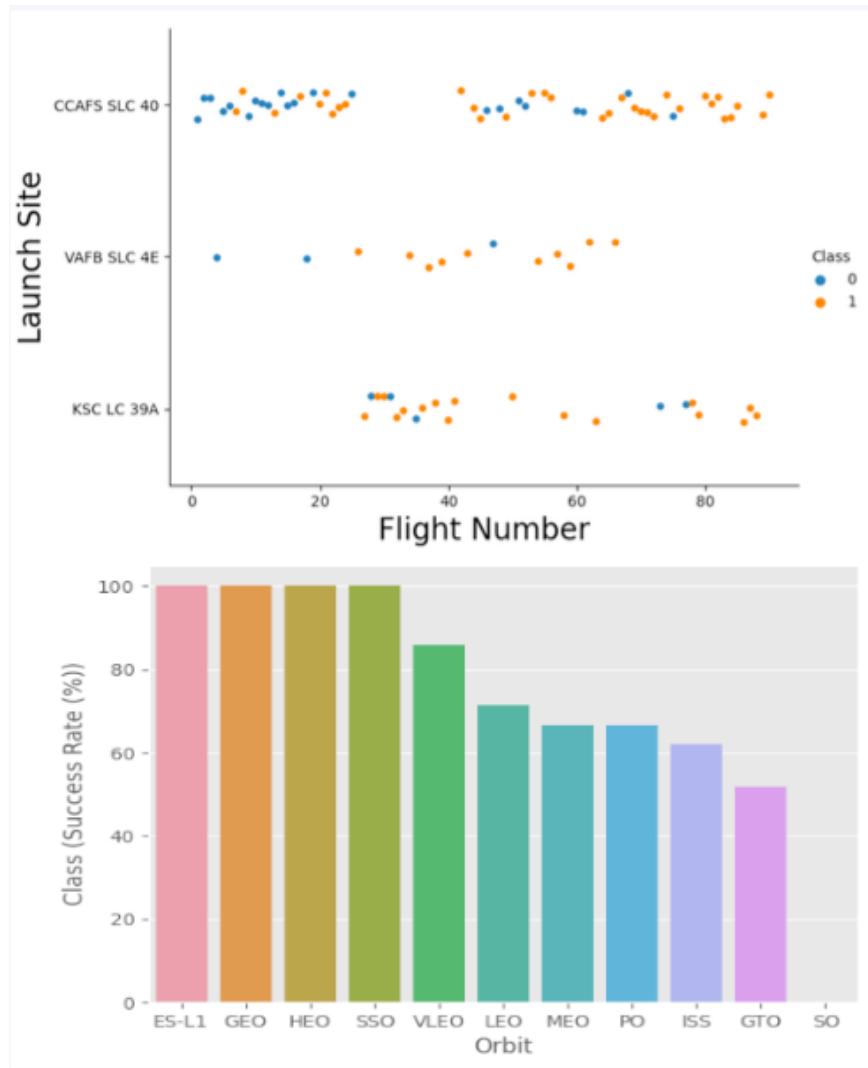
```
[26]: landing_class=df['Class']  
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA with Data Visualization

- Performed data Analysis and Feature Engineering using Pandas and Matplotlib.i.e.
 - Exploratory Data Analysis
 - Preparing Data Feature Engineering
- Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit type, Payload and Orbit type.
- Used Bar chart to Visualize the relationship between success rate of each orbit type
- Line plot to Visualize the launch success yearly trend.
- [Here](#) is the GitHub URL of your completed EDA with data visualization notebook,

EDA with Data Visualization



EDA with SQL

- The following SQL queries were performed for EDA

Display 5 records where launch sites begin with the string 'CCA'

```
[9]: %sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

```
[11]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Vers
```

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

```
[10]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)'
```

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

```
[8]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

EDA with SQL (Cont)

- The following SQL queries were performed for EDA

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

Hint: Use min function

```
[12]: %sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";  
* sqlite:///my_data1.db
```

Task 6

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

```
[13]: %sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYL  
* sqlite:///my_data1.db
```

Task 7

List the total number of successful and failure mission outcomes

