IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

M Irfan Avianto
January 2026

Outline

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

Executive Summary

- Landing success is strongly influenced by mission characteristics

Launch site, orbit type, and payload mass significantly affect recovery outcomes.

- Operational performance improved over time

Success rates increased year-over-year, reflecting technological and procedural learning.

- Geographic factors matter

Proximity to coastlines and safe landing zones plays a role in recovery feasibility.

- Machine learning can predict landing success with good accuracy (~83%)

Models were able to distinguish between successful and failed landings using only pre-launch variables.

- Best-performing model provides a practical decision-support tool

Predictive insights can support mission planning, cost estimation, and risk assessment.

Introduction

Space Y is a private aerospace company aiming to make space travel more affordable and accessible. A major factor influencing launch cost is rocket reusability. SpaceX has significantly reduced launch expenses by successfully recovering and reusing the Falcon 9 first-stage booster.

In this project, SpaceX launch data is analyzed to better understand the factors that influence first-stage landing success. By predicting whether the first stage will land successfully, Space Y can estimate the potential for booster reuse, which directly impacts launch cost efficiency and pricing strategy.

Project Objective: Develop a predictive model to estimate the likelihood of first-stage landing success using historical SpaceX launch data.

Section 1

Methodology

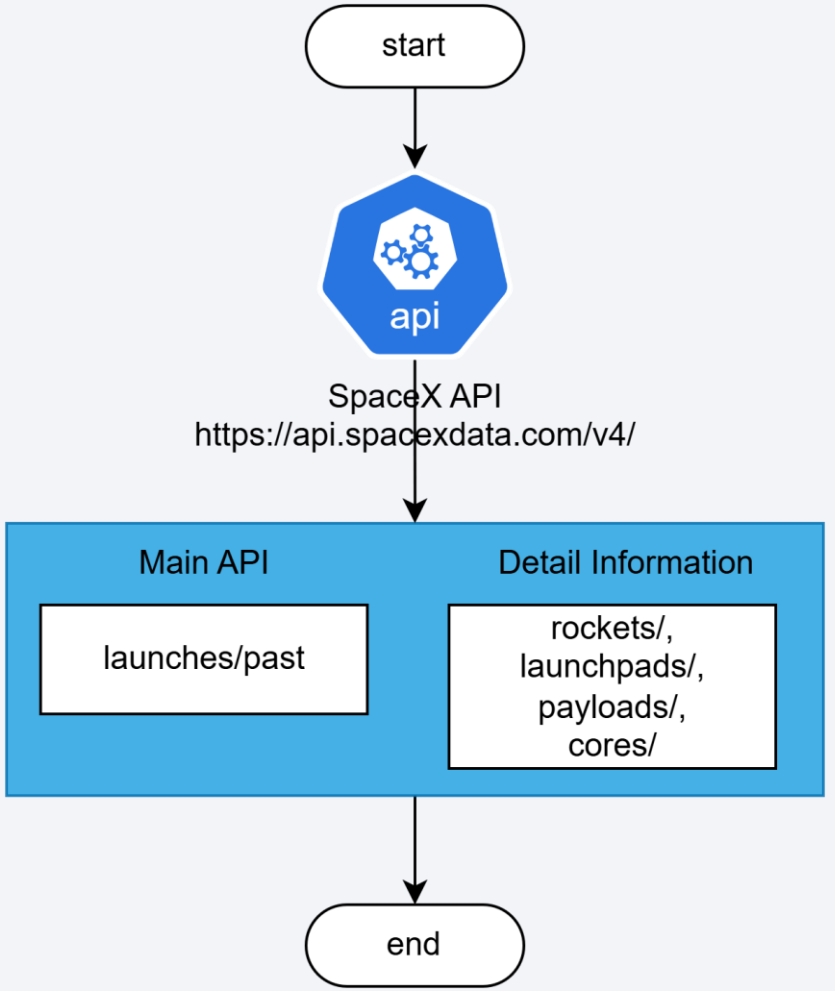
Methodology

Executive Summary

- Data collection methodology:
  - Data collected from SpaceX Rest API, SQL Server and web scrapping from Wikipedia
- Perform data wrangling
  - Performing ETL (extract, transform, and load) to create high-quality data prior to training.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Develop several classification models, then tune and evaluate their performance using the appropriate metrics for this case, such as accuracy

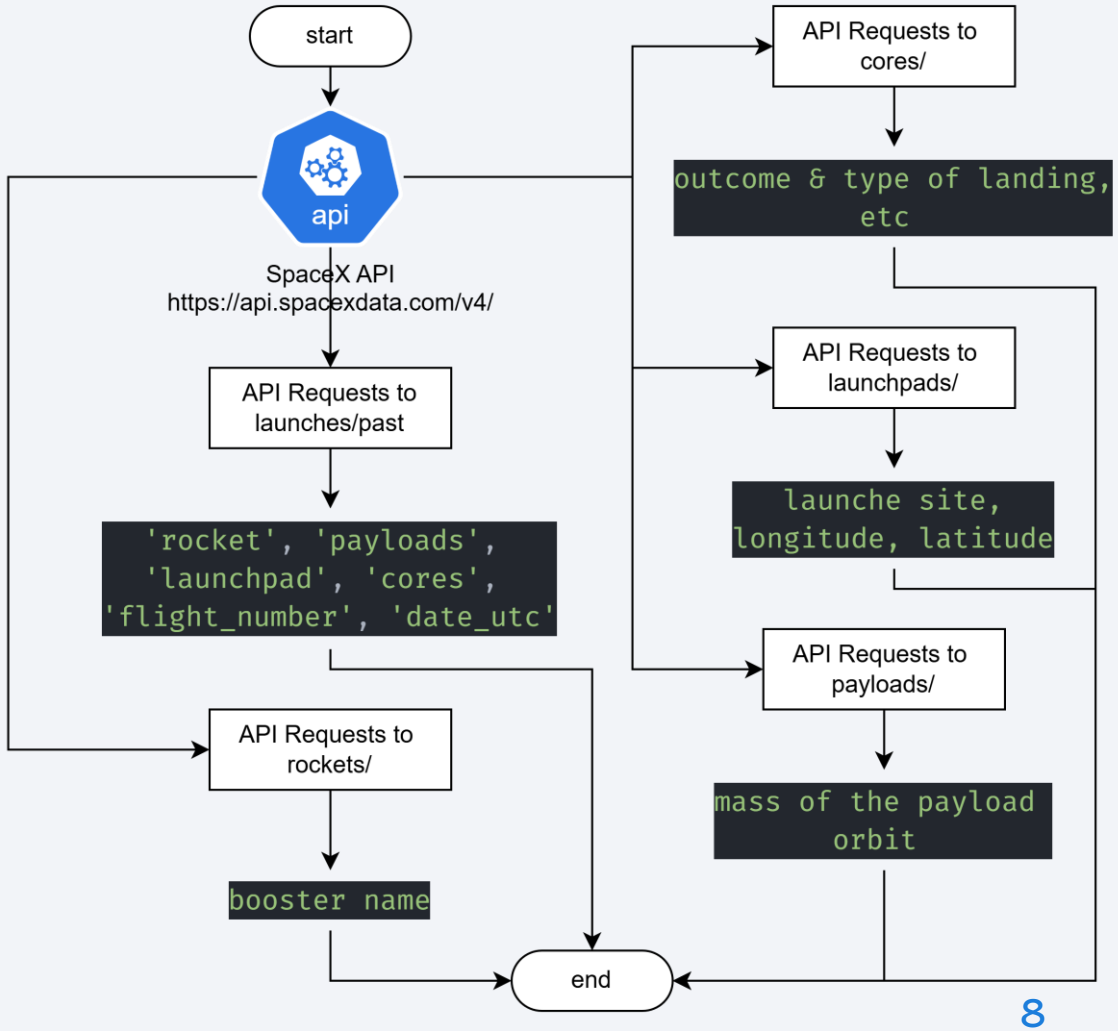
Data Collection

Data collected using SpaceX API at https://api.spacexdata.com/v4/launches/past as main API. Detailed information from rockets/, launchpads/, payloads/, cores/ endpoint also included to gain more information and create appropriate dataset for modeling.



