IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

THACH VAN MAI

July 15th, 2022

Outline

Executive Summary

❖Summary of methodologies

- Data Collection via API, SQL and Web Scraping
- Data Wrangling and Analysis
- Interactive Maps with Folium
- Predictive Analysis for each classification model

❖Summary of all results

- Data Analysis along with Interactive Visualizations
- Best model for Predictive Analysis

Introduction

- **Project background and context**

Here we predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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 a start-up company wants to bid against SpaceX for a rocket launch.

- **Problems you want to find answers**
  - With what factors, the rocket will land successfully?
  - The effect of each relationship of rocket variables on outcome.
  - Conditions which will aid SpaceX have to achieve the best results.

Identify the business question you'd like to answer.

Section 1

Methodology

Methodology

Executive Summary

1.1/ Data collection methodology:

- ❑Via SpaceX Rest API
- ❑Web Scrapping from Wikipedia

1.2/ Perform data wrangling

- ❑One hot encoding data fields for machine learning and dropping irrelevant columns (transforming data for Machine Learning)

2/ Perform exploratory data analysis (EDA) using visualization and EDA with SQL

- ❑Scatter and bar charts to show patterns between data

3-4/ Perform interactive visual analytics using Folium and dashboard with Plotly Dash

5/Perform predictive analysis using classification models

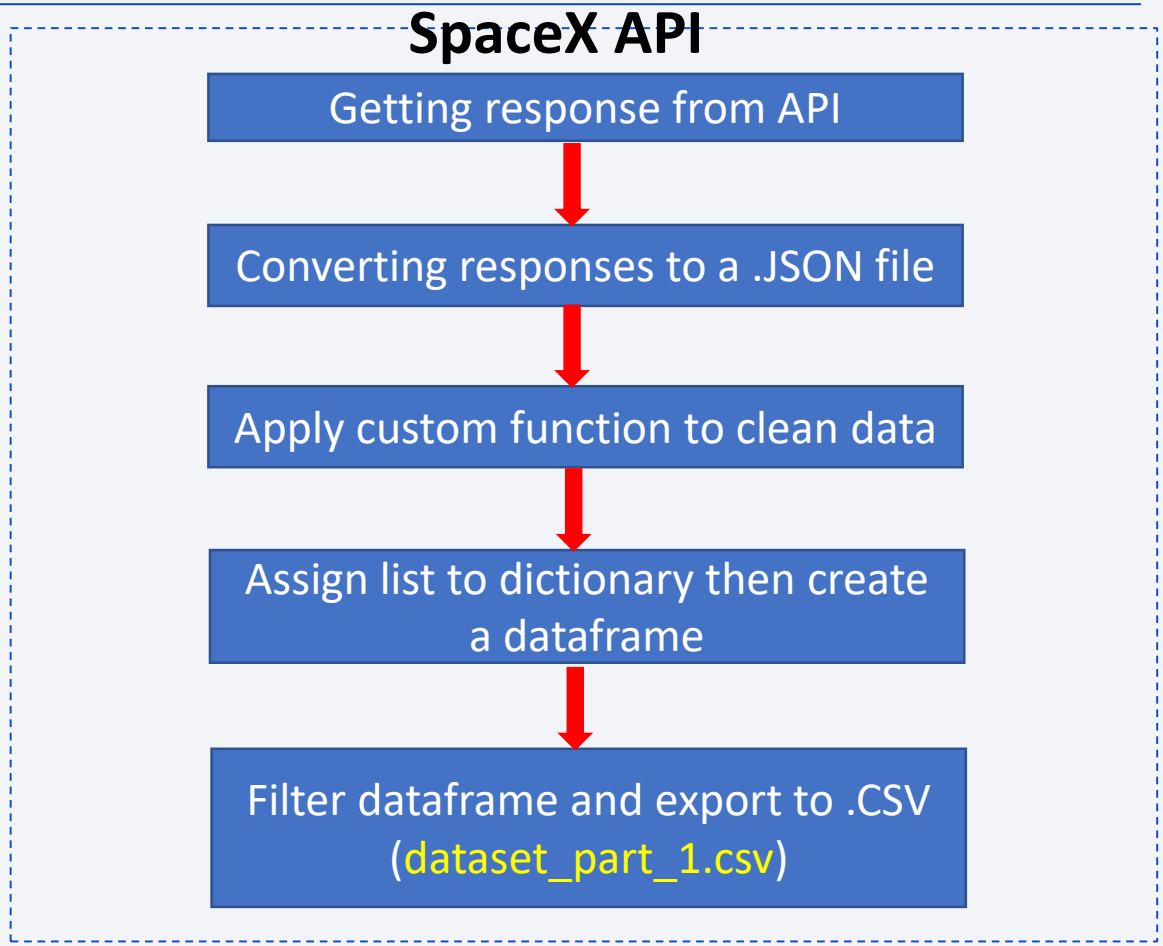
- ❑build, tune, evaluate classification models

Data Collection

- **The data was collected using various methods**
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with ***BeautifulSoup***.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

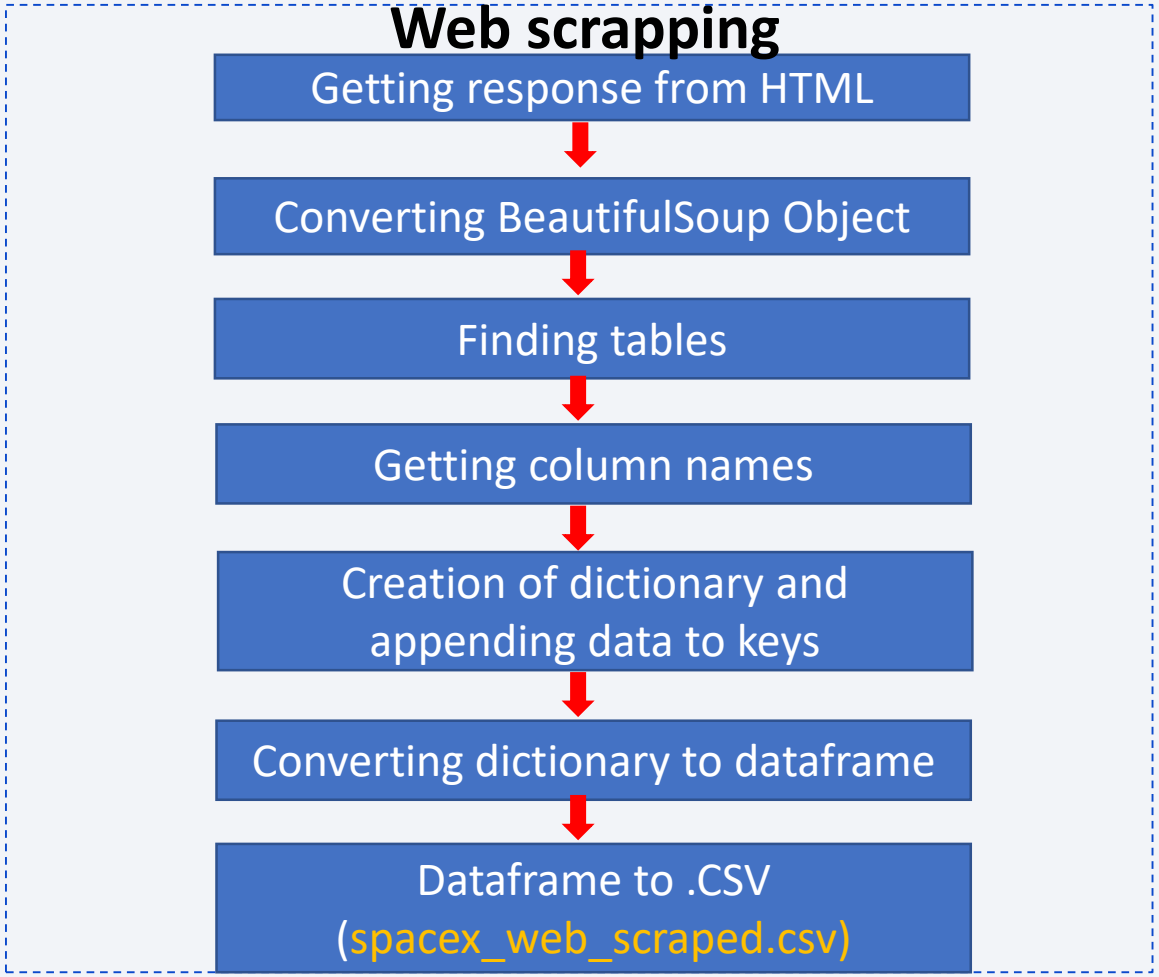
Data Collection - SpaceX API

- 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/vanthachvn80/data_science/blob/87126e844469afd13b2cb22f4374fe1cb1df90b1/Ibm_DS_capstone/course10_w1_Data%20Collection_SpaceX%20API.ipynb