```
[14]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";  
+-----+-----+
```

- [Here](#) is the github link

Build an Interactive Map with Folium

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Created a launch set outcomes (failure=0 or success=1)
- [Here is the GitHub URL](#) of the completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotly dash by:
- Adding a Launch Site Drop-down Input Component
- Adding a callback function to render success-pie-chart based on selected site dropdown
- Adding a Range Slider to Select Payload
- Adding a callback function to render the success-payload-scatter-chart scatter plot
- [Here](#) is the github link

Predictive Analysis (Classification)

- In order to find the best ML model/ method that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;
- First created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.
- For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find best Hyperparameter.
- After fitting the training set, we output GridSearchCV object for each of the models, then displayed the best parameters using the data attribute `best_params_` and the accuracy on the validation data using the data attribute `best_score_`.
- Finally using the method `score` to calculate the accuracy on the test data for each model and plotted a confusion matrix for each using the test and predicted outcomes.

Predictive Analysis (Classification)

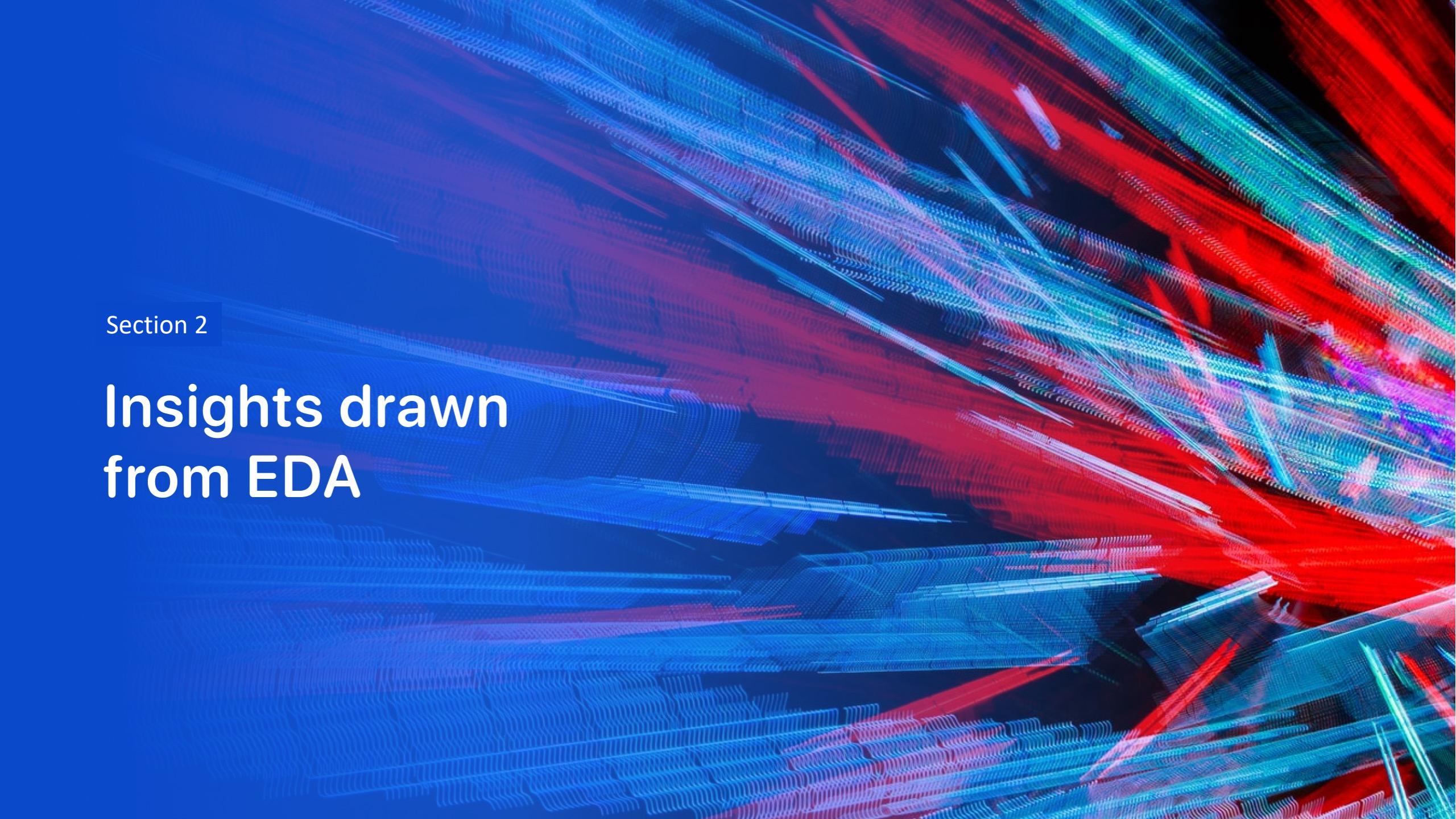
- The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