Data Collection – SpaceX API

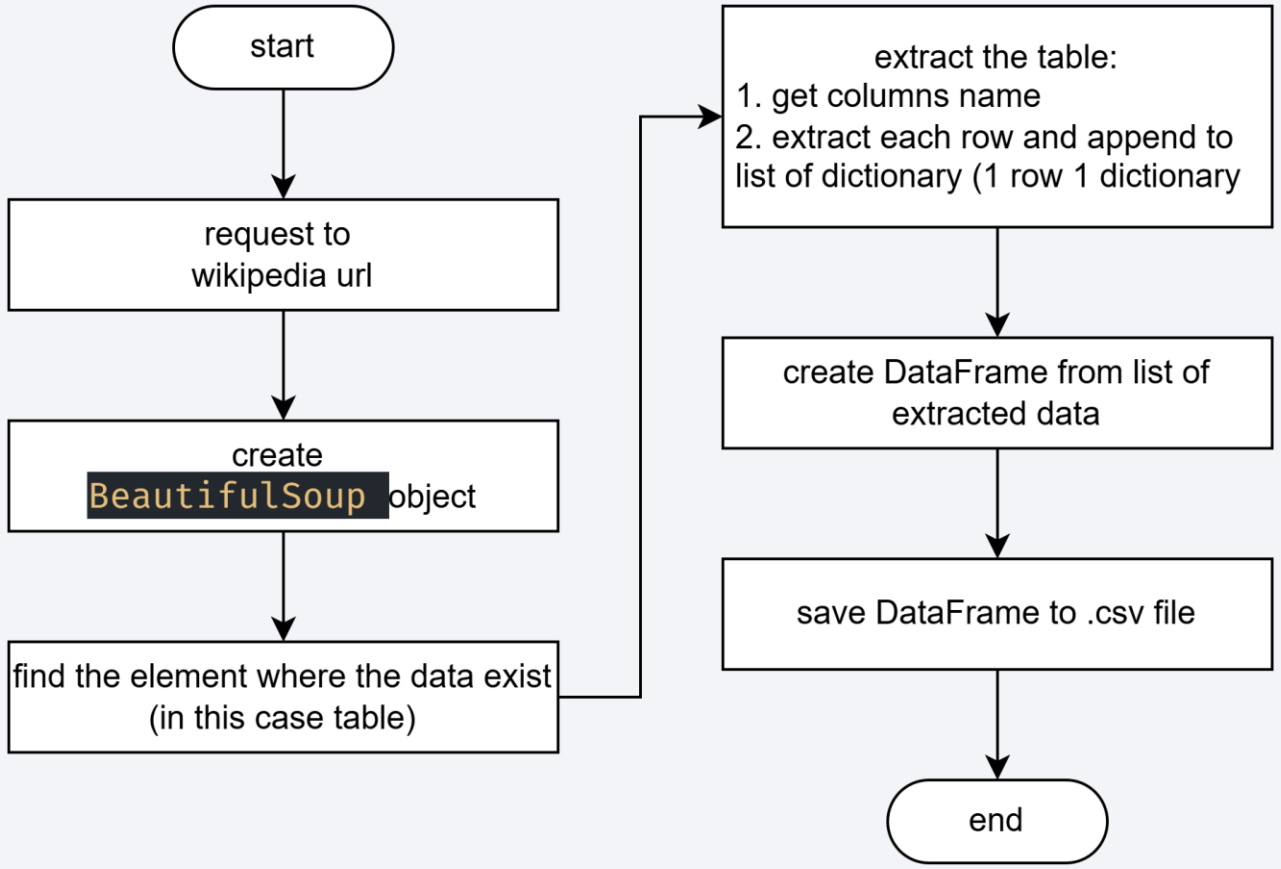
- From the rocket we would like to learn the booster's name
- From the payload we would like to learn the mass of the payload and the orbit that it is going to
- From the launchpad we would like to know the name of the launch site being used, the longitude, and the latitude.
- From cores we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether grid fins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core which is a number used to separate version of cores, the number of times this specific core has been reused, and the serial of the core.
- [GitHub URL]



Data Collection - Scraping

- Data crawled from Wikipedia

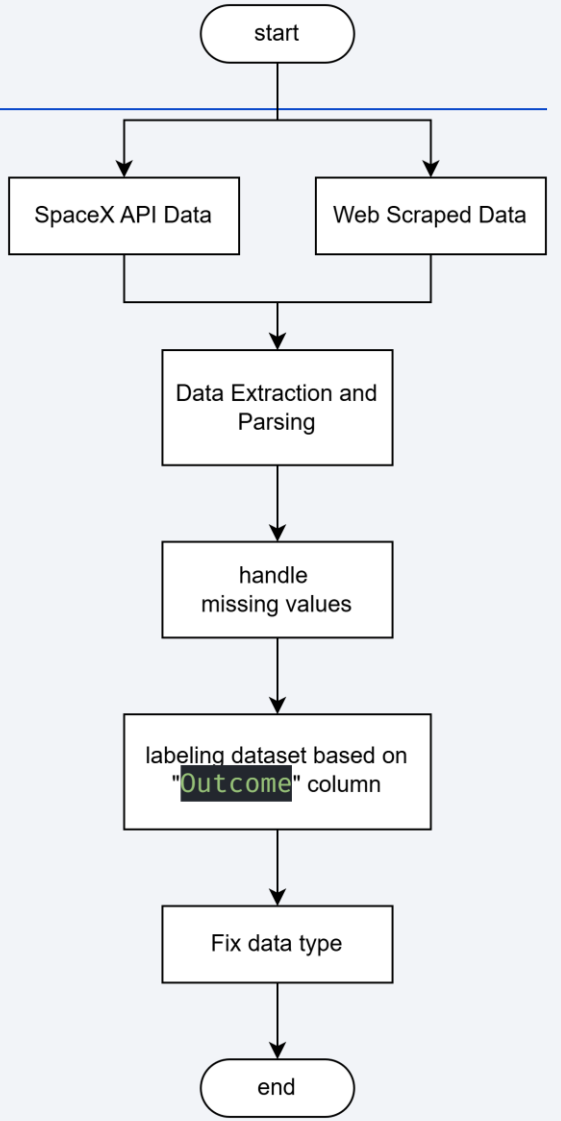
- GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Data Wrangling

- The data wrangling process involved collecting launch data from the SpaceX API and supplementary sources through web scraping. The raw datasets were converted into structured Pandas DataFrames for processing. Data cleaning steps included handling missing values, removing duplicate records, correcting data types, and standardizing column formats.

- Jupyter Notebook available on GitHub



EDA with Data Visualization

- Bar Charts – Launch Success Rate by Category

Used to compare landing success rates across categorical variables such as orbit type. These charts help identify which conditions are associated with higher or lower recovery success.

- Scatter Plots – Payload Mass vs Landing Outcome

Used to explore the relationship between payload mass and first-stage landing success. This helps assess whether heavier payloads reduce the likelihood of recovery due to fuel and trajectory constraints.

- Jupyter Notebook available on GitHub

EDA with SQL

- SQL queries were used to perform structured exploratory analysis directly on the launch dataset. Aggregation and grouping functions were applied to calculate launch frequencies, payload statistics, and landing success rates across different sites, orbits, and years. Filtering and conditional queries enabled comparisons between successful and failed landings. This SQL-based exploration provided high-level operational insights and helped identify key variables influencing first-stage recovery outcomes.
- Jupyter Notebook available on GitHub

Build an Interactive Map with Folium

- **Markers:** Added markers to represent each launch site location. These help identify where launches are geographically distributed.
- **Circle Markers:** Used circle markers to visualize launch outcomes, with different colors indicating successful or failed landings. This allows quick visual comparison of recovery performance across sites.
- **Popups:** Attached popups to markers displaying key launch details such as launch site name, mission outcome, and payload mass. This provides interactive access to detailed information without cluttering the map.
- **Lines (Distance Lines):** Drew lines between selected launch sites and nearby infrastructure (e.g., coastline, highways, railways) to show proximity and logistical advantages. Distance labels were added to quantify how close these features are.
- GitHub URL

Build a Dashboard with Plotly Dash

- An interactive dashboard was developed using Plotly Dash to enable dynamic exploration of launch performance data. The dashboard includes a pie chart summarizing overall landing success rates, a bar chart comparing outcomes by launch site, a scatter plot examining the relationship between payload mass and landing success, and a line chart showing performance trends over time.
- Interactive filters such as a launch site dropdown and a payload mass range slider allow users to customize the analysis and observe how mission characteristics influence recovery outcomes. This interactive approach transforms static analysis into a decision-support tool that helps stakeholders better understand operational patterns and factors affecting rocket reusability.
- GitHub url

Predictive Analysis (Classification)

Multiple classification algorithms were trained and evaluated to predict first-stage landing success. Although all models produced similar performance metrics on the small test dataset, cross-validation and theoretical generalization considerations were used to guide model selection.

Support Vector Machine (SVM) was selected as the final model because of its ability to create a robust decision boundary and reduce overfitting risk in small datasets. The model demonstrated stable performance during cross-validation and maintained balanced error types, making it suitable for predicting landing outcomes that influence cost estimation and launch planning.