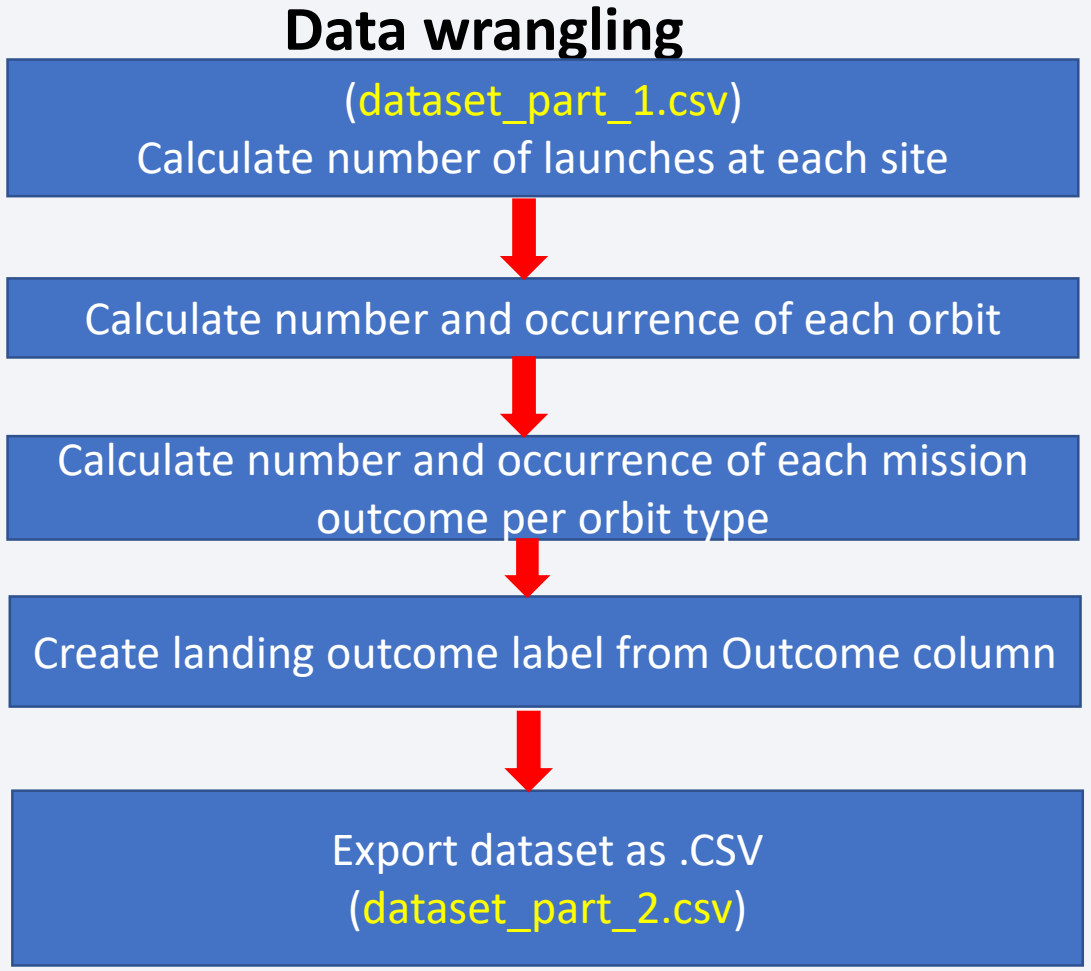
Data Collection - Scraping

- We applied web scrapping to web scrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/main/Ibm_DS_capstone/course10_w1_Data%20Collection_Scraping.ipynb



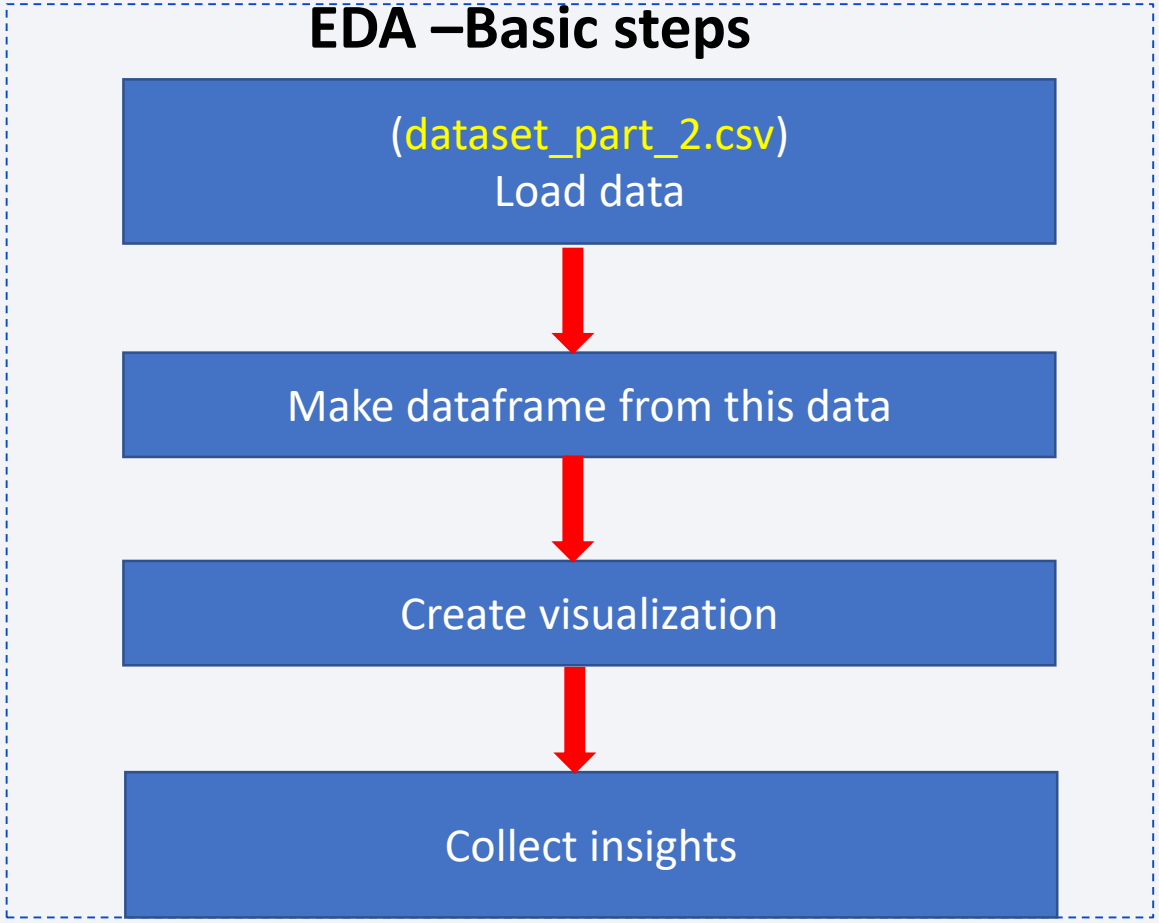
Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/main/Ibm_DS_capstone/course10_w1_Data%20Wrangling.ipynb



EDA with Data Visualization

- We explored the data 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
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/87126e844469afd13b2cb22f4374fe1cb1df90b1/Ibm_DS_capstone/course10_w2_EDA%20with%20Data%20Visualization.ipynb



EDA with SQL

- We loaded the dataset (Spacex.csv) into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - 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 failed mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/87126e844469afd13b2cb22f4374fe1cb1df90b1/Ibm_DS_capstone/course10_w2_EDA%20with%C2%A0SQL.ipynb

Build an Interactive Map with Folium

- We load the dataset (spacex_launch_geo.csv), then we marked all launch sites, and added map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to classes 0 and 1. i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have a relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some questions for instance:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep a certain distance away from cities?
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/87126e844469afd13b2cb22f4374fe1cb1df90b1/Ibm_DS_capstone/course10_w3a_Interactive%20Map%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We load the dataset (spacex_launch_dash.csv)
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by certain sites
- We plotted a scatter graph showing the relationship between Outcome and Payload Mass (Kg) for the different booster version
- The link to the notebook is: https://github.com/vanthachvn80/data_science/blob/87126e844469afd13b2cb22f4374fe1cb1df90b1/Ibm_DS_capstone/course10_w3b_Build%20a%20Dashboard%20with%20Plotly%20Dash.ipynb

Predictive Analysis (Classification)

The link is: GitHub URL

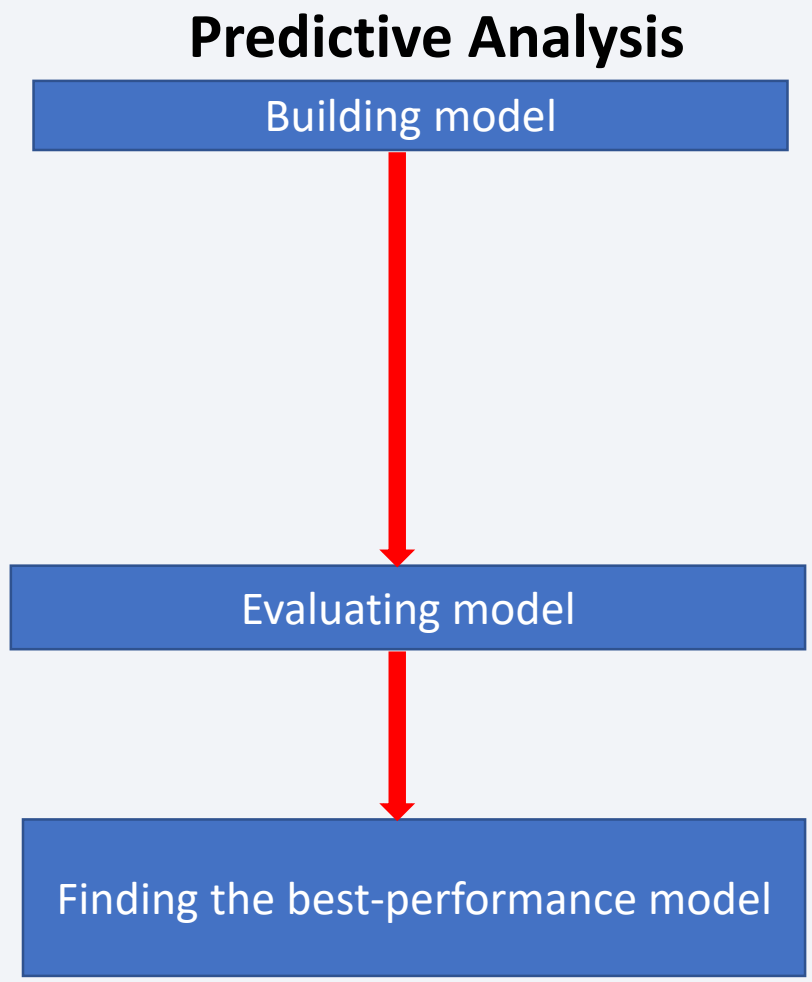
Building model:

- Load our feature-engineered data (dataset_part_2.csv) into dataframe.
- Transform it into NumPy arrays
- Standardize and transform data
- Split data into training and test data sets
- Check how many test samples have been created
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our model

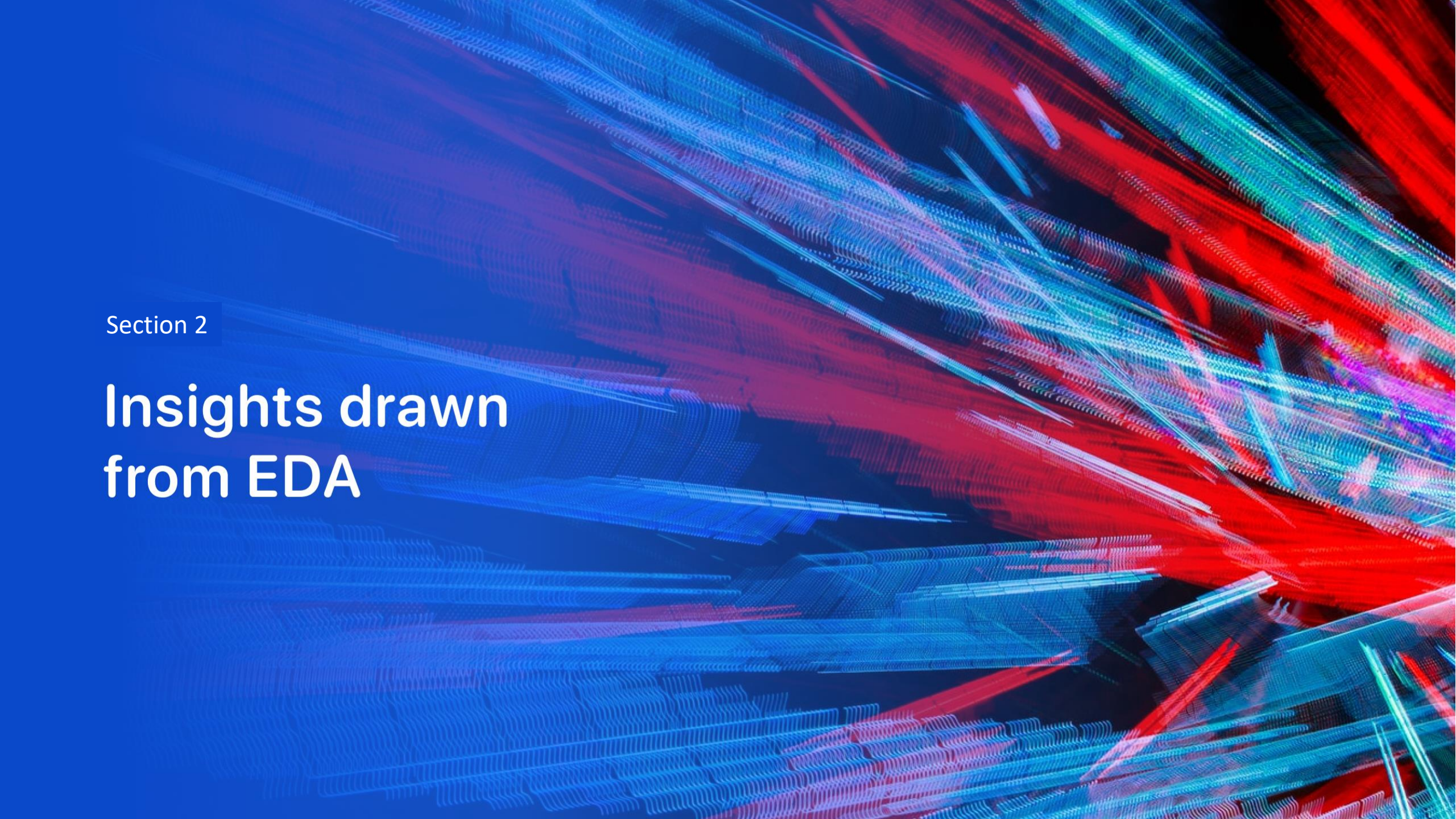
Evaluating model:

- Check accuracy for each model
- Get the best hyperparameters for each type of algorithms
- Plot Confusion Matrix

