Winning Space Race with Data Science

Hoang Duc Mai
Sep 20th, 2024

Outline

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

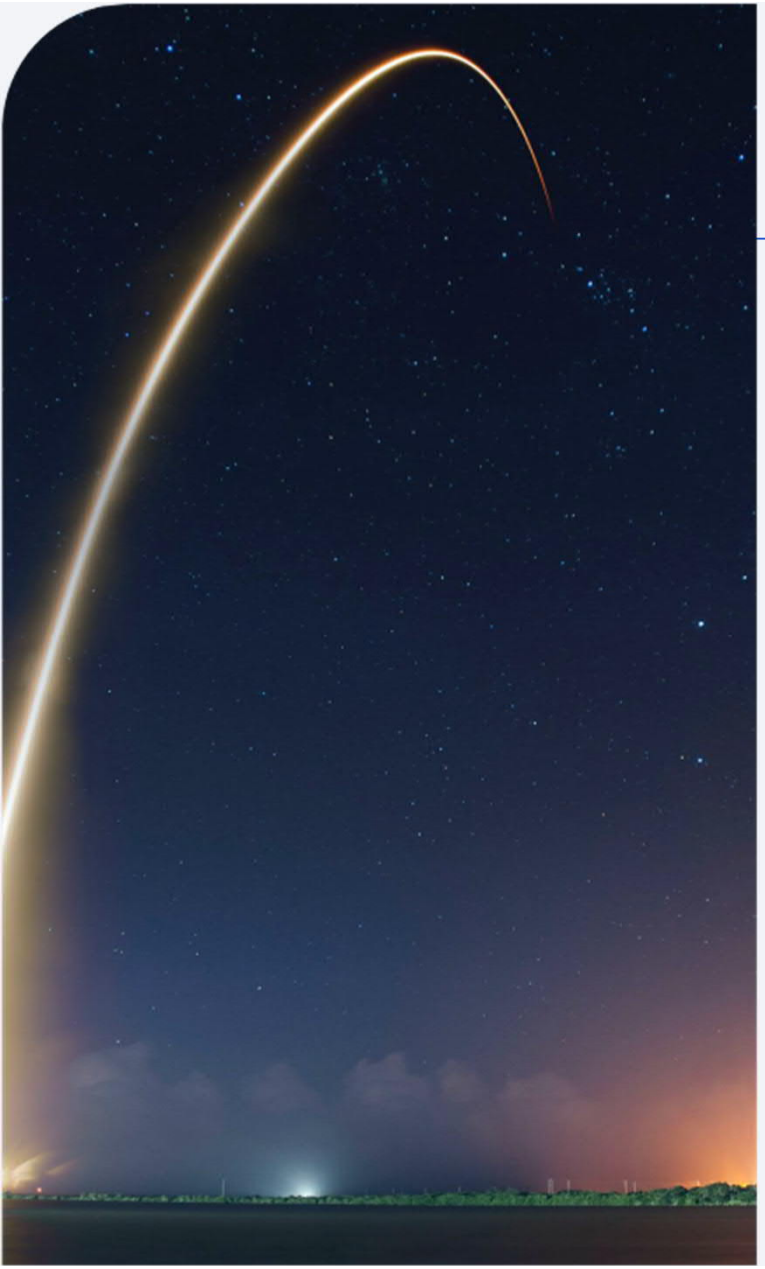
Executive Summary

- Summary of methodologies

The research attempts to identify the factors for a successful SPACEX rocket landing

  - Data collection from SpaceX REST API and relevant Wikipedia page
  - Data wrangling => Exploratory Data Analysis and determine Training Lab
  - Data exploration and analysis => factors impact mission outcomes
  - Visualisation of launch sites with the most success rates and payload ran
  - Machine learning techniques to predict landing outcomes

- Summary of all results
  - Launch success rate over time
  - Site and orbit with highest successful rate
  - Features of launch site
  - Performance of prediction models



Introduction

BACKGROUND AND CONTEXT

- SpaceX's reusable first stage rocket
- Significantly reduction of cost of space launch
- Space exploration more accessible.

PROBLEMS

Which factors influence the Reusability of SpaceX’s first stage rocket?

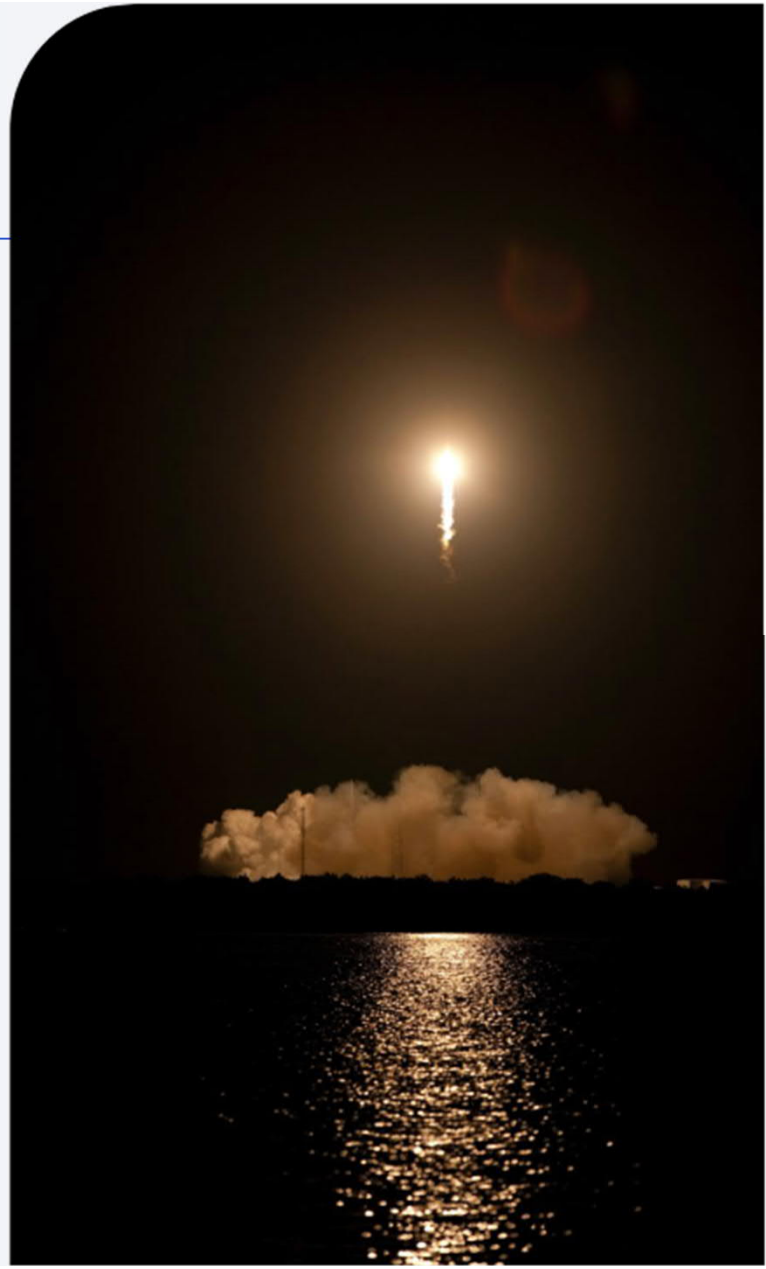
Section 1

Methodology

Methodology

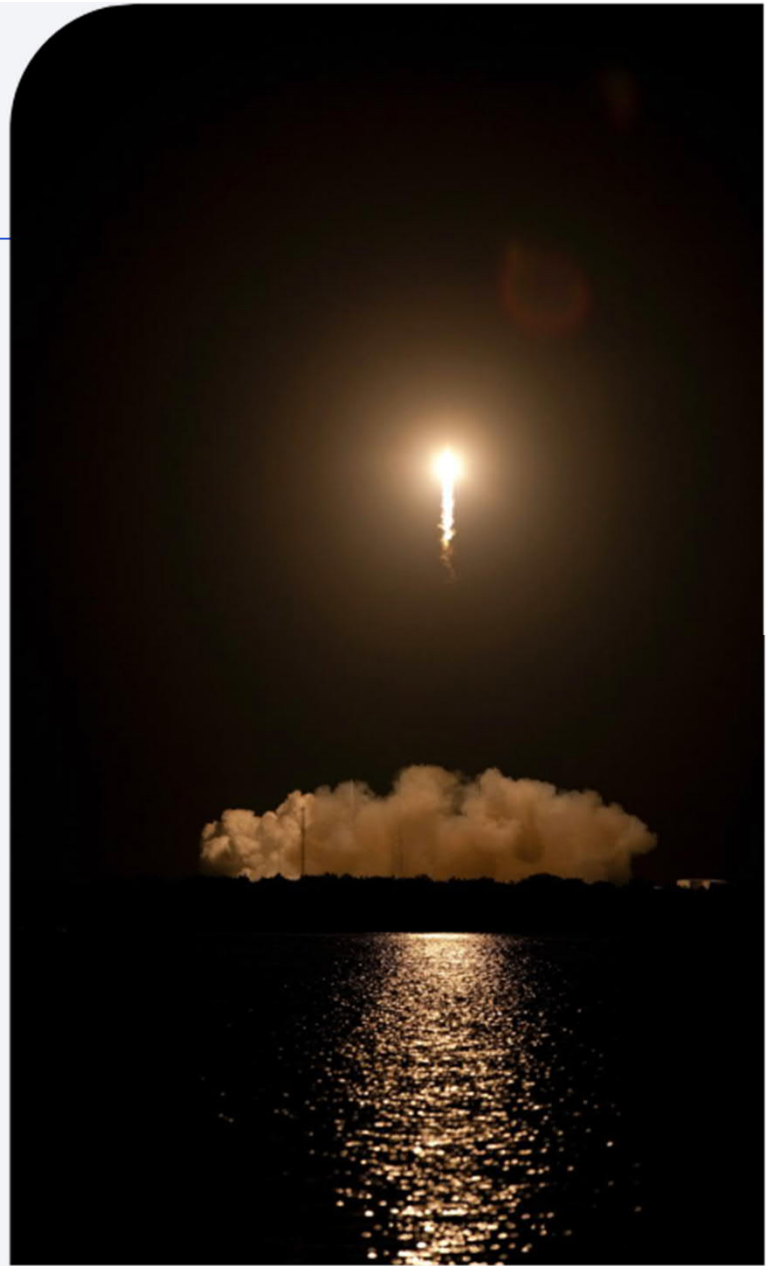
Executive Summary

- Data collection methodology:
  - REST API
  - Web scraping (from Wikipedia)
- Perform data wrangling
  - Filter data to keep the Falcon 9 launches
  - Remove missing values
  - Determine Training Label



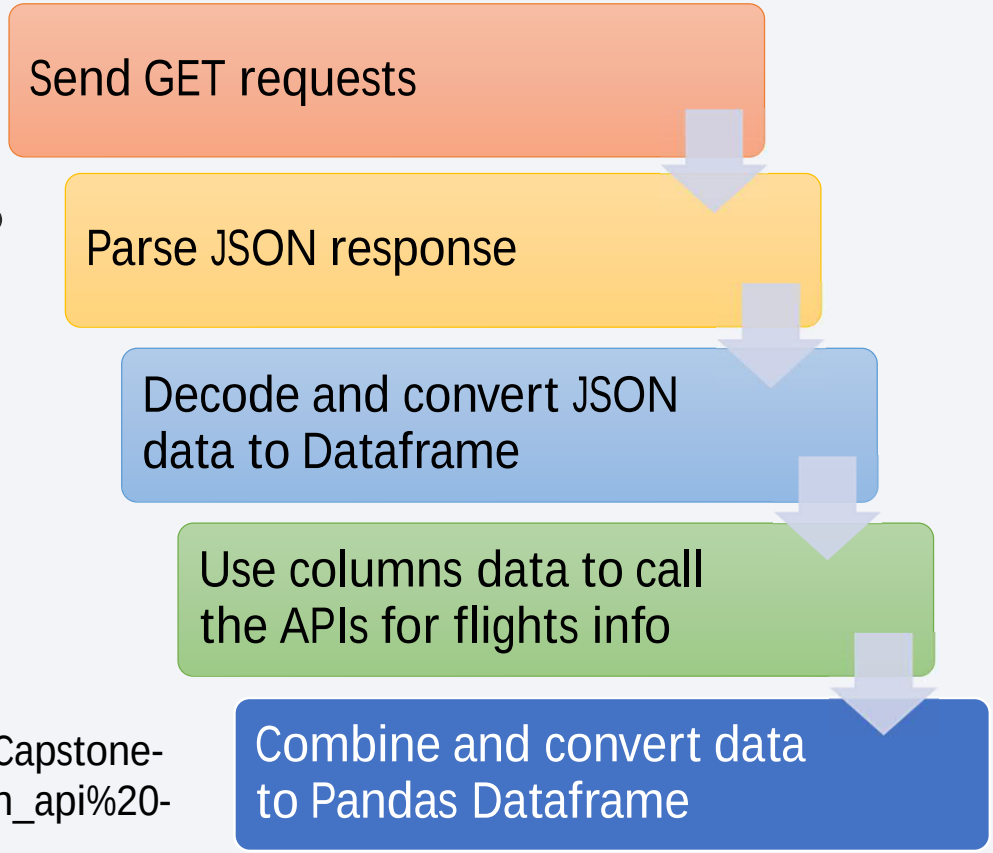
Methodology (cont.)