- [Here](#) is the github link

Results

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

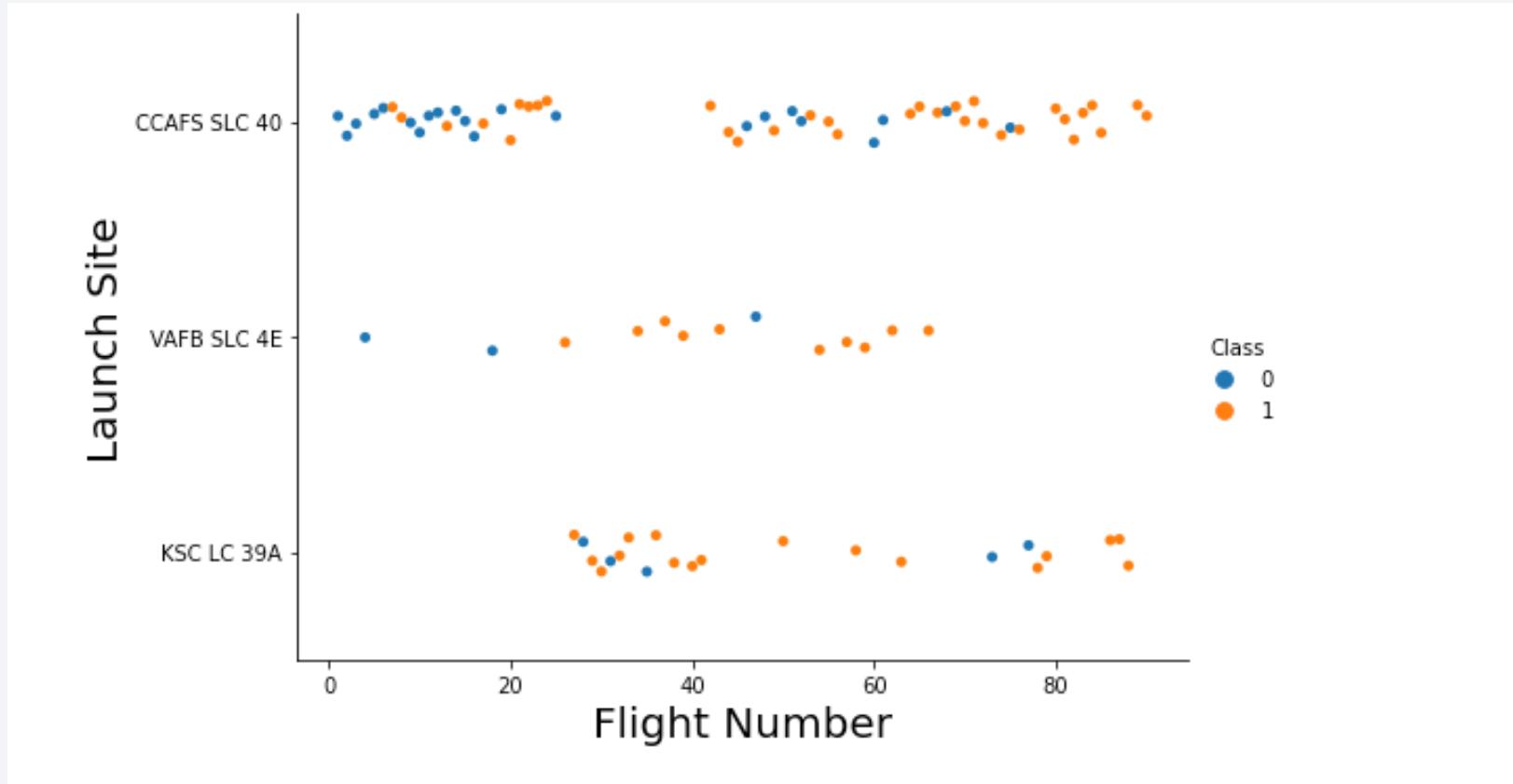
The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



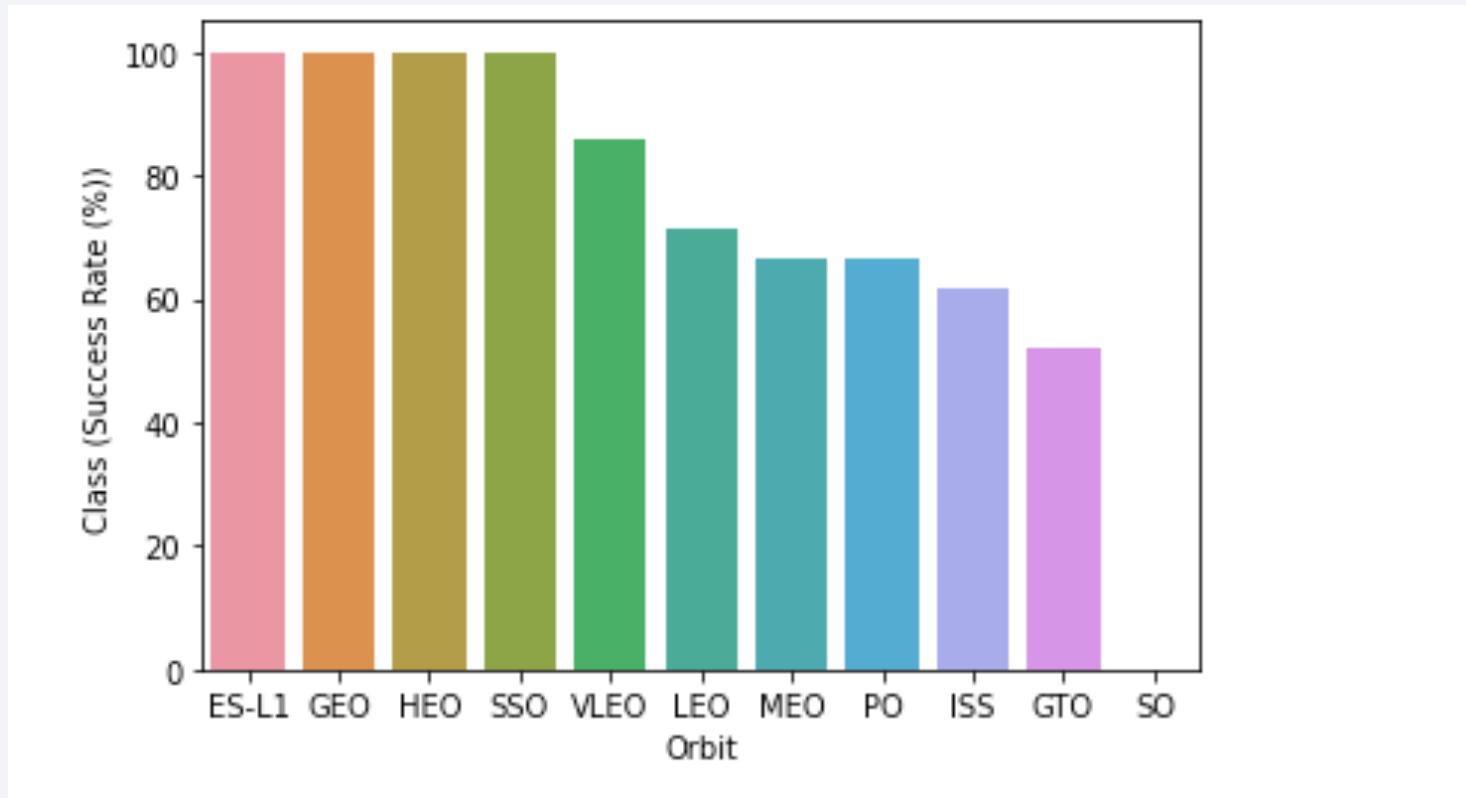
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



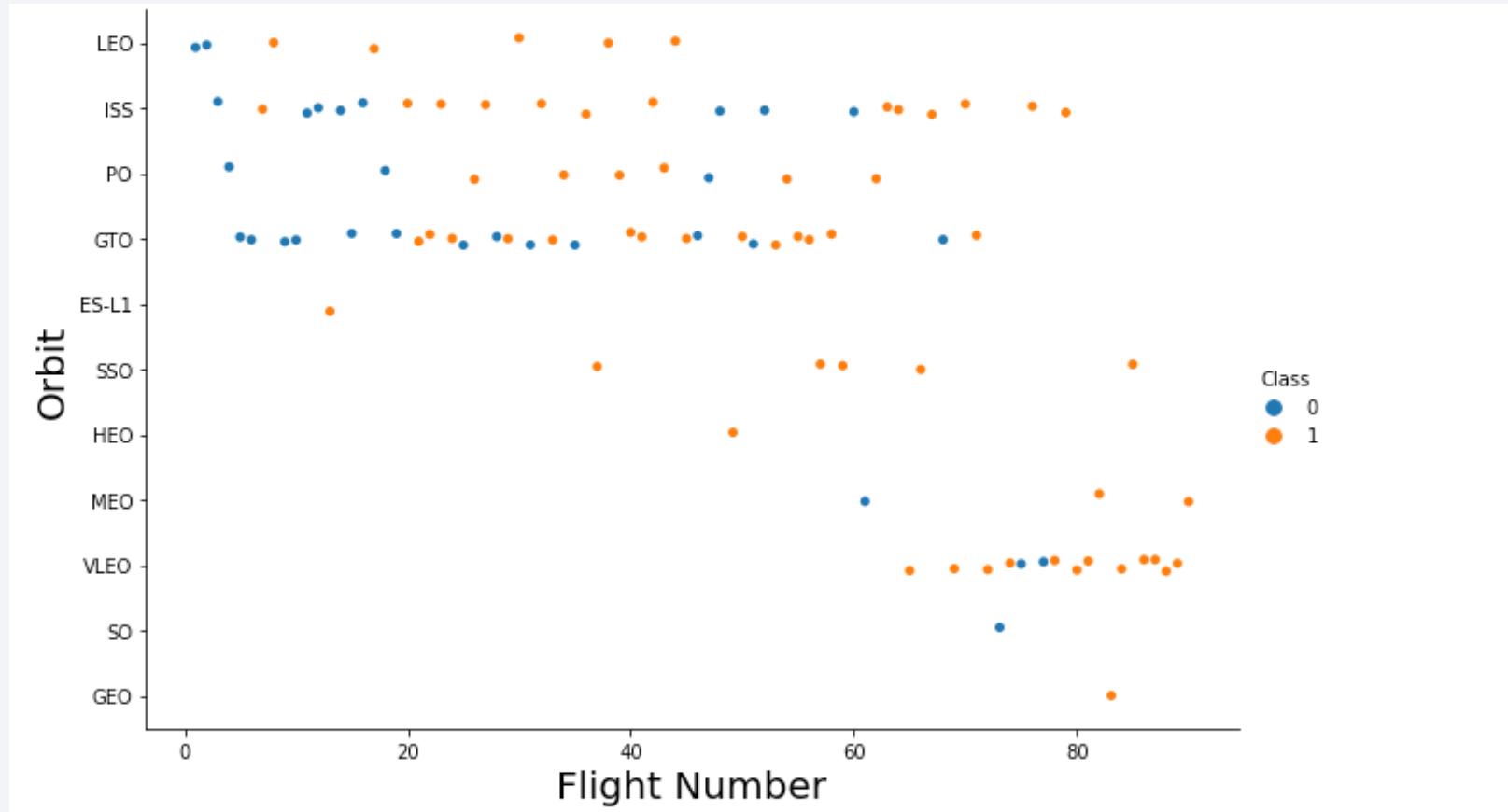
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type



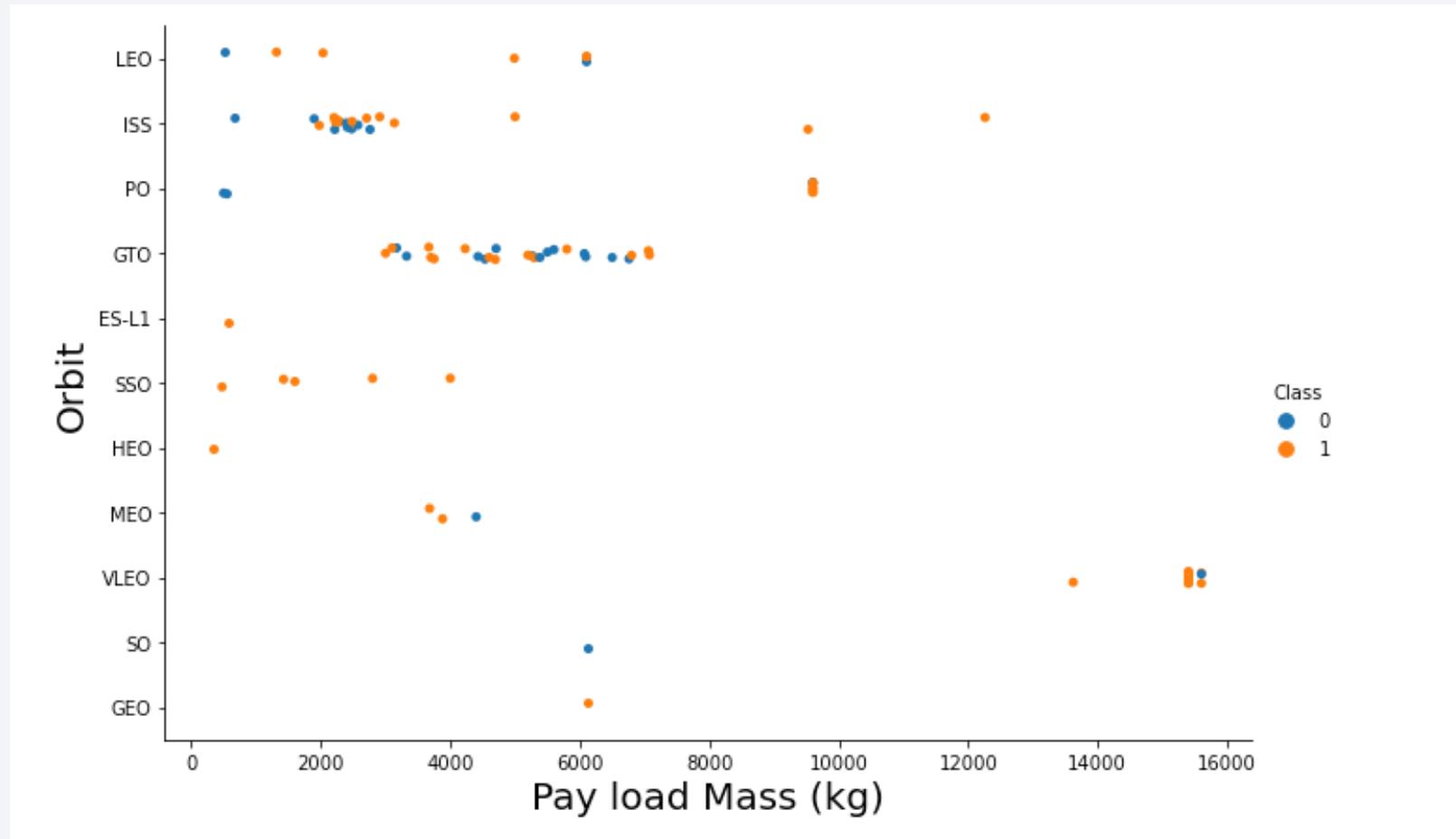
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



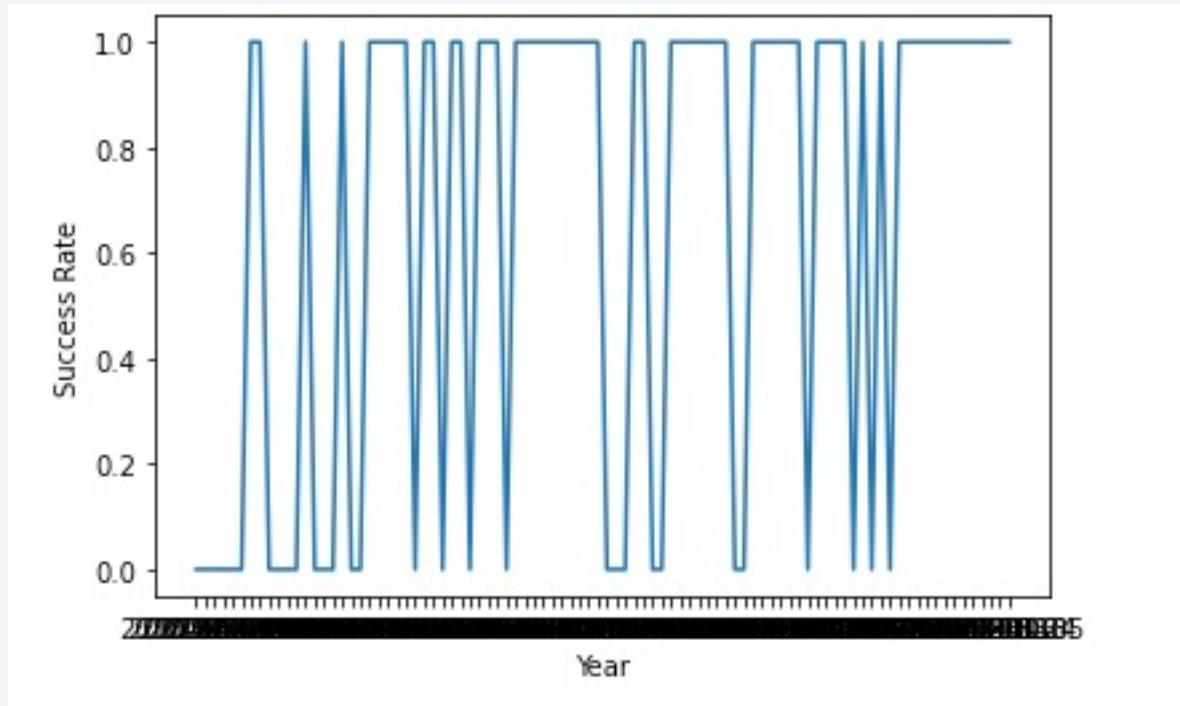
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



Launch Success Yearly Trend

- Show a line chart of yearly average success rate



All Launch Site Names

- Find the names of the unique launch site

