IBM-Data-Science-Capstone-SpaceX

Brent Williams

https://github.com/Brent-W/IBM

04/04/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API
  - Data Collection via Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with SQL
  - Exploratory Data Analysis (EDA) with Data Visualization
  - Interactive Visual Analytics with Folium and Plotly Dash
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis (EDA) results
  - Interactive Analytics Images
  - Predictive Analytics Result

Introduction

- Project background and context

SpaceX advertises the $ 62 million Falcon 9 rocket launch on its website. The other providers will cost more than $ 165 million, and much of the savings come from being able to reuse the first steps. Therefore, if you can determine if the first stage will land, you can determine the cost of the launch. This information can be used if an alternative company wants to bid on SpaceX for a rocket launch. The goal of this project is to build a machine learning pipeline and predict whether the first stage will land successfully.

- 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.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1: Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping.
- Performed data wrangling
  - One-hot encoding method was applied to the categorical features in the data.
  - Classifying landings as successful or unsuccessful.
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - Models were tuned using GridSearchCV.

Data Collection

- The data was collected using various methods:
  - Data collection was through the SpaceX API using the get request method.
  - We decoded the response content as a Json using .json() function call and converted it into a Pandas data frame using the .json_normalize().
  - Then we cleaned the data thoroughly by checking for missing values and filled in missing values where it was necessary.
  - We also performed web scraping from Wikipedia for Falcon 9 launch records by using the BeautifulSoup package.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

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.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%201%20:%20SpaceX%20Falcon%20Data%20Collection-Wrangling.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-Skill
```

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