Finding the best-performance model

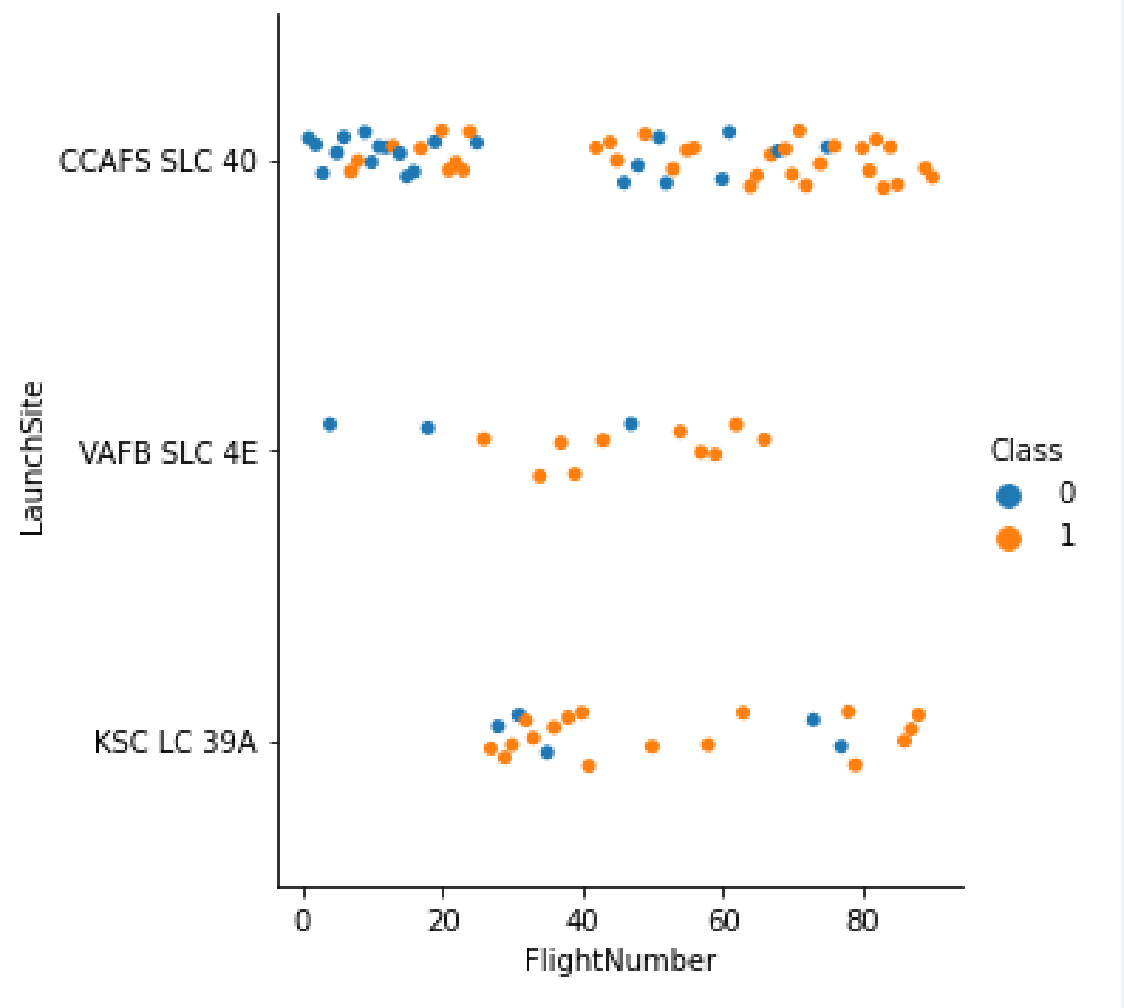


Results


Section 2
Insights drawn from EDA

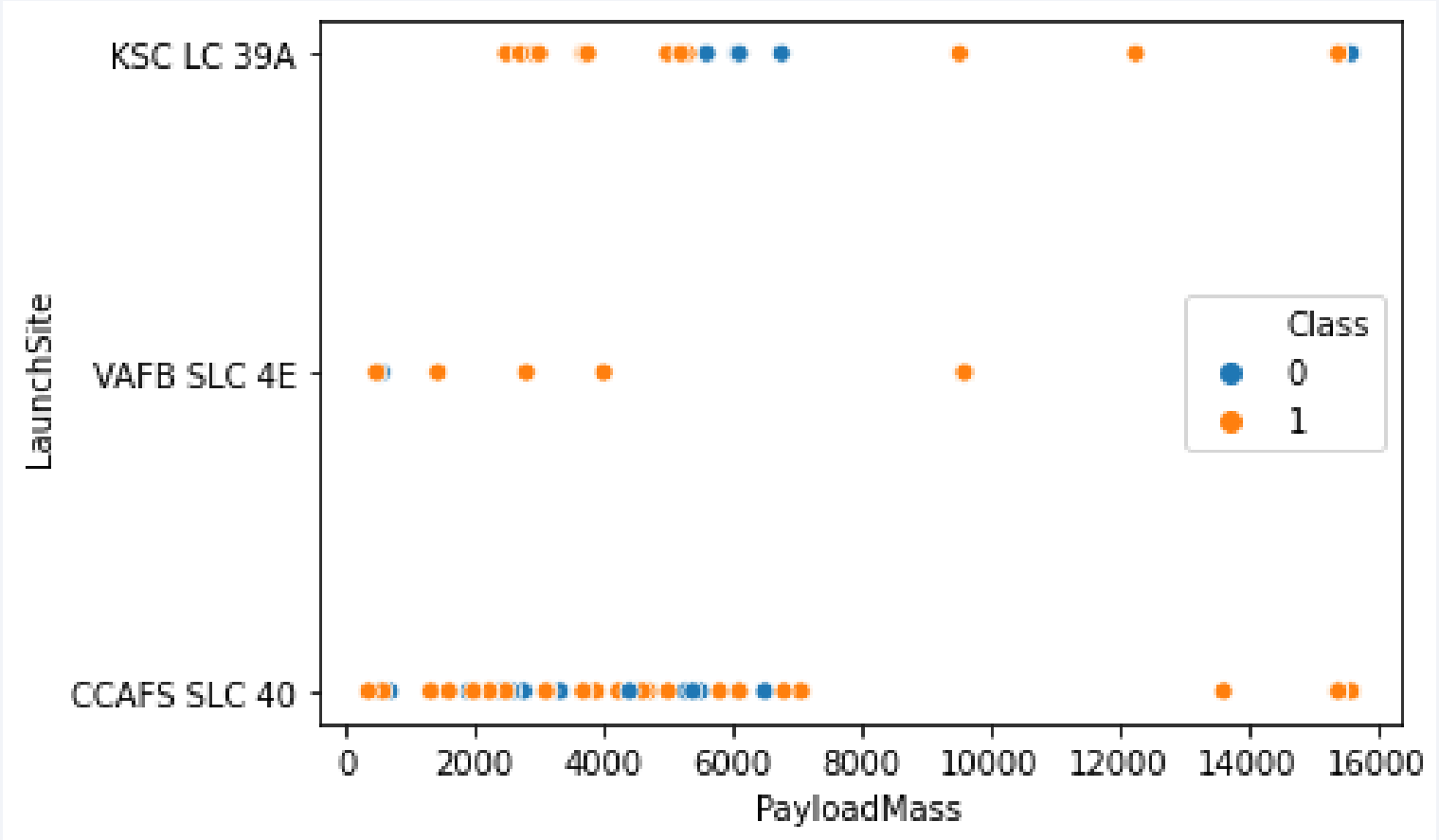
Flight Number vs. Launch Site

- With higher flight numbers (> 30) the success rate for the rocket is increasing.



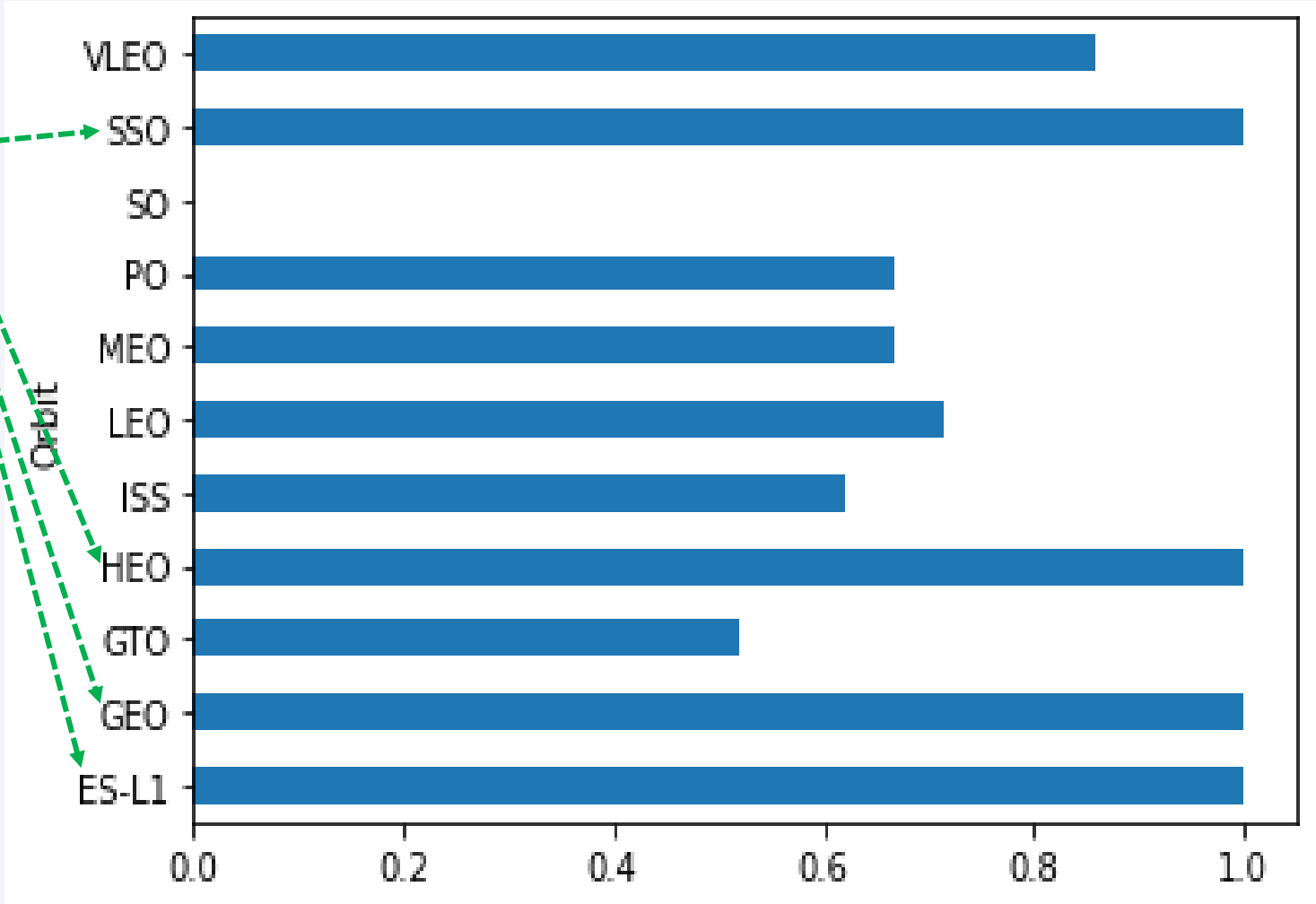
Payload vs. Launch Site

- The greater the payload mass (> 7000kg) *higher* the success rate for the rocket.



