Winning Space Race with Data Science

Luis Fernando Paolucci
23/03/2022

Outline

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

Executive Summary

- Summary of methodologies
  - ✓ Data collection API
  - ✓ Data collection with web scraping
  - ✓ Data wrangling
  - ✓ Exploratory data analysis (EDA) with SQL
  - ✓ Exploratory data analysis (EDA) with visualization
  - ✓ Data visualization with Folium
  - ✓ Interactive Dashboard with Plotly Dash
  - ✓ Machine learning prediction

- Summary of all results
  - ✓ Predictive models can distinguish between the different classes. The major problem is false positives.

Introduction

SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars; other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

We want to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Wikipedia site data was extracted using scrapping framework of BeautifulSoup.
- Perform data wrangling
  - One-hot encoding was applied to transform categorical variables to numerical values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We used Scikit-learn pipelines with various classification models. Optimal hyperparameters are detected by grid searches.

Data Collection

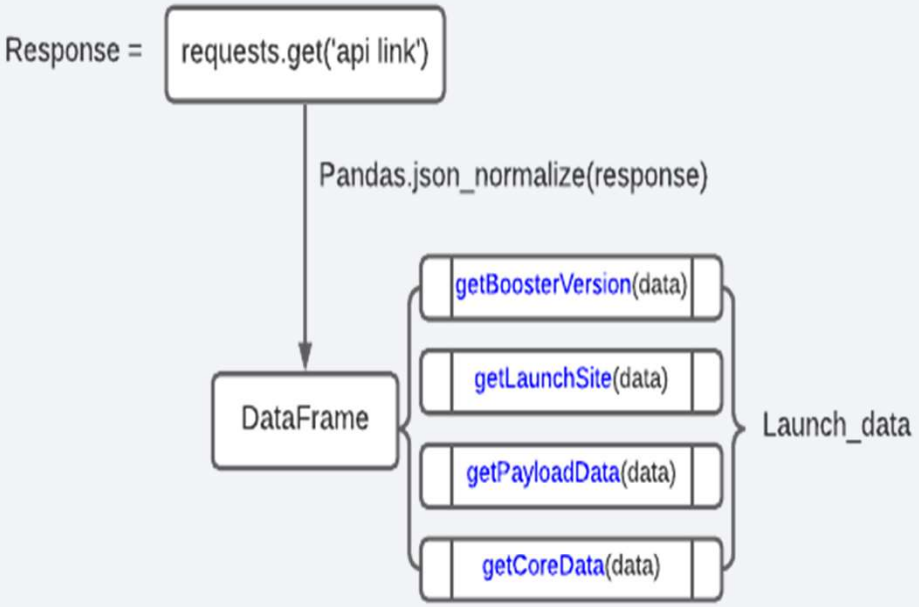
- Wikipedia site data was extracted using scrapping framework of BeautifulSoup.
- Data collection was done using get request to the SpaceX API.
- The objective was to extract the launch records as HTML table, parse the table and convert it to pandas DataFrame for further analysis.

Data Collection – SpaceX API

- We requested and parsed the SpaceX launch data using the GET request from the SpaceX API url.

- Add the GitHub URL of the completed SpaceX API calls notebook:

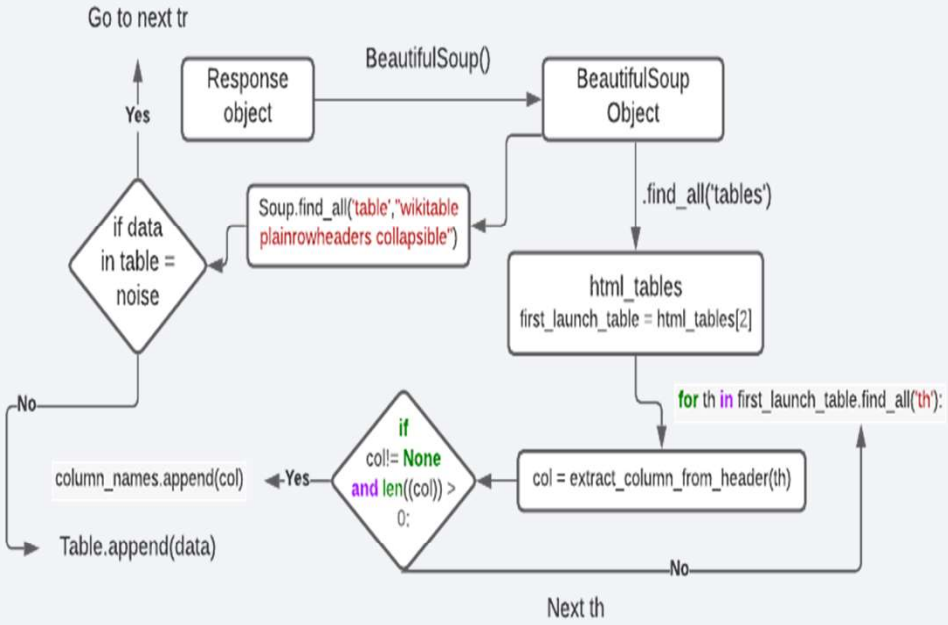
https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%201%20-%20N1%20-%20Data%20Collection%20API.ipynb



Data Collection - Scraping

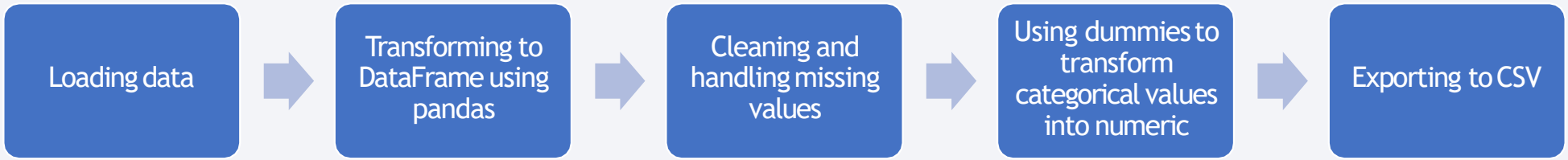
- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into pandas DataFrame.
- Add the GitHub URL of the completed web scraping notebook:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%201%20-%20N2%20-%20Data%20Collection%20with%20Web%20Scraping.ipynb



Data Wrangling

- Loaded Falcon 9 data
- Transformed to DataFrame using pandas
- Missed values in Payload are filled by its average
- Created a landing outcome label
- Exported to CSV



- Add the GitHub URL of your completed data wrangling related notebooks:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%201%20-%20N3%20-%20EDA.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Data was explored by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

- Add the GitHub URL of your completed EDA with data visualization notebook:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%202%20-%20N2%20-%20EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- Loaded the SpaceX dataset into a PostgreSQL database and applied EDA with SQL to get insight from the data. Summary of queries:

  - ✓ The names of unique launch sites in the space mission.
  - ✓ The total payload mass carried by boosters launched by NASA (CRS)
  - ✓ The average payload mass carried by booster version F9 v1.1
  - ✓ The total number of successful and failure mission outcomes
  - ✓ The failed landing outcomes in drone ship, their booster version and launch site names.

EDA with SQL

- Add the GitHub URL of your completed EDA with SQL notebook:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%202%20-%20N1%20-%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as circles, markers, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1
- Using the color-labeled marker clusters, identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities to answer questions:
  - ✓ Are launch sites near railways, highways and coastlines.
  - ✓ Do launch sites keep certain distance away from cities.
- Add the GitHub URL of your completed interactive map with Folium map:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%203%20-%20N1%20-%20Interactive%20Visual%20Analytics%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- I added those plots and interactions for a more dynamic exploratory analysis and to give the client the ability to explore what is in the data and draw insight
- Add the GitHub URL of your completed Plotly Dash lab:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%203%20-%20N2%20-%20Dashboard%20Application%20Plotly%20Dash.ipynb

Predictive Analysis (Classification)

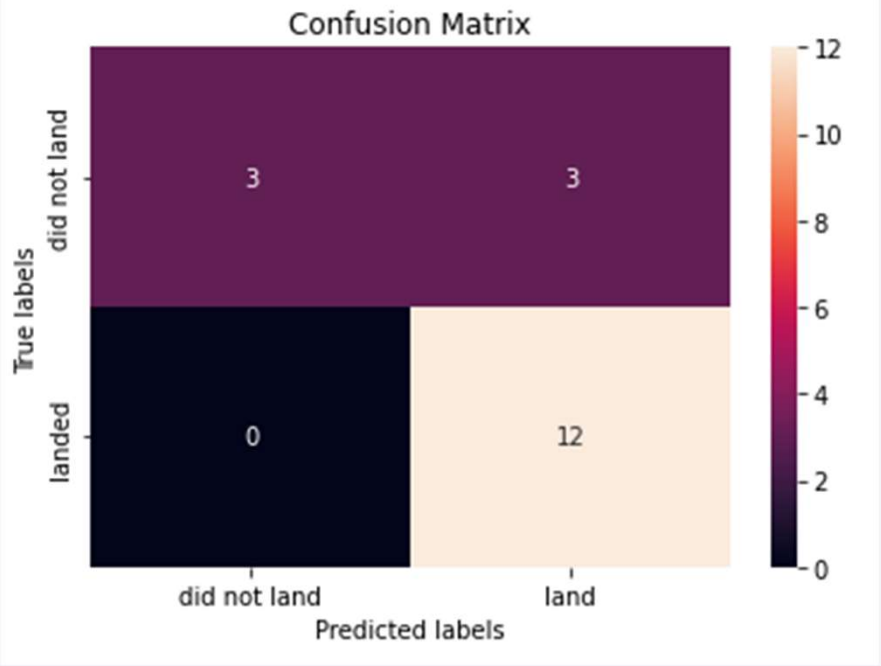
- We used Scikit-learn and made pipelines to implement various classification models. Models are validated by cross validation and tested.
- We used grid-search to find an optimal hyperparameters of each model and compared the performances.
- Add the GitHub URL of your completed predictive analysis lab:

https://github.com/luispaolucci/capstone/blob/eace8c907feecb77b4376fbcf44b3d33585439f1/Week%204%20-%20N1%20-%20Machine%20Learning%20Prediction.ipynb