```
response.status_code
```

```
200
```

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

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

Data Collection - Scraping

- We used web scrapping with BeautifulSoup to collect Falcon 9 launch records from Wikipedia.
- We used the html parser and converted it into a Pandas data frame.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%201:%20Data%20Collection%20with%20Web%20Scraping.ipynb

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

Next, request the HTML page from the above URL and get a `response` object

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

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

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

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

Data Wrangling

- Through this process we used exploratory data analysis (EDA) and determined exactly what labels were the training labels.
- During this process we also calculated the amount of launches that occurred at each site, and the number and occurrence of each orbits.
- We also created a landing outcome label from the outcome column and exported the final results to a csv file.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%201:%20EDA%20(Data%20Wrangling).ipynb

EDA with Data Visualization

- Here we explored the relationship between:
- Flight number and Launch Site
- Payload and Launch site
- Success rate of each orbit type
- Flight number and Orbit Type
- And finally, the launch success yearly trend
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%202:%20EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- We loaded the SpaceX dataset into IBM Db2 database and connected to it using Jupyter Notebooks.
- We applied EDA with SQL to get insights from the data. We wrote queries to find out for instance:
  - The unique launch sites in the space mission.
  - The total amount of payload mass that was carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total amount of mission outcomes that were successful or a failure
  - The landing outcomes in drone ship, their booster version and launch site names that failed.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%202:%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Using Folium, we marked all the launch sites, and added objects like markers, circles, lines to mark and show the success or failure of launches for each site on the Folium map.
- We assigned the launch outcomes to 0 and 1. 0 for failure and 1 for successful.
- We identified which launch sites have relatively high success rate by using color-labeled marker clusters.
- We used Folium to calculate the distances between a launch site to certain places. For example:
  - How near are the launch sites torailways, highways and coastlines?
  - What is the approximate distance that the launch sites are from cities?
- Link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%203:%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

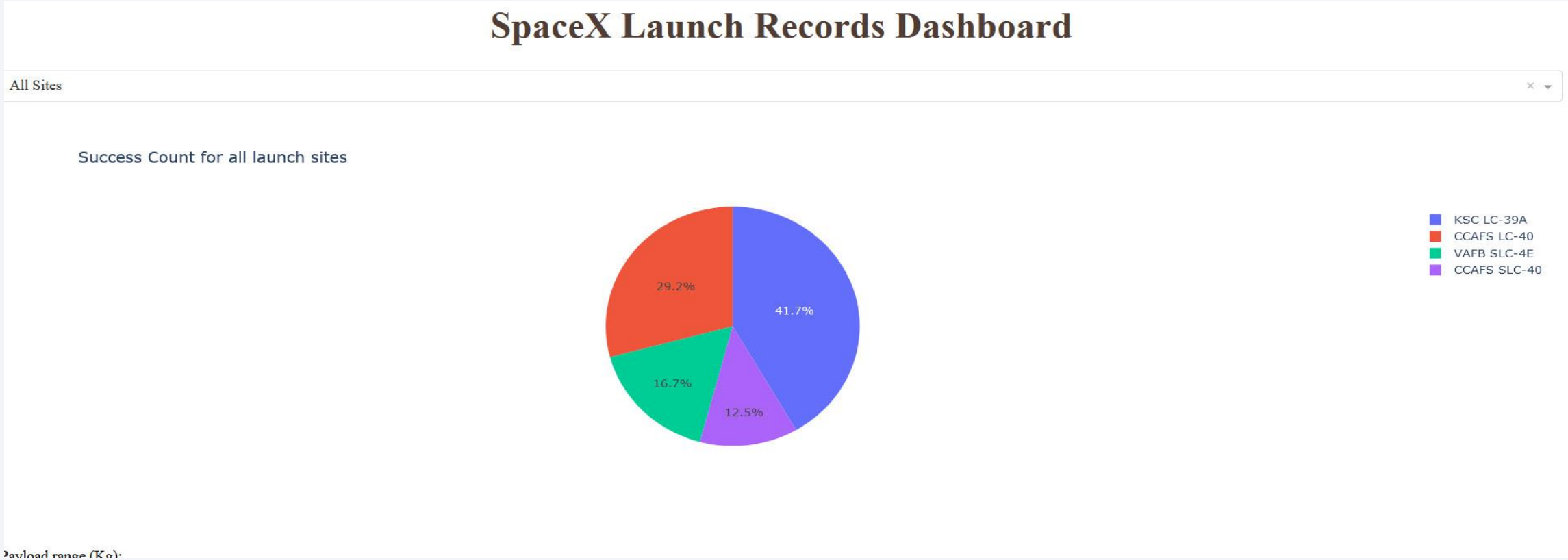
Build a Dashboard with Plotly Dash

- Using Plotly Dash, we built an interactive dashboard of the data.
- We used pie charts to show the total launches that occurred in each site.
- We also plotted a scatter graph that visualized the relationship between the Outcome and Payload Mass (Kg) for all the different booster versions.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%203:%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py