```
[8]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;  
      * sqlite:///my\_data1.db  
Done.  
[8]: Launch_Sites  
-----  
    CCAFS LC-40  
    VAFB SLC-4E  
    KSC LC-39A  
    CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
[9]: %sql SELECT * FROM 'SPACEXTBL' 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Task 3

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

```
[10]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)  
* sqlite:///my_data1.db  
Done.  
[10]: Total Payload Mass(Kgs)      Customer  
      45596  NASA (CRS)
```

Used the ‘SUM()’ function to return and dispaly the total sum of ‘PAYLOAD_MASS_KG’ column for Customer ‘NASA(CRS)’)

Average Payload Mass by F9 v1.1

Task 4

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

```
[11]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE  
      * sqlite:///my_data1.db  
Done.  
[11]:   Payload Mass Kgs  Customer  Booster_Version  
       2534.6666666666665    MDA      F9 v1.1 B1003
```

Used the 'AVG()' function to return and display the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
[12]: %sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)"
```

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

```
[12]: MIN(DATE)
```

```
None
```

Used the ‘MIN()’ function to return and dispaly the first (oldest) date when first successful landing outcome on ground pad ‘Success (ground pad)’happened.

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

```
[26]: # %sql SELECT * FROM 'SPACEXTBL'
```

```
[27]: %sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS_*
* sqlite:///my_data1.db
Done.
```

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Used ‘Select Distinct’ statement to return and list the ‘unique’ names of boosters with operators >4000 and <6000 to only list booster with payloads btween 4000-6000 with landing outcome of ‘Success (drone ship)’.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[14]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Used the ‘COUNT()’ together with the ‘GROUP BY’ statement to return total number of missions outcomes

Boosters Carried Maximum Payload

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

```
[15]: %sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS_KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS_KG_" = (SELECT MAX("PAYLOAD_MASS_KG_")  
* sqlite:///my_data1.db  
Done.
```

Booster_Version	Payload	PAYLOAD_MASS_KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0. GPS III-04	15600

Using a Subquery to return and pass the Max payload and used it list all the boosters that have carried the Max payload of 15600kgs

2015 Launch Records

Task 9

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.

Note: SQLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
[68]: %sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome"
      * sqlite:///my_data1.db
Done.

[68]:
```

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

Used the ‘substr()’ in the select statement to get the month and year from the date column where substr(Date,7,4)='2015' for year and Landing_outcome was ‘Failure (drone ship)’ and return the records matching the filter.

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

Task 10

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

```
[74]: %sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY
```