Results

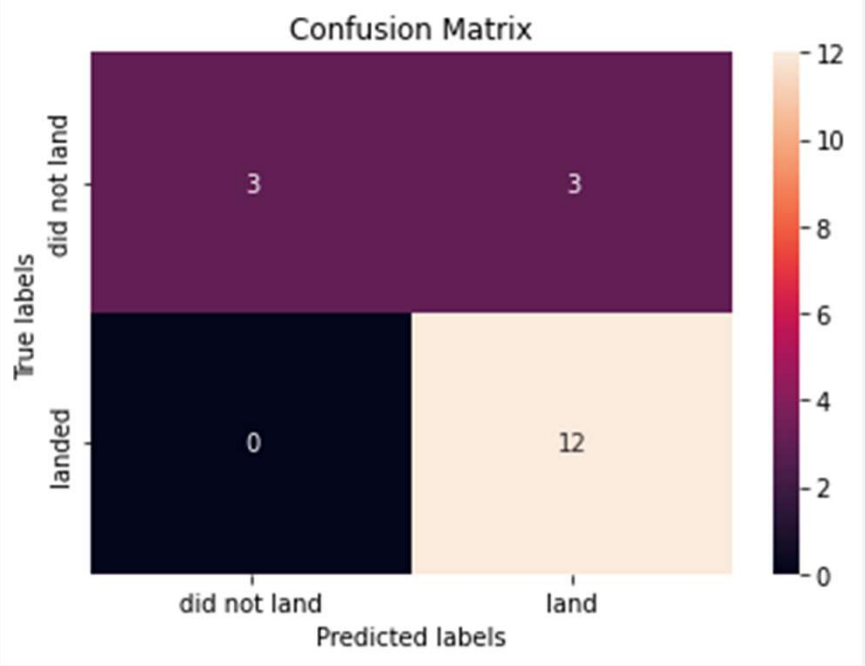
- Logistic Regression:

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



- SMV regression:

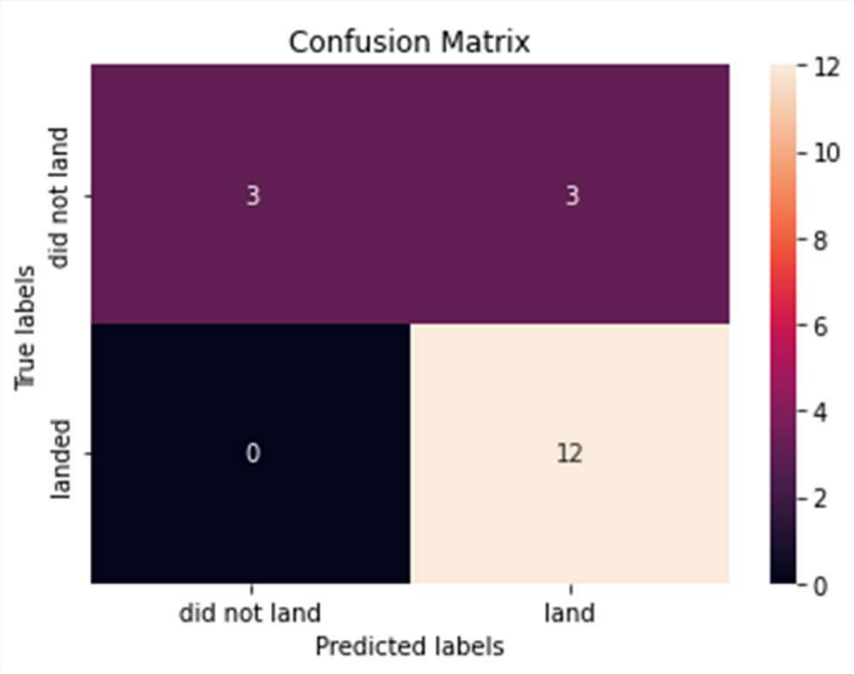
```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Results

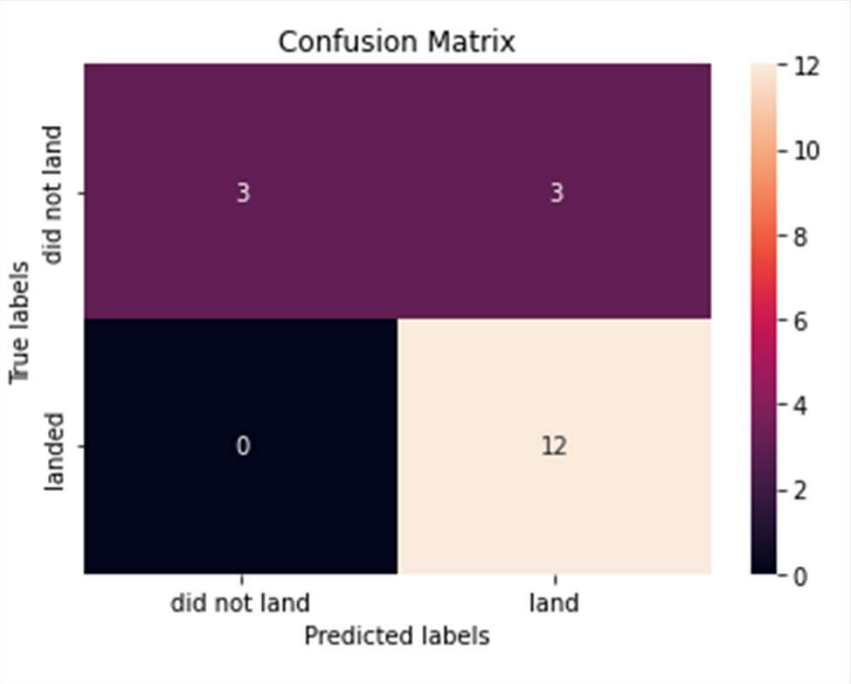
- Tree regression:

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



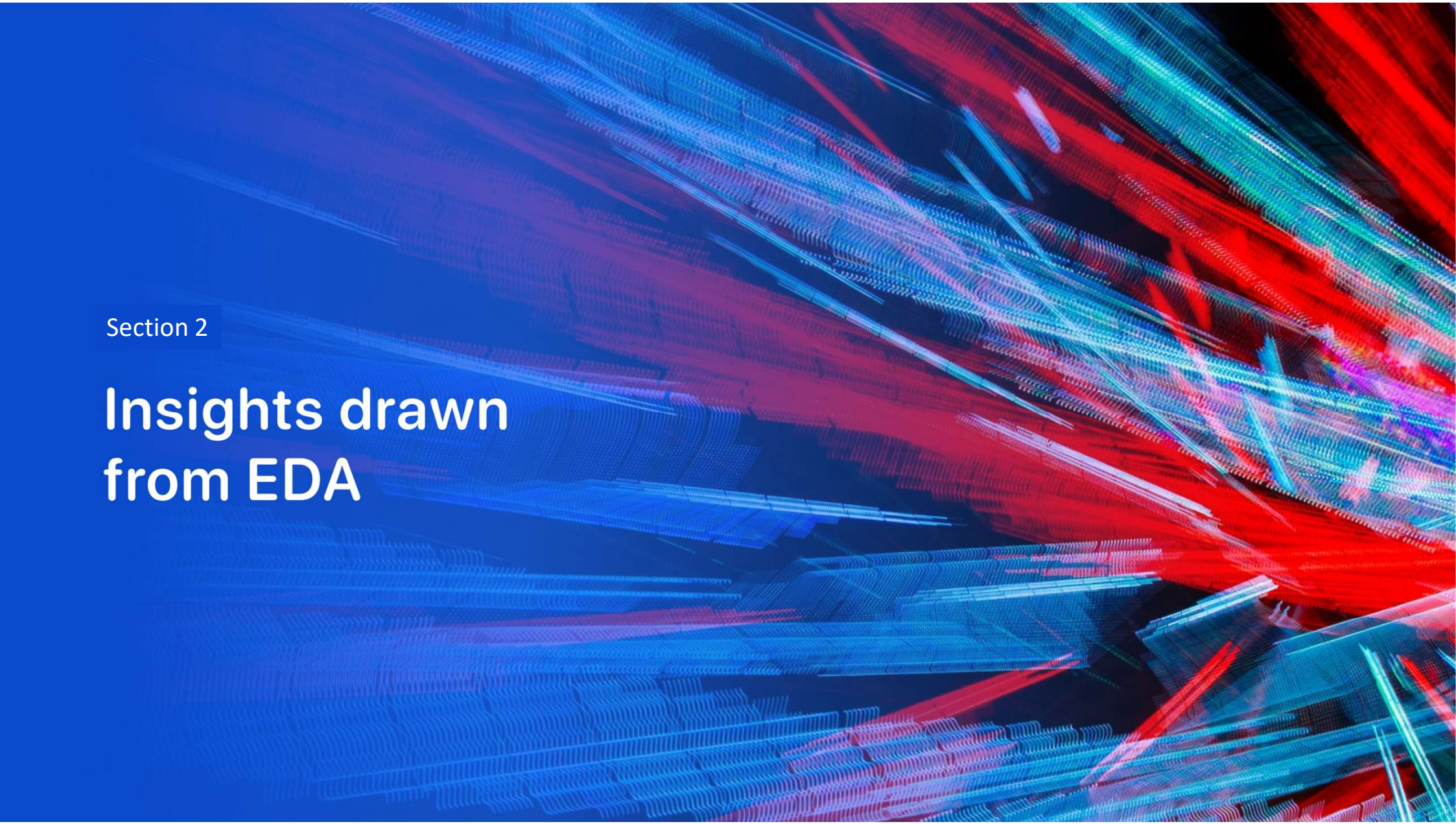
- KNN regression:

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



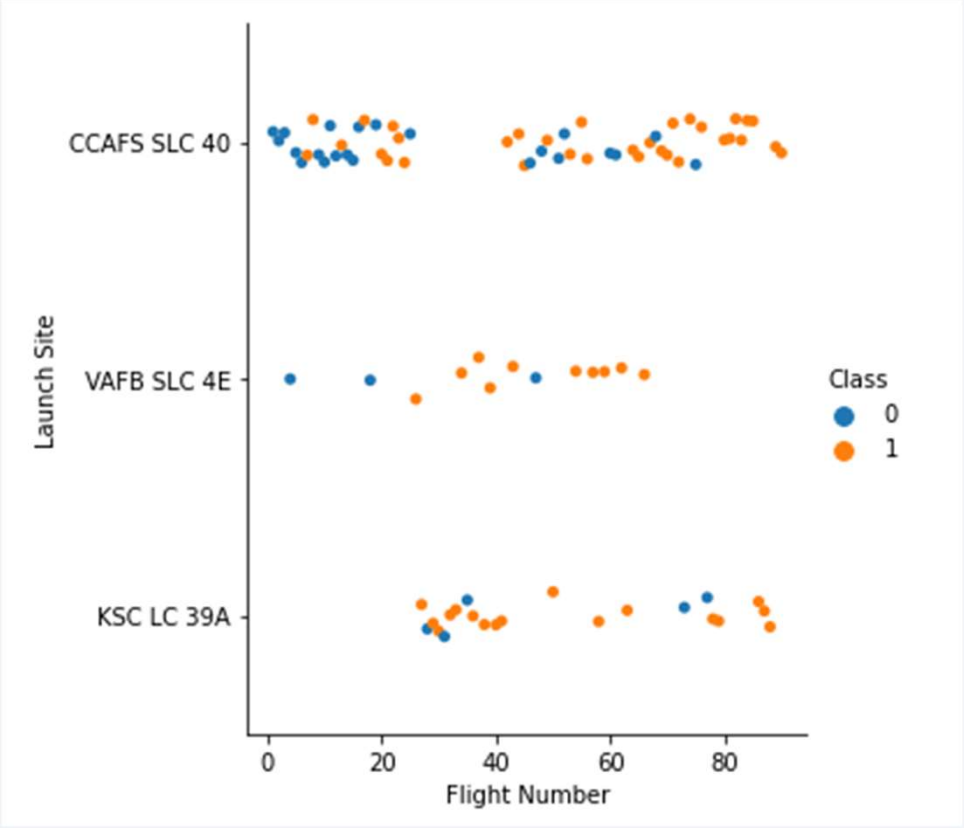
Results

- The method which performs best is "Decision Tree" with a score of 0.8892857142857145


Section 2
Insights drawn
from EDA

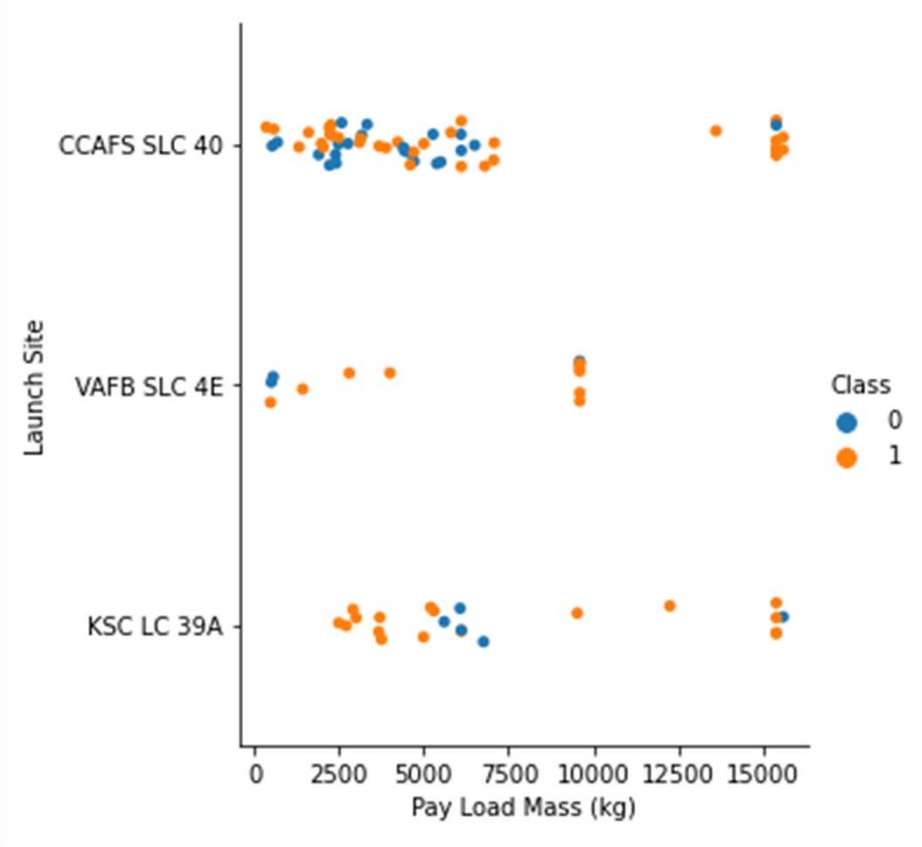
Flight Number vs. Launch Site

- From the plot, it can be seen that the larger the flight amount at a launch site, the greater the success rate at a launch site.
  - ✓Class 0: failure
  - ✓Class 1: success



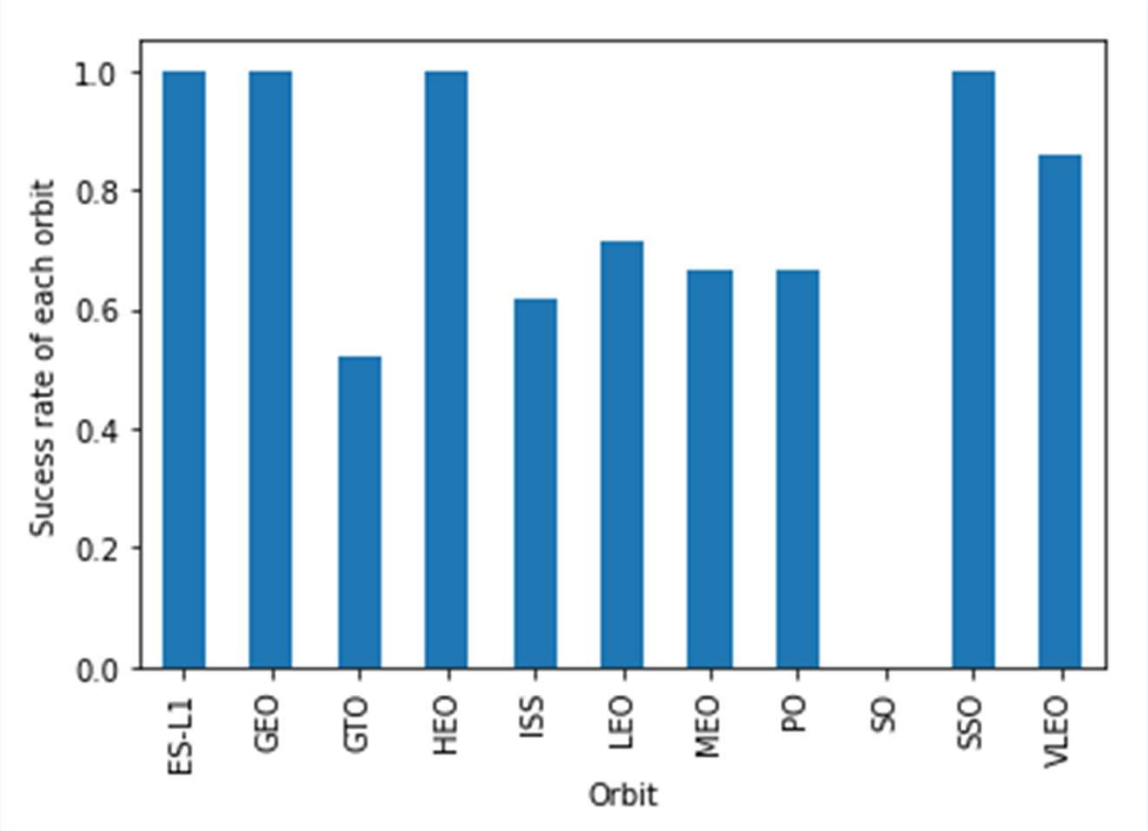
Payload vs. Launch Site

- CCAFS SLC 40 had a high chances of success, when the payload mass was lower



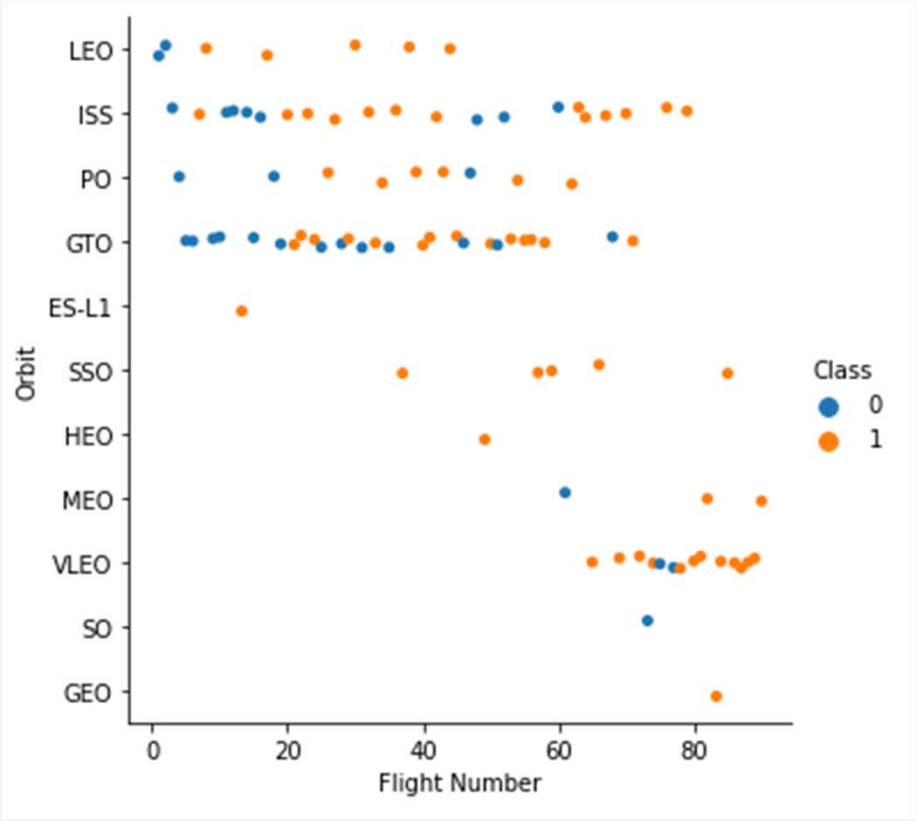
Success Rate vs. Orbit Type

- ES-P1, GEO, HEO and SSO orbits have a higher chances of success



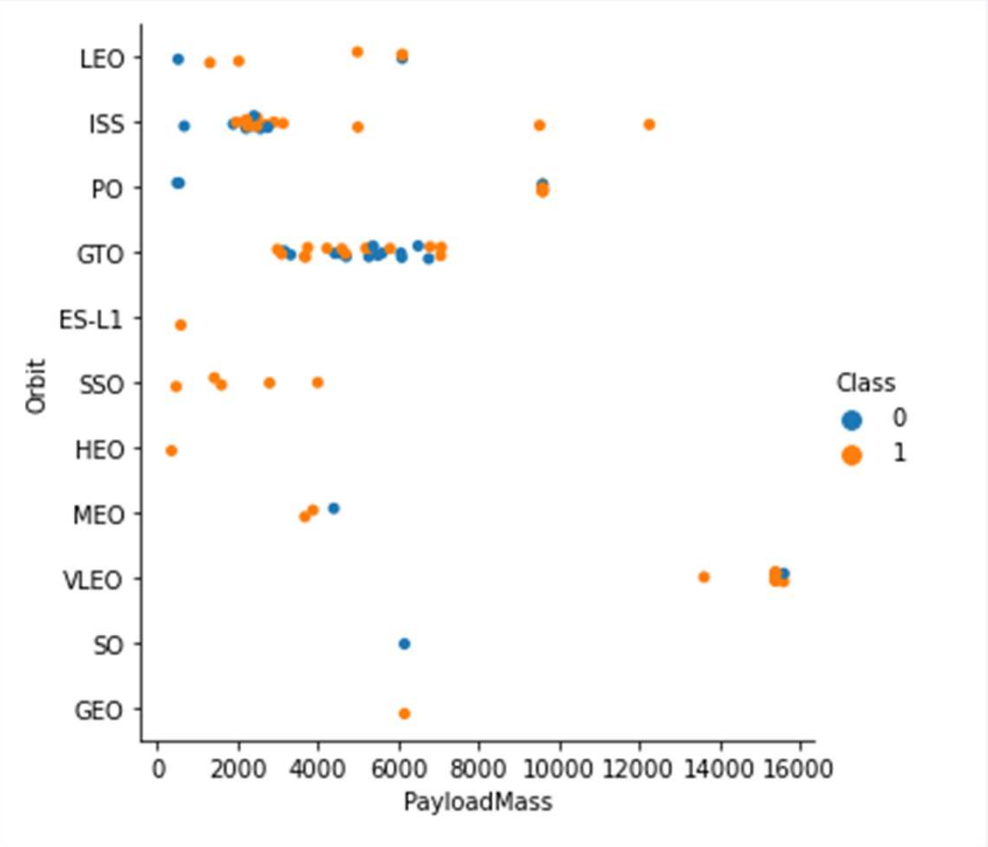
Flight Number vs. Orbit Type

- 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
  - ✓ Class 0: failure
  - ✓ Class 1: success



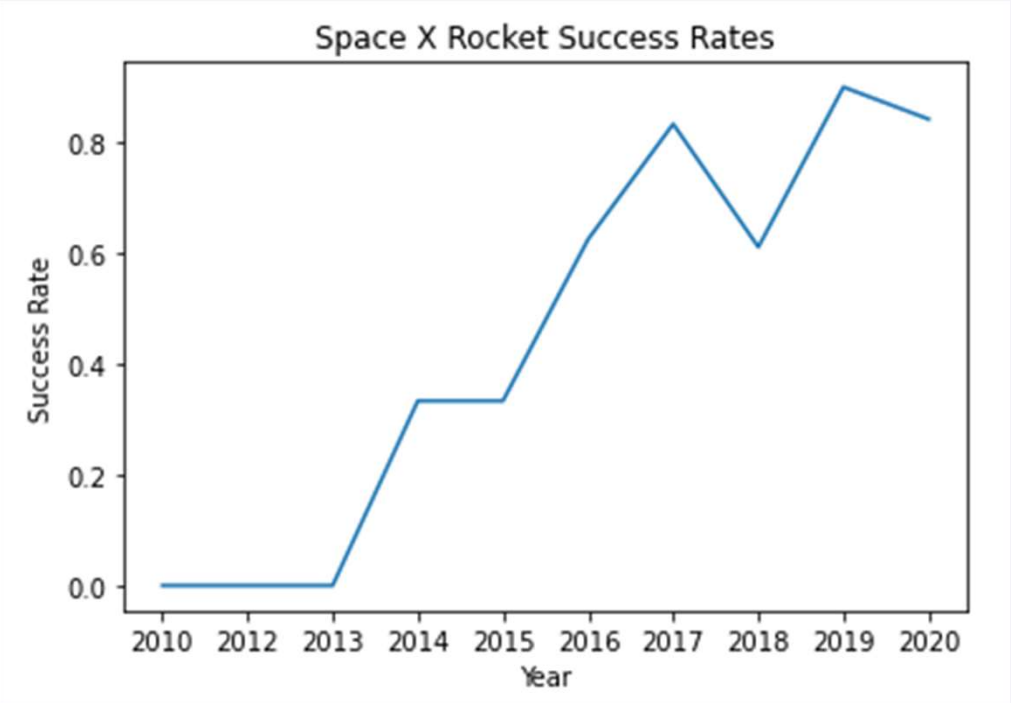
Payload vs. Orbit Type

- LEO, GTO, ES-L1, SSO, HEO, MEO, SO, GEO are used for light payload.
- VLEO is only for heavy payload.
- ISS are used for both light and heavy ones.
  - ✓ Class 0: failure
  - ✓ Class 1: success



Launch Success Yearly Trend

- Sucess rate since 2013 kept increasing till 2019. It shows how much progress had space x done in recent times



All Launch Site Names

- There are four launch sites. Their latitudes and longitudes are listed below:

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

Launch Site Names Begin with 'CCA'

- The term LIKE ‘CCA%’ was used to identify records that start with CCA.

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

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/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

Total Payload Mass

- The total payload carried by boosters from NASA is 45596 kg, including heavy payload.

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

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

total_payload_mass_kg_by_nasa_crs
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2928 kg.

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

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

average_payload_mass_kg
2928

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad is 2015-12-22 and MIN(DATE) was used to filter out the first successful landing

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

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

first_successful_ground_pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE clause was used 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