Predictive Analysis (Classification)

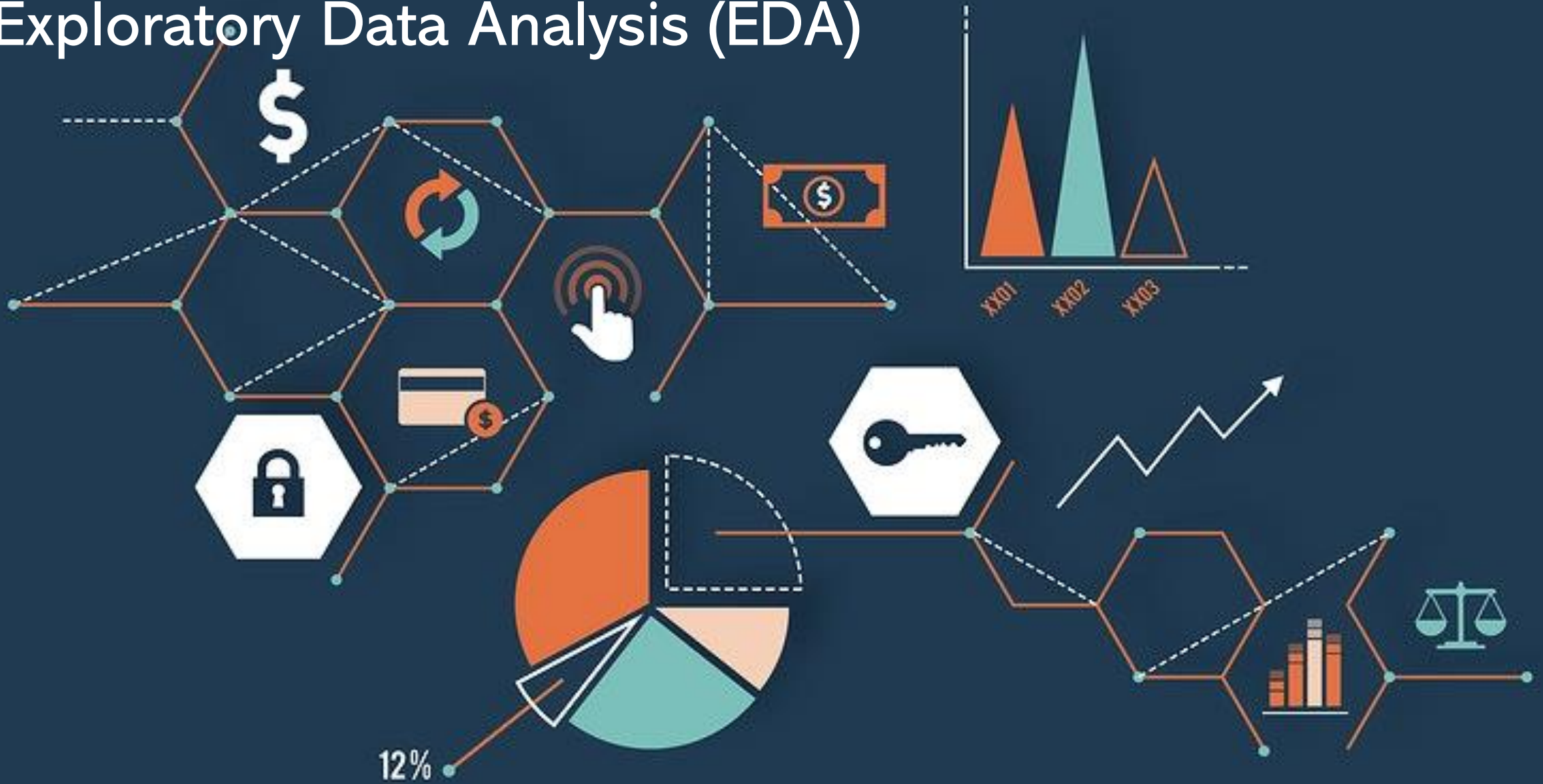
- In this phase we used Pandas and NumPy to transform the data, and we also used Scikit-Learn to split the data into training and test sets.
- We used different machine learning classification models and tuned different hyperparameters using the GridSearchCV.
- Using accuracy as an indicator of the model, we improved the model through feature engineering and algorithm tuning.
- We also determined which classification model performed the best.
- The link to the notebook is https://github.com/Brent-W/IBM/blob/master/Week%204:%20Machine%20Learning%20Prediction.ipynb

Results



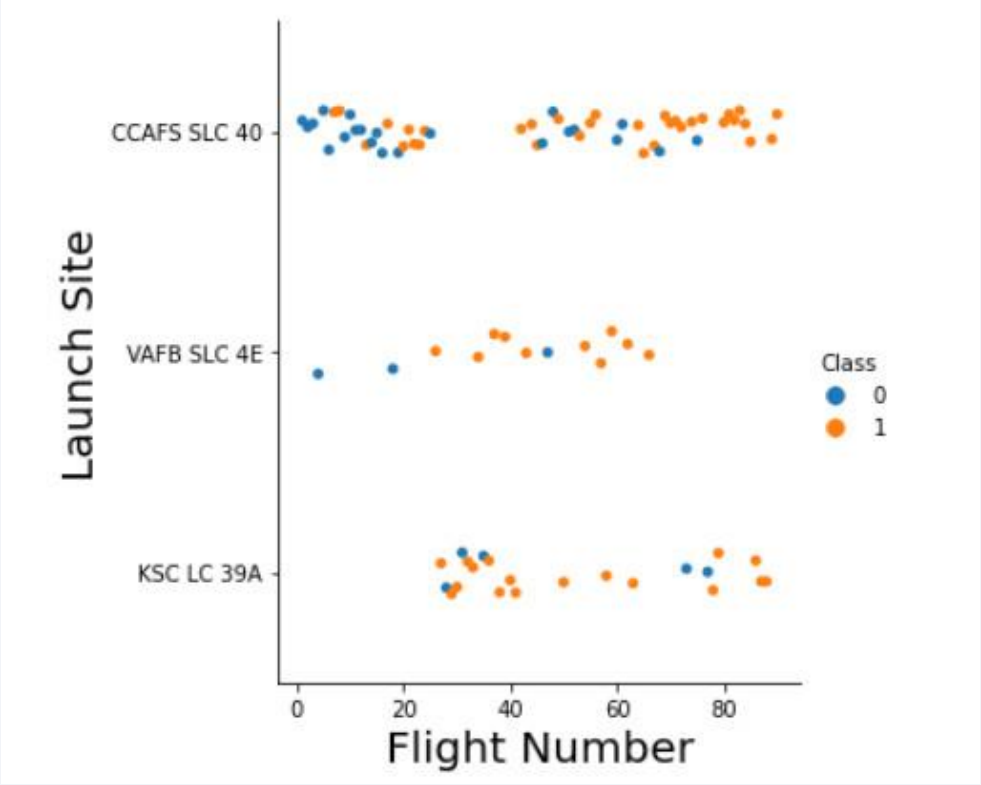
This image is a preview of the Plotly dashboard and the following slides will show the results of the exploratory data analysis (EDA) with the visualizations, SQL, the interactive map with Folium and the result of our classification model.

Section 2: Insights drawn from Exploratory Data Analysis (EDA)



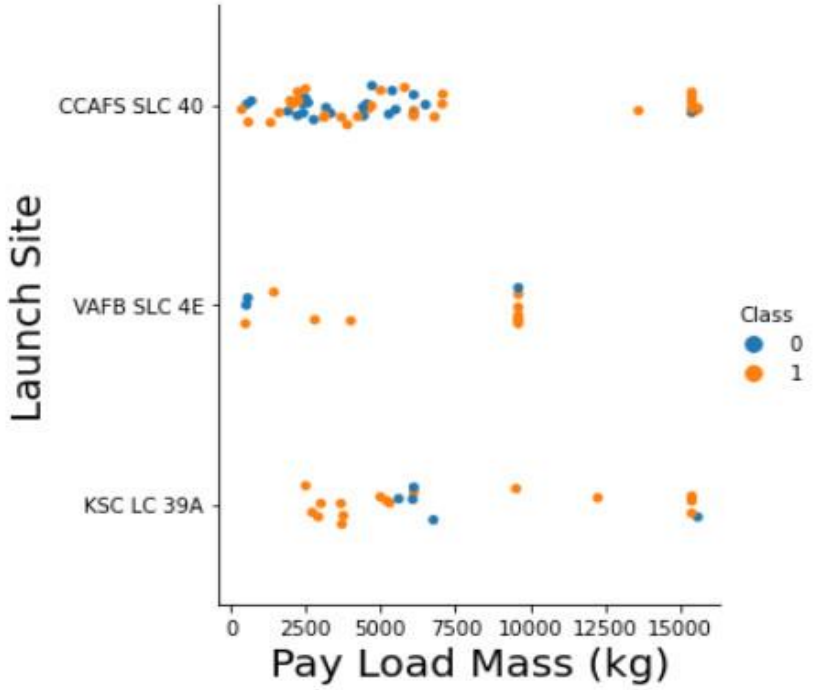
Flight Number vs. Launch Site

- From the scatter plot we find that the higher the number of flights at the launch site, the higher the success rate at the launch site.
- CCAFS appears to be the main launch site as it shows the most number of launches.
- In the graph, blue indicates an unsuccessful launch and orange indicates a successful launch.



Payload vs. Launch Site

- When looking at the mass of the payload mass, it mostly seems to fall between 0-6000 Kg.
- It also seems to appear that different launch sites vary in the mass that they use.
- 0 indicates an unsuccessful launch and 1 indicates a successful launch.



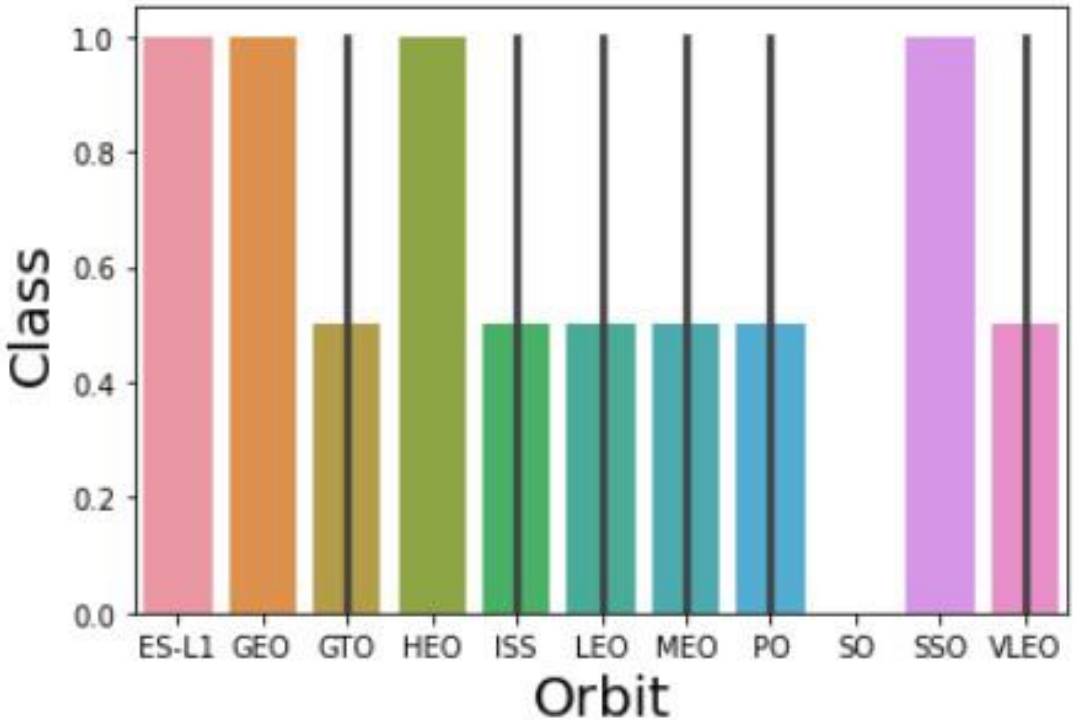
Success Rate vs. Orbit type

In the graph, the scale is as follows:

- 0 indicates a 0% success rate.
- 0.4 indicates a 40% success rate
- 1 indicates a 100% success rate

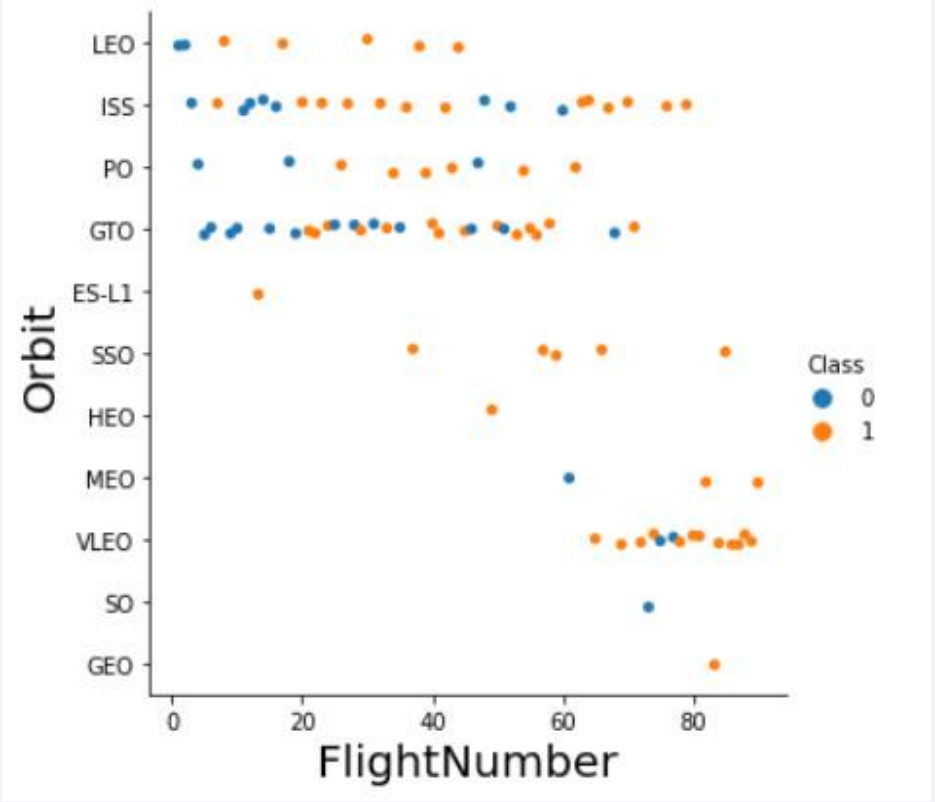
- ES-L1, GEO, HEO, and SSO has a 100% success rate

- GTO, ISS, LEO, MEO, PO, and VLEO all have a success rate of approximately 50%



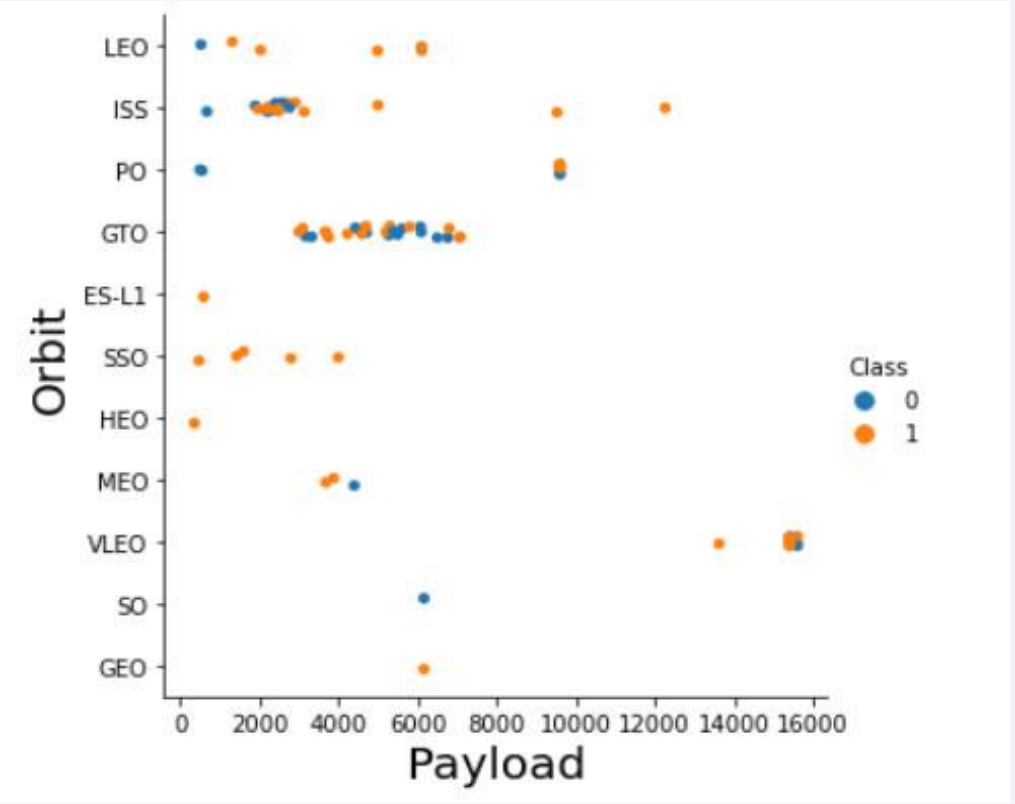
Flight Number vs. Orbit Type

- The scatter plot shows the Flight Number vs. Orbit type.
- According to the plot, Launch Orbit preferences changed over Flight Number.
- We note that SpaceX appears to perform better in lower orbits.



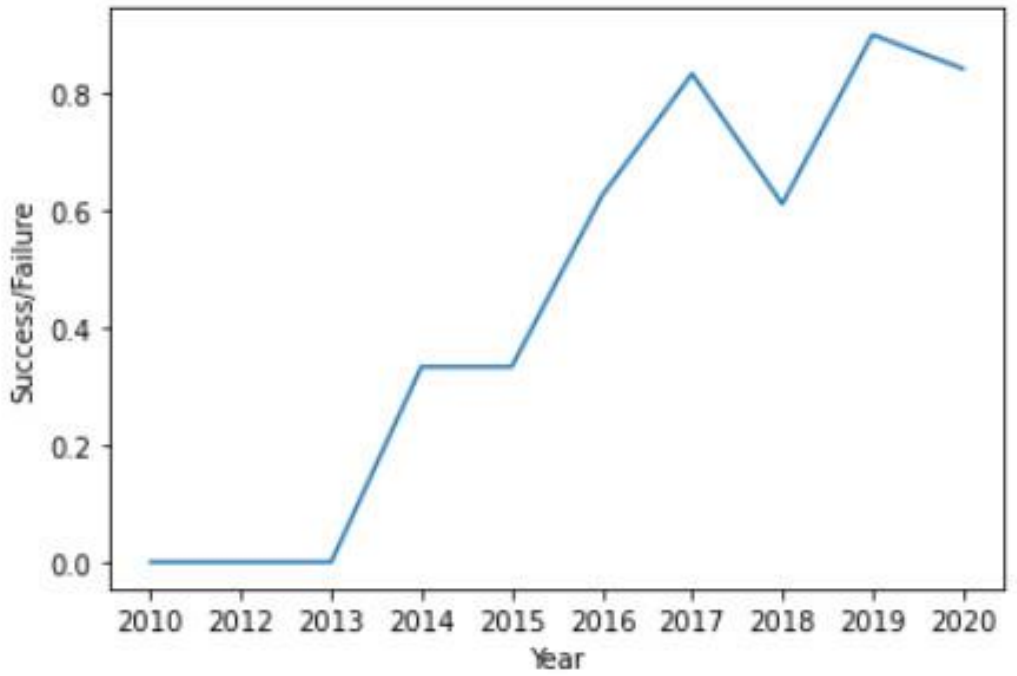
Payload vs. Orbit Type

- From the scatter plot, we can see that there's a strong correlation between the payload and the orbit.
- We see that the orbit VLEO only has a payload mass in the higher end of the range.
- 0 indicates an unsuccessful launch, and 1 indicates a successful launch.



Launch Success Yearly Trend

- Based on the graph we can see success generally increases over time since 2013 with a slight dip in 2018
- We can see that the success in recent years at around 80%
- By 2019 we see another increase that goes slightly above 80%



EDA with SQL

Exploratory data analysis (EDA) with SQL using IBM Db2 and Jupyter Notebook

Distinct Launch Site Names

- Using SQL, we used the keyword **DISTINCT** to search the table for distinct launch site names.

- When looking at CCAFS LC-40 and CCAFS SLC-40 could be the exact same launch site, but someone made a data entry error that caused the names to differ.

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

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-
Done.
```