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Factors influence to landing outcomes (site, booster type, orbits, payload mass)
- Perform interactive visual analytics using Folium and Plotly Dash
  - Successful rate for each factors (launch site, payload)
  - Map of site locations and their success/failed launches
- Perform predictive analysis using classification model
  - Support Vector Machine, Logistic Regression, KNN, Decision Tree



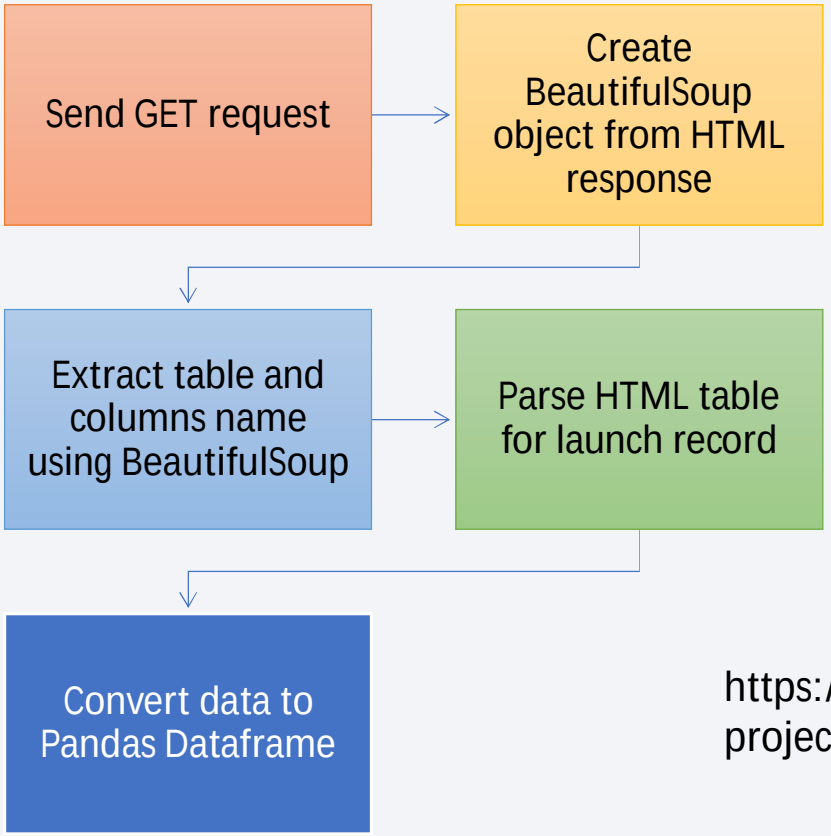
Data Collection – Using SpaceX's REST API

- Request and parse the SpaceX launch data using the GET request
- Decode and turn the response content to Pandas Dataframe using JSON method
- Call flight information using APIs obtained from column data
- Append data to the lists
- Combine lists of data and convert to Pandas Dataframe



https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/jupyter_labs_spacex_data_collection_api%20-%20DMA.ipynb

Data Collection – Scraping from Wikipedia



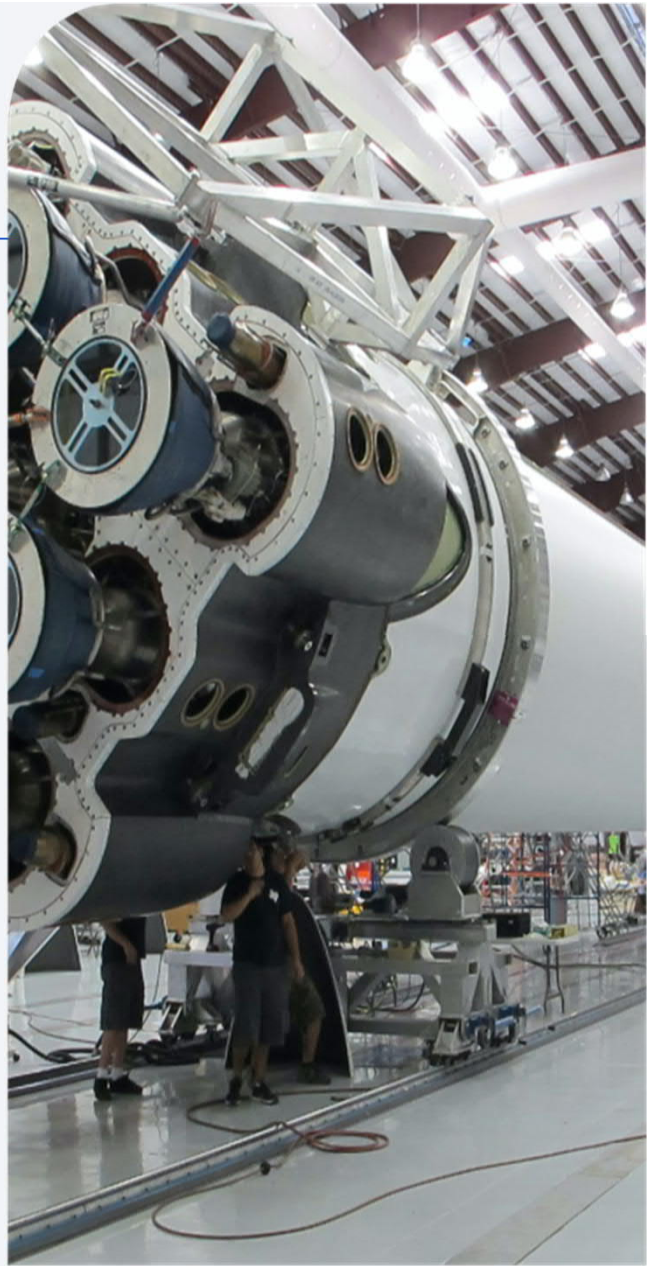
- HTTP GET method to request the Falcon9 Launch HTML Wikipedia page
- Use find_all method of BeautifulSoup to parse table and columns
- Extract data from HTML tables to fill in the dictionary
- Covert dictionary to Pandas dataframe using pd.Dataframe method()

https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/jupyter_labs_webscraping%20-%20DMA.ipynb

Data Wrangling

- Filter Dataset to only include data of interest (Falcon 9 launches)
- Understand Dataset (column type, missing values)
- Replace missing values (Landing Pad)
- Cast all numeric columns to Float 64
- Create dummy variables to categorical columns (Orbits, Launch site, Landing pad, Serial)
- Explore independent variables (launch sites, orbits, occurrence of mission)
- Explore and prepare dependent variable (Landing Outcome)
- Create categorical dependent variable (Landing class)
- Export Data to CSV file for further exploratory tasks

https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/labs_jupyter_spacex_Data_wrangling%20-%20DMA.ipynb



EDA with Data Visualization

Identify trends or patterns between launch features

- Flight Number vs Payload Mass
- Flight Number vs. Launch Site
- Payload Mass (kg) vs. Launch Site
- Success rate of each orbit type
- Flight number vs Orbit type
- Payload Mass vs Orbit type
- Launch Success trend

Identify factors influence dependent variable

https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/edadataviz%20-%20DMA.ipynb

EDA with SQL

- Name of unique launch sites
- Payload Mass by booster type
  - Booster launch by NASA (CRS)
  - Booster version F9 v1.1
- Date of first successful landing
- Booster which have success in drone ship and have payload mass in range of 4000 – 6000kg
- Total number of successful and failure outcomes
- Booster carried the maximum payload mass
- Count of landing outcomes for a certain period

https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/jupyter_labs_eda_sql_coursera_sqllite%20-%20DMA.ipynb

Build Interactive Map with Folium

A map was prepared to provide information on:

- Location of NASA – blue circle with marker
- Launch sites – orange circles with markers
- Launch record with color-labeled marker for success / failure
- Distance from a launch site (CCAFS SLC-40) to the nearest populated area/highway/railway or coastline

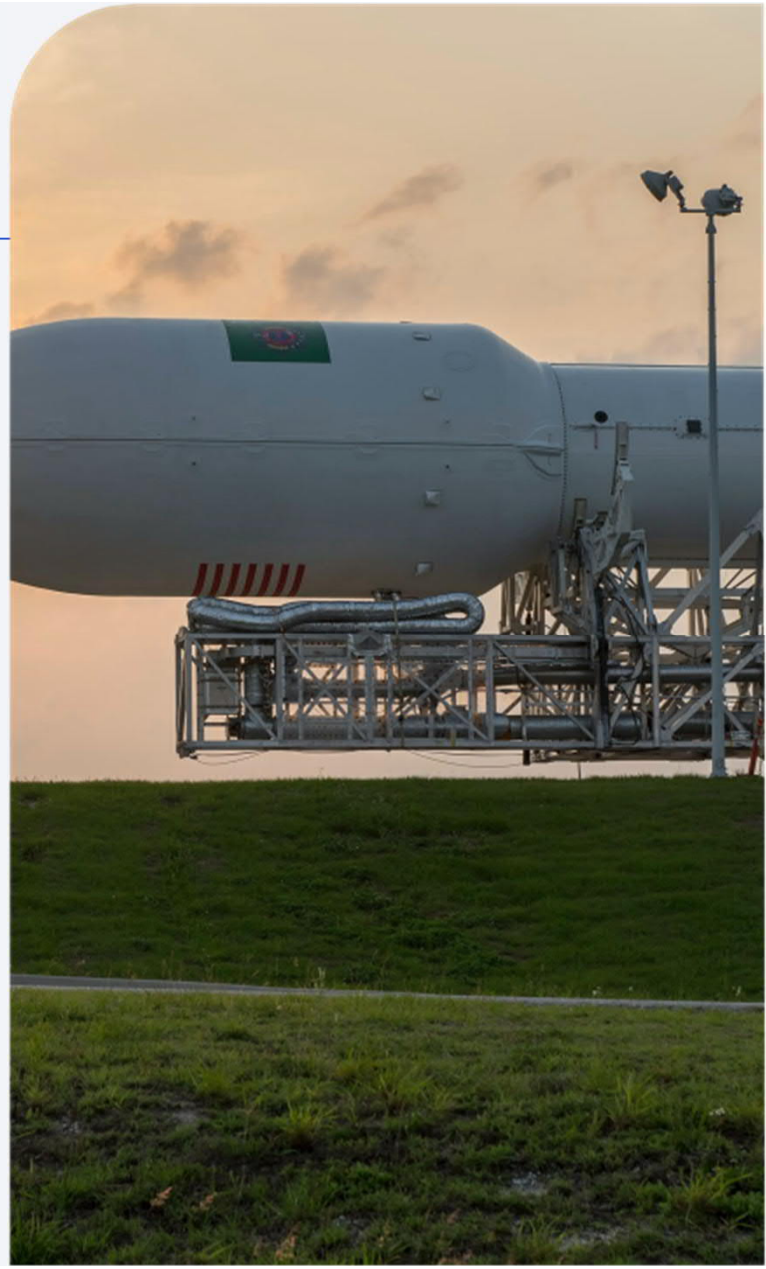
https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/lab_jupyter_launch_site_location%20-%20DMA.ipynb