```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
  AND PAYLOAD_MASS__KG_ > 4000 \
  AND PAYLOAD_MASS__KG_ < 6000 \
ORDER BY 1;
```

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Wildcard like '%' to filter for WHERE Mission_Outcome was a success or a failure.

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS NUMBER \
FROM SPACEXTBL \
WHERE UPPER(MISSION_OUTCOME) LIKE '%SUCCESS%' \
   OR UPPER(MISSION_OUTCOME) LIKE '%FAILURE%' \
GROUP BY MISSION_OUTCOME \
ORDER BY 1;
```

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

mission_outcome	number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- A subquery was used to find the names of boosters that carried the maximum payload

```
%sql SELECT DISTINCT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) \
                            FROM SPACEXTBL) \
ORDER BY 1;
```

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE, TO_CHAR(DATE,'MON') AS month \
FROM SPACEXTBL \
WHERE TO_CHAR(DATE,'YYYY') = '2015' \
  AND LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

booster_version	launch_site	MONTH
F9 v1.1 B1012	CCAFS LC-40	JAN
F9 v1.1 B1015	CCAFS LC-40	APR

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

- Used COUNT of landing outcomes from the data and the WHERE query to filter for landing outcomes BETWEEN 2010-06-04 to 2017-03-20.
- Applied the GROUP BY query to group the landing outcomes and the ORDER BY to order the grouped landing outcome in descending order.