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

Launch Site Names Begin with 'CCA'

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

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

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

We used this query to collect 5 records where the launch site name began with ‘CCA’

Total Payload Mass by NASA (CRS)

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

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

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lc
Done.
```

1
45596

- This query sums the total payload mass in kg where NASA was the customer, and the total amounted to 45596 kg.
- CRS stands for Commercial Resupply Services which shows that those payloads had been despatched to the International Space Station (ISS).

Average Payload Mass by F9 v1.1

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

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

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde0(
Done.
```

1
2928

- Using this query we calculated the average payload mass that was carried by booster version F9 v1.1.
- We see that the average falls on the low end range of the payload mass.

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing__Outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00
Done.
```

1
2015-12-22

Using this query, we extracted the date of the first successful landing outcome on ground pad which occurred on the 22nd December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE Landing__Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MA
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

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

In this query, we used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%' GROUP BY MISSION_OUTCOM
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

1
1
99
1

- We used the symbol like '%' to filter in the **WHERE** clause whether the Mission Outcome was a success or a failure.
- There was 1 failure (in flight)
- 99 successful mission outcomes.
- 1 success but the payload status is unclear.

Boosters that Carried Maximum Payload

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

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

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3069
Done.
```

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

- This query returns the booster versions that carried the highest payload mass which amounted to 15600 kg.

- We can see that all booster versions are almost alike and all are of the F9 B5 B10xx.x category.

2015 Launch Records That Failed

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databa
Done.
```

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

- We used a combinations of the **WHERE**, **LIKE**, and **AND** clause to filter for failed landing outcomes in drone ships, their booster versions, and launch site names that occurred in the year 2015.
- We find that there were 2 occurrences in January and April.

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

```
%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS NO_OUTCOME \
FROM SPACEXTBL \
WHERE (LANDING__OUTCOME LIKE '%Success%') AND DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING__OUTCOME \
ORDER BY NO_OUTCOME DESC
```

```
 * ibm_db_sa://zwf76444:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databa
Done.
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

- We selected Landing outcomes and the **COUNT** of landing outcomes from the dataset and used the **LIKE** and **WHERE** clause to filter for successful landing outcomes **BETWEEN** 2010-06-04 **AND** 2017-03-20.