GitHub url

Results

- Payload mass impacts landing outcome

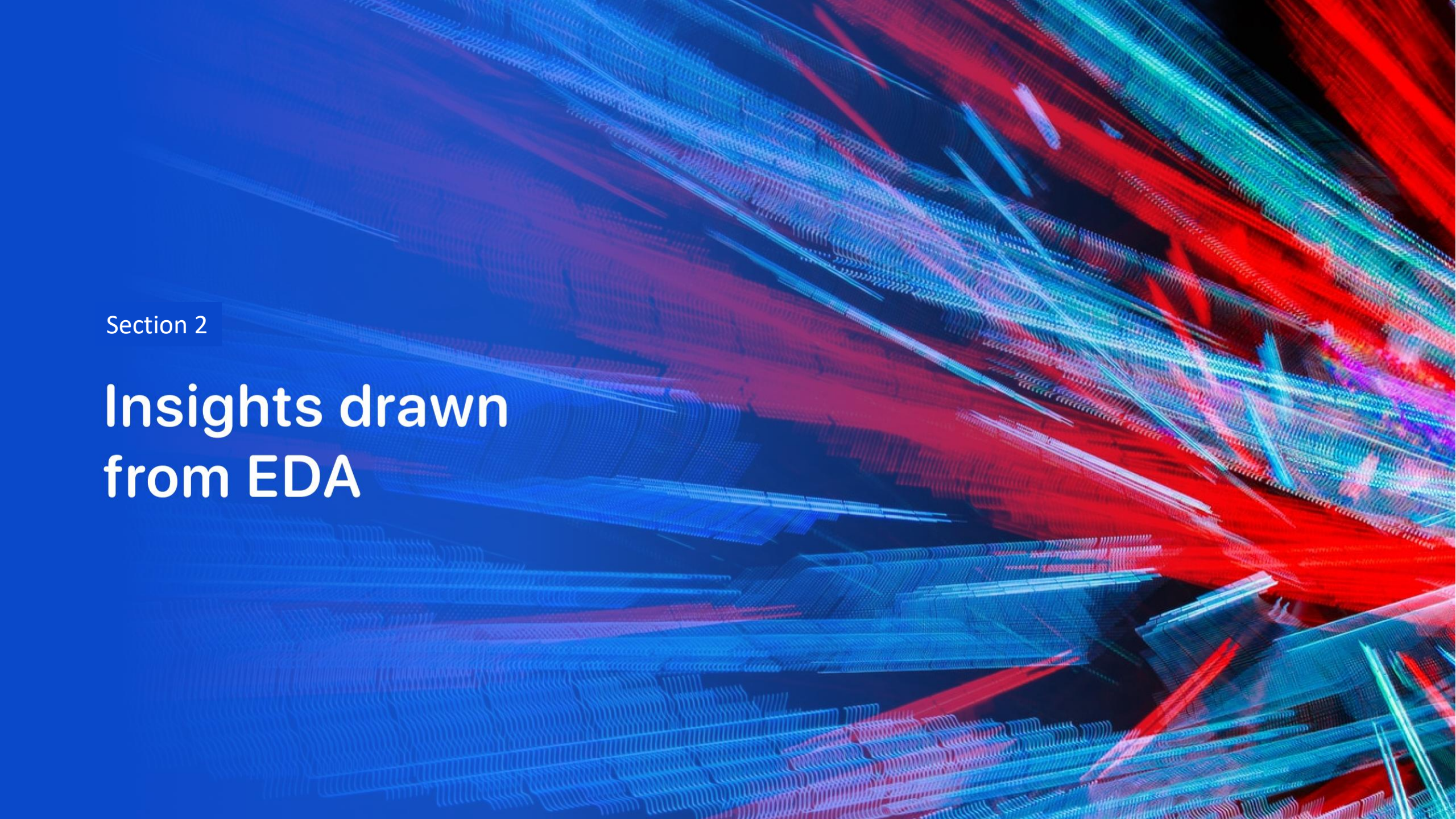
Very light and very heavy payloads had lower success probability, while mid-range payloads showed more stable landing performance.

- Orbit type correlates with landing success

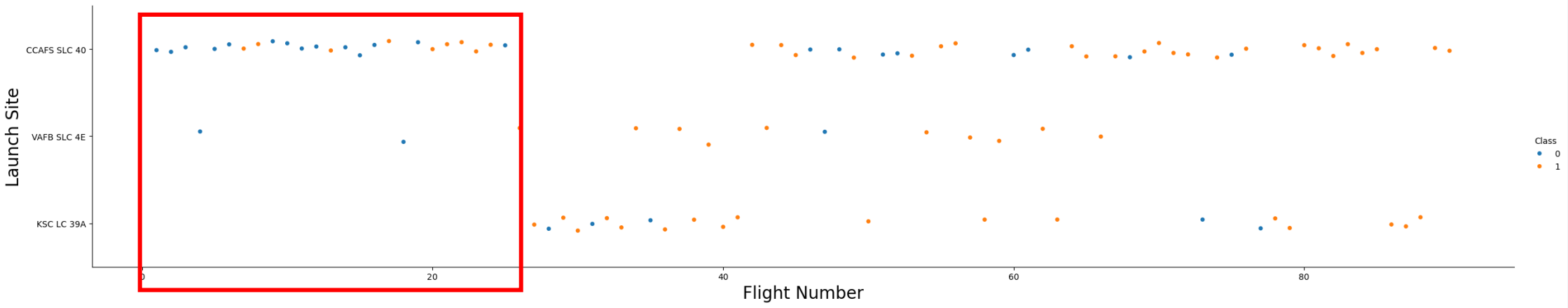
Specific orbit categories were associated with higher success rates, indicating mission profile affects recovery feasibility.

- Year-over-year improvement observed

Landing success rate improved over time, showing learning effects and technological refinement.

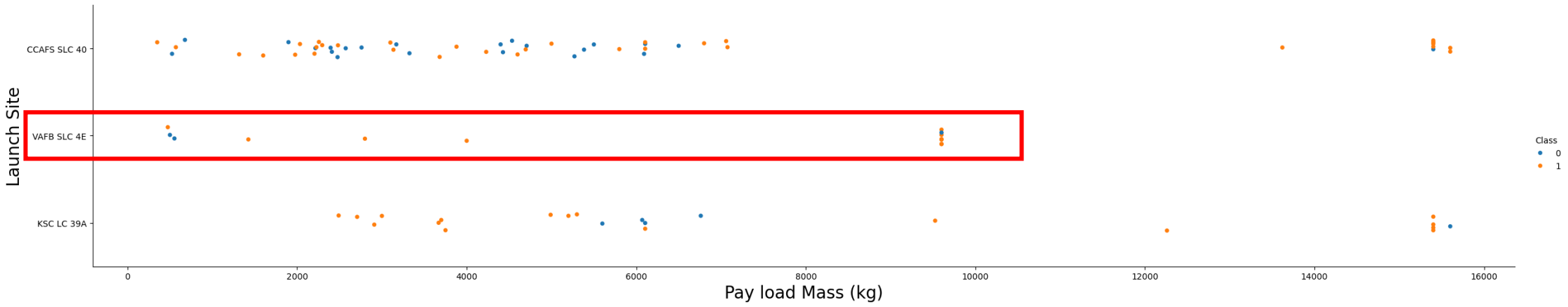

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



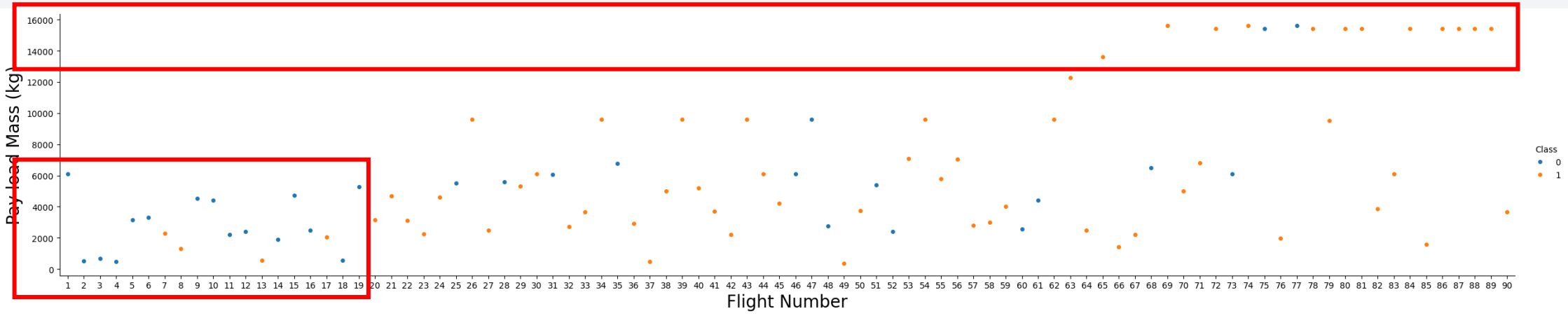
- The rocket experienced several landing failures during the early stages of flight. The majority of rocket launches after the 60th launch were successful.
- CCAFS SLC-40 has become most used launch site than the others, followed by KSC LC 39A and VAFB SLC 4E