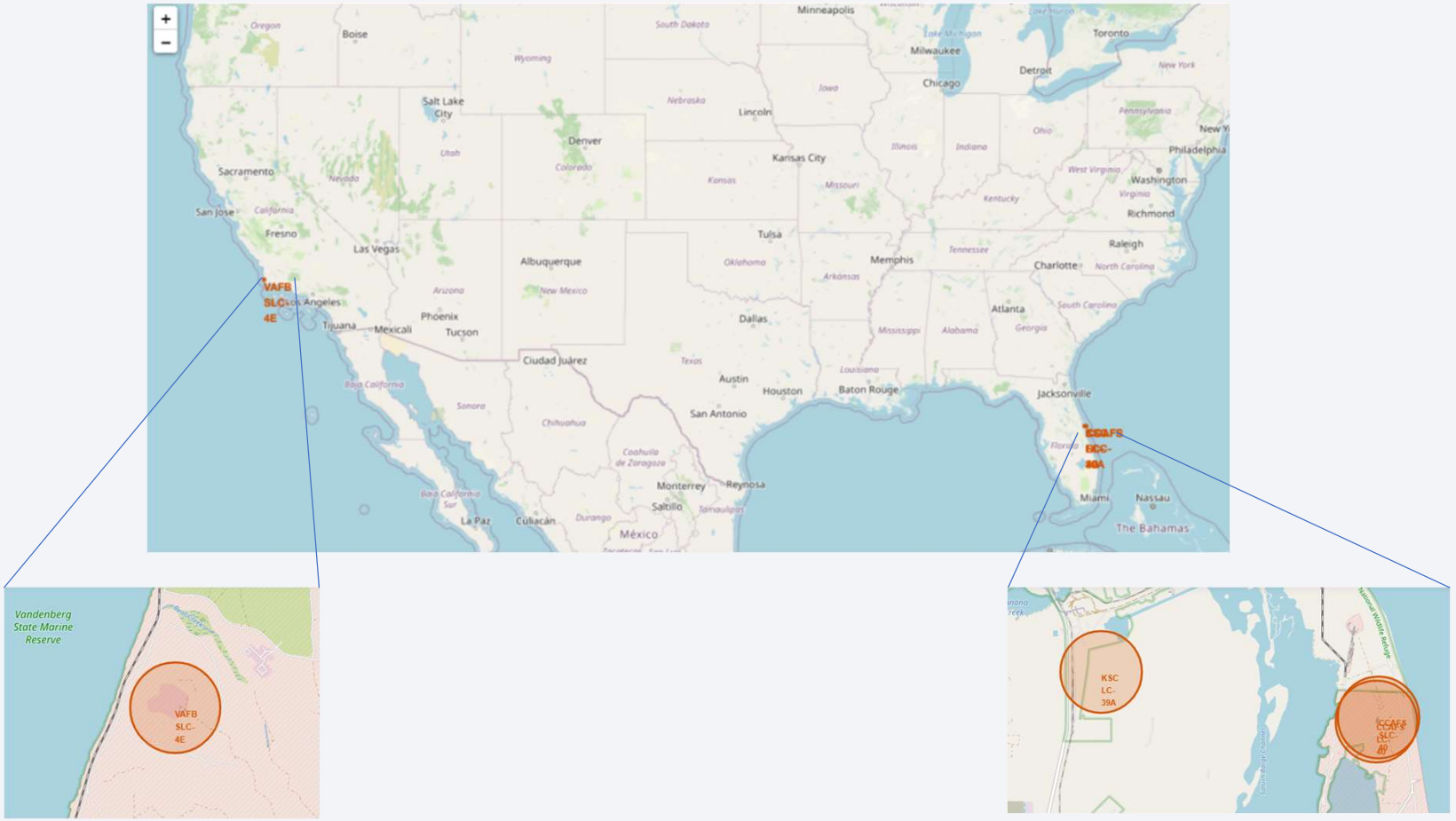
```
%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS COUNT \
FROM SPACEXTBL \
WHERE DATE BETWEEN TO_DATE('04/06/2010','DD/MM/YYYY') \
  AND TO_DATE('20/03/2017','DD/MM/YYYY') \
GROUP BY LANDING__OUTCOME \
ORDER BY 2 DESC;
```

```
 * ibm_db_sa://pyc29210:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

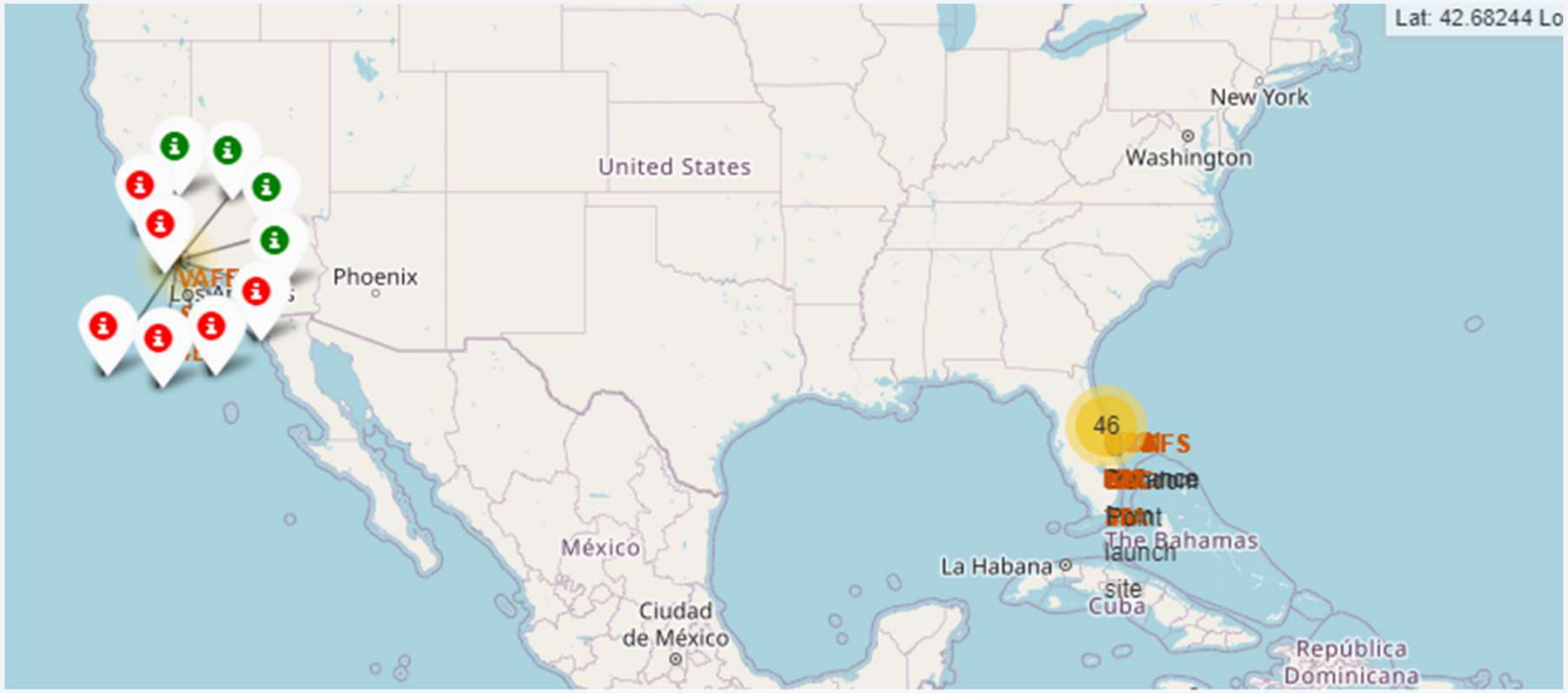
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

Launch sites of SpaceX space rockets

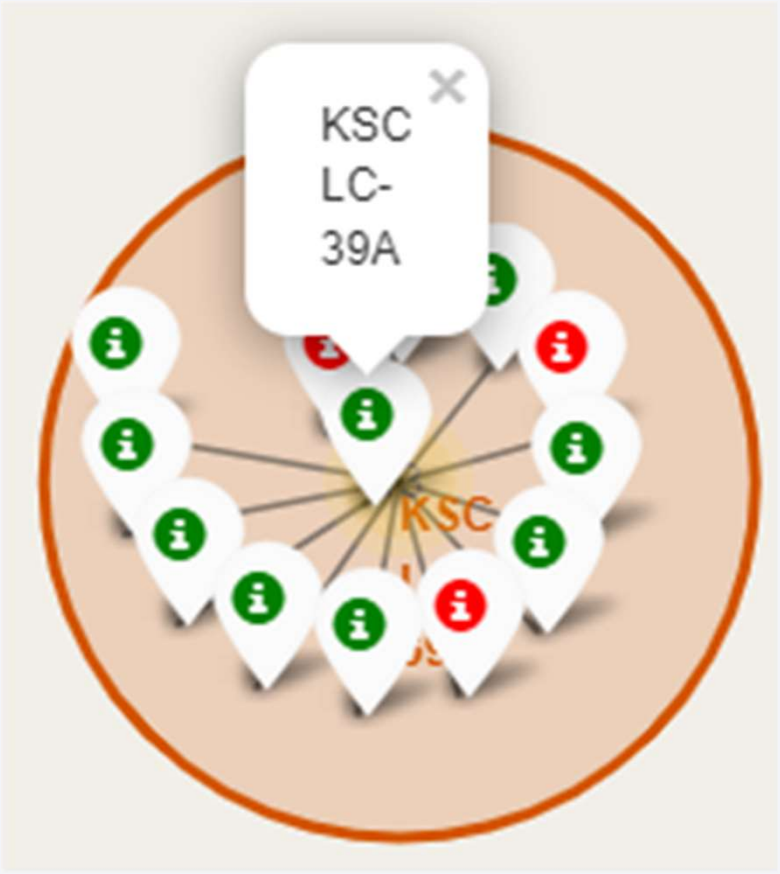


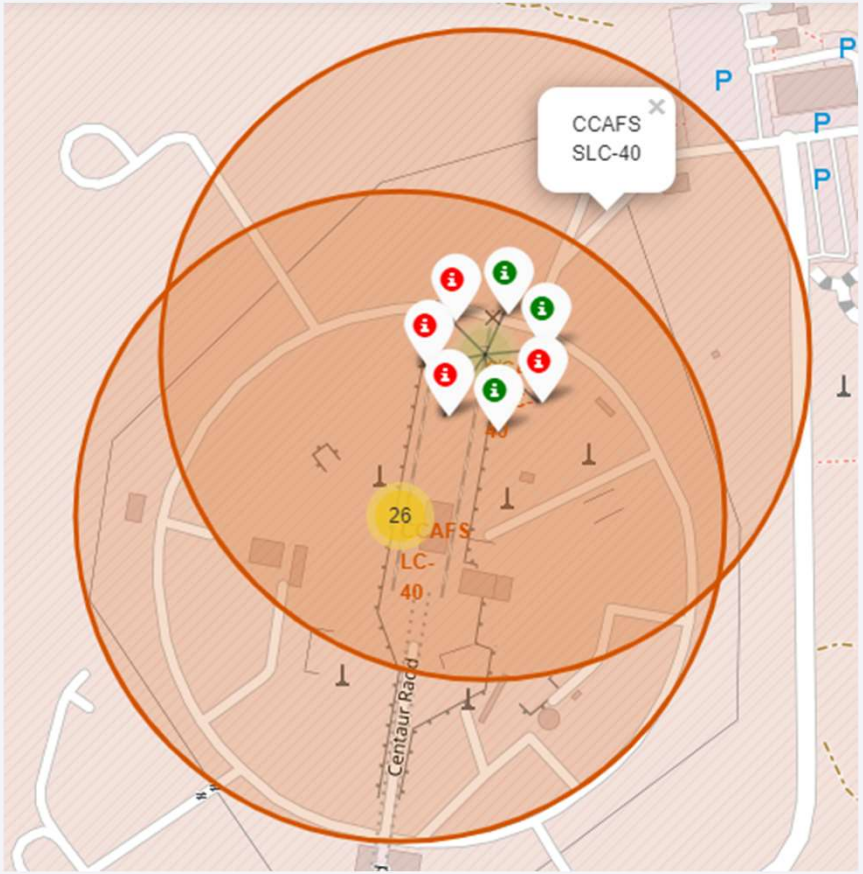
Clusters of launch sites and success rate – part 1 of 3



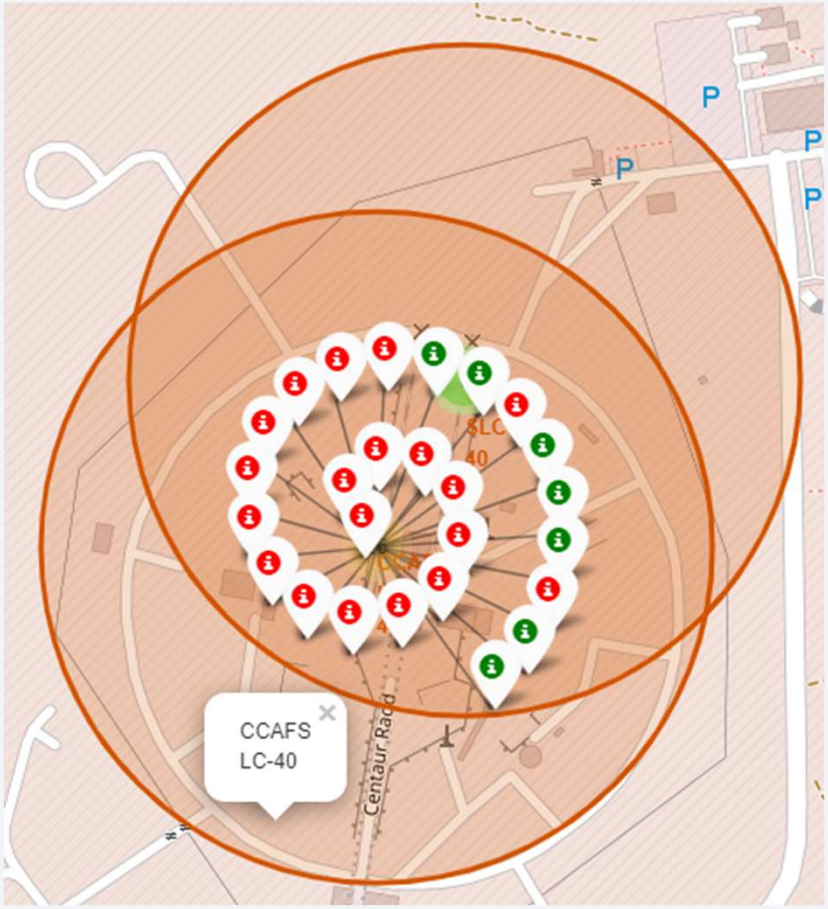
- More launches are conduced in Florida than California.
- KSC-LC 39A has the best success rate.

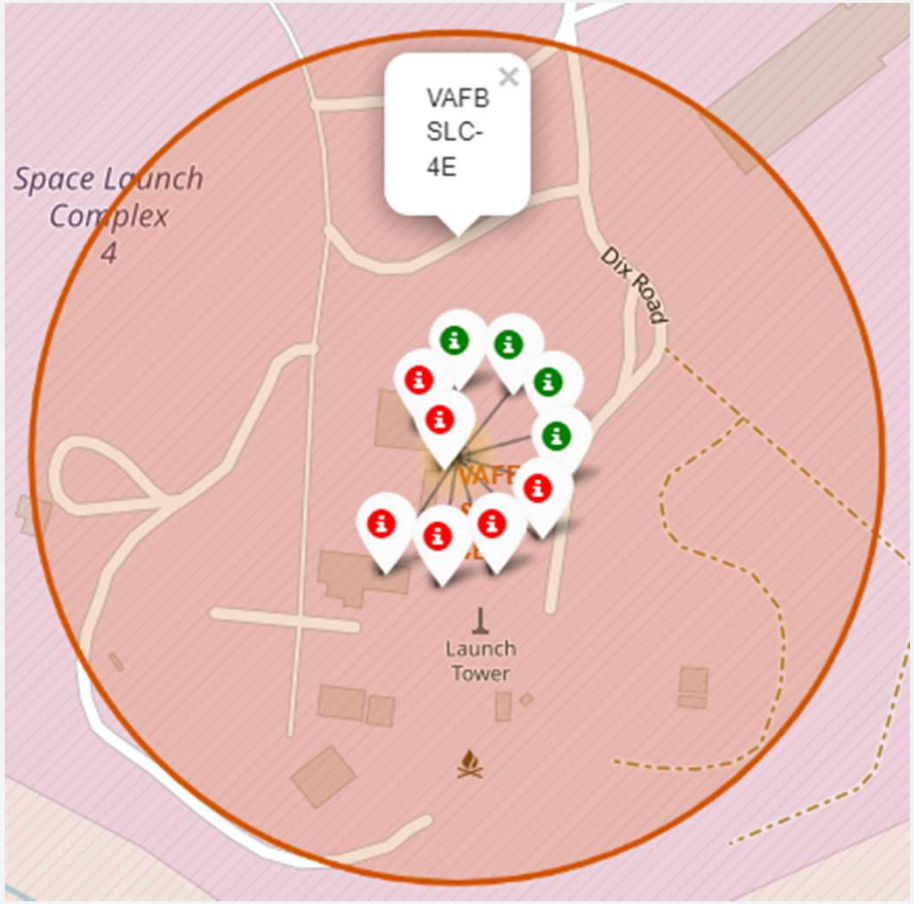
Clusters of launch sites and success rate – part 2 of 3



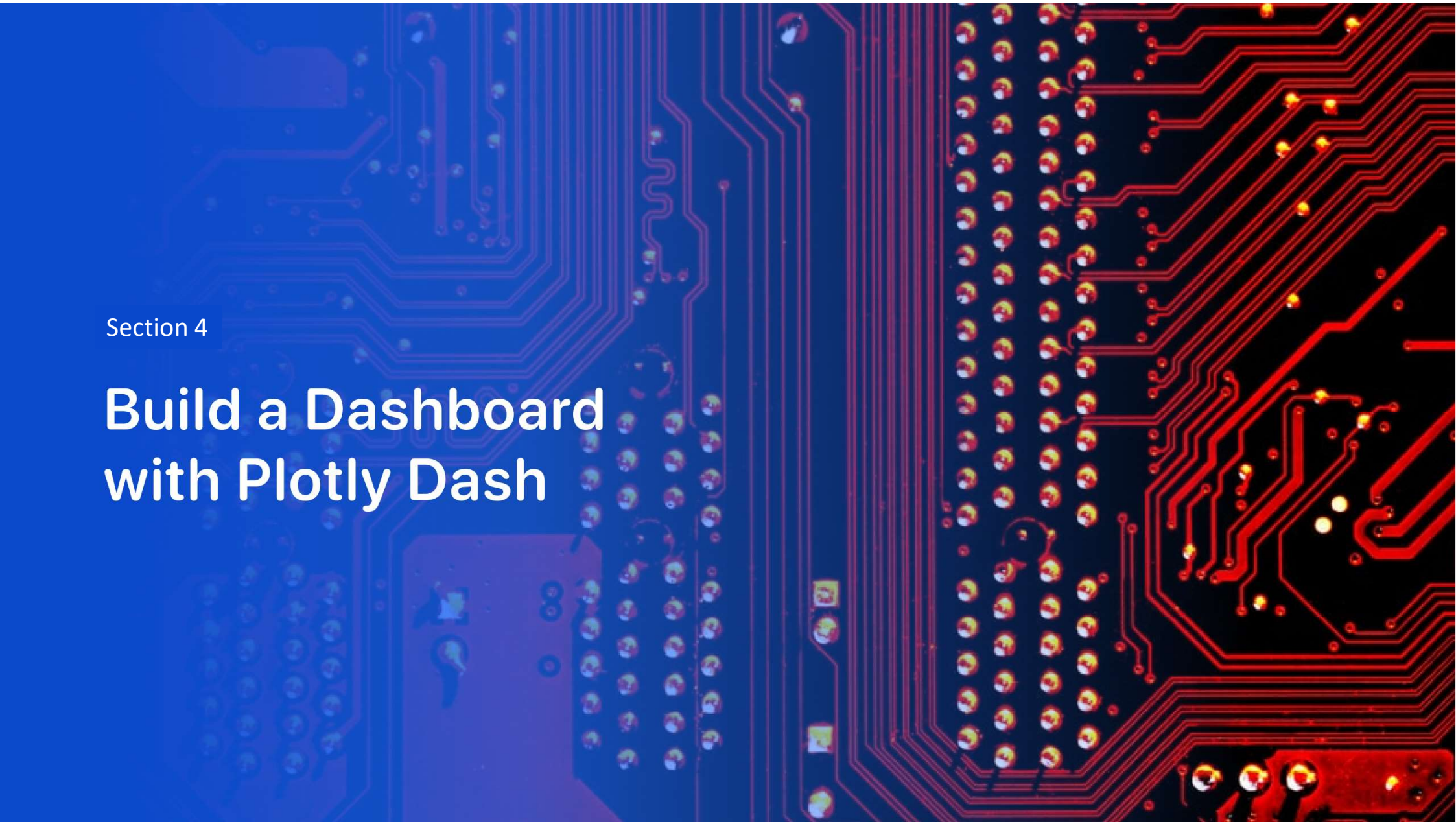


Clusters of launch sites and success rate – part 3 of 3