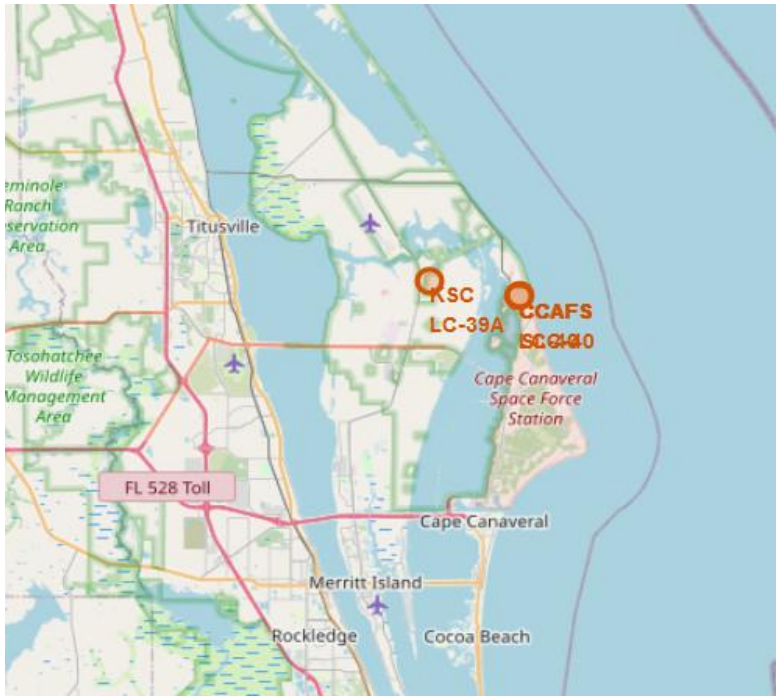
- We used the **GROUP BY** clause to group by the different successful landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 3: Launch Sites Proximities Analysis

Launch Site Locations

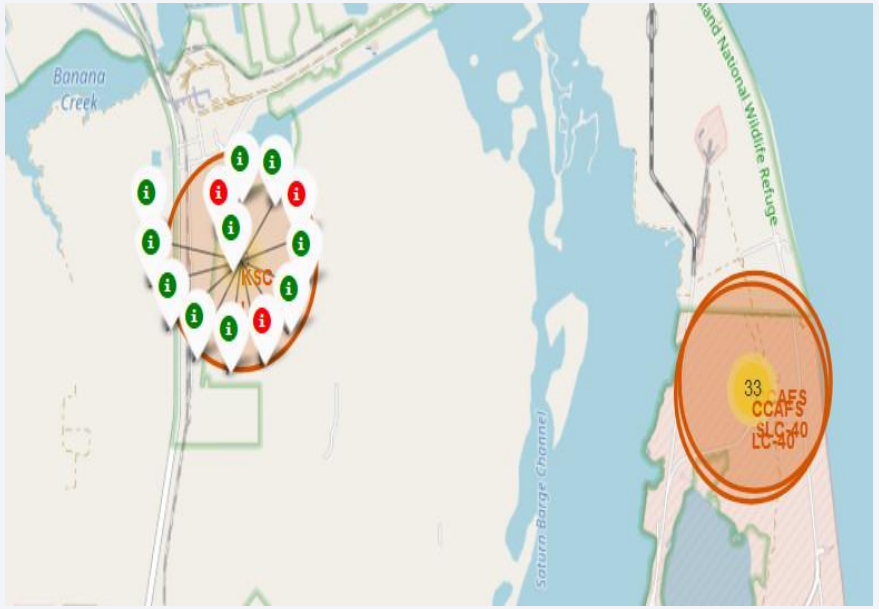
This map shows all the launch sites on the US map.

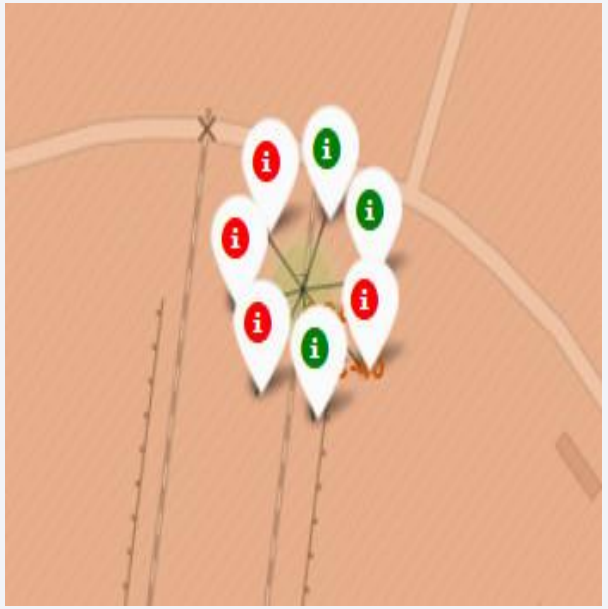
Launch Site Locations – Continued

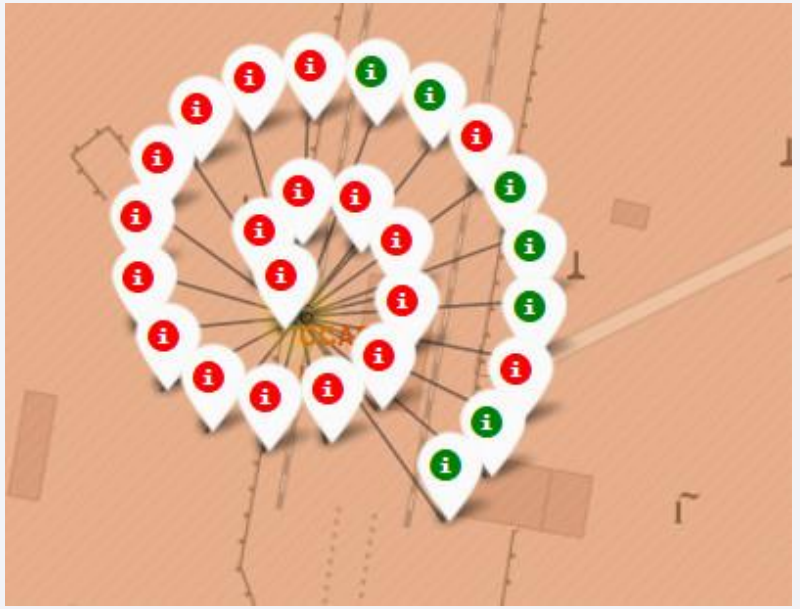


- The map on the left side shows the launch sites that are located in Florida.
- The map on the right side shows the launch site that are located in California.