Payload vs. Launch Site



- VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

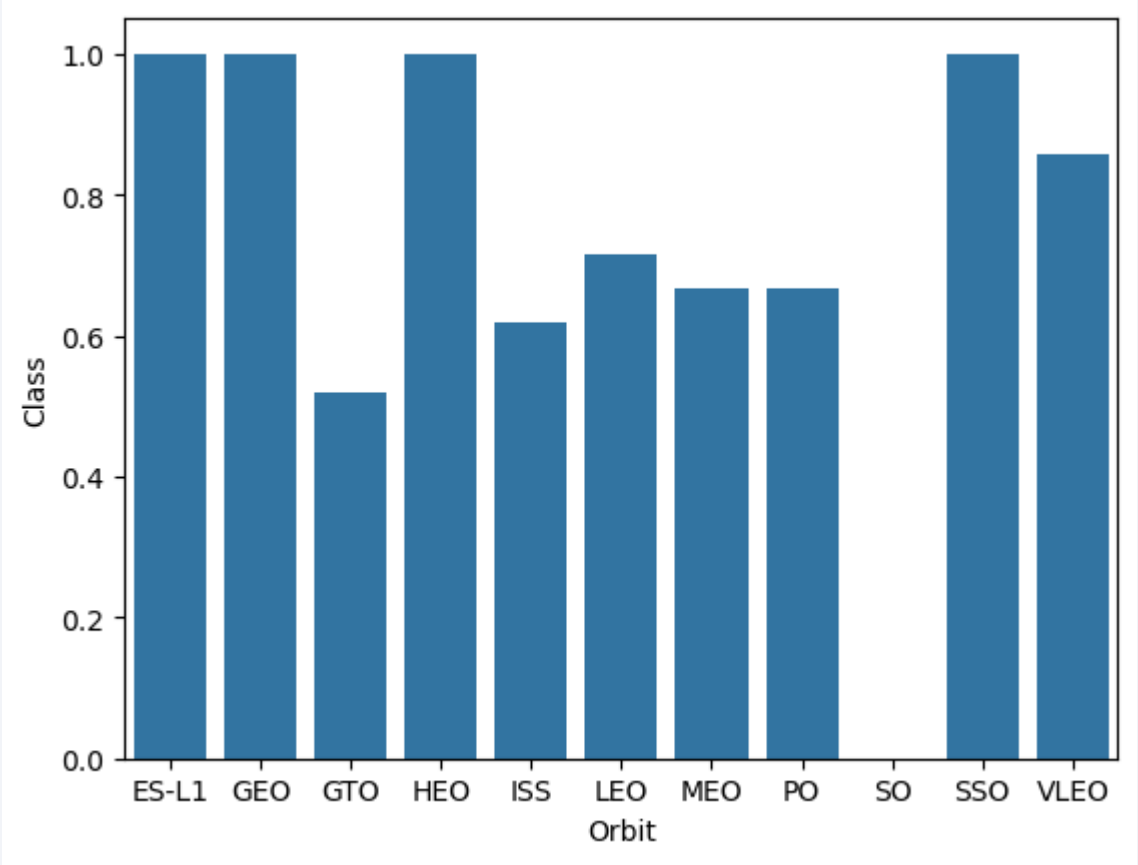
[Additional Analysis] Payload vs. Flight Number



- In the early days of rocket launches, the majority failed to land despite carrying light payloads. However, over time, rockets have successfully landed with payloads of more than 14,000 kg.

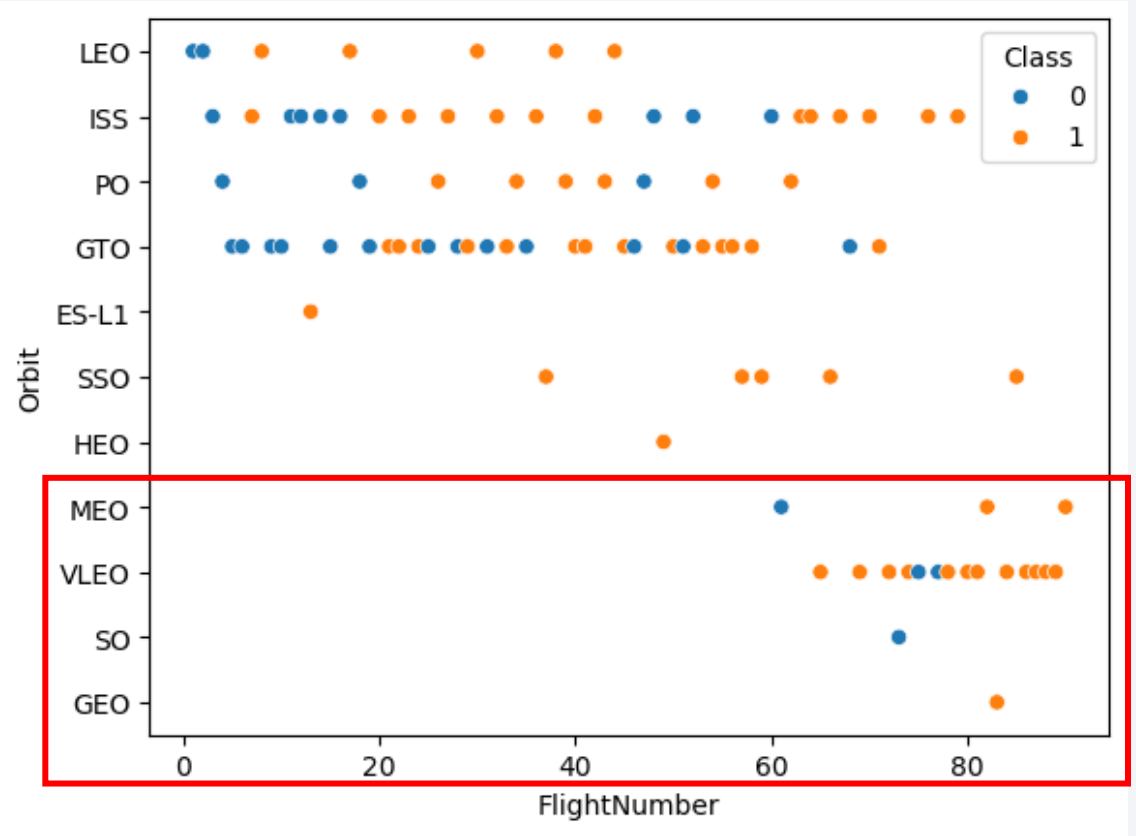
Success Rate vs. Orbit Type

- Rockets has 100% success rate to reach ES-L1, GEO, HEO, SSO and VLEO orbit type.
- Rockets always fail to reach SO Orbit.



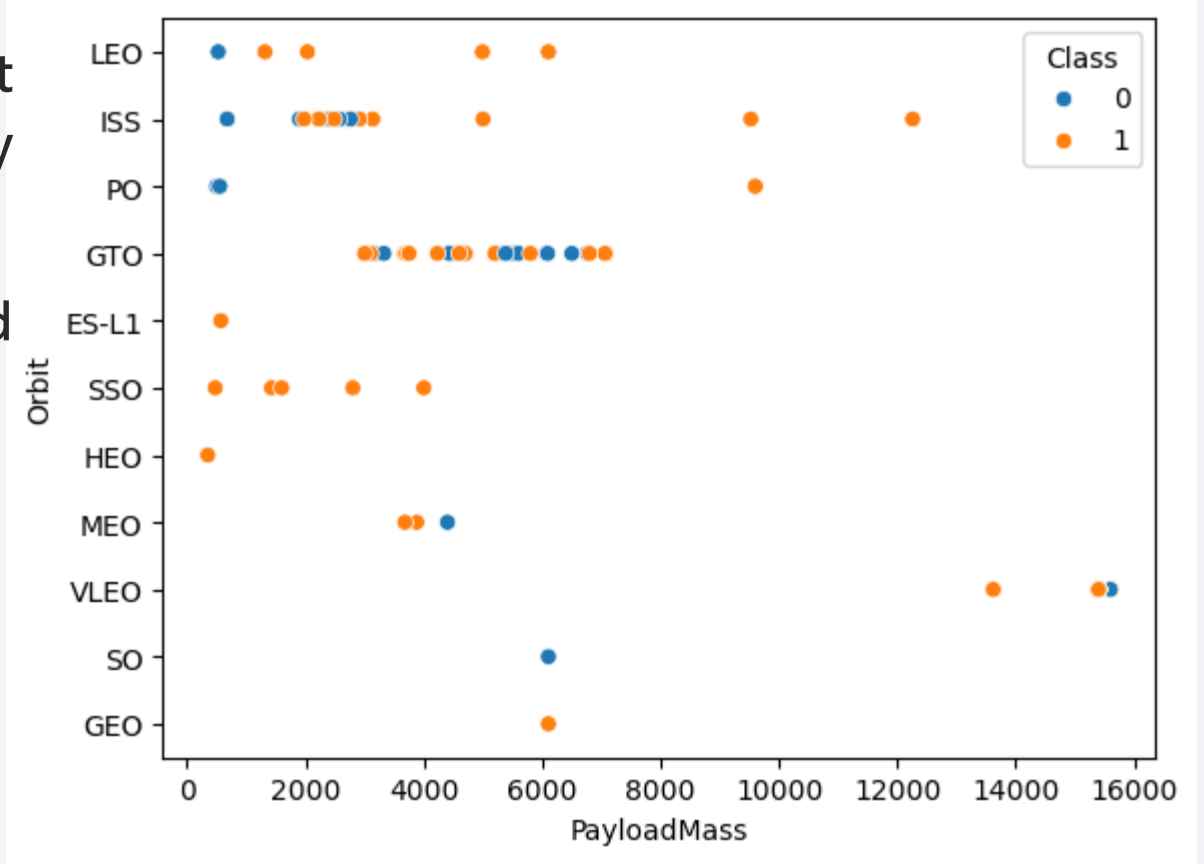
Flight Number vs. Orbit Type

- GTO, ISS and VLEO becomes the most frequently reached orbit.
- After 60th launches, the company trying to reach MEO, VLEO, SEO and GEO orbit.
- The rocket successfully landed twice after failing to reach MEO orbit on its first launch.
- The rocket successfully reach GEO orbit and landed at the first-time attempt