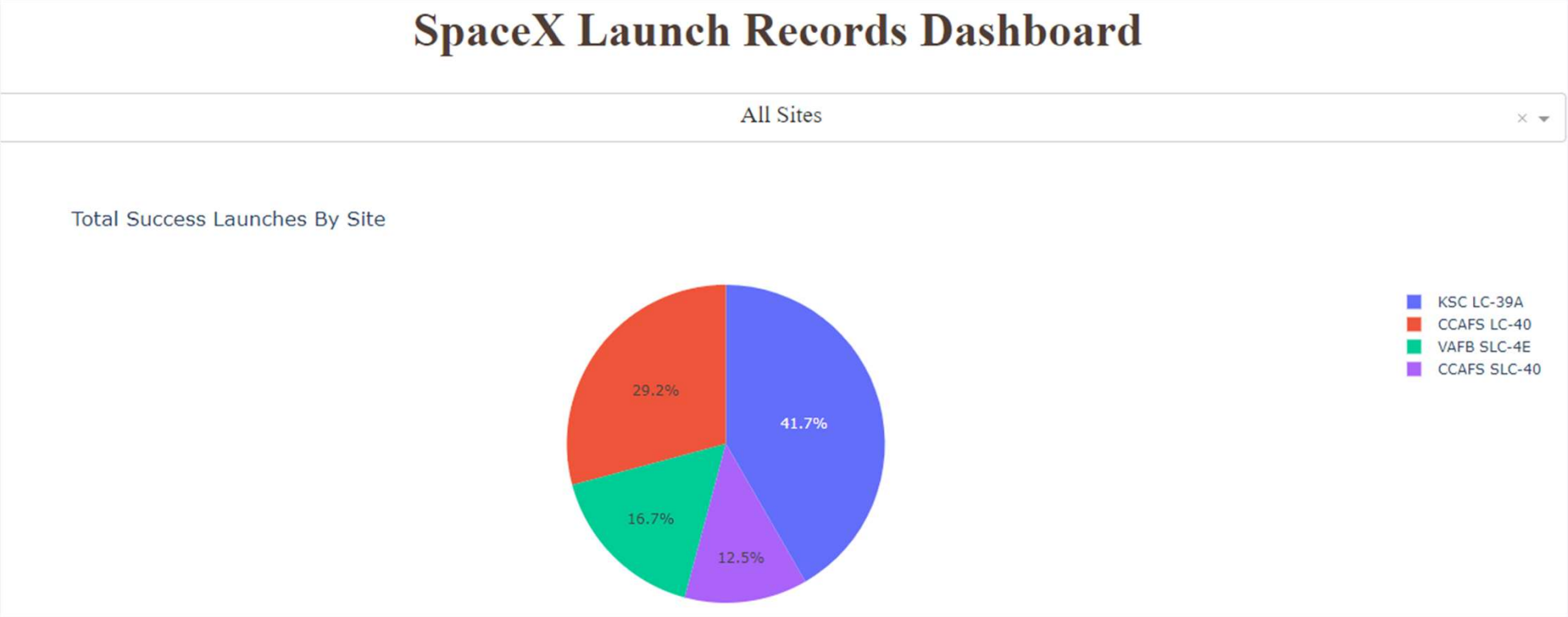




Proximities of a launch site

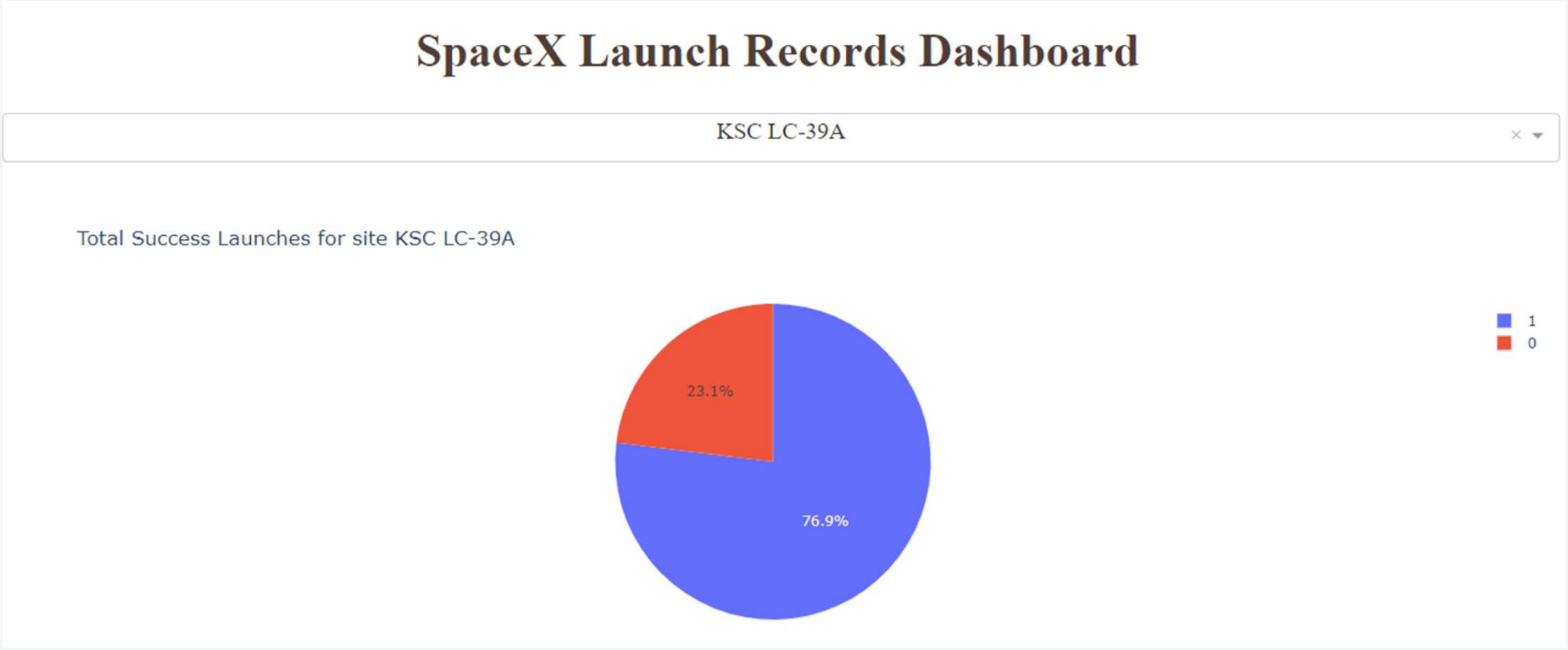

Section 4
Build a Dashboard with Plotly Dash

Total success launches by site



- KSC LC-39A has the most successful launches and CCAFS SLC-40 has the least one.

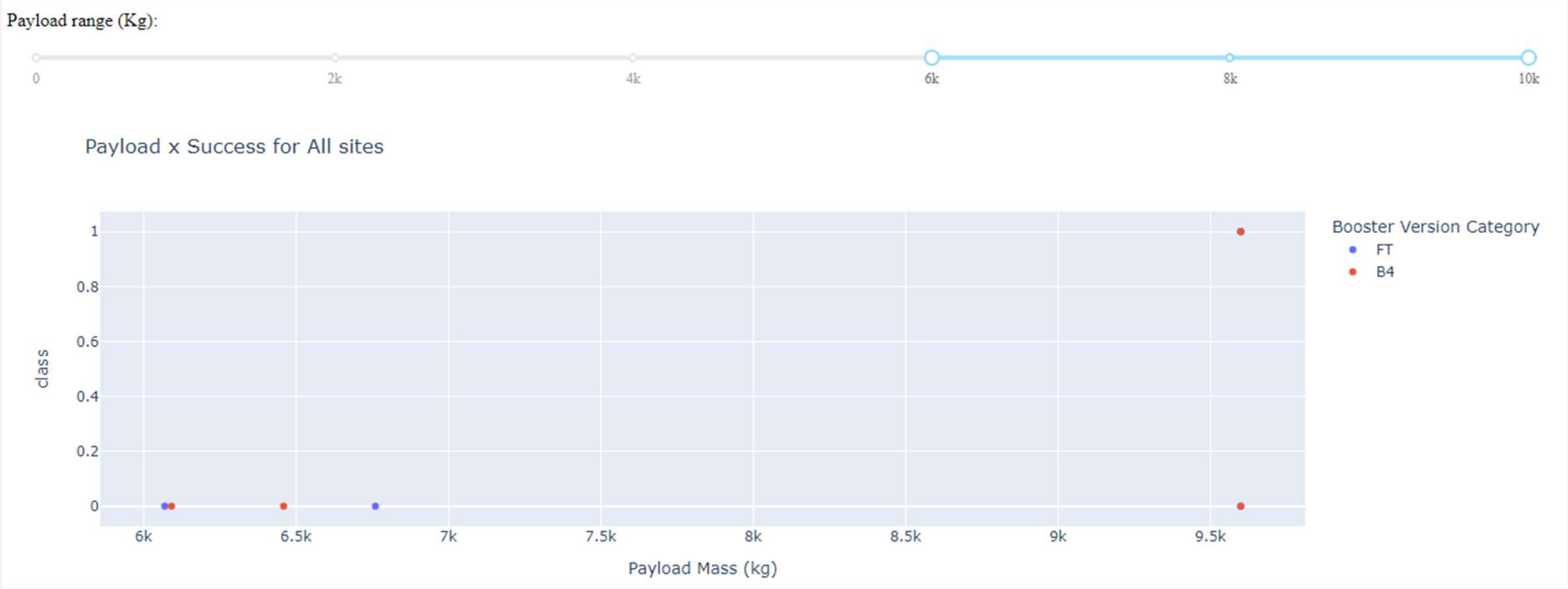
Most successful launch site (KSC LC-39A)



- Success rate in KSC LC-39A: 0=failure (23,1%), 1=success (76,9%)

Payload vs Launch Outcome (0 – 6000 kg)

Payload vs Launch Outcome (6000 – 10000 kg)



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualizing model accuracy for all built classification models, in a bar chart
- From chart it is clear Decision Tree has a higher performance accuracy as compared with other approaches

Find the method performs best:

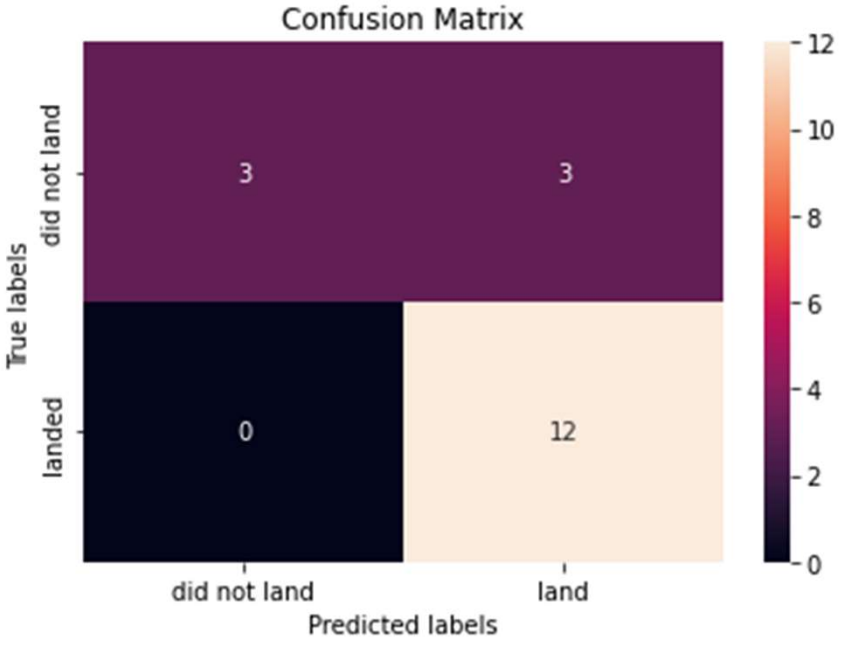
```
algorithms = {'KNN':knn_cv.best_score_,'Decision Tree':tree_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