Markers showing launch sites with color labels – Florida

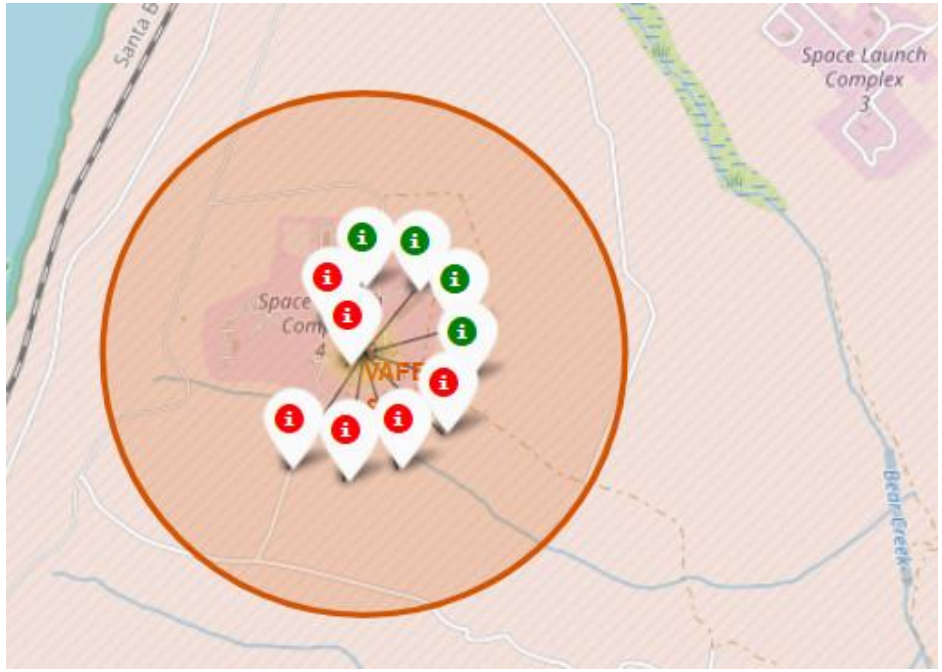






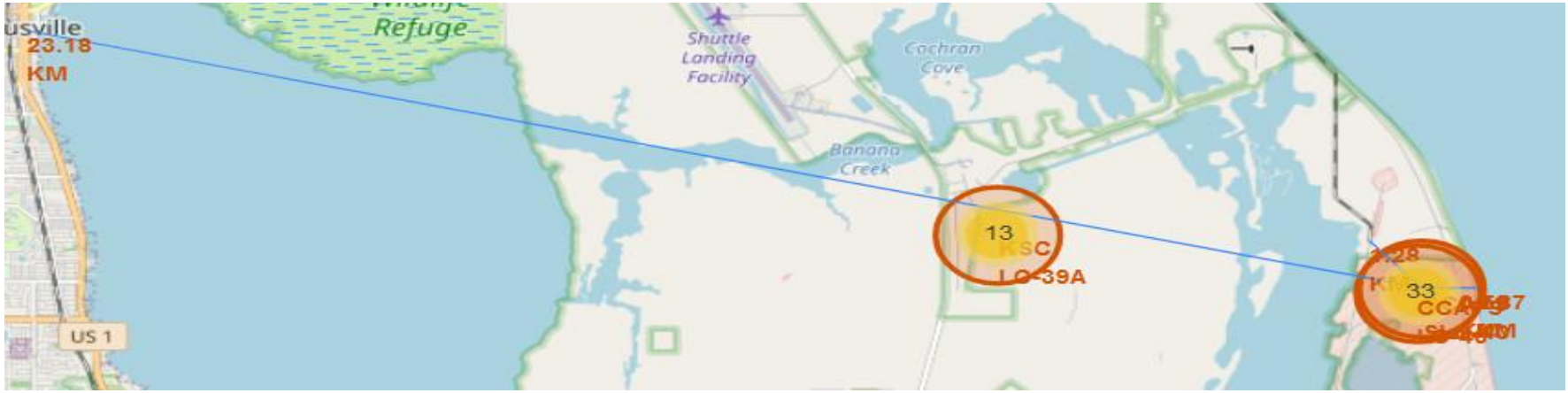
- On the marker shown Green indicates a successful landing, while a red marker shows a failed landing.

Markers showing launch sites with color labels – California

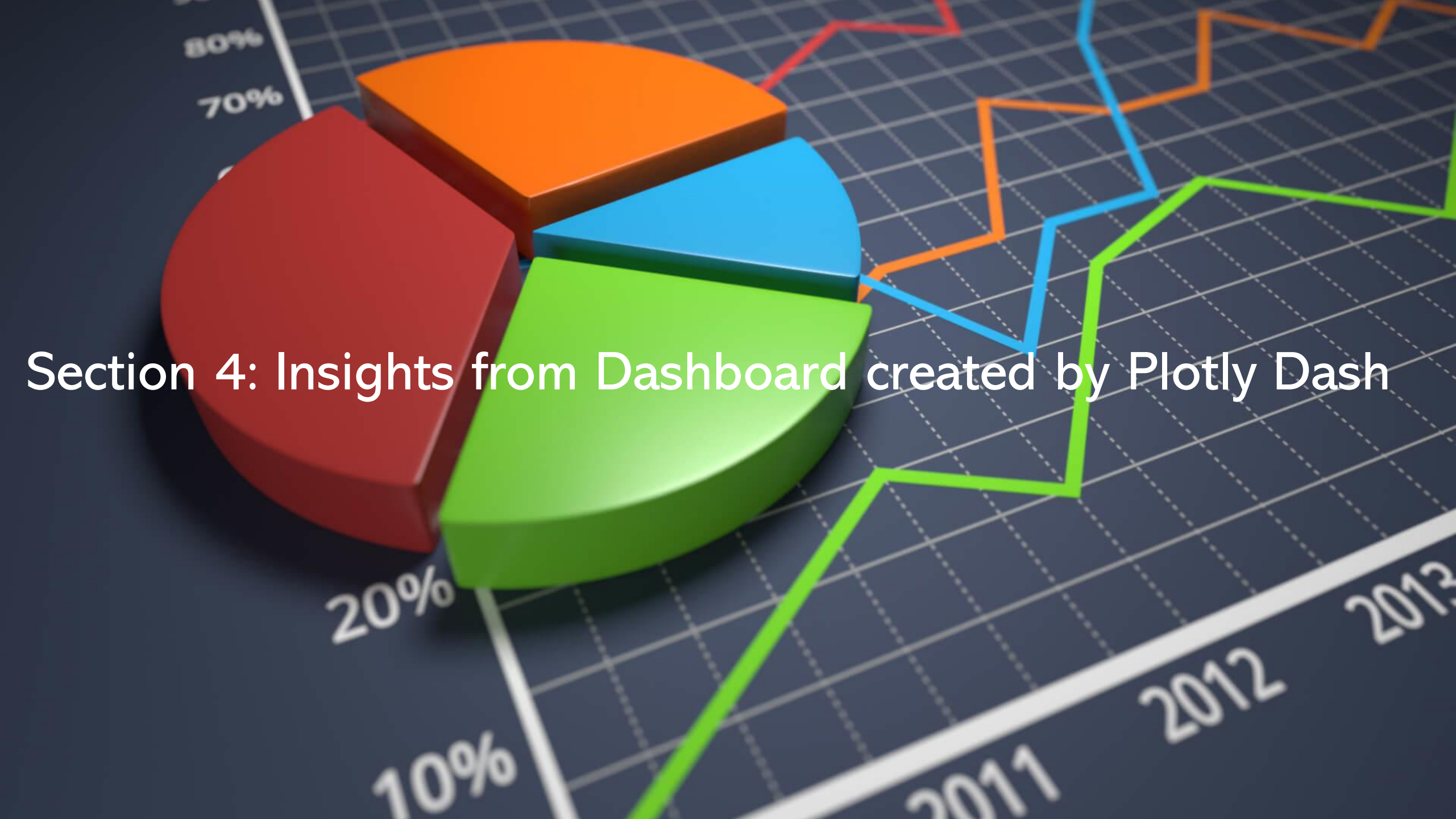


- This map shows the markers of the launch site in California.
- The green markers indicates a successful landing while a red marker indicates a failed landing.
- By this launch site, we see that there were 4 landings that were successful while there were 6 failed landings.