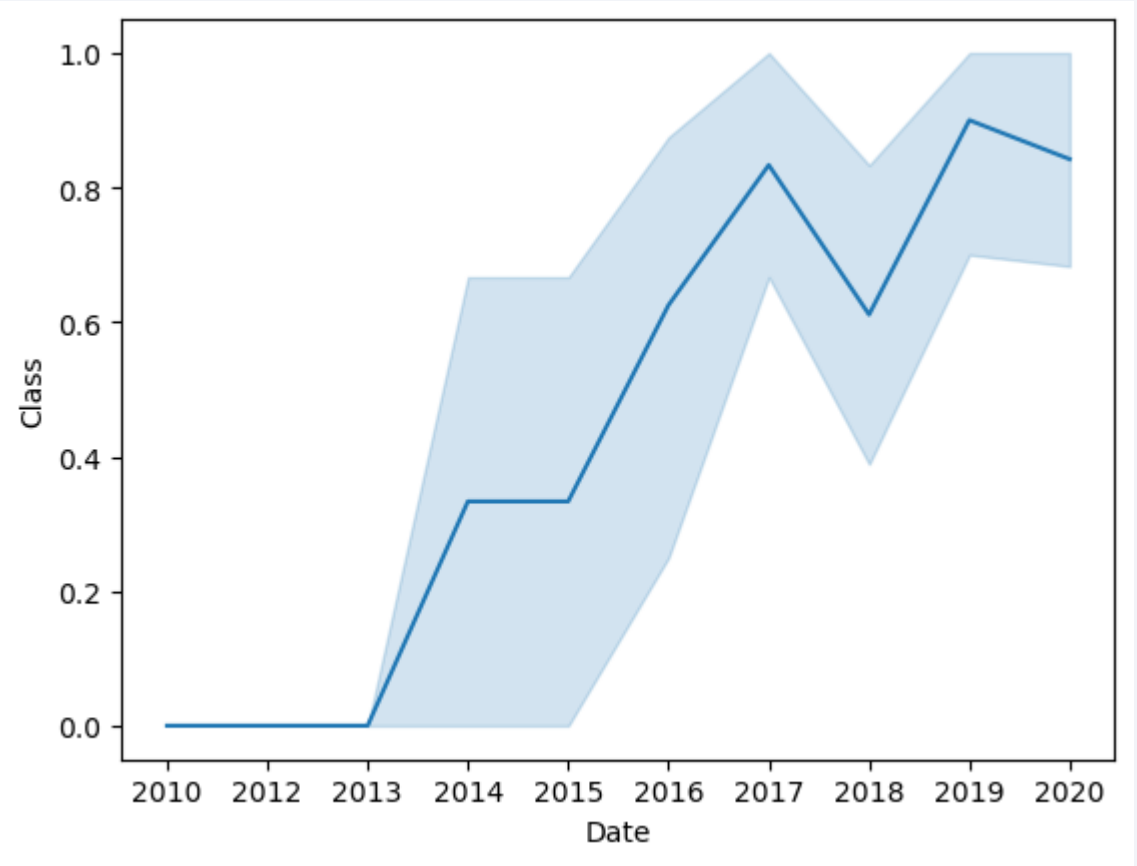
Payload vs. Orbit Type

- The rocket has 75% success rate (3 out of 4 attempt) when carrying heavy payload.
- The rocket always carrying heavy payload when traveled to VLEO orbit



Launch Success Yearly Trend

- In general, the success rate of rocket launches has increased year by year.
- The success rate for 2017-2018 has decreased significantly but rebounded in 2019.



All Launch Site Names

```
1  %sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
```

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

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

Query: SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
Instead of using “SELECT [column_name]”, use “SELECT DISTINCT [column_name]” to get unique value. From that query, we get 4 unique Launches Site

Launch Site Names Begin with 'CCA'

```
1  %sql select * from SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

Python

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

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

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

Total Payload Mass

```
1  %sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload FROM SPACEXTABLE WHERE "Customer"="NASA (CRS)"
```

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

total_payload
45596

- Query: `SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload FROM SPACEXTABLE WHERE "Customer"="NASA (CRS)"`
- Result: Total payload carried by boosters from NASA is 45,596Kg

Average Payload Mass by F9 v1.1

```
1  %sql select AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1"
```

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

AVG("PAYLOAD_MASS__KG_")
2928.4

Query:

```
SELECT AVG("PAYLOAD_MASS__KG_")
FROM SPACEXTABLE
WHERE "Booster_Version" LIKE "F9 v1.1"
```

- Result: the average payload mass carried by booster version F9 v1.1 is 2928.4Kg

First Successful Ground Landing Date

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

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

MIN("Date")
2015-12-22

Query:

```
SELECT MIN("Date")
FROM SPACEXTABLE
WHERE Landing_Outcome="Success (ground pad)"
```

- Result: December 22, 2015, was the first successful ground landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE
    (Landing_Outcome="Success (drone ship)") AND
    (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

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

```

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

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

- Calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
SELECT DISTINCT "Booster_Version"
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
  SELECT MAX("PAYLOAD_MASS__KG_“
) FROM SPACEXTABLE)
ORDER BY "Booster_Version"
```

```
1  %%sql
2  SELECT DISTINCT "Booster_Version"
3  FROM SPACEXTABLE
4  WHERE PAYLOAD_MASS__KG_ = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
5  ORDER BY "Booster_Version"
```

```
 * sqlite:///my_data1.db
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
- Result: In 2015, SpaceX failed twice to land on a drone ship

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

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

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

```
%%sql
select Landing_Outcome, count(*) as outcome_count
from SPACEXTBL
where
    date between '2010-06-04' and '2017-03-20'
    and "Landing_Outcome" like '%Success%'
group by Landing_Outcome
order by outcome_count desc
```

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

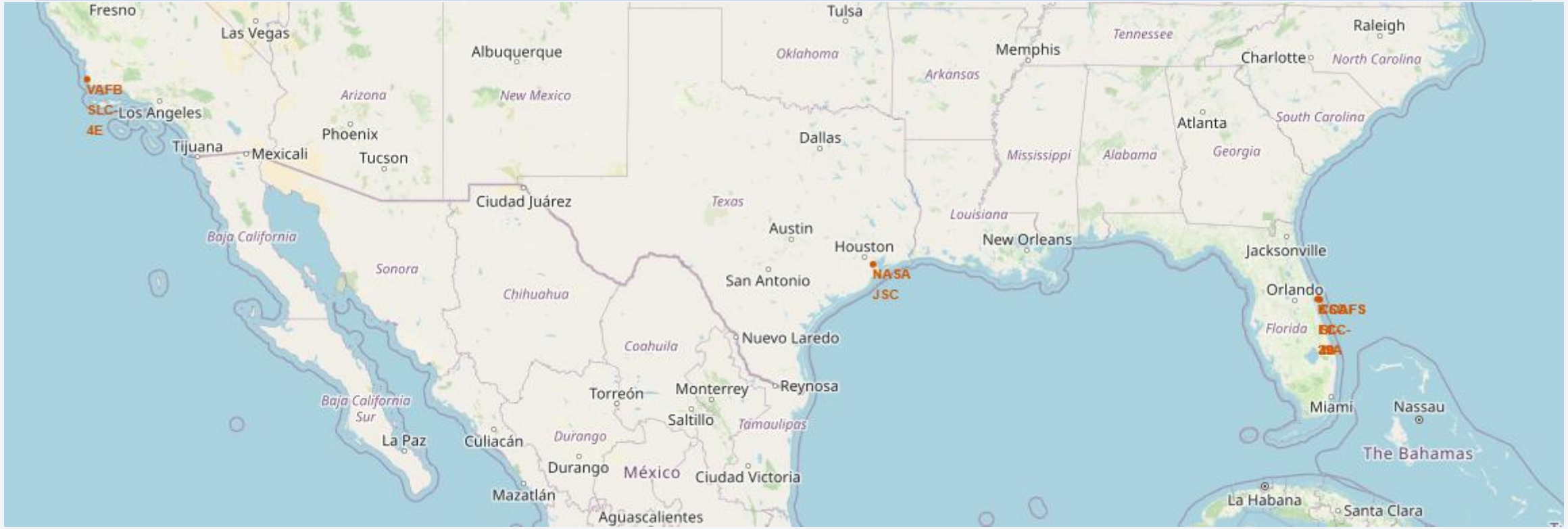
Landing_Outcome	outcome_count
Success (drone ship)	5
Success (ground pad)	3

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Between June 4, 2010, and March 20, 2017, SpaceX had 8 successful landings (5 on drone ships and 3 on ground pads).