Build Dashboard with Plotly Dash

The dashboard providing overview on success rate for payload and launch site was developed with the following items:

- A drop-down menu for selection of a specific launch site or all sites
- A pie chart visualizing launch success rate for the selected launch site option
- A range slider to select payload up to 10,000 kg
- A scatter plot visualising selected payload range for the nominated launch site option

https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/spacex_dash_app%20-%20DMA.py



Predictive Analysis (Classification)

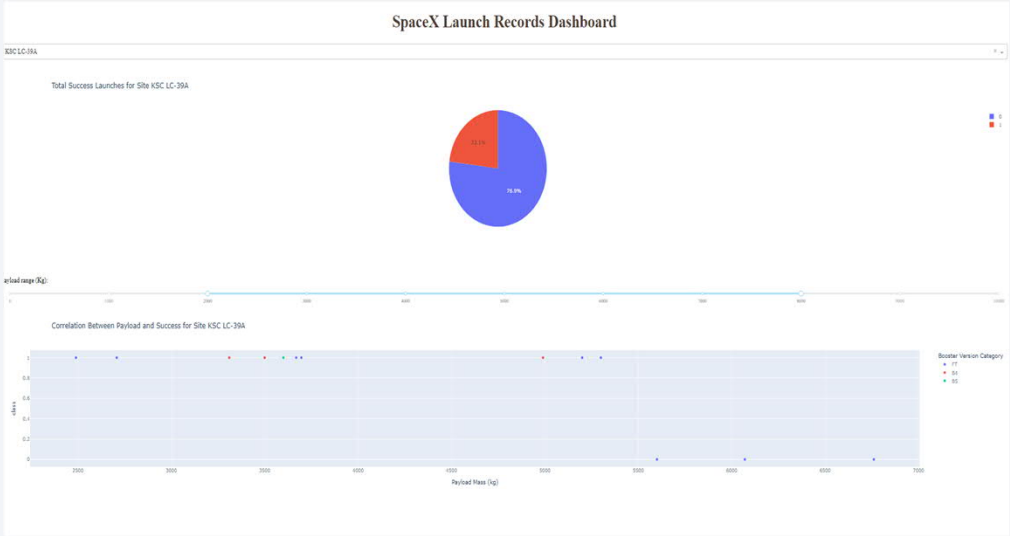
- Dependent variable: (Outcome class)
- Independent variables: Flight Number, Payload Mass, Orbit, Launch site, flights, Reused, etc. are converted from categorical data into numerical data by creating binary columns for each unique category.
- Independent variables are standardized to improve the performance of machine learning algorithms
- Dataset is split to 2 subsets: train (80%) and test (20%)
- Machine learning algorithms applied: Logistic regression, Support vector machine, Decision tree, K Nearest Neighbour
- Compare accuracy using method score to select the best algorithms

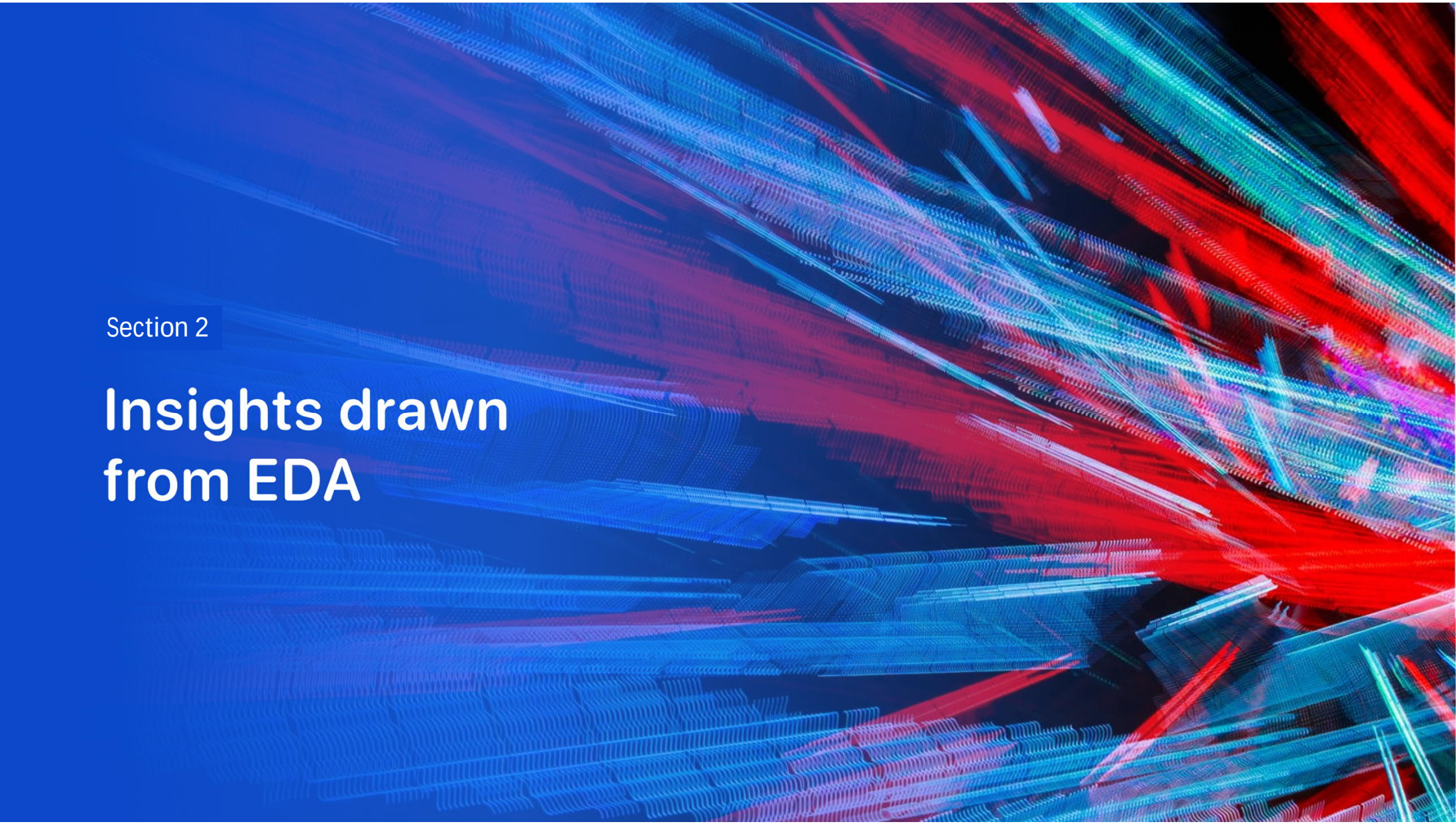
https://github.com/HoangDucMai/IBM-Data-Science---Capstone-project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5%20-%20DMA.ipynb

Results

- Exploratory data analysis results
  - Launch success improved over time
  - KSC LC-39A has the highest success rate among landing sites (76.9%)
  - Orbits ES-L1, GEO, HEO and SSO have a 100% success rate
- Predictive analysis results
  - All models have same accuracy score (83.33%)
  - Decision tree has the highest accuracy (87.5%)