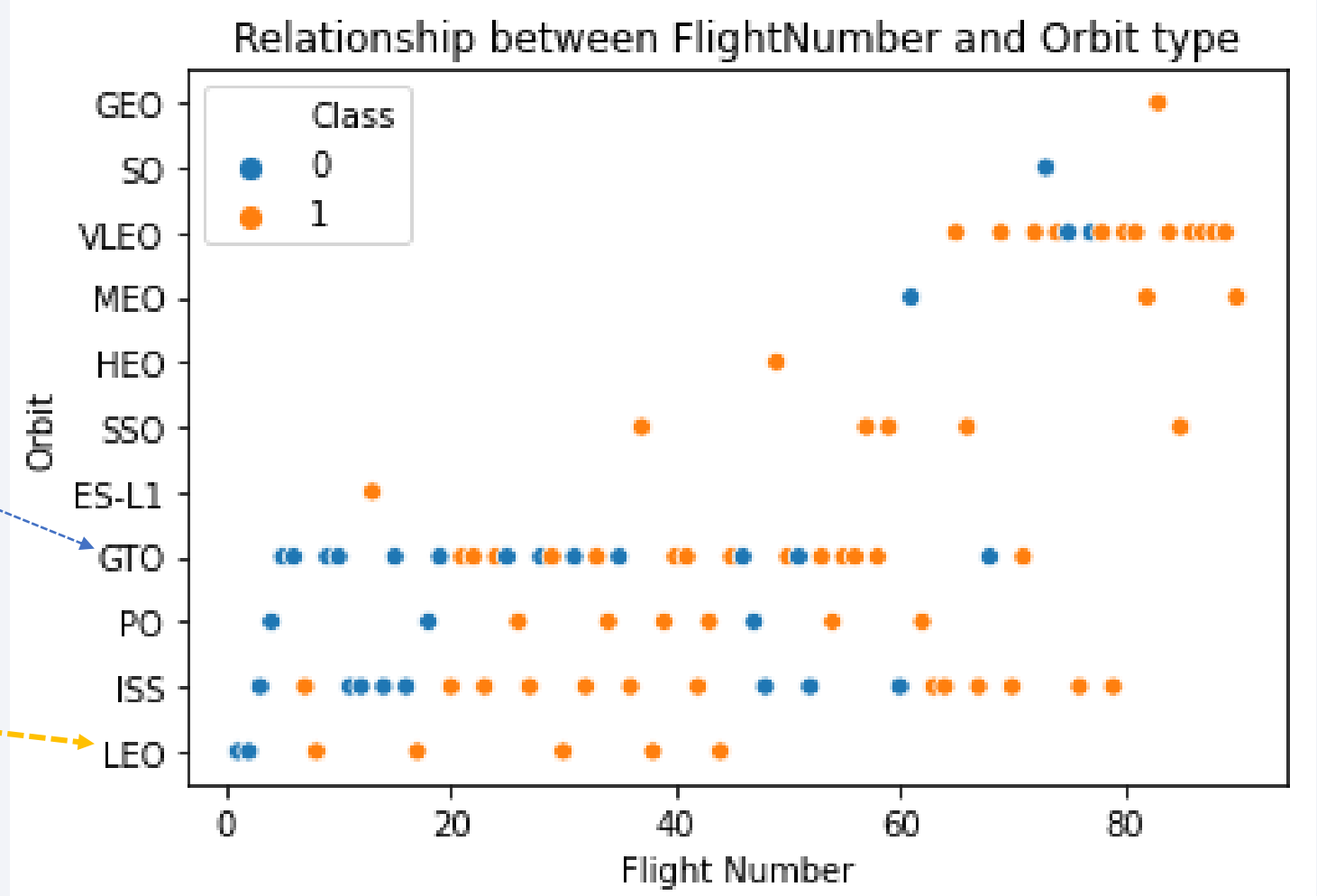
Success Rate vs. Orbit Type

- With SSO, HEO, GEO, ES-L1 with highest success rate



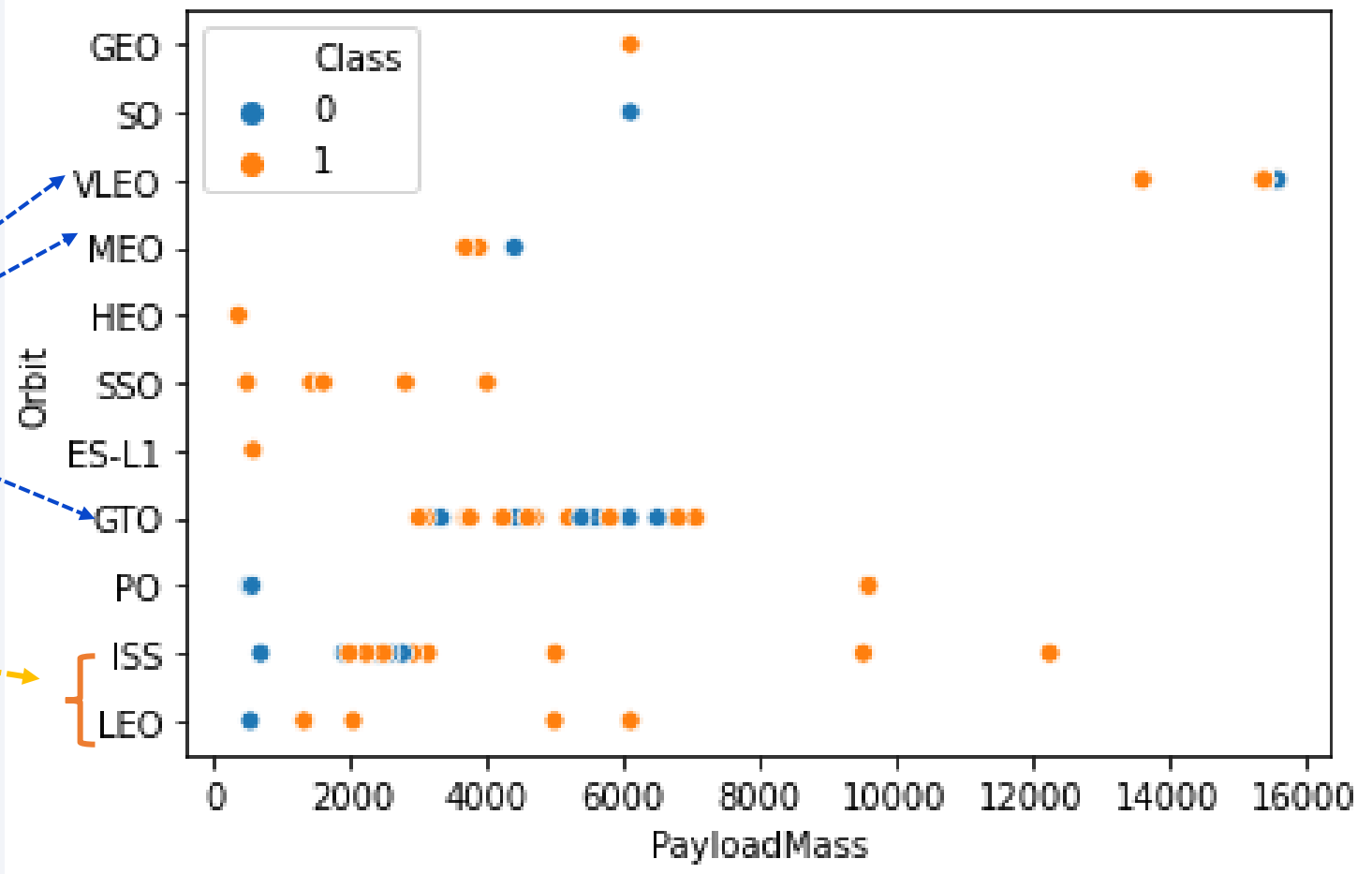
Flight Number vs. Orbit Type

- With it seems no relationship between flight number and the GTO orbit.
- On contrary, LEO orbit, the success rate increases with the number of flights.



Payload vs. Orbit Type

- The heavy payload has a negative influence on VLEO, MEO, GTO,
- but has a positive influence on ISS, LEO



Launch Success Yearly Trend

- There is an *increasing trend* in the success rate from 2013 to 2020, but there is a fluctuation since 2017.

EDA with SQL

I use https://labs.cognitiveclass.ai/v2/tools/jupyterlab to finish this assignment.

All Launch Site Names

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

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

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

Launch Site Names Begin with 'CCA'

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

We used the query above to display 5 records where launch sites begin with `CCA`

```
%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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

We calculated the total payload carried by boosters from NASA as 45596 using the query above

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_F9_V1_1 FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_F9_V1_1 FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

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

AVG_PAYLOAD_MASS_F9_V1_1
2928.4

First Successful Ground Landing Date

%%sql SELECT MIN(DATE) FROM SPACEXTBL

WHERE LANDING__OUTCOME = 'Success (ground pad)'

I use https://labs.cognitiveclass.ai/v2/tools/jupyterlab to finish my final assignment, so there is an unexpected result as below:

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

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT MIN(DATE) FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

%%sql SELECT Booster_Version FROM SPACEXTBL

WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'

I use https://labs.cognitiveclass.ai/v2/tools/jupyterlab to finish my final assignment, so there is an unexpected result as below:

```
%%sql SELECT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

Total Number of Successful and Failure Mission Outcomes

%%sql SELECT SUM(CASE WHEN MISSION_OUTCOME LIKE '%Success%' THEN 1 ELSE 0 END) AS Success,

SUM(CASE WHEN MISSION_OUTCOME LIKE '%Failure%' THEN 1 ELSE 0 END) AS Failure

FROM SPACEXTBL

```
%%sql SELECT SUM(CASE WHEN MISSION_OUTCOME LIKE '%Success%' THEN 1 ELSE 0 END) AS Success,
SUM(CASE WHEN MISSION_OUTCOME LIKE '%Failure%' THEN 1 ELSE 0 END) AS Failure
FROM SPACEXTBL
```

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

Success	Failure
100	1

Boosters Carried Maximum Payload

```
%%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
FROM SPACEXTBL)
```

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

2015 Launch Records

```
%%sql SELECT substr(Date, 4, 2) AS 'Month', BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE substr(Date,7,4)= '2015' AND LANDING__OUTCOME= 'Filuare (drone ship)'
```

```
%%sql SELECT substr(Date, 4, 2) AS 'Month', BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE substr(Date,7,4)='2015' AND LANDING__OUTCOME='Failure (drone ship)'
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT substr(Date, 4, 2) AS 'Month', BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE substr(Date,7,4)='2015' AND LANDING__OUTCOME='Failure (drone ship)']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

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

%%sql SELECT LANDING__OUTCOME AS 'Landing outcome', COUNT(LANDING__OUTCOME) AS 'Total count' FROM SPACEXTBL

WHERE (Date BETWEEN '2010-06-04' and '2017-03-20') AND LANDING__OUTCOME LIKE '%Success%'

GROUP BY Date

ORDER BY COUNT(LANDING__OUTCOME) DESC

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%%sql SELECT LANDING__OUTCOME AS 'Landing outcome', COUNT(LANDING__OUTCOME) AS 'Total count' FROM SPACEXTBL
WHERE (Date BETWEEN '2010-06-04' and '2017-03-20') AND LANDING__OUTCOME LIKE '%Success%'
GROUP BY Date
ORDER BY COUNT(LANDING__OUTCOME) DESC
```