Section 3

Launch Sites Proximities Analysis

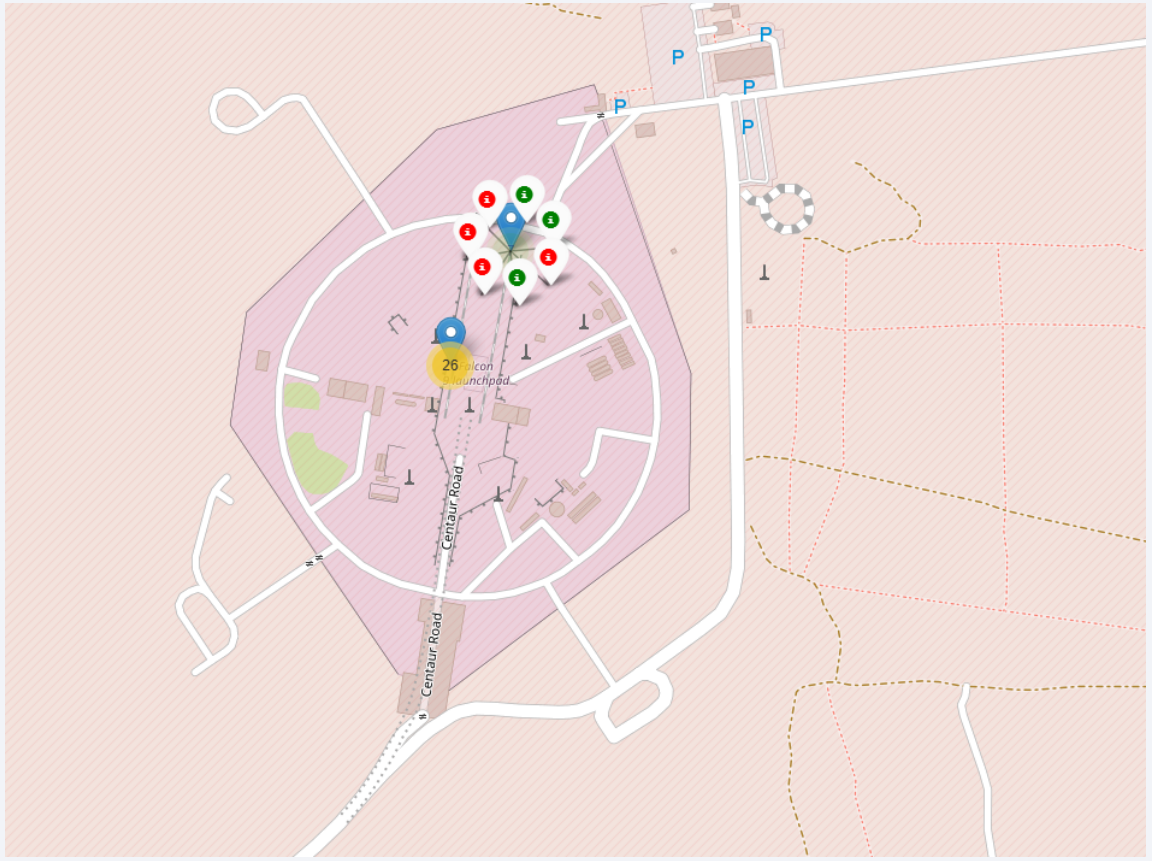
All Launch Sites



- The markers show that launch activity is concentrated primarily in the United States, with major sites located along the east and west coasts. Coastal positioning is strategically important because it allows rockets to launch over open ocean, improving safety and enabling first-stage boosters to land on offshore drone ships.

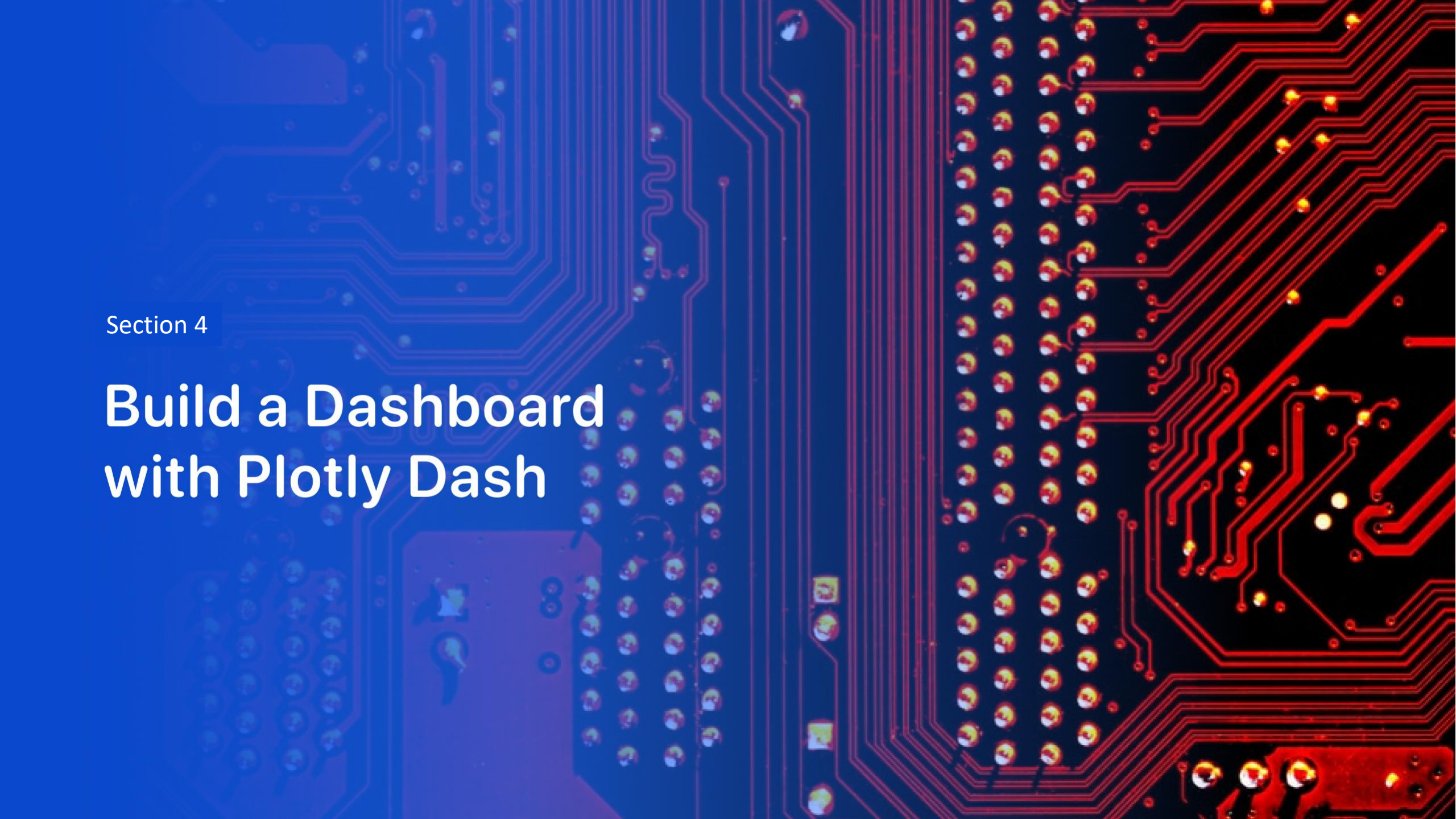
Launch Site Locations Colored by First-Stage Landing Outcome

- Certain launch sites show a higher concentration of successful landings, indicating that location plays an important role in recovery feasibility. Coastal launch sites are particularly favorable, as they allow boosters to land on autonomous drone ships positioned downrange in the ocean. This expands the range of missions where recovery is possible.
- NOTE:
  - Red marker = Failed Launch
  - Green marker = Successful launch



Proximity of Launch Site to Key Infrastructure and Coastline

- This map highlights the CCAFS LC-40 launch site and its proximity to critical infrastructure and geographic features. The site is located close to the coastline (0.96Km), which is strategically important because it allows rockets to launch over open water and enables first-stage boosters to land safely on offshore drone ships. This significantly improves the feasibility of rocket recovery and reuse.
- The nearest International Airport is Melbourne Orlando International Airport (less than 52Km)
- The launch site is ~79Km from downtown Orlando.