Florida Launch Site Distance to Different Landmarks

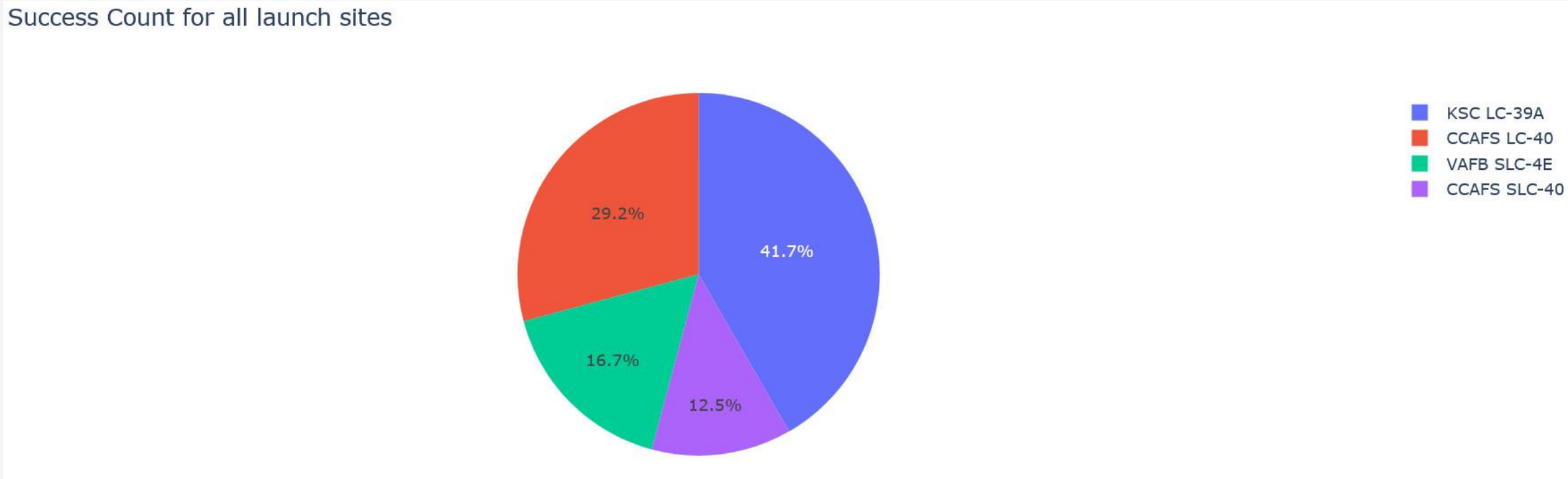


- We can see that the launch site is very close to the railway for supply transportation
- The launch site is also close to the coast but keeps a distance from cities for safety reasons.



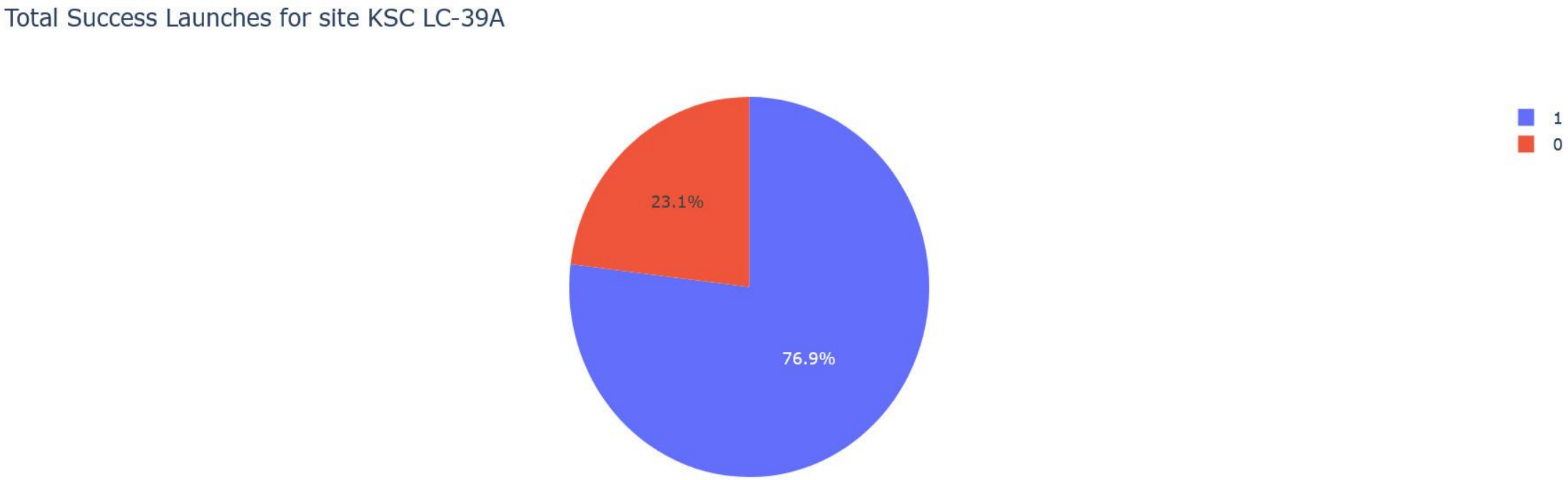
Section 4: Insights from Dashboard created by Plotly Dash

Success Percentage achieved by each launch site using a Pie Chart



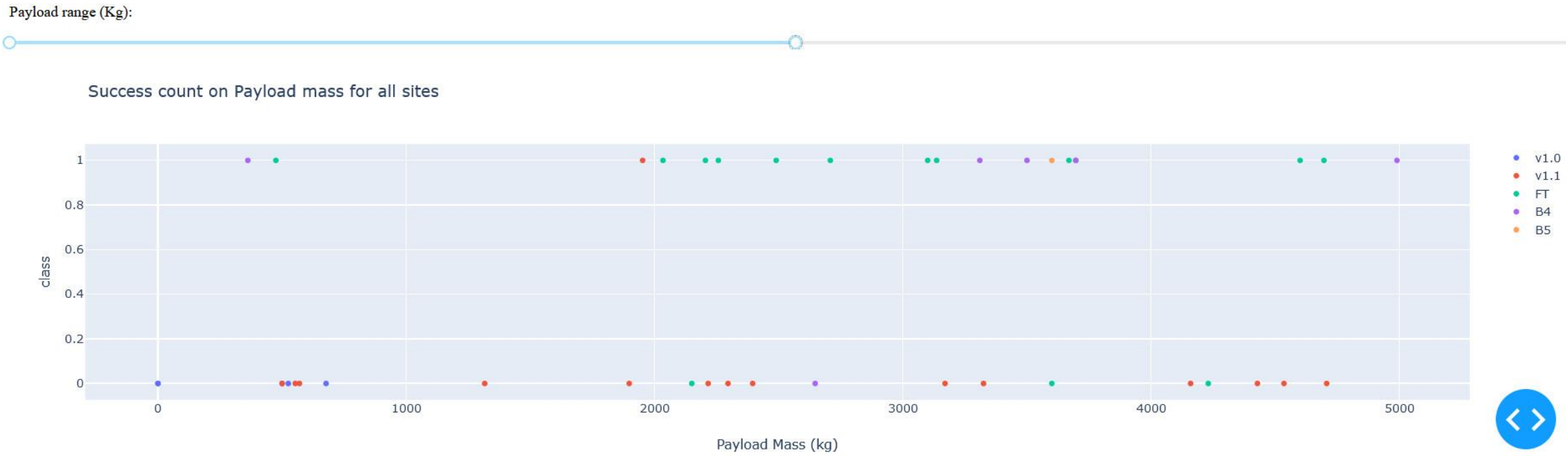
We can see that the launch site KSC LC-39A had the most success rate with 41.7%