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

```
Done.
```

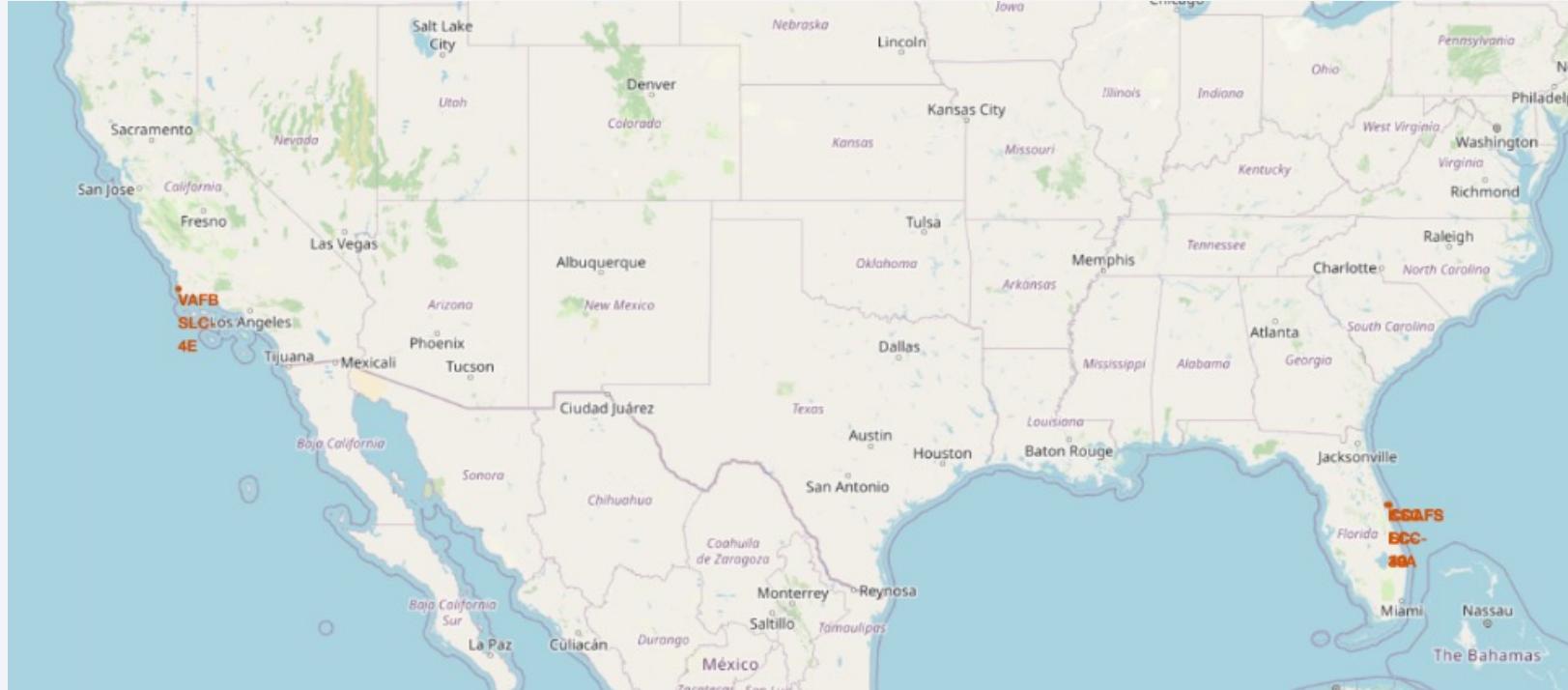
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against a dark blue-black void of space. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper right, the green and yellow glow of the aurora borealis is visible. The atmosphere of the Earth is thin and hazy, appearing as a light blue band near the horizon.

Section 3

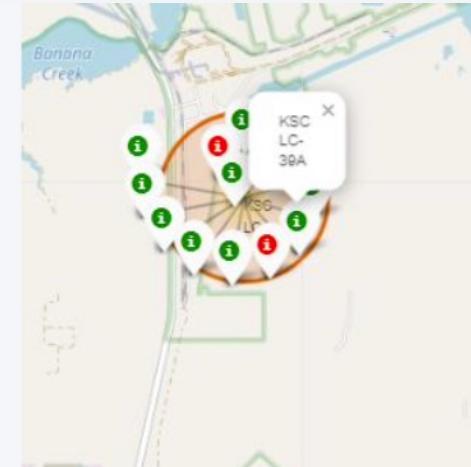
Launch Sites Proximities Analysis

Markers of all launch sites on global map



All launch sites are in proximity to the Equator, (located southwards of the US map). Also all the laumch sites are in very close proximity to the coast.

Launch outcomes for each sites

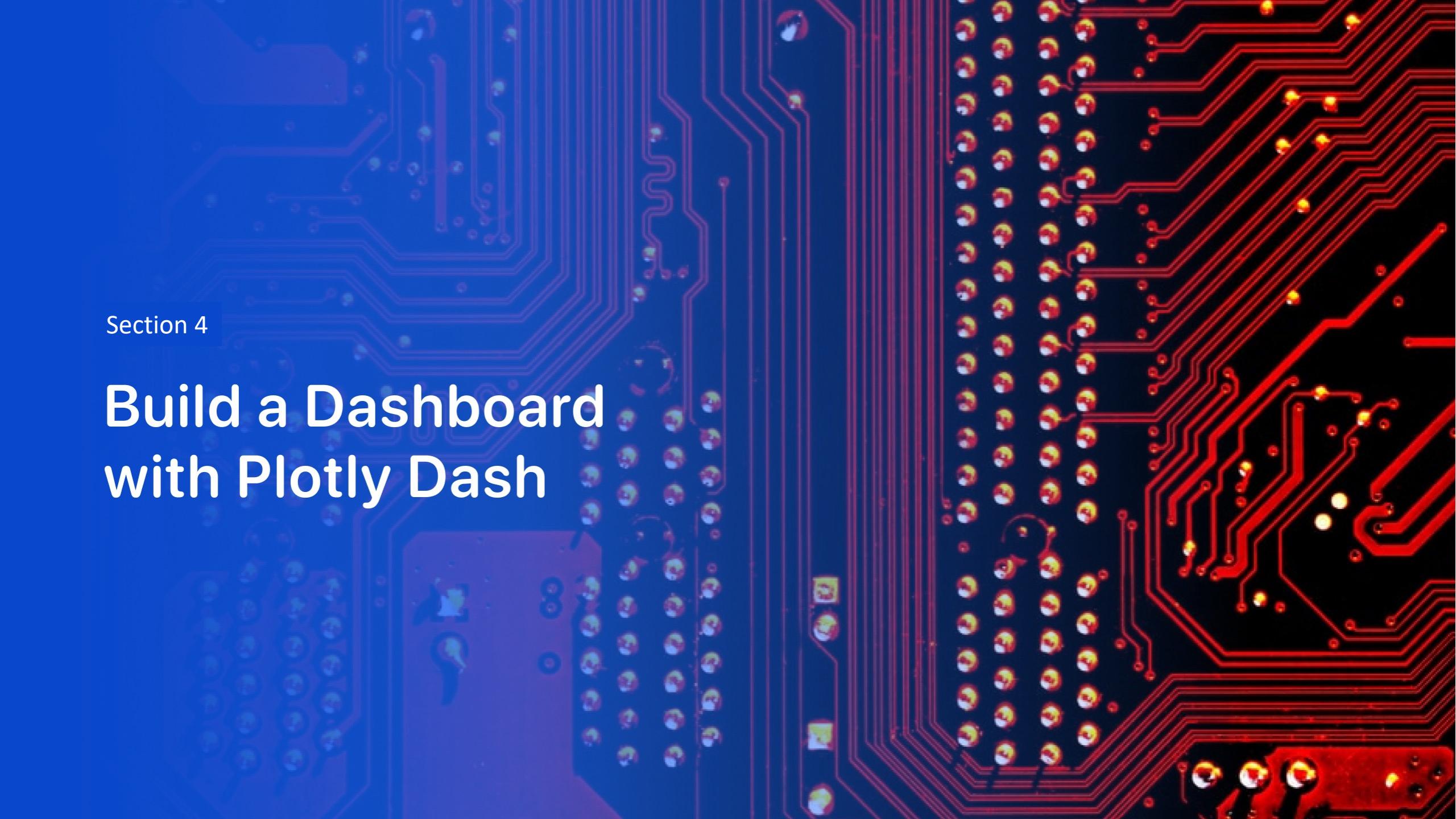


In the Eastern coast (Florida) Launch site KSC LC-39A has relatively high success rates compared to CCAFS SLC-40 & CCAFS LC-40.