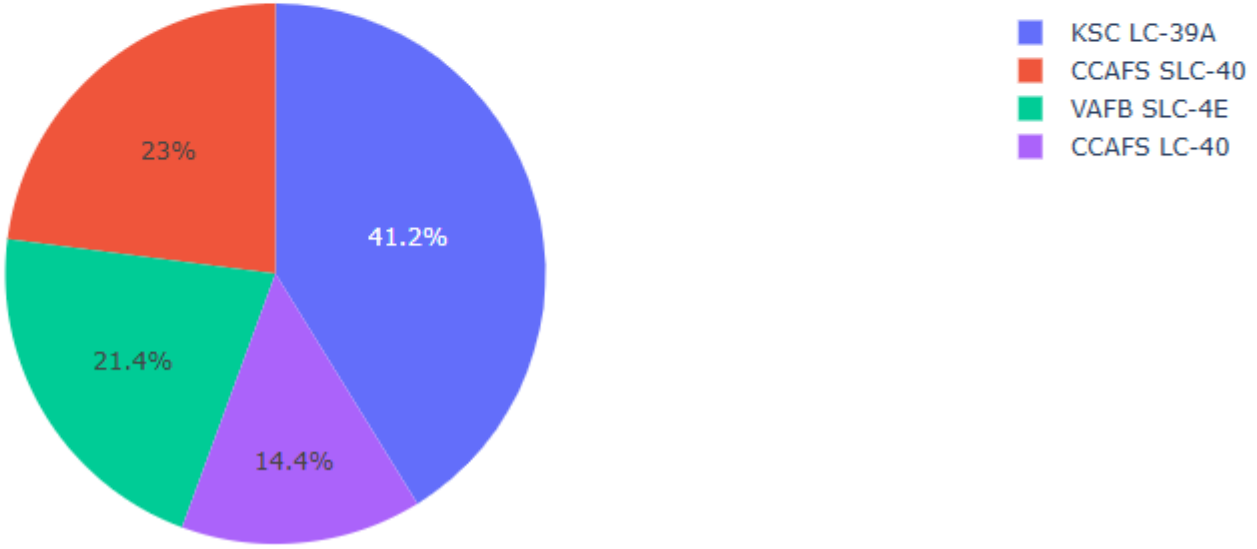

Section 4
Build a Dashboard with Plotly Dash

Success Rate for All Sites

SpaceX Launch Records Dashboard

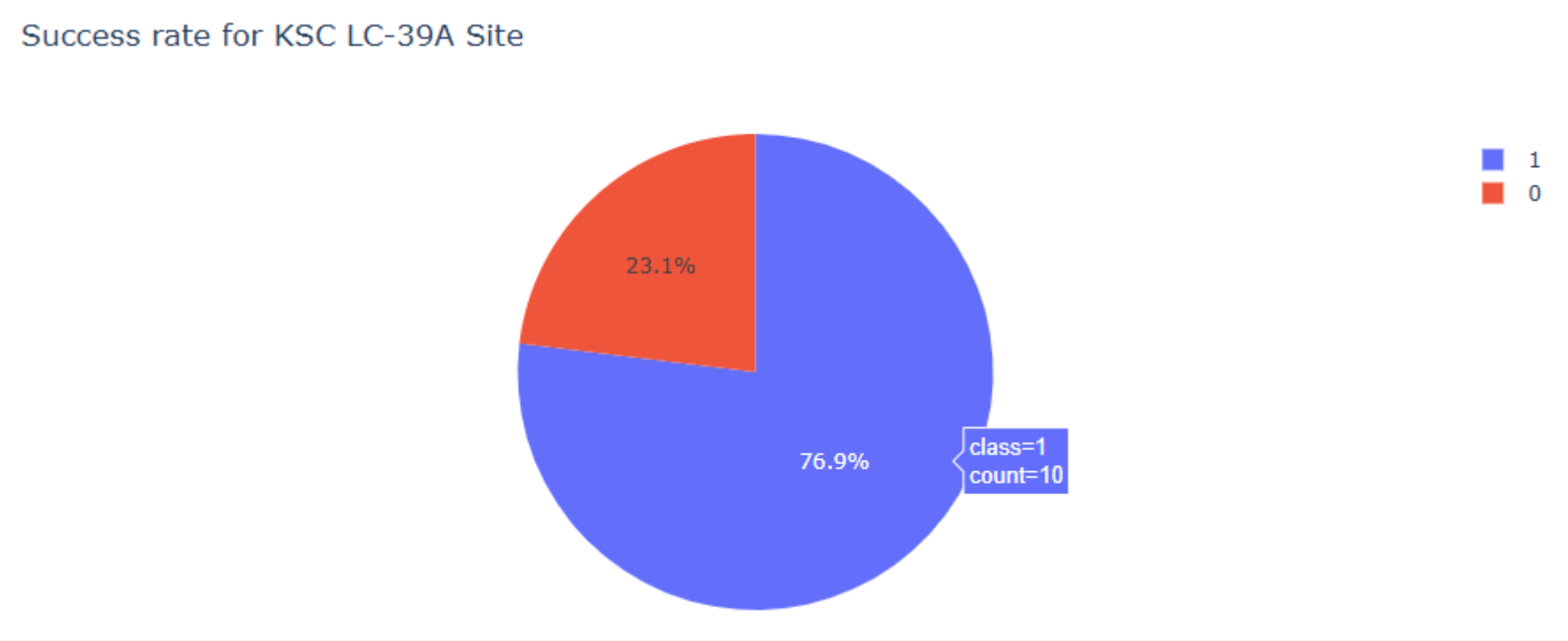
All Sites

Success rate for All Sites



- KSC LC-39A has the highest success rate

Highest Lunch Site Success Ratio



- KSC LC-39A has the highest success rate ratio with 76.9%

Success rate by Payload Mass (all sites)

- From graph above, payload with around 2.000Kg-5.500Kg has higher success rate

Section 5

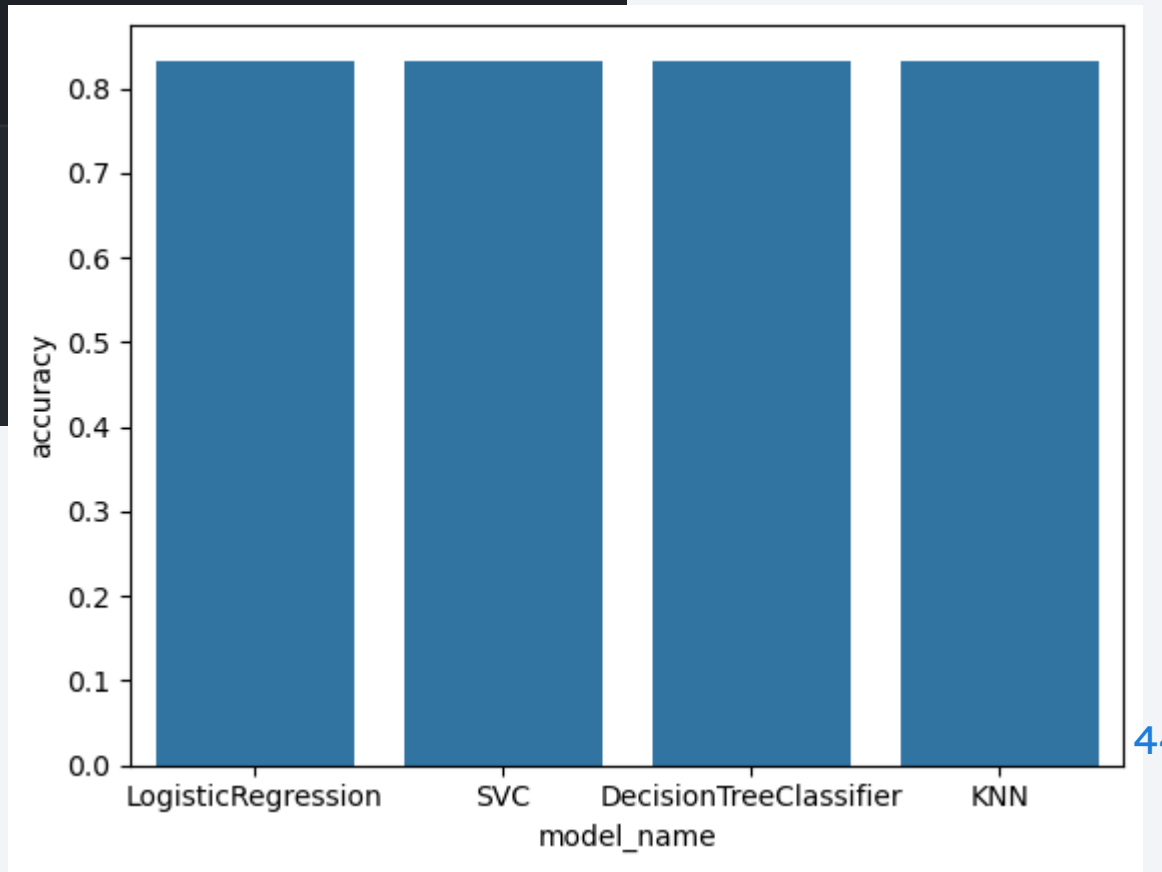
Predictive Analysis (Classification)

Classification Accuracy (at test data)

```python
accuracy_data = [
    {'model_name': 'LogisticRegression', 'accuracy': logreg_cv.score(X_test, Y_test)},
    {'model_name': 'SVC', 'accuracy': svm_cv.score(X_test, Y_test)},
    {'model_name': 'DecisionTreeClassifier', 'accuracy': tree_cv.score(X_test, Y_test)},
    {'model_name': 'KNN', 'accuracy': knn_cv.score(X_test, Y_test)}
]
df_model_acc = pd.DataFrame(accuracy_data)
df_model_acc
```

✓ 0.0s

	model_name	accuracy
0	LogisticRegression	0.833333
1	SVC	0.833333
2	DecisionTreeClassifier	0.833333
3	KNN	0.833333



- The accuracy from all model are identical. This could happen because the data used for testing was too small (18 data points).