Interactive analytics demo

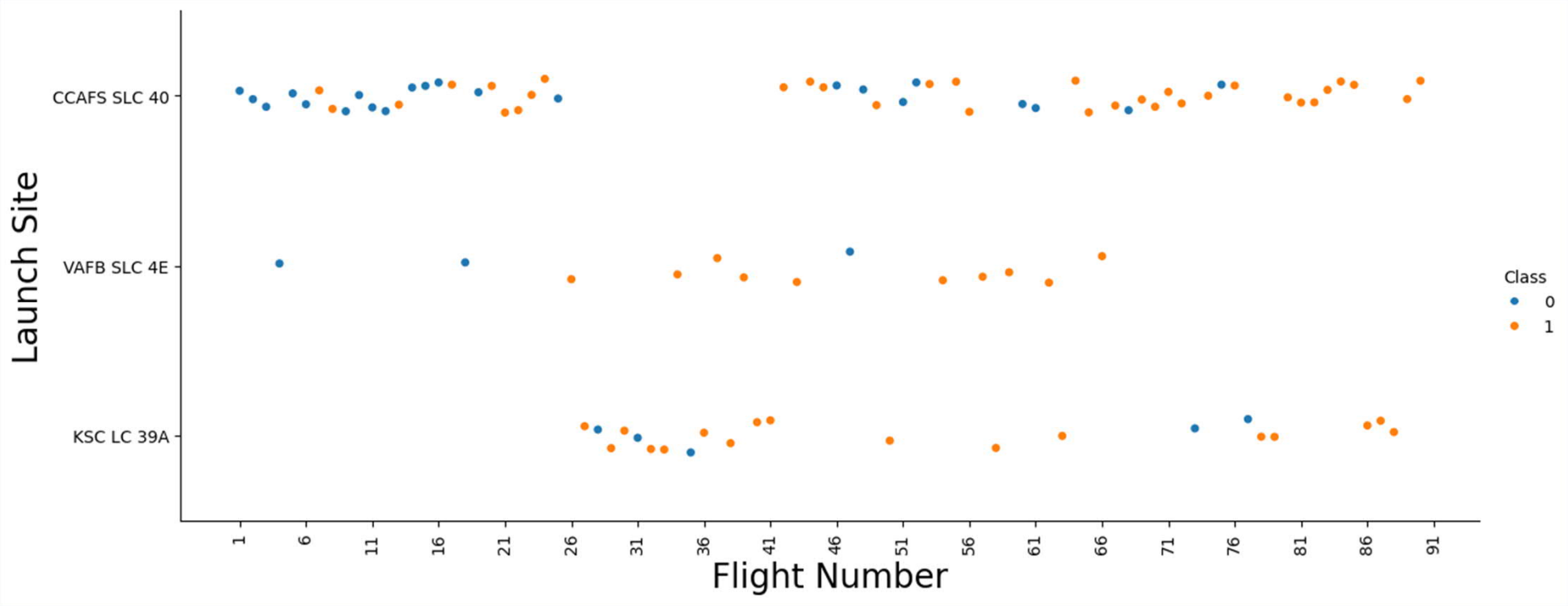




Section 2

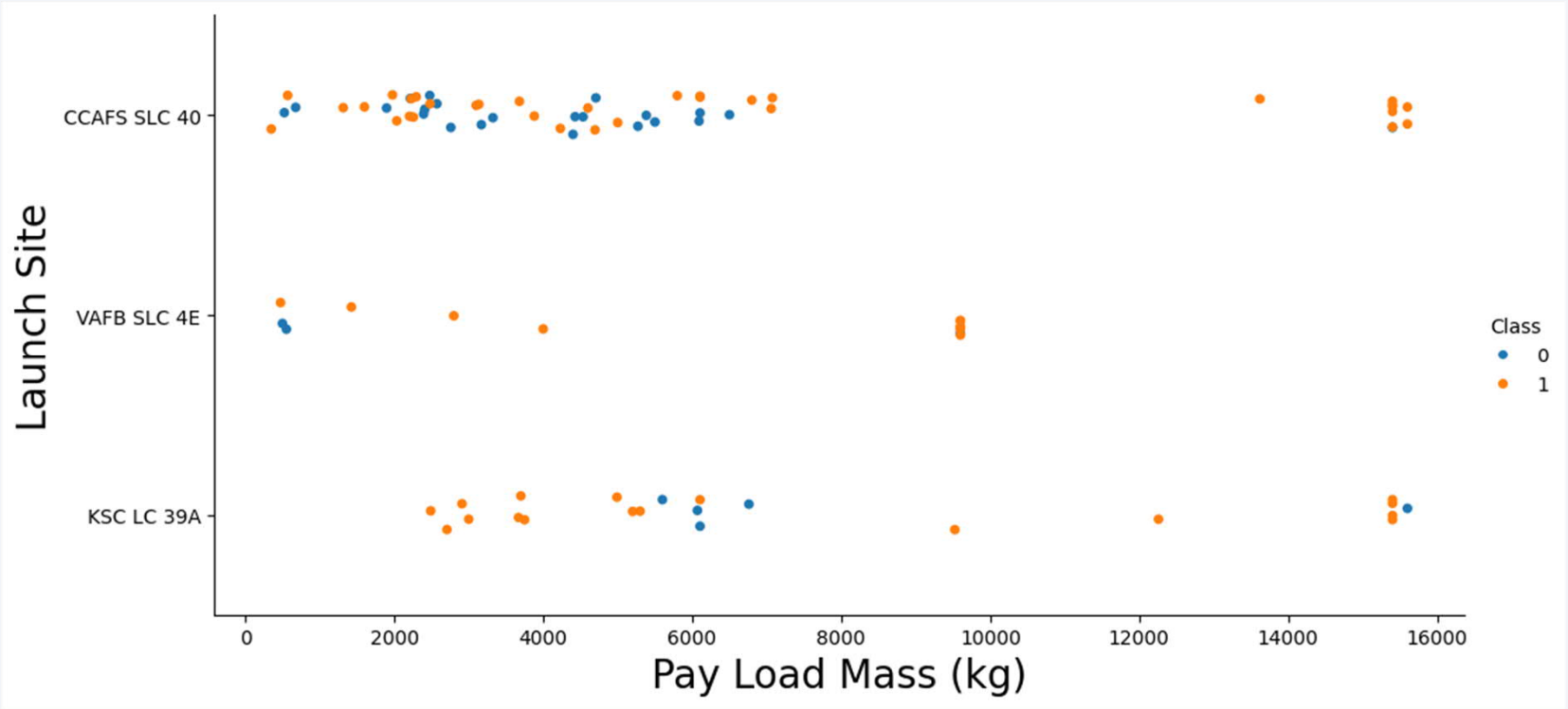
Insights drawn from EDA

Flight Number vs. Launch Site



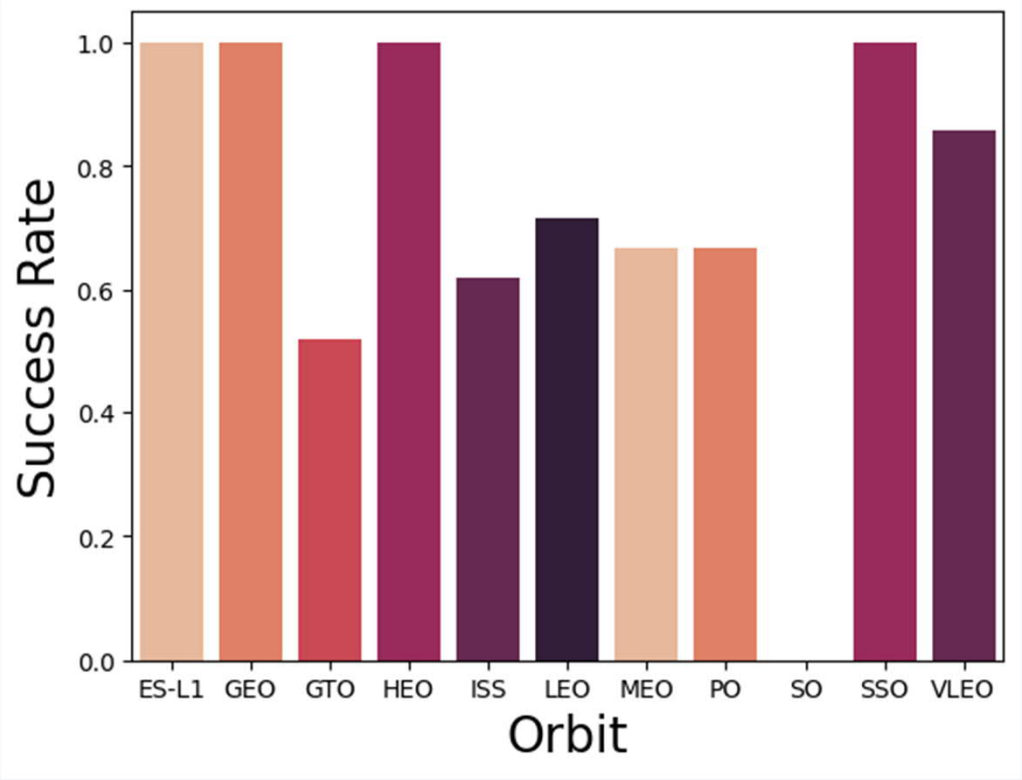
- Success rate for each site improves over time
- CCAFS SLC 40 has the most launch

Payload vs. Launch Site



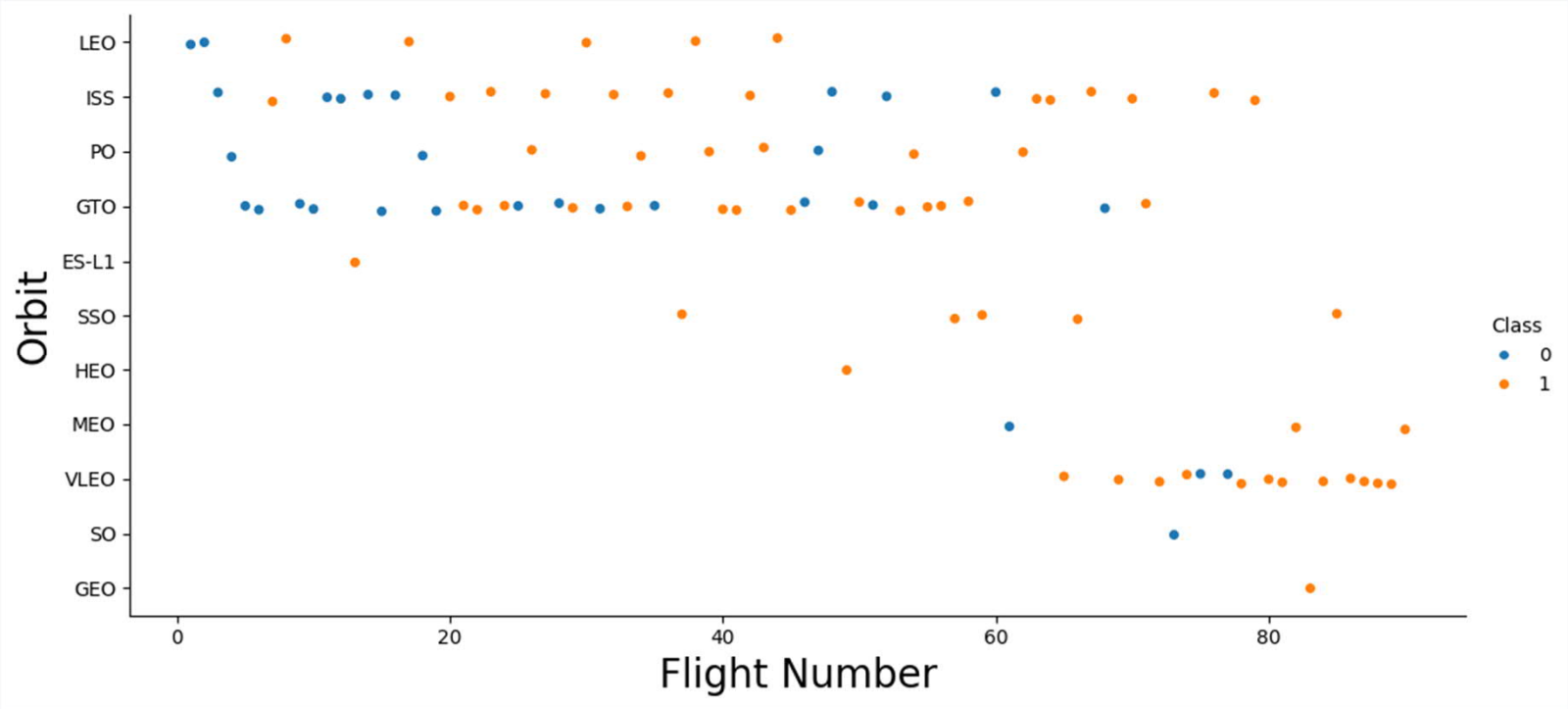
- CCAFS SLC 40 has higher success rate for heavier payload mass
- KSC LC 39A has 100% success rate for launches less than 5,500kg
- Most launches with a payload greater than 7,000 kg were successful

Success Rate vs. Orbit Type



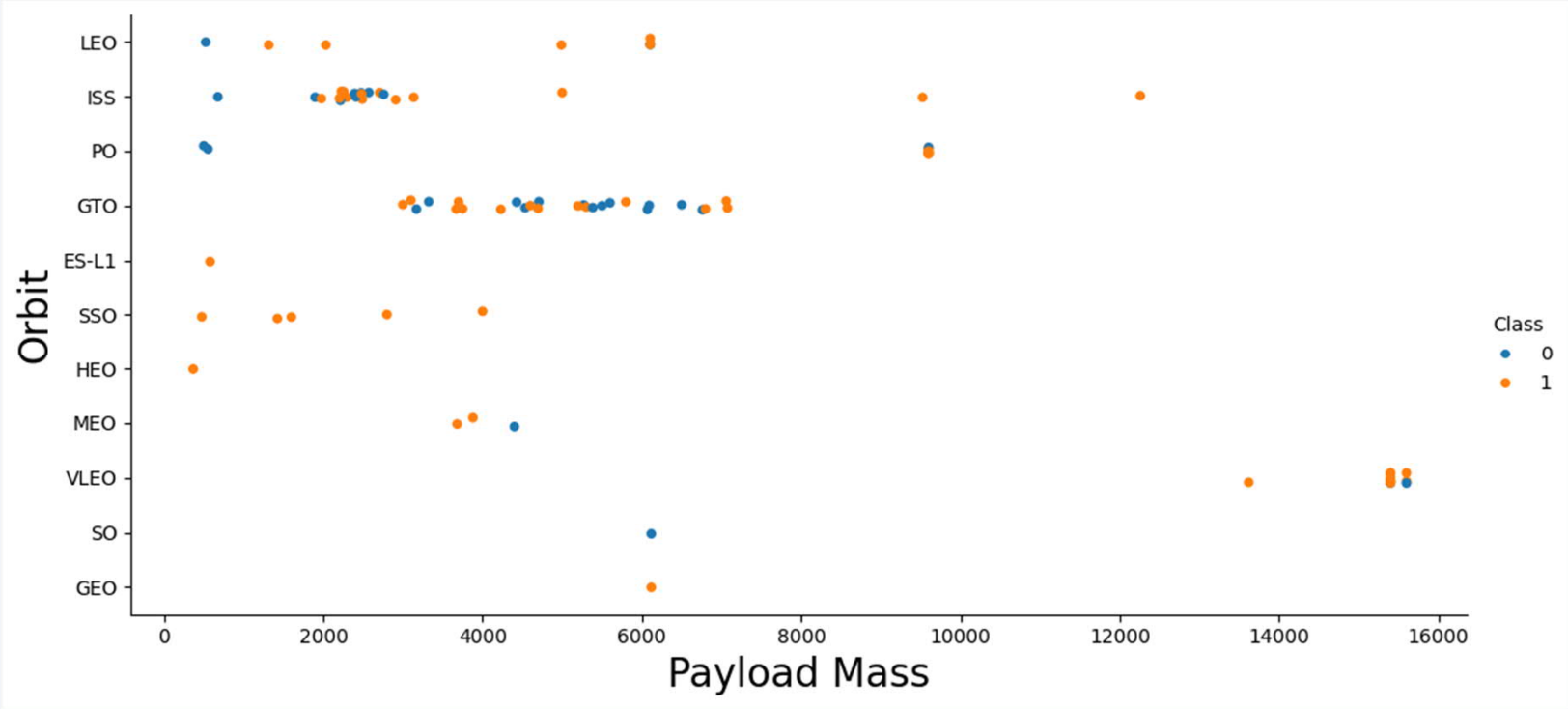
- Mission(s) to SO was not successful
- Except mission to SO, other orbit has at least 50% successful rate
- Most successful orbits: SSO, GEO, HEO and ES-L1