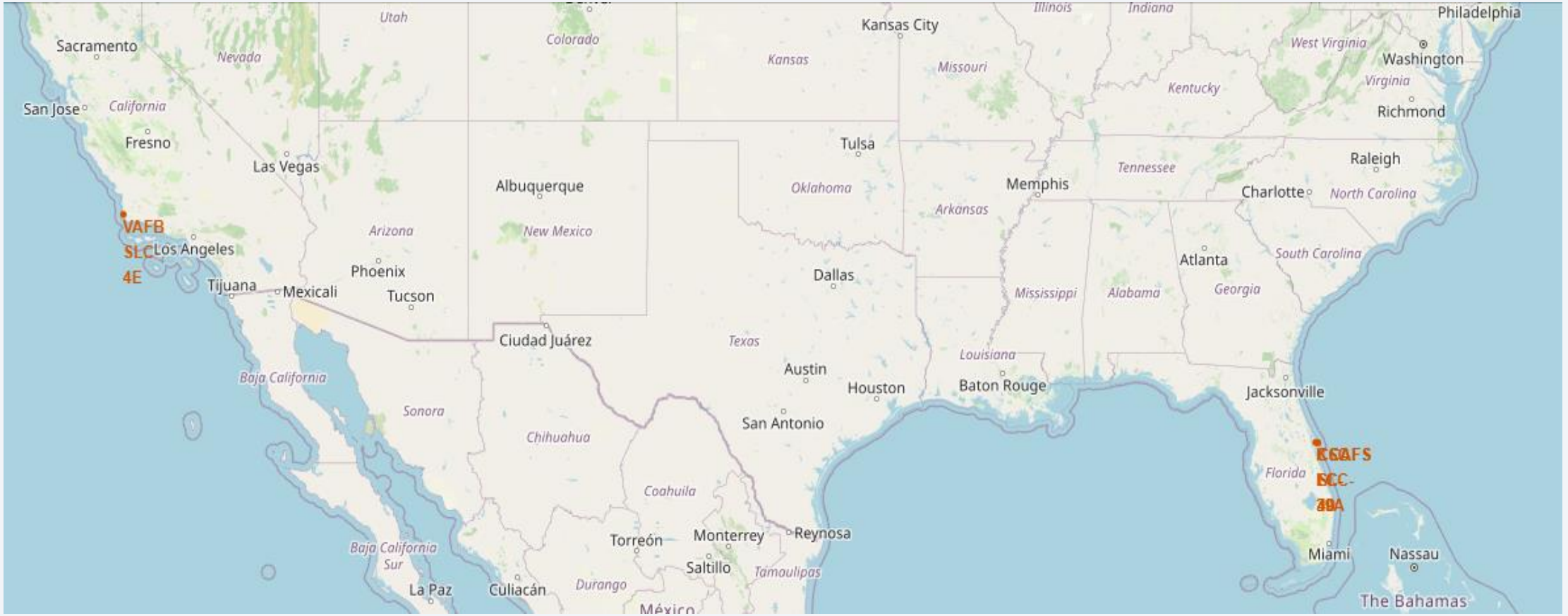
```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT LANDING__OUTCOME AS 'Landing outcome', COUNT(LANDING__OUTCOME) AS 'Total count' FROM SPACEXTBL
WHERE (Date BETWEEN '2010-06-04' and '2017-03-20') AND LANDING__OUTCOME LIKE '%Success%'
GROUP BY Date
ORDER BY COUNT(LANDING__OUTCOME) DESC]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

Section 3

Launch Sites Proximities Analysis

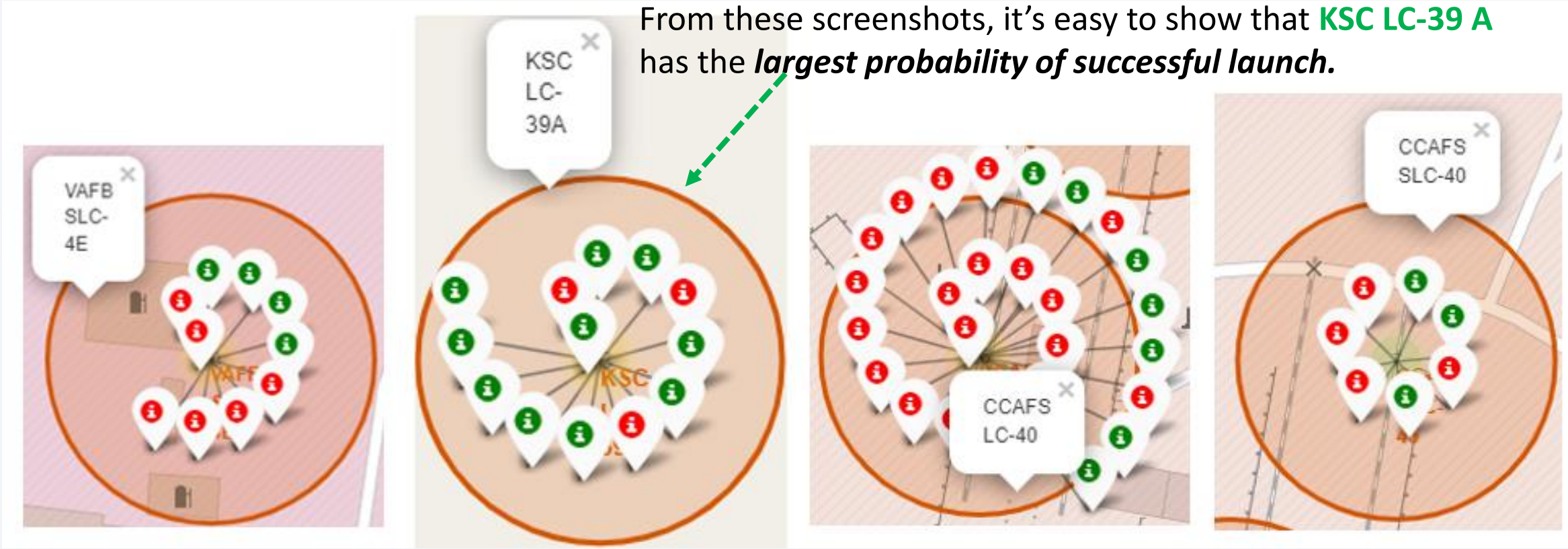
All launch sites on Folium Map

We can see that all SpaceX launch sites are near to US coasts (Florida, California)



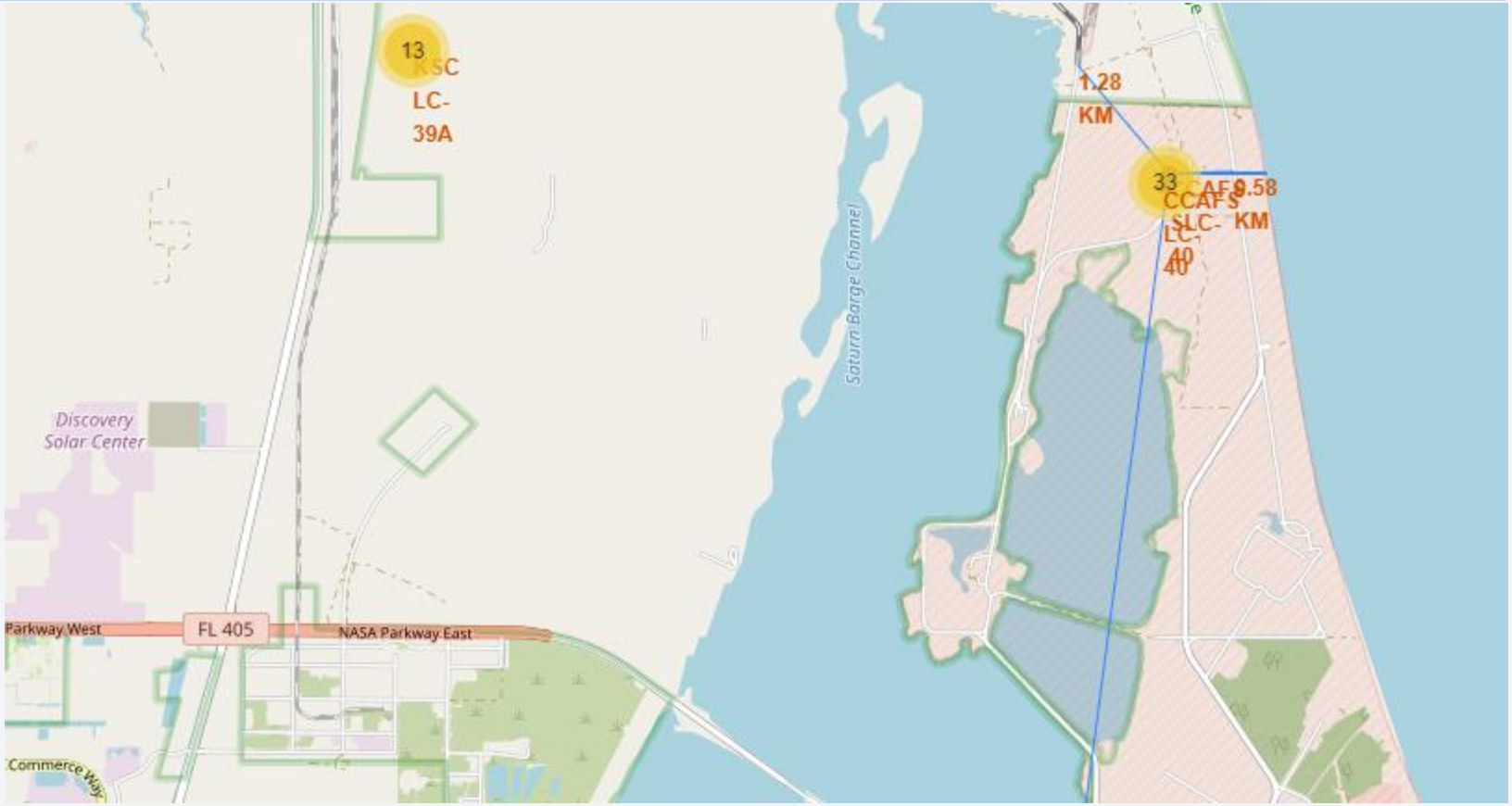
Color label launch records

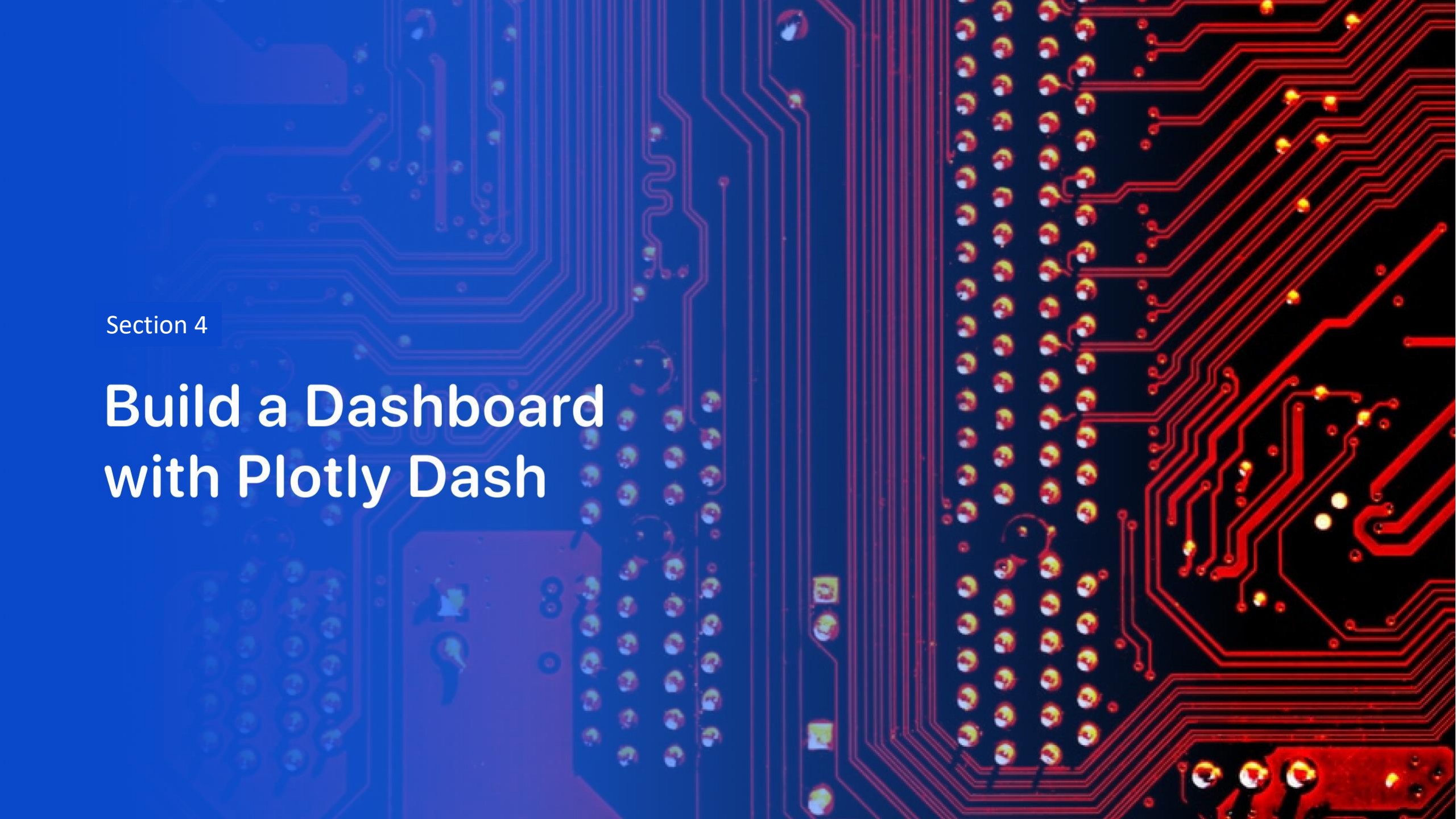