Distances between a launch site to its proximities



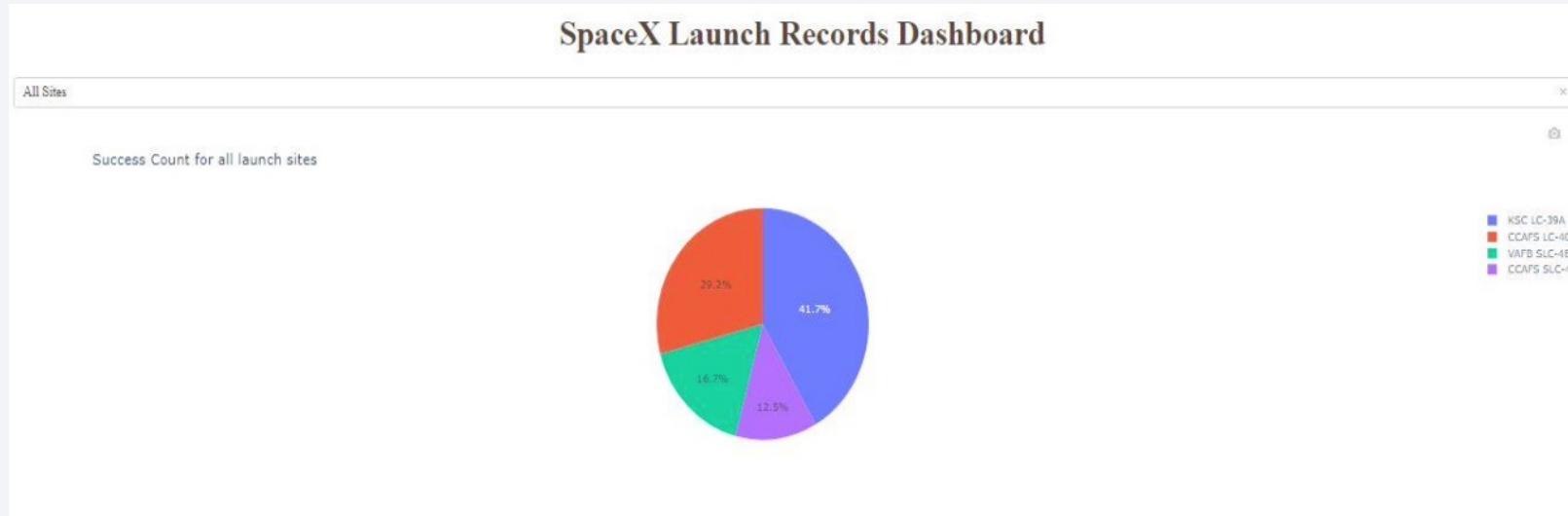
Launch site CCAFS SLC-40 proximity to coastline is 0.90 km

The background of the slide features a detailed image of a printed circuit board (PCB). The left side of the image is tinted blue, while the right side is tinted red. The PCB is populated with various electronic components, including resistors, capacitors, and integrated circuits, all connected by a complex network of red and blue printed circuit lines.