Flight Number vs. Orbit Type



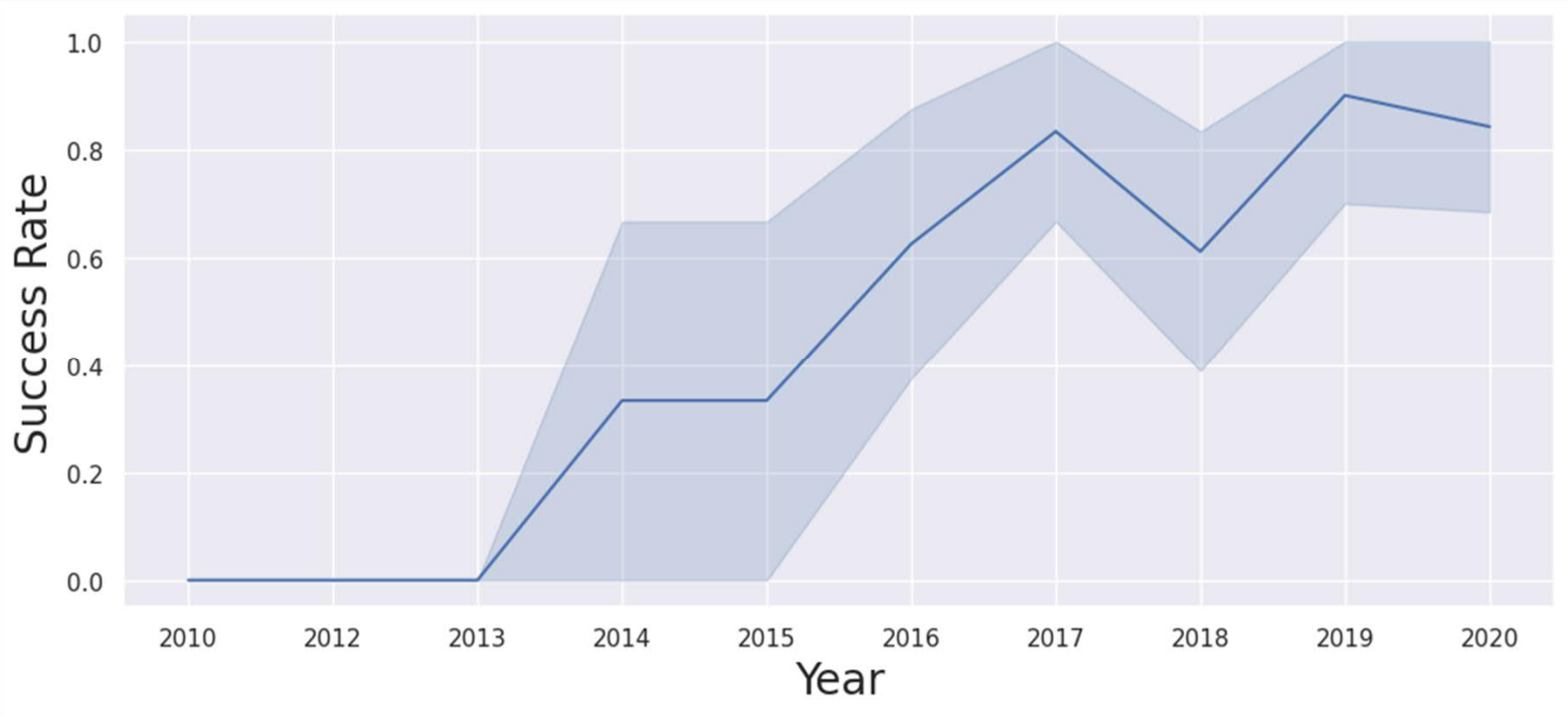
- Most common orbits are ISS and GTO
- SSO, ES-L1, HEO and GEO have high successful rate but low number of flight
- Success rate typically increases with the number of flight for each orbit except GTO

Payload vs. Orbit Type



- Correlation between orbit type and payload mass
- Wider range of payload mass to ISS
- VLEO has highest payload mass flights

Launch Success Yearly Trend



- Success rate of Falcon 9 booster typically improves over time
- 5 failures in 2018 dipped the success rate for that year

All Launch Site Names

Use SQL to query launch site from the table

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT DISTINCT: select unique values from the table
- FROM: nominate source of query

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

Launch Site Names Begin with 'CCA'

Use SQL to query launch site from the table

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT *: selects all columns from the table
- FROM: specifies the table from which to retrieve data
- WHERE column_name: filters the results to only include rows where the nominated column satisfy condition
- LIKE ‘string’: operator allows for ‘string’ matching
- LIMIT number: limits the number of rows return to the first nominated number

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Use SQL to query launch site from the table

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT function(‘column_name’): selects only calculated value from the ‘column_name’. The value will be calculated by math function SUM()
- FROM: specifies the table from which to retrieve data
- WHERE column_name: filters the results to only include rows where the nominated column satisfy condition
- = ‘string’: operator to match ‘string’

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

SUM(PAYLOAD_MASS__KG_)

Average Payload Mass by F9 v1.1

Use SQL to query average payload mass carried by booster version F9 v1.1

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT function(‘column_name’): selects only calculated value from the ‘column_name’. The value will be calculated by math function AVG()
- FROM: specifies the table from which to retrieve data
- WHERE column_name: filters the results to only include rows where the nominated column satisfy condition
- = ‘string’: operator to match ‘string’

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Use SQL to query average payload mass carried by booster version F9 v1.1

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT function('column_name'): selects only calculated value from the 'column_name'. The value will be calculated by math function MIN()
- FROM: specifies the table from which to retrieve data
- WHERE column_name: filters the results to only include rows where the nominated column satisfy condition
- = 'string': operator to match 'string'

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Use SQL to query average payload mass carried by booster version F9 v1.1

- WHERE column_name: filters the results to only include rows where the nominated column satisfy condition
- = ‘string’: operator to match ‘string’
- AND: operator to ensures that all conditions must be met simultaneously
- < , >: math operator to compare column values with the conditions

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success
(drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

Use SQL to query average payload mass carried by booster version F9 v1.1

- %sql: magic command in Jupyter Notebook to indicate the code is an SQL query
- SELECT column: selects value from the ‘column_name’.
- COUNT (*): count all rows in each group regardless of their values
- FROM: specifies the table from which to retrieve data
- GROUPBY column_name: group the rows based on the values in column ‘column_name’

```
%sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

Boosters Carried Maximum Payload

Use SQL query and subquery to return booster version that carried the heaviest payload
Subquery (i.e. (SELECT MAX (PAYLOAD_MASS__KG_) FROM SPACEXTABLE)) is used to create more complex filtering conditions.

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

Use SQL to query failure landing outcomes in drone ship, booster versions, launch site for the months in 2015.

- Substr(Date, 0, 5) = ‘2015: check if the first 5 characters of the ‘Date’ column are equal to 2015. This will filter records for the year 2015
- Landing_Outcome = 'Failure (drone ship)’: filter the results to only include rows where the landing outcome is ‘Failure’

```
%sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version,
Launch_Site FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome =
'Failure (drone ship)';
```

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Use SQL to query and rank the count of landing outcomes between the dates in descending order

- Date BETWEEN '2010-06-04' AND '2017-03-20': filter records between the date range
- GROUP BY Landing_Outcome: groups the results by the 'Landing-outcome' column
- ORDER BY COUNT(*) DESC: sort by descending order