print('The method which performs best is \"',best_algorithm,'\" with a score of',algorithms[best_algorithm])
```

```
The method which performs best is " Decision Tree " with a score of 0.8892857142857145
```

Confusion Matrix

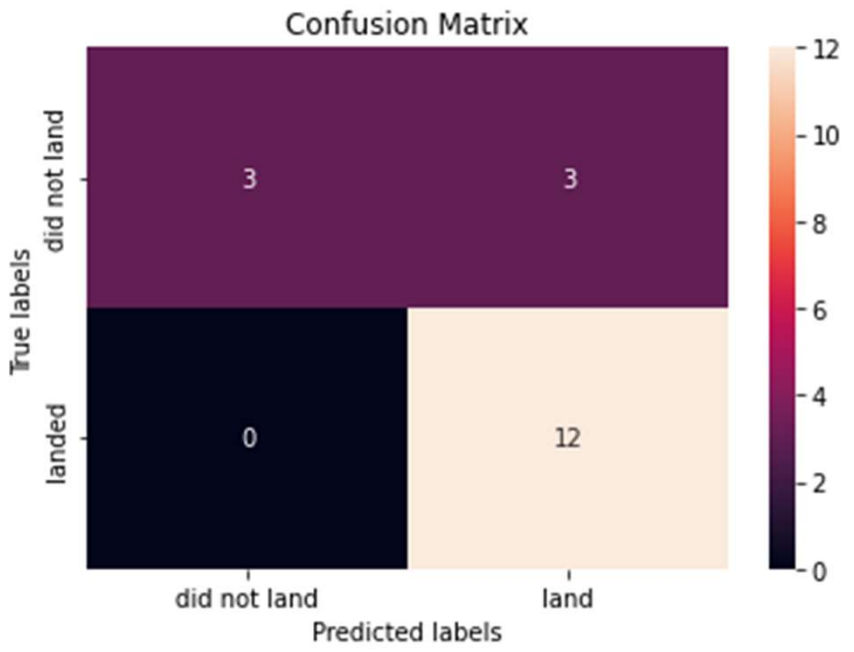
- Logistic Regression:

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



- SMV regression:

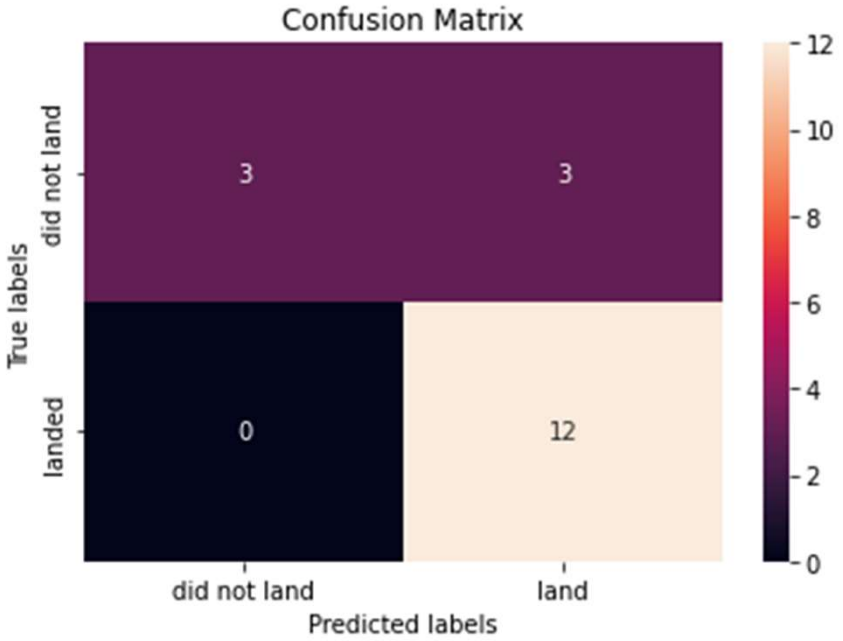
```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Confusion Matrix