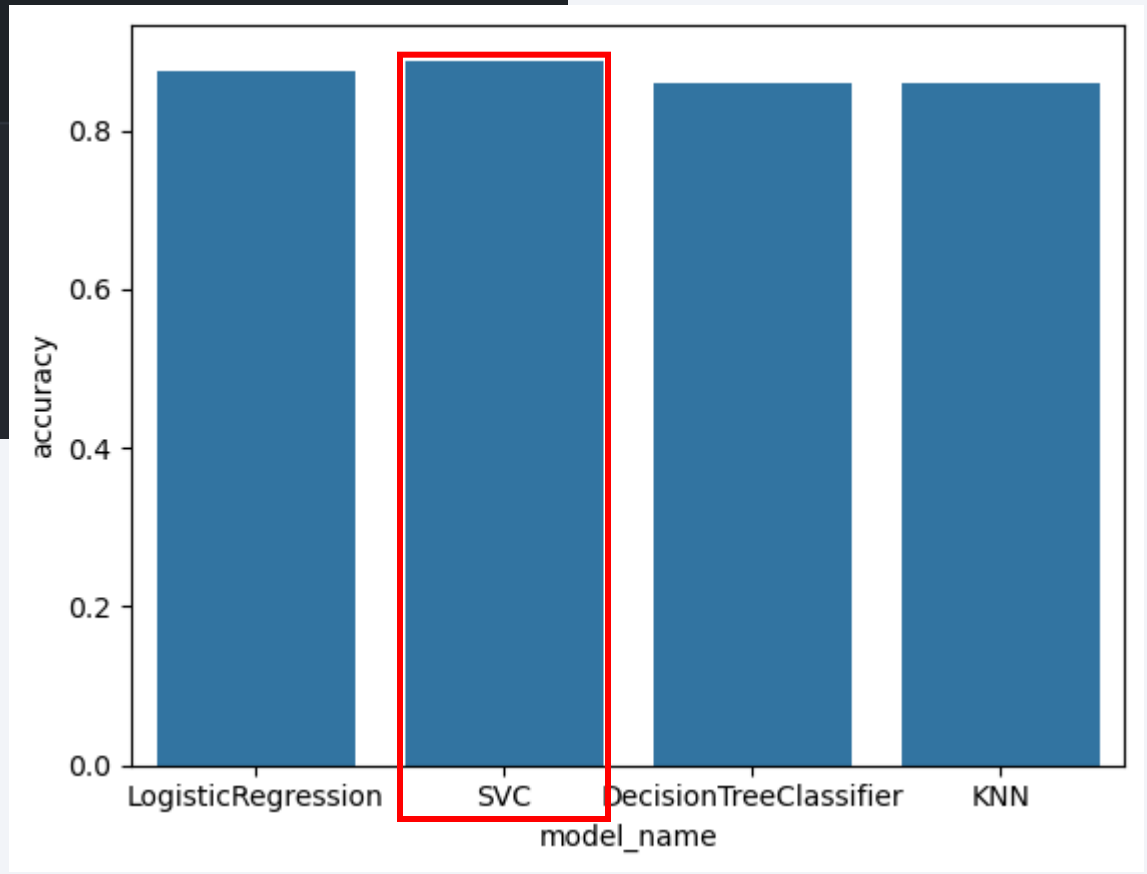
Classification Accuracy (at train data)

```
accuracy_data = [
    {'model_name': 'LogisticRegression', 'accuracy': logreg_cv.score(X_train, Y_train)},
    {'model_name': 'SVC', 'accuracy': svm_cv.score(X_train, Y_train)},
    {'model_name': 'DecisionTreeClassifier', 'accuracy': tree_cv.score(X_train, Y_train)},
    {'model_name': 'KNN', 'accuracy': knn_cv.score(X_train, Y_train)}
]
df_model_acc = pd.DataFrame(accuracy_data)
df_model_acc
```

✓ 0.0s

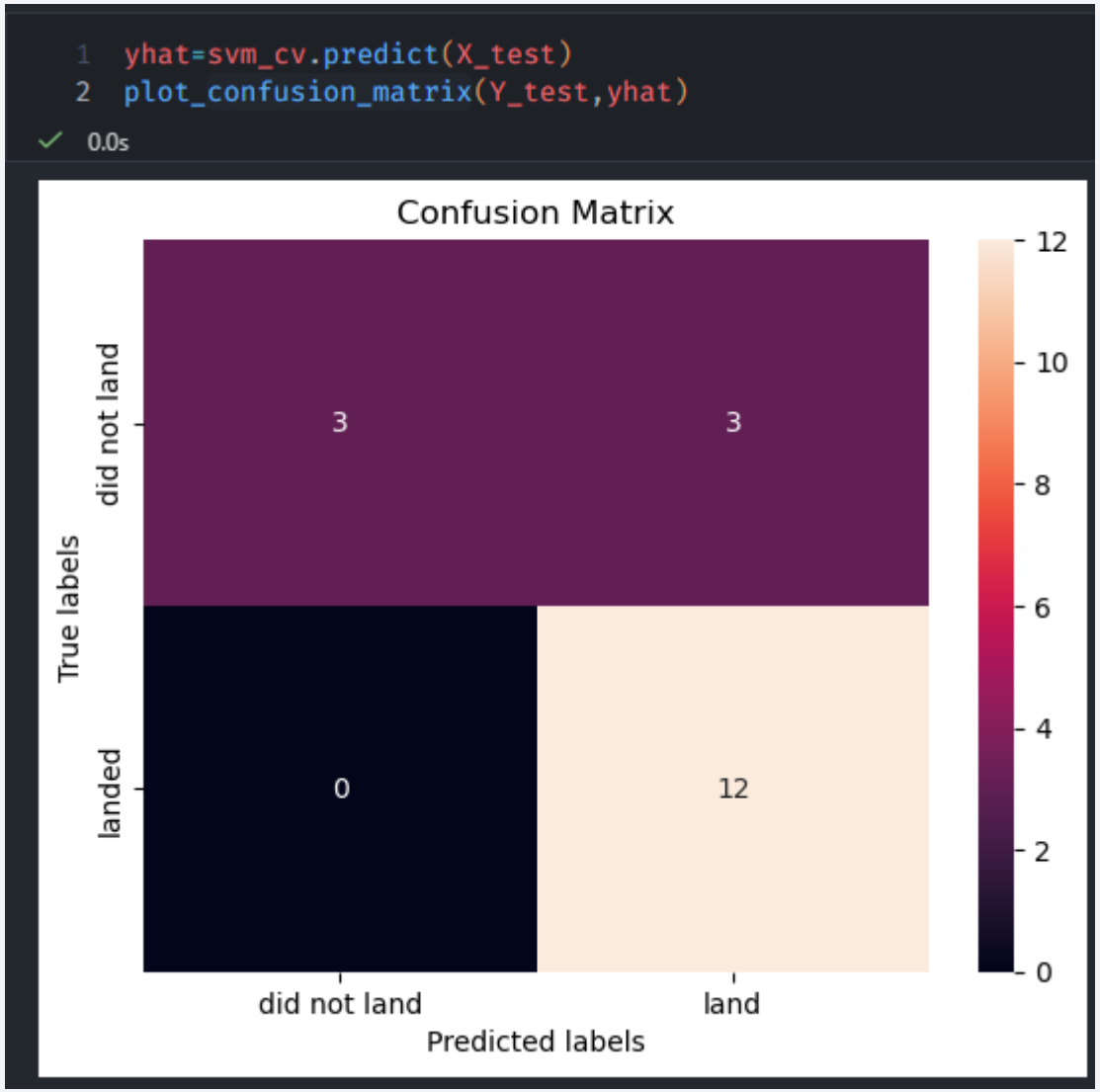
	model_name	accuracy
0	LogisticRegression	0.875000
1	SVC	0.888889
2	DecisionTreeClassifier	0.861111
3	KNN	0.861111

- But when we use training data to calculate accuracy, SVM model has the highest accuracy with 88.8%



Confusion Matrix

```
1  yhat=svm_cv.predict(X_test)
2  plot_confusion_matrix(Y_test,yhat)
✓ 0.0s
```



- All evaluated models produced identical performance metrics on the test dataset, including the same accuracy and confusion matrix values. This outcome is largely due to the small test set size of only 18 observations. With such a limited sample, different models can make predictions on the same data points and yield indistinguishable evaluation results.

Conclusions

- Because of this, the current test data does not provide sufficient evidence to conclusively determine which model performs best. Instead, model selection must consider generalization capability and robustness. Support Vector Machines are known to perform well with limited and high-dimensional data by maximizing the margin between classes, which reduces the risk of overfitting.
- Therefore, SVM is selected as the preferred model, not because it outperformed others on this small test set, but because its learning mechanism offers stronger theoretical generalization for this type of classification problem.

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!