```
%sql SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC;
```

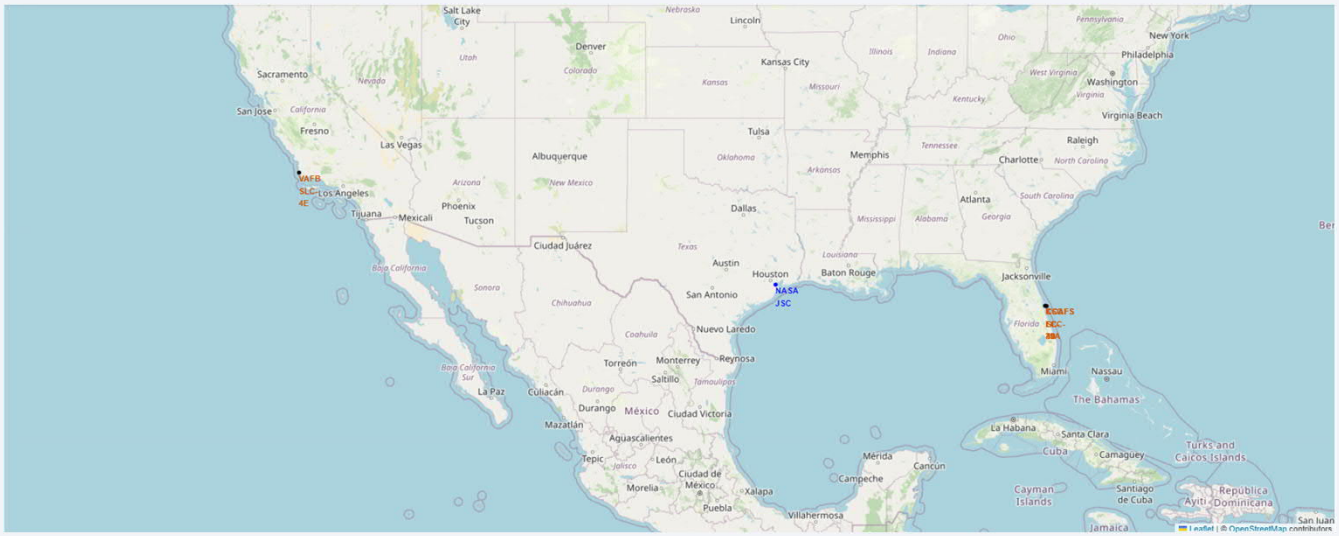
Landing_Outcome	COUNT(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

LAUNCH SITES

Launch sites:

- Near equator: increase launch velocity & reduce orbital inclination
- Close to coast: recovery of first stage rockets & minimise risk of failure

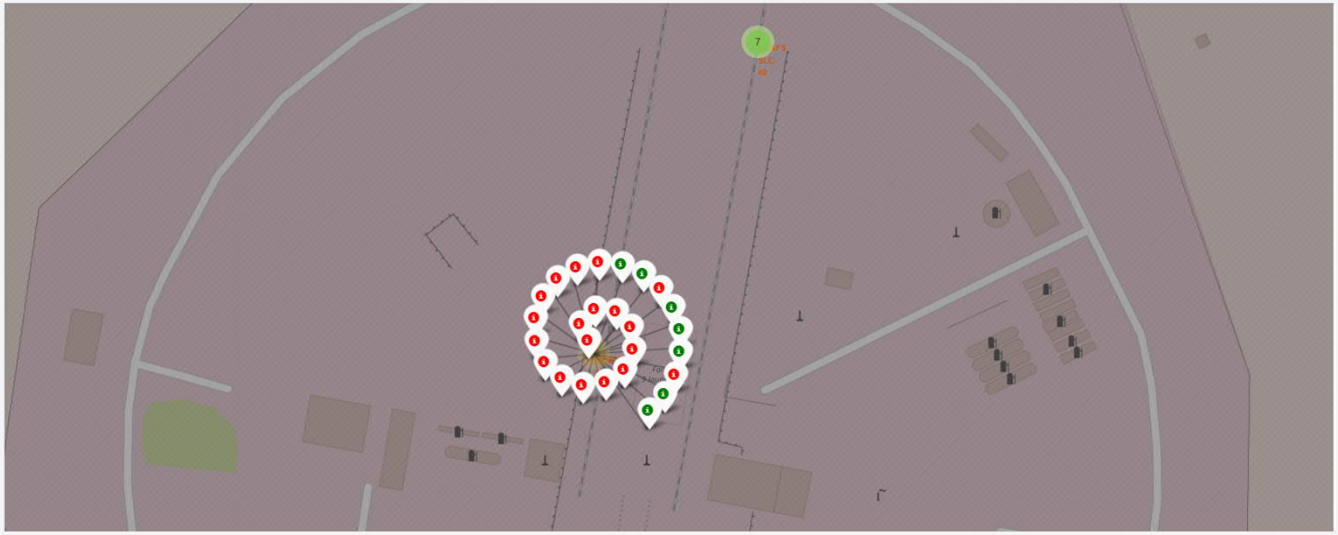


Launch Outcomes

At each launch site

- Green: successful launches
- Red: unsuccessful launches

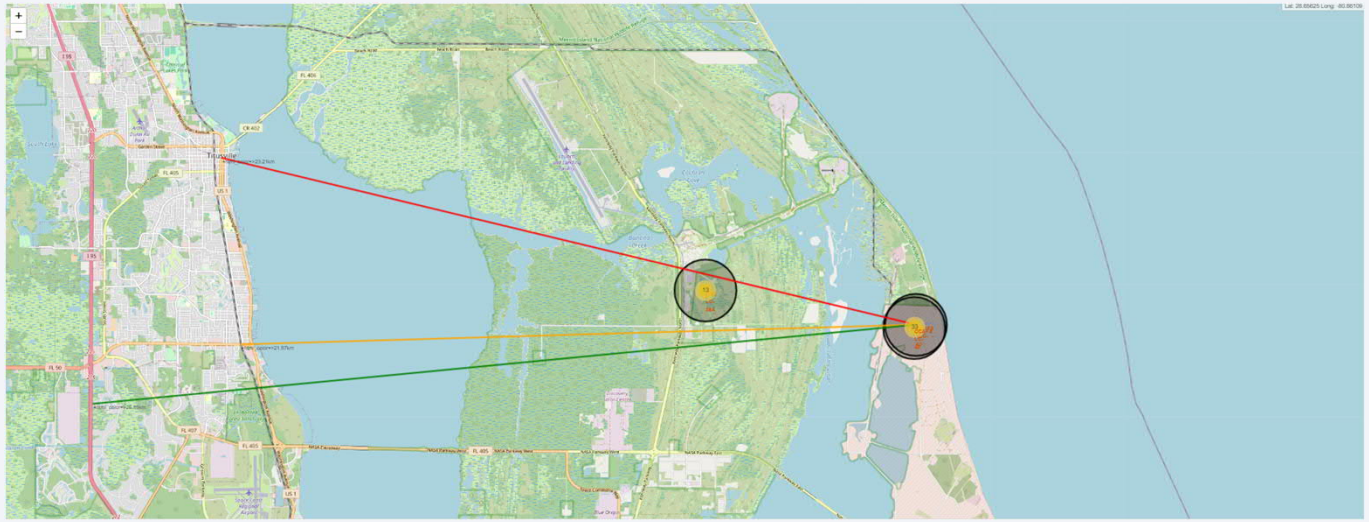
For the CCAFS SLC-40 site, 7 successful and 19 failure launches

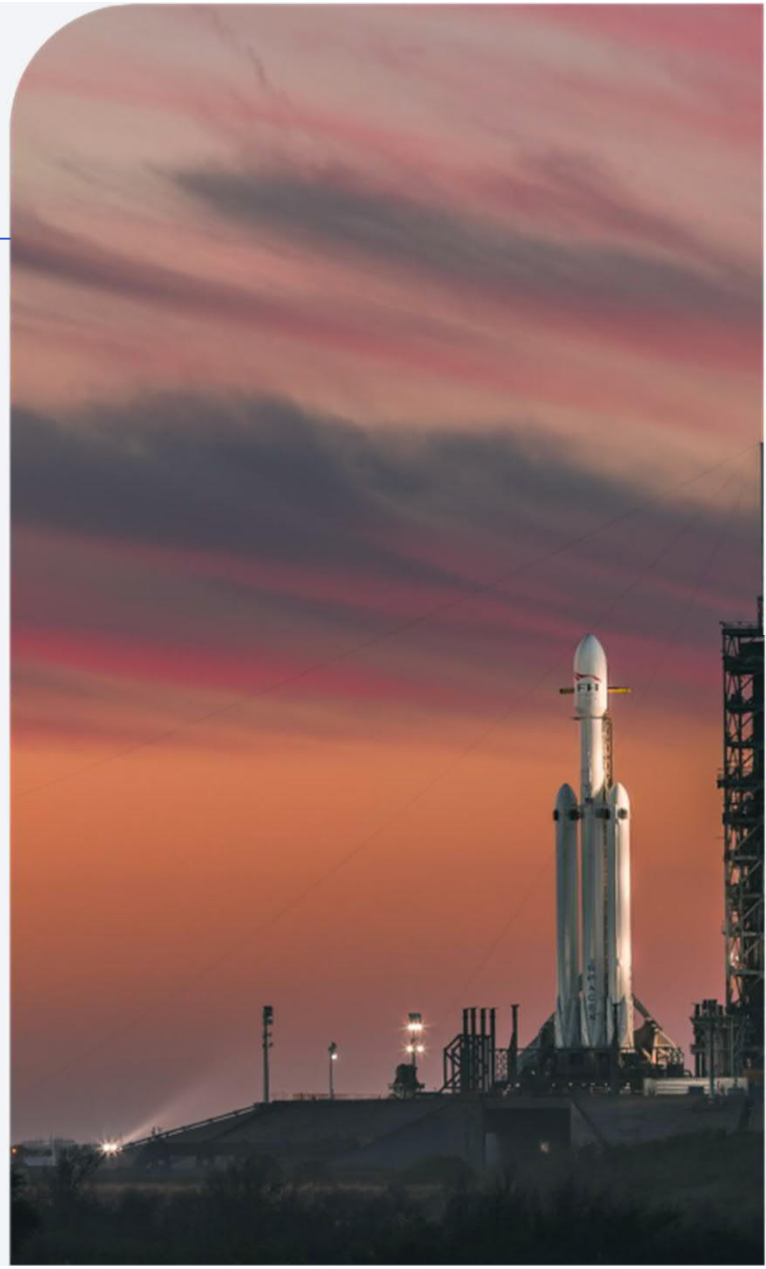


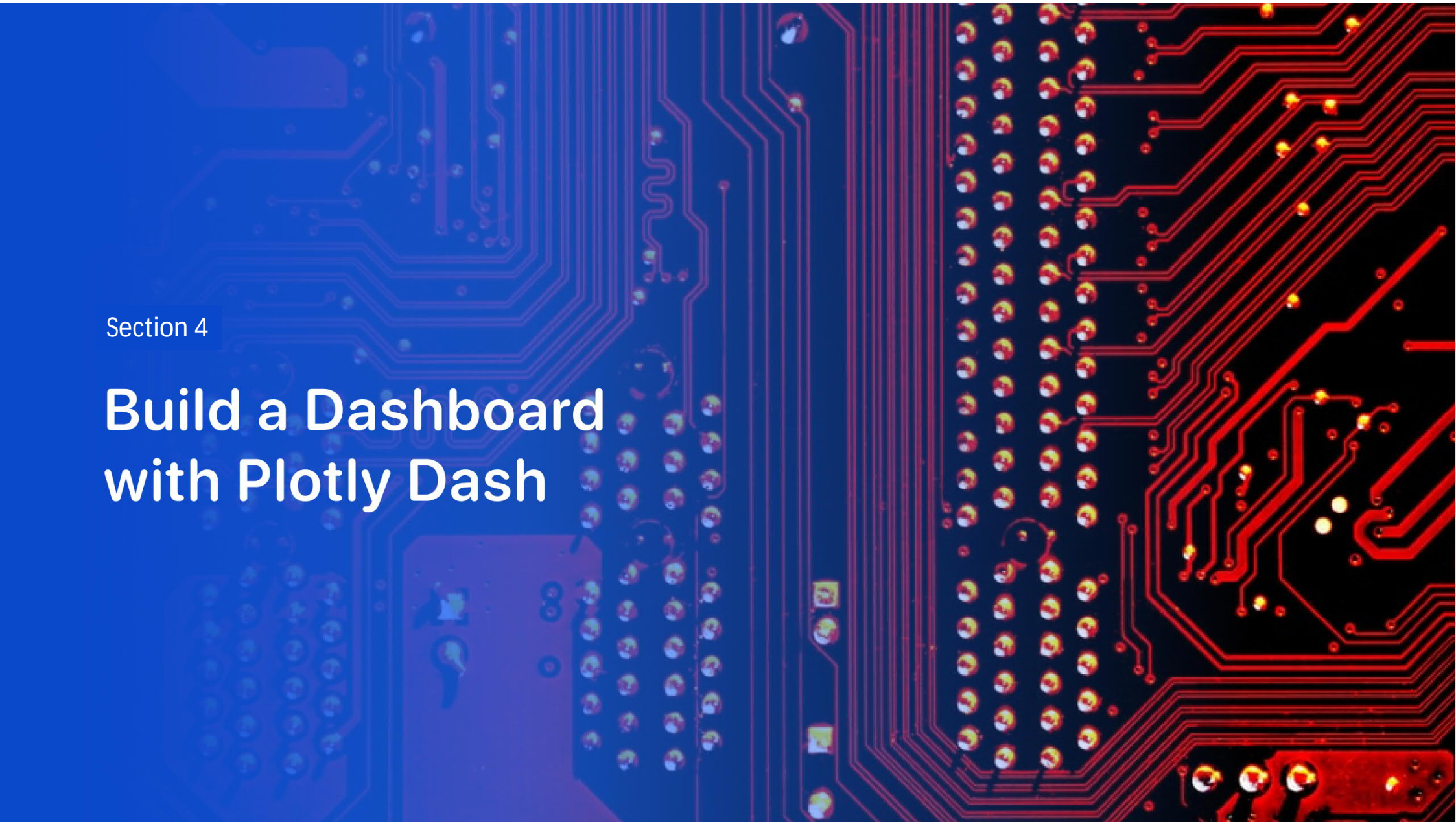
Distance to Proximities

CCAFS SLC-40

- To nearest coastline: 0.86km
- To nearest railway: 21.96km
- To nearest city: 23.23km
- To nearest highway: 26.88km



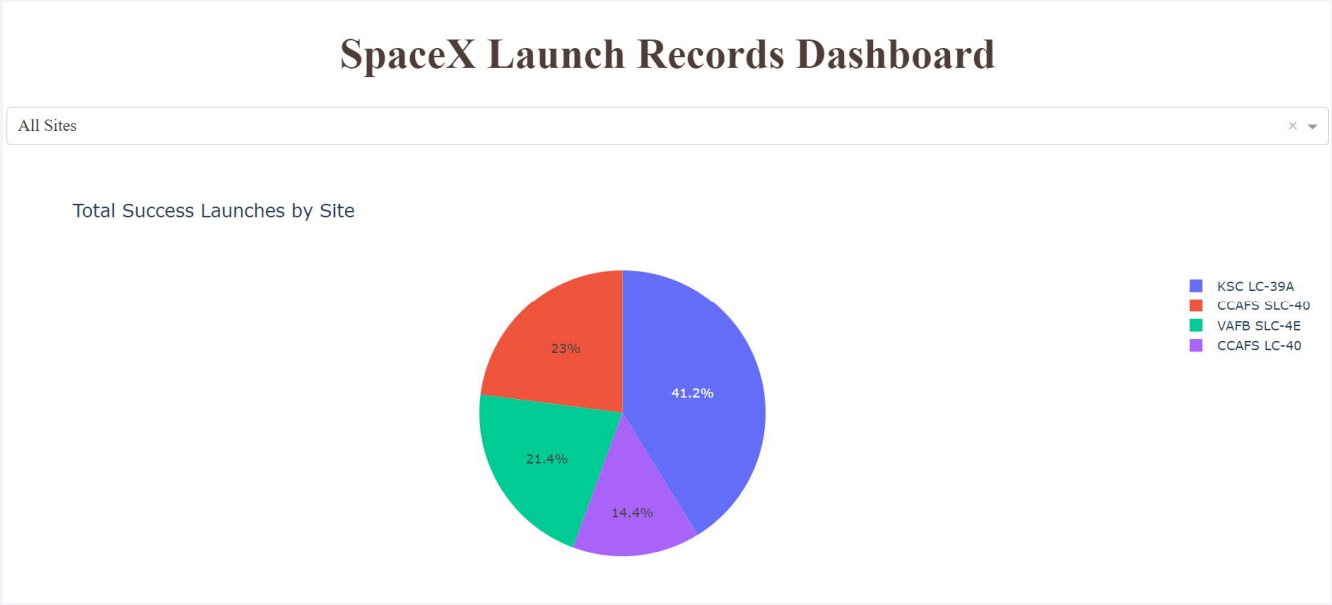



Section 4
Build a Dashboard with Plotly Dash

LAUNCH SUCCESS BY SITE

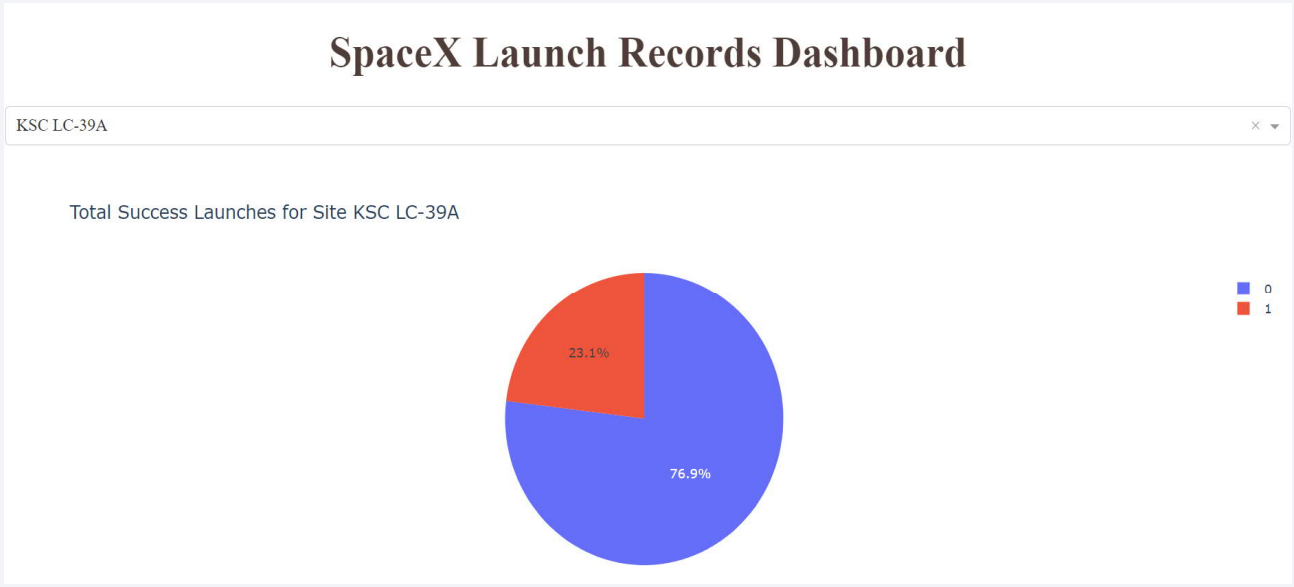
Total success launches by site

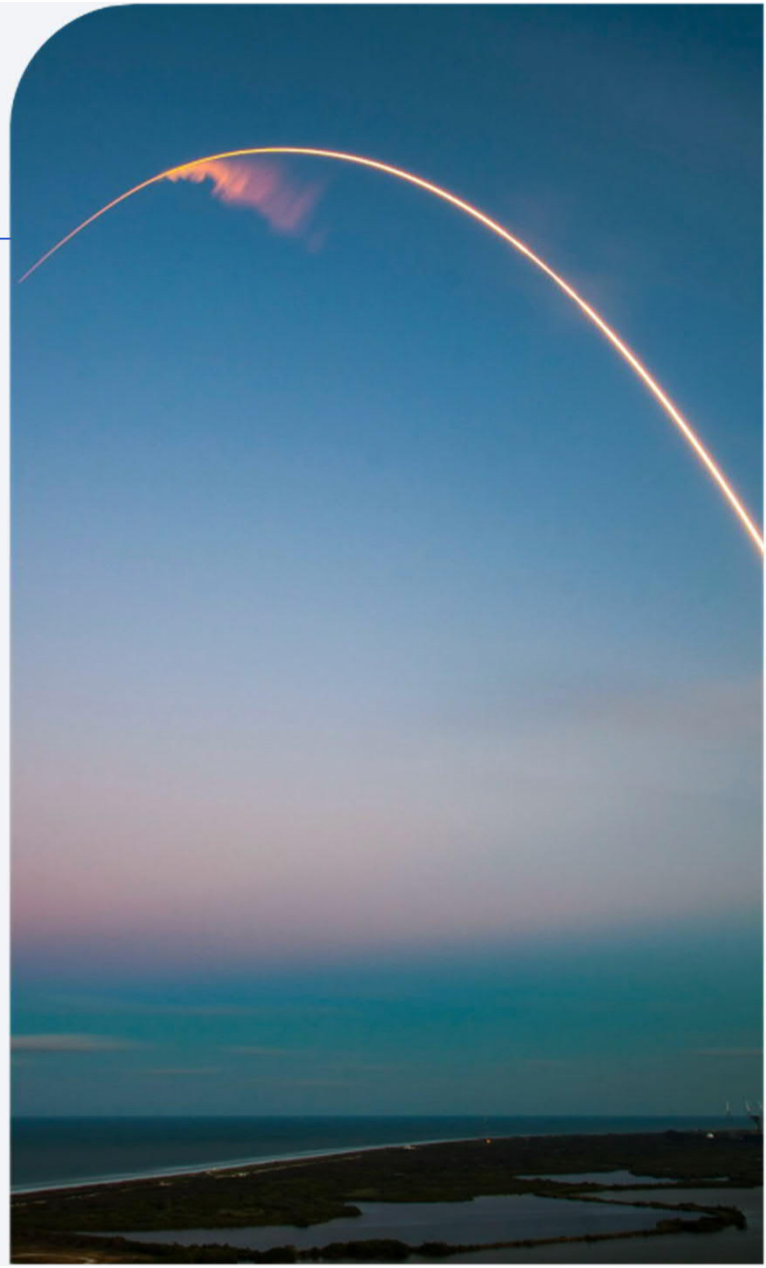
- KSC LC-39A has the most successful launches amongst launch sites (41.2%)



LAUNCH SUCCESS (KSC LC-29A)

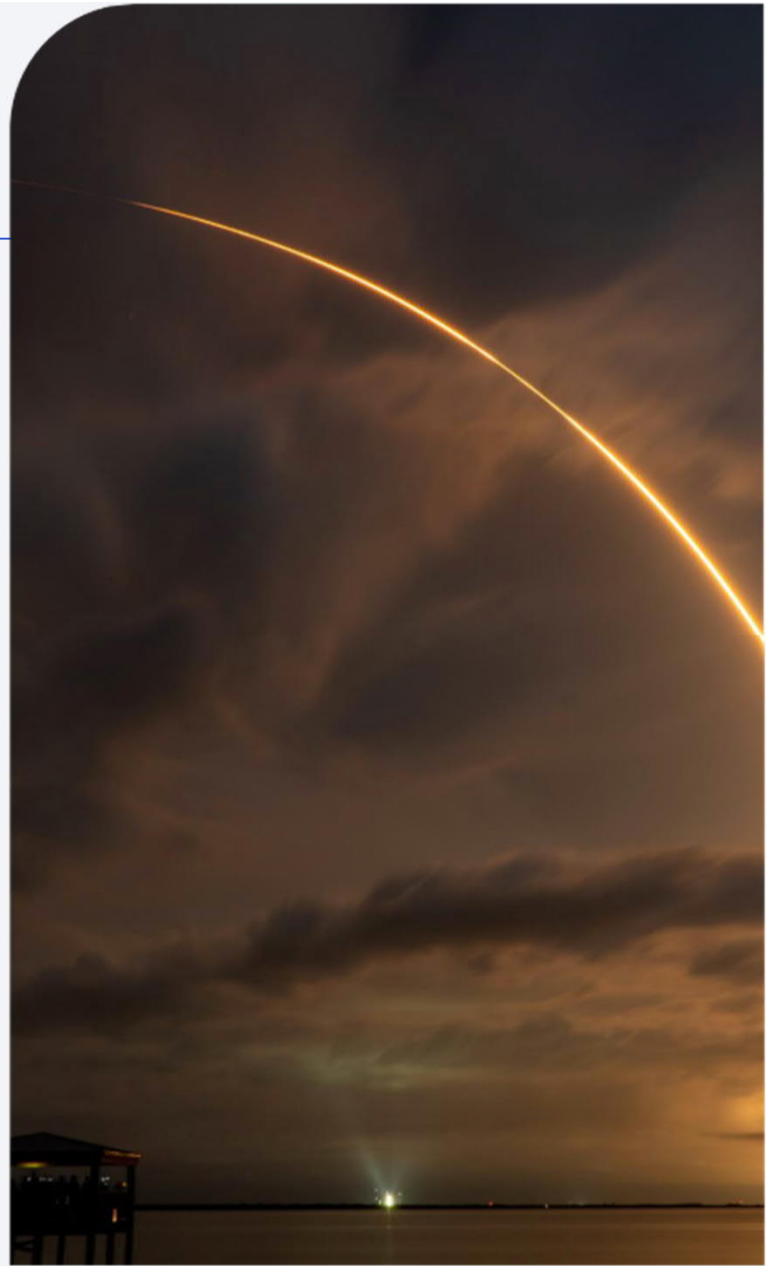
- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches





PAYLOAD MASS AND SUCCESS

- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome



Section 5

Predictive Analysis (Classification)

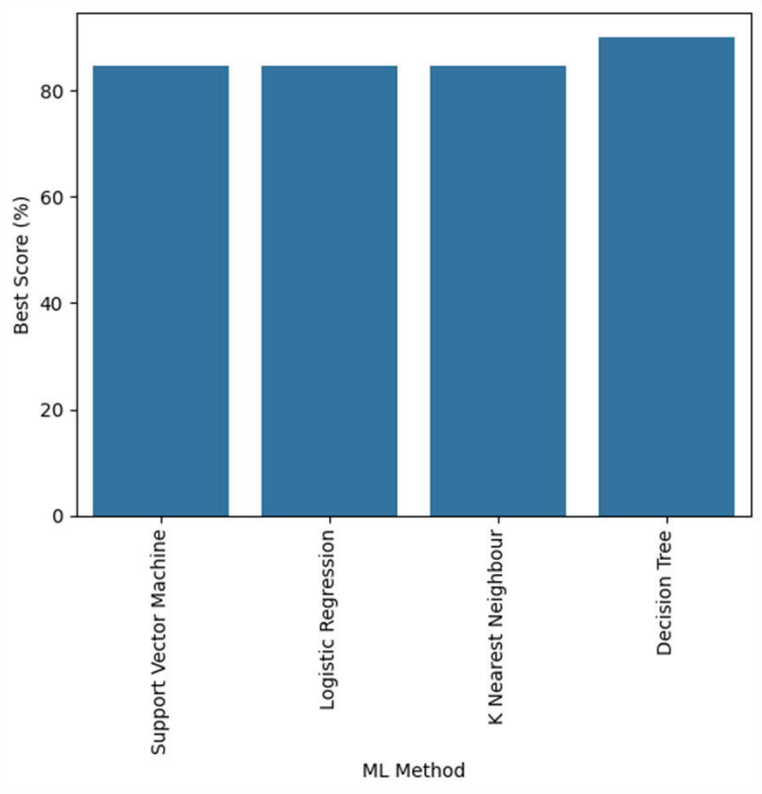
Classification Accuracy

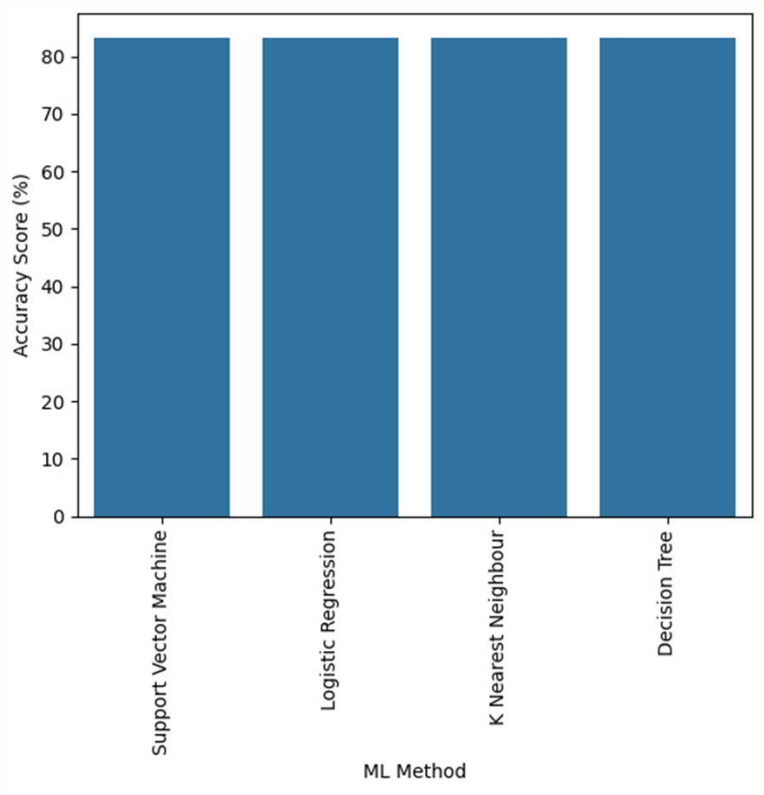
Feature	Accuracy Score	Best Score
Definition	Proportion of correct predictions	Highest achievable accuracy
Calculation	Based on actual performance	Theoretical maximum
Interpretation	Measures model's performance	Benchmark for comparison

- Accuracy = (Number of correct predictions) / (Total number of predictions)
- In some machine learning libraries, like scikit-learn, the "best score" often refers to the highest score achieved during cross-validation or grid search processes.

Classification Accuracy

Best score: Decision Tree model slightly outperformed





Accuracy score: all models perform at about same level

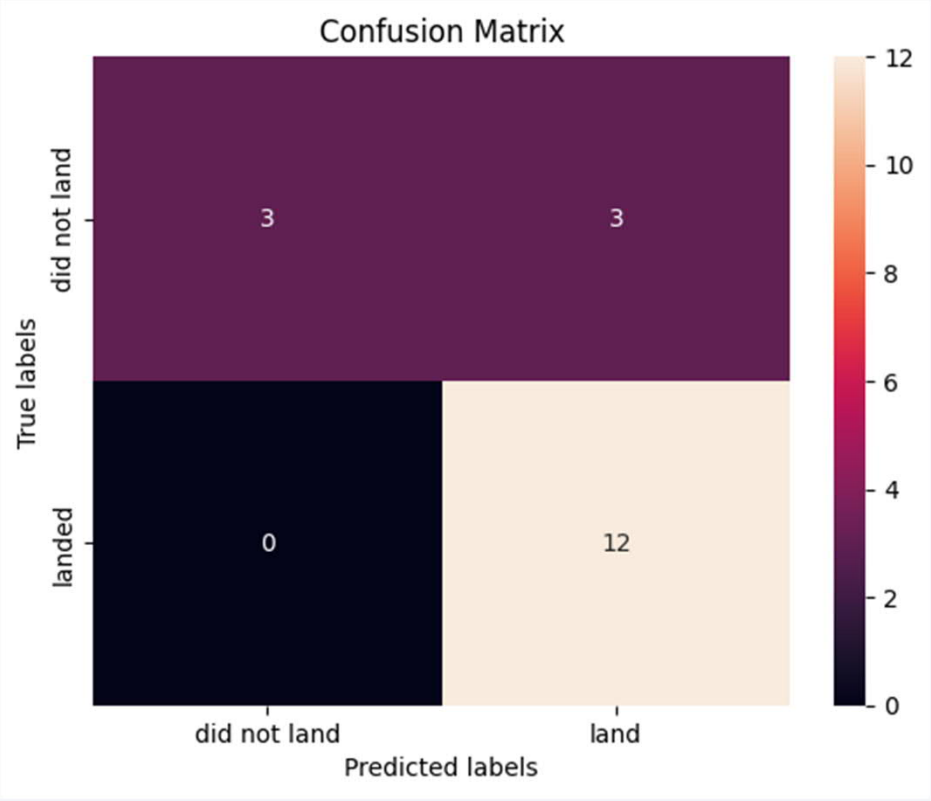
Confusion Matrix of the Decision Tree model

- True Positives (TP): 12. The model correctly predicted that 12 instances would land.
- True Negatives (TN): 3 - The model correctly predicted that 3 instances would not land.
- False Positives (FP): 3 - The model incorrectly predicted that 3 instances would land.