From these screenshots, it's easy to show that **KSC LC-39 A** has the ***largest probability of successful launch.***



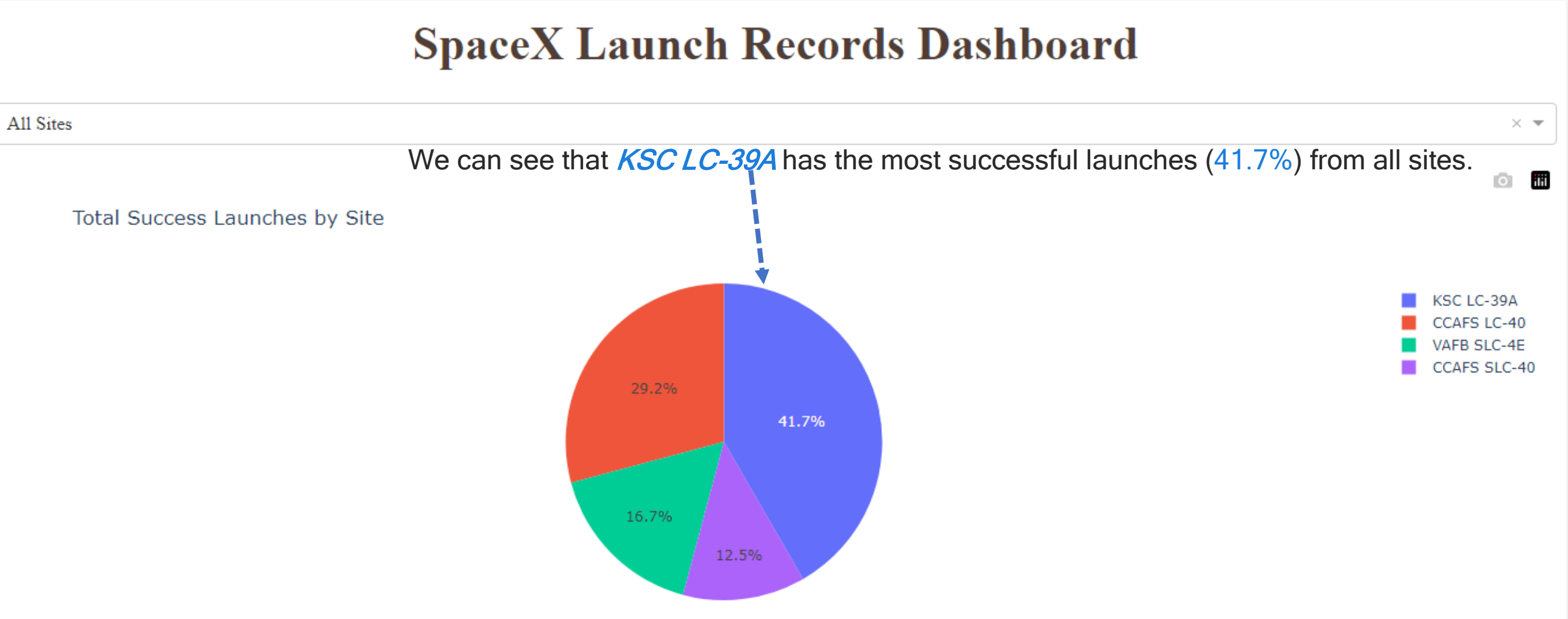
Launch Site distance to landmarks

Distance to railway, highway, city

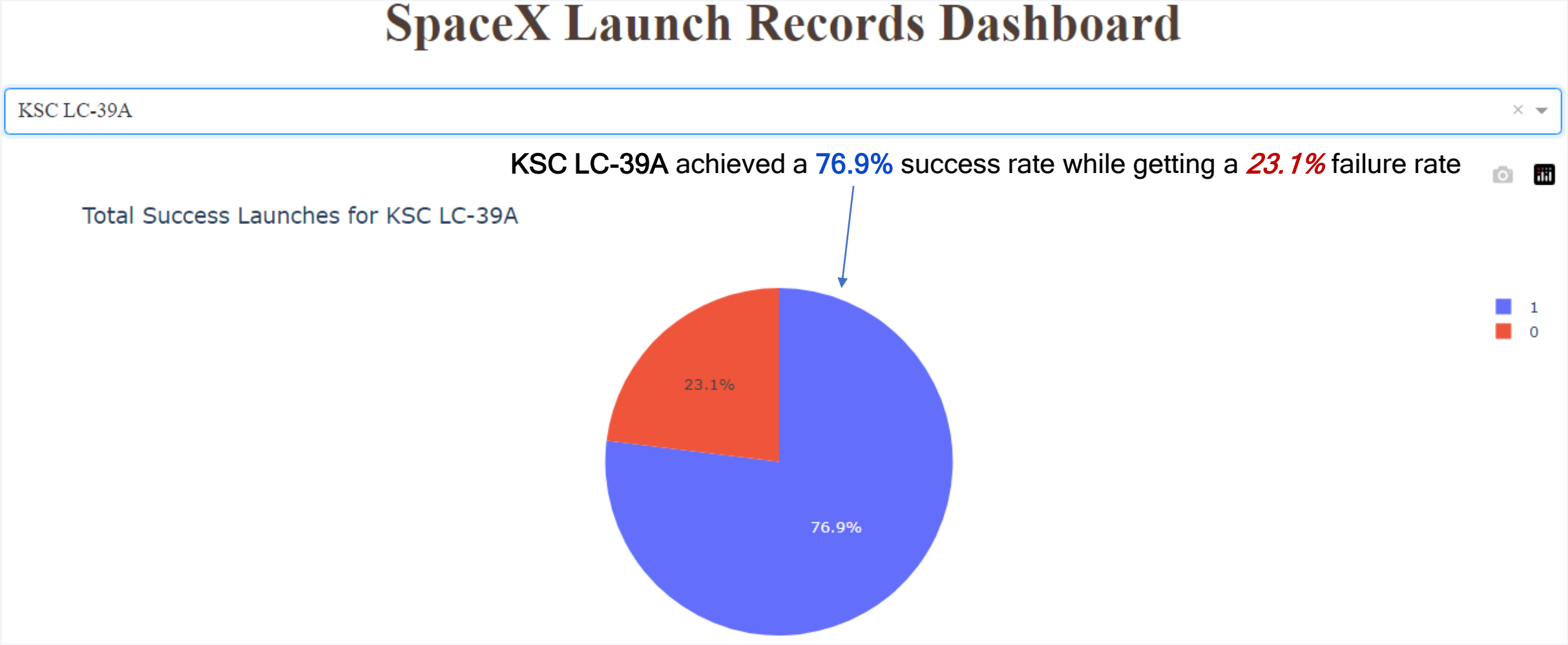



Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard



Launch site with the highest launch success ratio



Correlation between Payload and Success for all Sites

We can see the success rates for the low weighted payloads are ***higher*** than the heavily weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

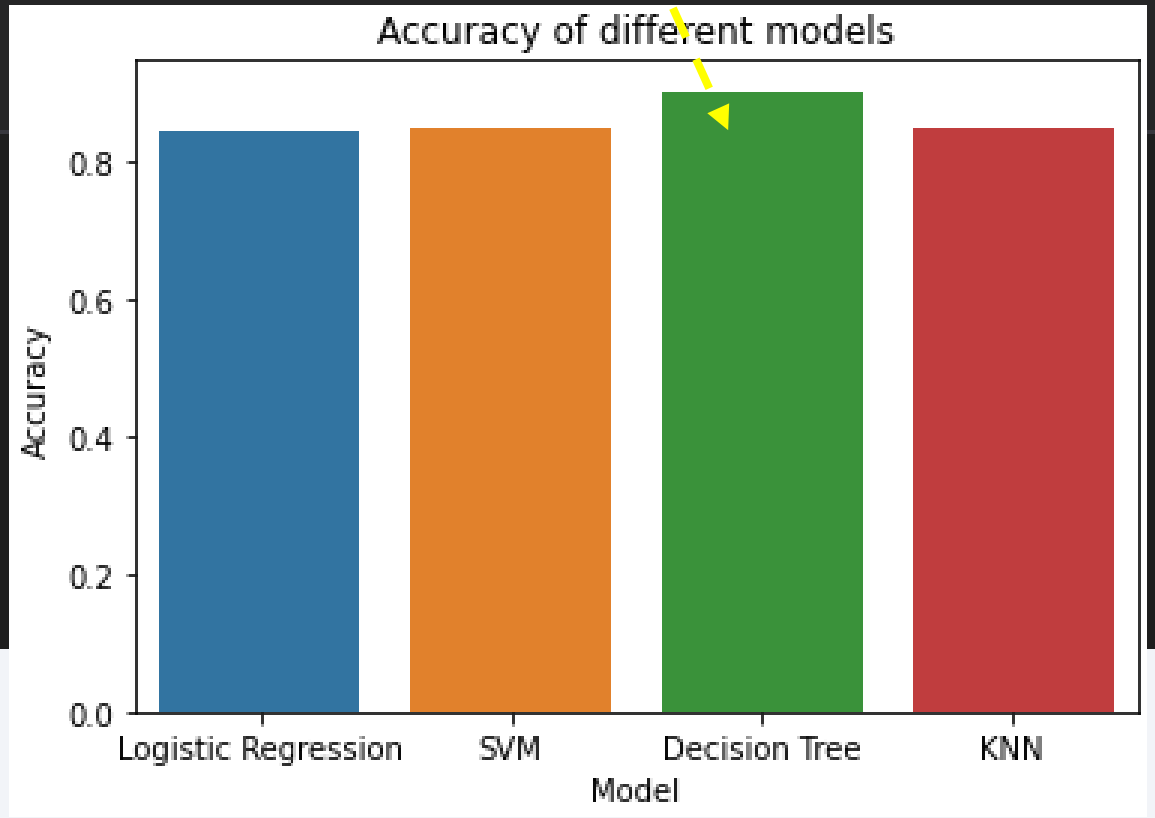
- The Decision Trees classifier is the model with the ***highest*** classification accuracy