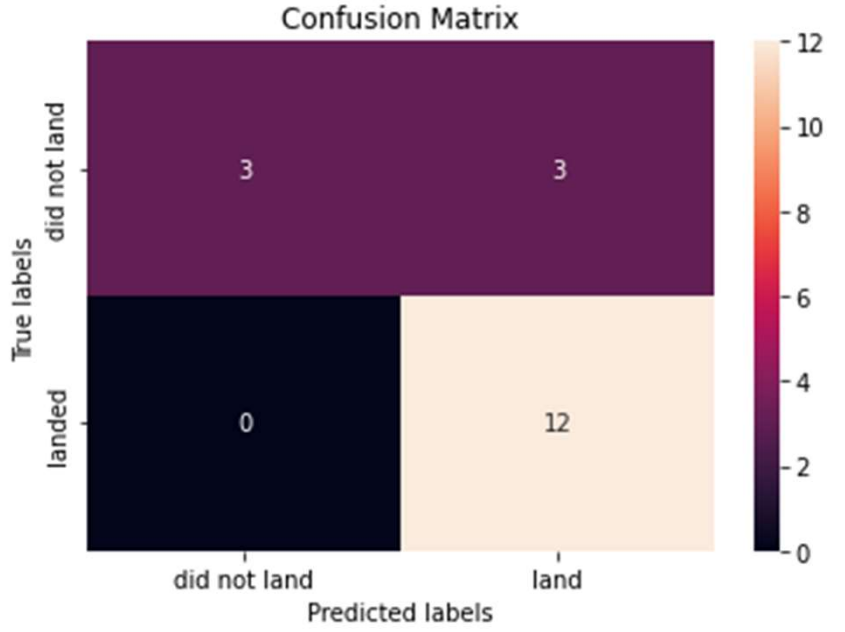
- Tree regression:

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



- KNN regression:

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Confusion Matrix

- All the models that are optimized in hyperparameters give the same confusion matrix.
- They are good at classifying the landed targets, while suffering from the false positives.

Conclusions

- The launch site KSC LC-39A has the most successful launches and the largest success rate.
- KSC LC-39A started launching later than other sites, which explains the higher success rate.
- The launches to orbits, GTO and ISS, are more difficult than other orbits.
- Predictive model needs improvement on false positives.
- Decision Tree has a higher performance accuracy as compared with other approaches

Thank you!