Section 4

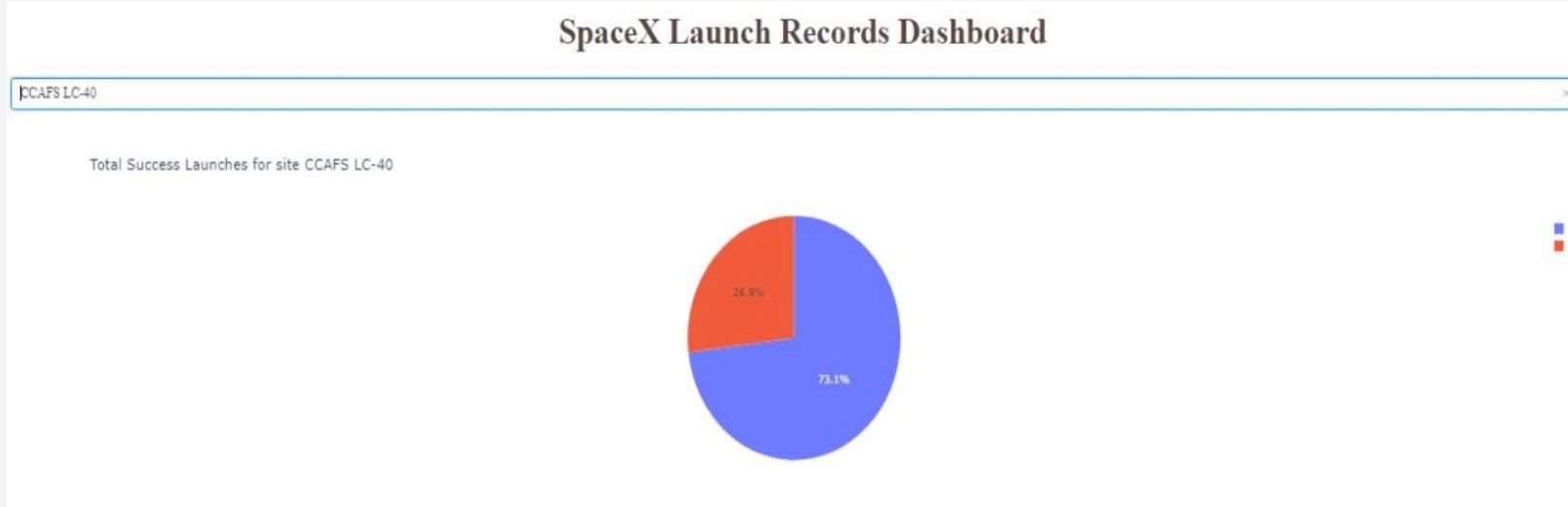
Build a Dashboard with Plotly Dash

Pie chart for launch success count for all sites



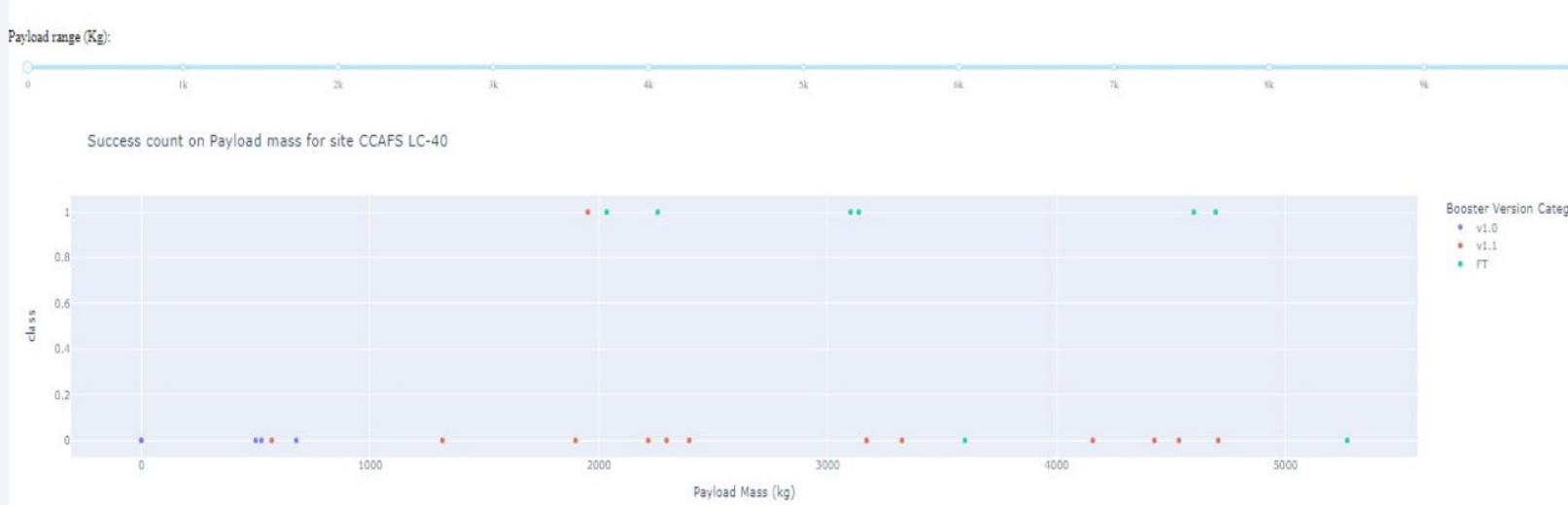
Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Pie chart for the launch site with 2nd highest launch success ratio



Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success against 27% failed launches

Payload vs launch outcome



For Launch site CCAFS LC-40 the booster version FT has the largest success rate from a payload mass of >2000kg

Section 5

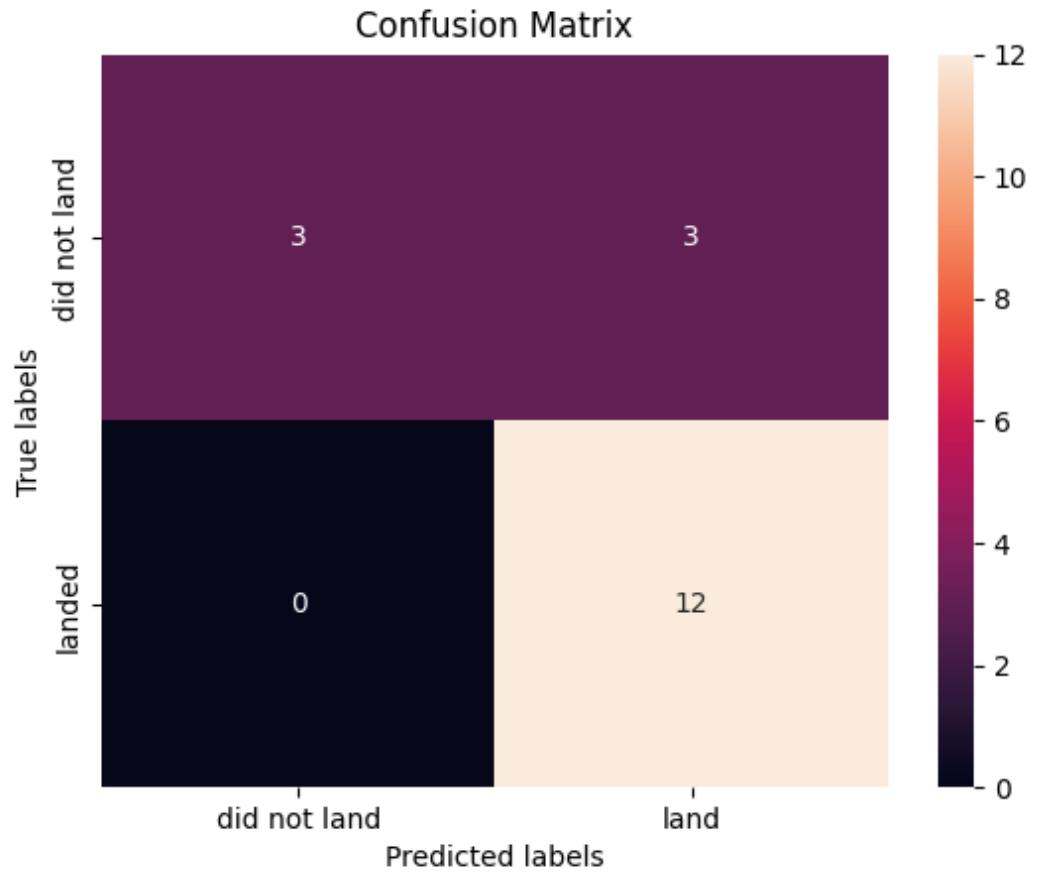
Predictive Analysis (Classification)

Classification Accuracy

[68]:

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix



All the 4 classification model had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is false positives for all the models.

Conclusions

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- We can deduce that, as the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight
- If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.
- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Thank you!