```
The best model is:  Decision Tree with the accuracy of  90.36 %
```

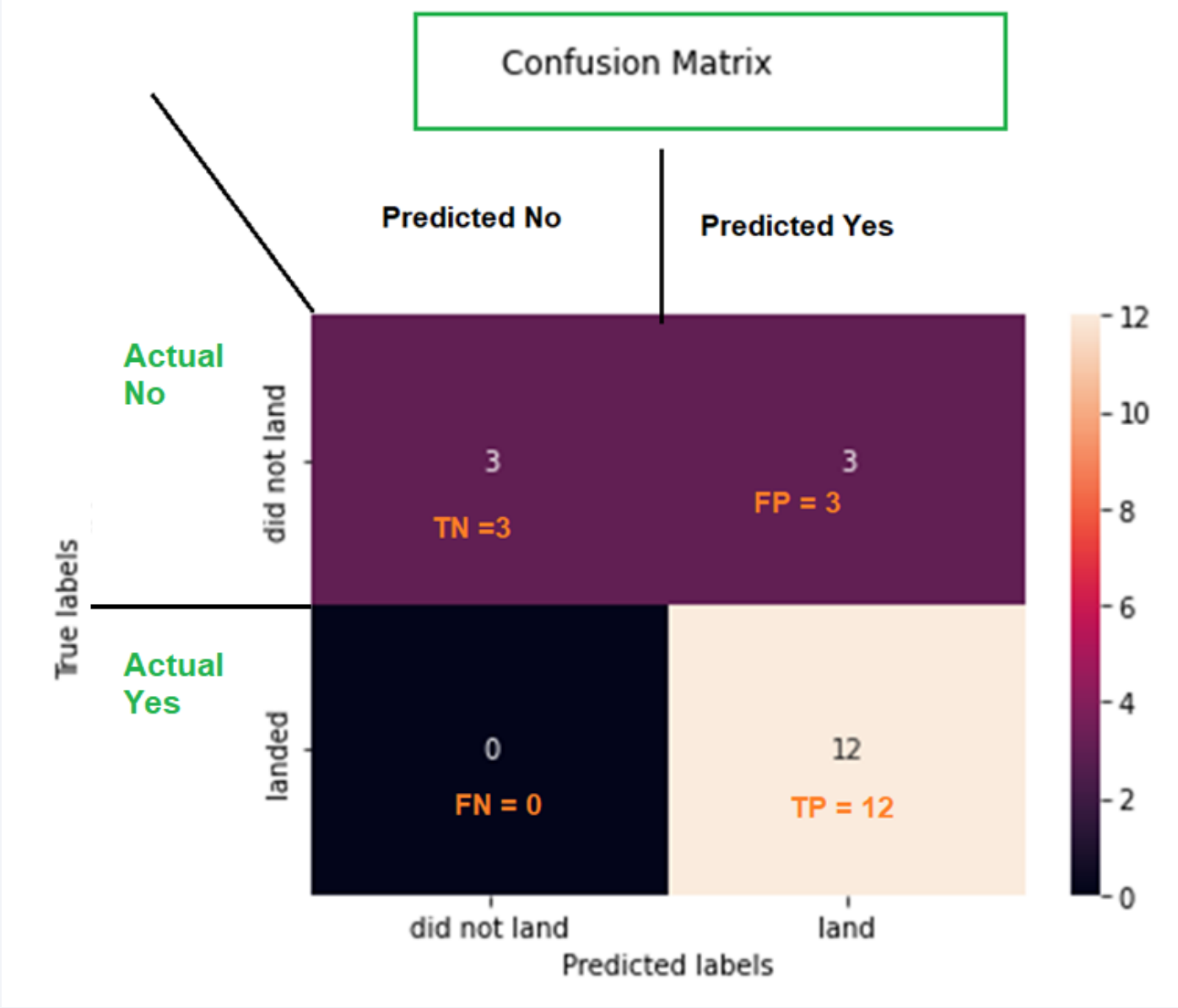
```
algo_df = pd.DataFrame.from_dict(scores, orient='index', columns=['Accuracy'])
algo_df.head()
```

✓ 0.1s

	Accuracy
Logistic Regression	0.846429
SVM	0.848214
Decision Tree	0.903571
KNN	0.848214



Confusion Matrix of the best model (Decision Trees)



Accuracy: (TP+TN)/Total = (3+12)/18 = **0.833**

Misclassification Rate: (FN+FP)/Total = (0+3)/18 = 0.167

True Positive Rate: TP/ Actual Yes = 12/(0+12) = 1

False Positive Rate: FP/Actual No = 3/(3+3)= 0.5

True Negative Rate: TN/Actual No = 3/(3+3)= 0.5

Precision: TP/ Predicted Yes = 12/ (12+3) = 0.8

Prevalence: Actual Yes/ Total = (0+12)/18 = 0.667

Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- The launch success rate started to increase in 2013 and till 2020.
- Orbits ES-L1, GEO, HEO, SSO, and VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any site.
- The Decision Tree classifier is the best machine learning algorithm for this task.

Appendix

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

Thank you!