- False Negatives (FN): 0 - The model incorrectly predicted that 0 instances would not land

Accuracy: (TP + TN) / (TP + TN + FP + FN) = 0.75
The model correctly predicted 75% of the cases.
Precision: TP / (TP + FP) = 0.8
When the model predicted "land," it was correct 80% of the time.
Overall Assessment:

- The model performs well in predicting "landed" outcomes (high recall)
- It's less accurate in predicting "did not land"

=> The model is more confident in predicting successful landings but less so in predicting failures.



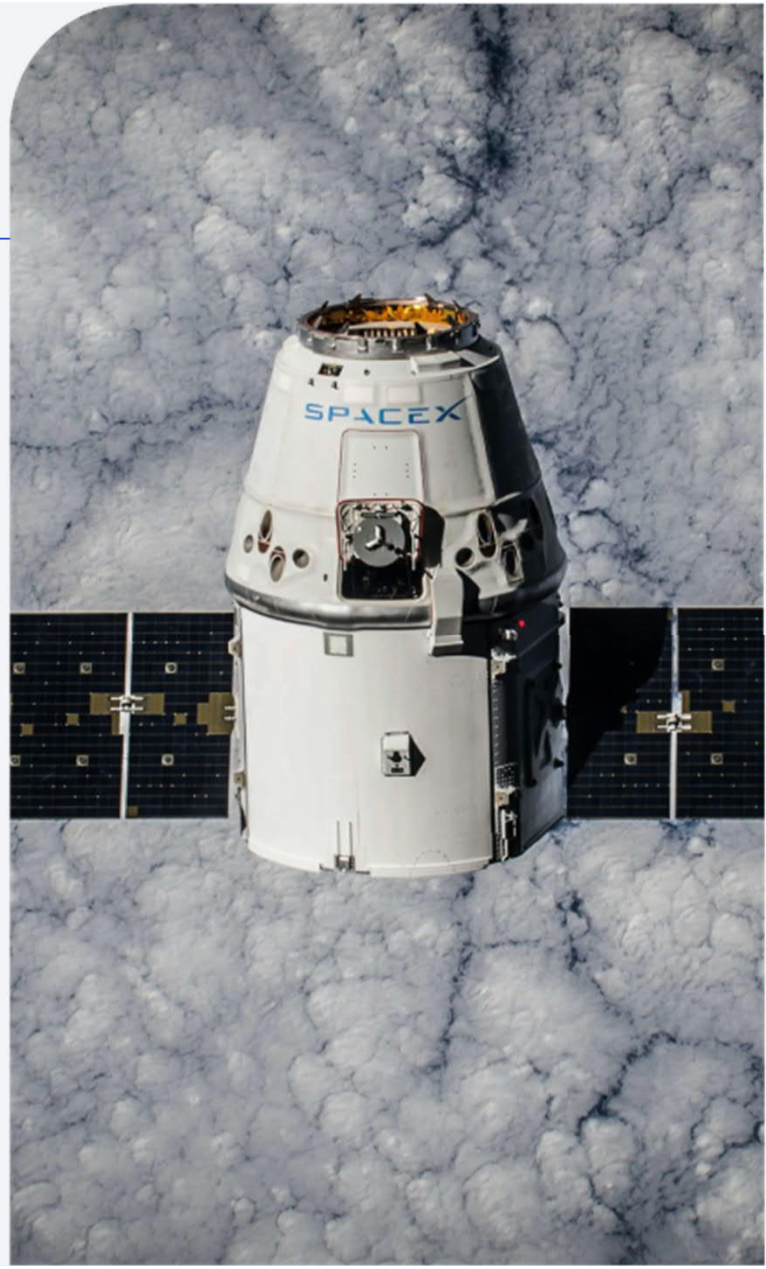
Conclusions

This study:

- Location of launch sites: near equator and coastline
- ES-L1, GEO, HEO and SSO orbits have high success rate (100%) based on the limited number of launch missions to those orbits
- KSC LC-39A has the highest success rate among launch sites, especially for mission with payload mass lighter than 5,500kg
- Model performance: The models performed similarly on the test set with the Decision Tree model slightly outperformed on best_score

Things to consider:

- Dataset: limited records over time
- Limited features have been collected and analysed
- Application of more model for higher accuracy score



Thank you!