Launch site with the Highest Launch Success Ratio



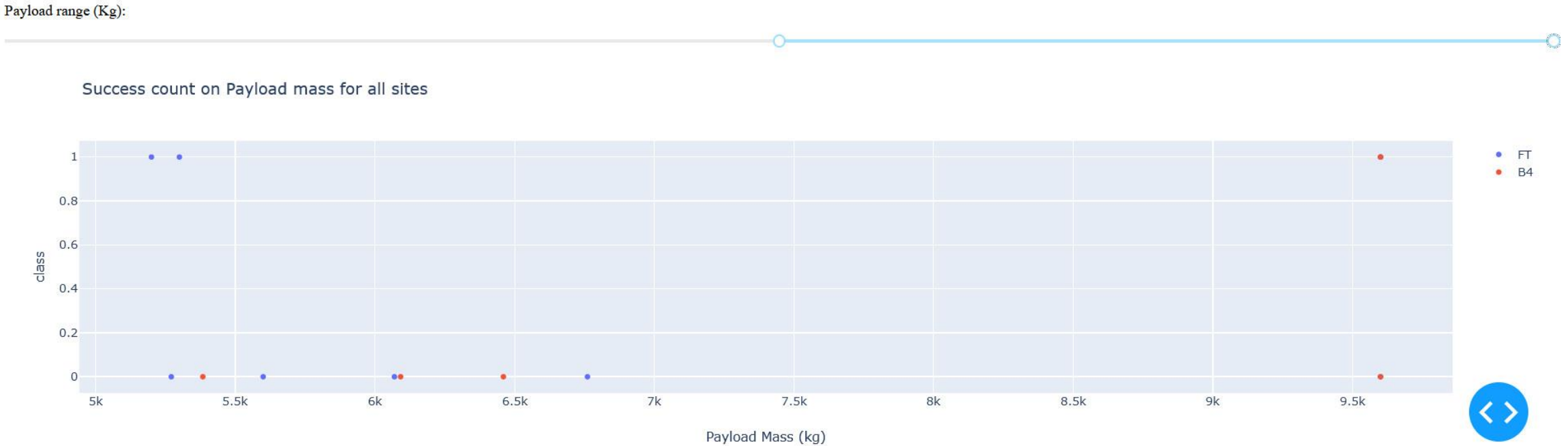
The launches for site KSC LC-39A had 10 successful launches and 3 failed launches.

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



- This scatter plot shows the launch outcomes for every site with a payload mass starting at 0 kg and going all the way up to 5000 kg.
- We see that a booster version v1.0 failed with a payload mass of 0.

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider - Continued



We see that as the payload mass increases from a range of 5000 kg to 10000 kg, There are more failed launches than successful launches.

Section 5: Predictive Analytics (Classification)

Classification Prediction Accuracy

- Using Scikit-Learn and GridSearchCV, we found that the Decision Tree Classifier performed the best with a score of 89%

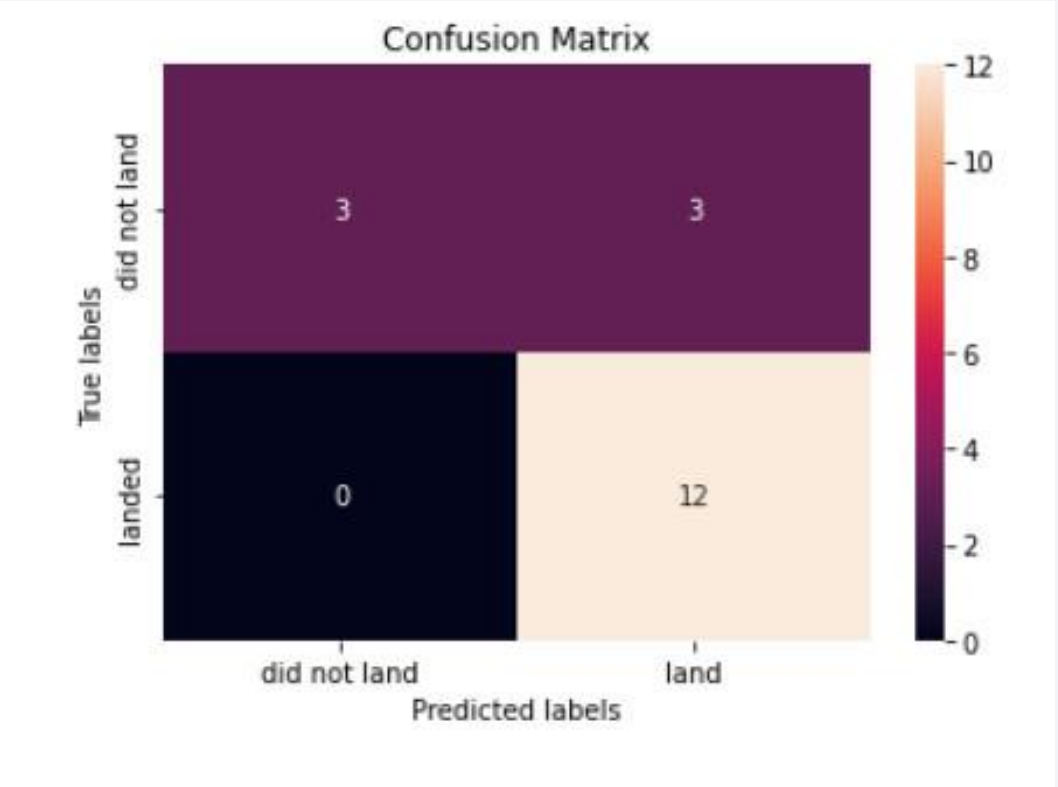
```
models = {'K-Nearest Neighbors':knn_cv.best_score_,
               'DecisionTree':tree_cv.best_score_,
               'LogisticRegression':logreg_cv.best_score_,
               'SupportVector': svm_cv.best_score_}

best_algorithm = max(models, key=models.get)
print('Best model is', best_algorithm,'with a score of', round(models[best_algorithm],2))
if best_algorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if best_algorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if best_algorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if best_algorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.89
Best params is : {'criterion': 'gini', 'max_depth': 16, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter
': 'best'}
```

Confusion Matrix – Decision Tree Classifier

- The confusion matrix for the decision tree classifier demonstrates the classifier's ability to distinguish between the different classes.

- However, there are problems like the false positives. These are unsuccessful landings that were marked as being a successful landing by the classifier itself.



Conclusion

We can conclude that:

- The larger the number of lights that occur at a launch site, the greater the success rate will be of the launch site.
- Based on the analysis done on the data we see that the success rate of launches increased from 2013 to 2020.
- Orbits ES-L1, GEO, HEO, and SSO had the highest success rates.
- Based on the Plotly dashboard, we see that the site KSC LC-39A had the highest launch success ratio than any other site.
- Using Machine Learning on the data, we determined that the Decision Tree Classifier was the best method to make predictions.

Appendix

- GitHub Repository URL:

https://github.com/Brent-W/IBM

- Instructors:

Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

- Special thanks to all the instructors of this course:

https://www.coursera.org/professional-certificates/ibm-data-science?&instructors#instructors


Thank You!