IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fahimeh Khorsand
21.05.2025

Outline

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

Executive Summary

The objective of this study is to examine SpaceX Falcon 9 launch data from multiple sources and utilize machine learning models to forecast the likelihood of a successful first-stage landing. This analysis aids other space organizations in determining whether they should compete with SpaceX in bidding opportunities.

- Summary of methodologies

Data Acquisition: Information was gathered through API calls and web scraping techniques.
Data Preprocessing: Structured and refined the dataset through data wrangling processes.
Exploratory Analysis: Investigated patterns using SQL queries and visual representations.
Geospatial Visualization: Created an interactive Folium map to examine the proximity of launch sites.
Dashboard Development: Designed a dynamic Plotly Dash interface for intuitive exploration of launch records.
Predictive Modeling: Built machine learning models to estimate the likelihood of a successful Falcon 9 first-stage landing.

- Summary of all results

Visual Insights: Comprehensive graphical representations highlight trends, correlations, and key metrics within the dataset.
Interactive Dashboards: Dynamic tools provide real-time exploration of launch records and predictive outcomes.
Predictive Model Outcomes: Machine learning algorithms assess the likelihood of successful landings, offering valuable decision-making insights.

Introduction

- **Project background and context**

➢ Private space travel is becoming more accessible, but high launch costs remain a major barrier for new competitors.

➢ SpaceX's first-stage reusability significantly reduces costs, offering a competitive advantage.

➢ SpaceX launches cost $62 million, while competitors spend over $165 million per launch, making competition challenging.

- **Problems you want to find answers**

➢ Landing Success Estimation: Develop models to predict whether the first stage of SpaceX Falcon 9 will successfully land.

➢ Influential Variables: Assess key parameters such as launch site, payload mass, and booster version to understand their impact on landing outcomes.

➢ Site-Based Correlations: Analyze the relationship between launch sites and historical success rates to uncover trends in landing performance.

Section 1

Methodology

Methodology

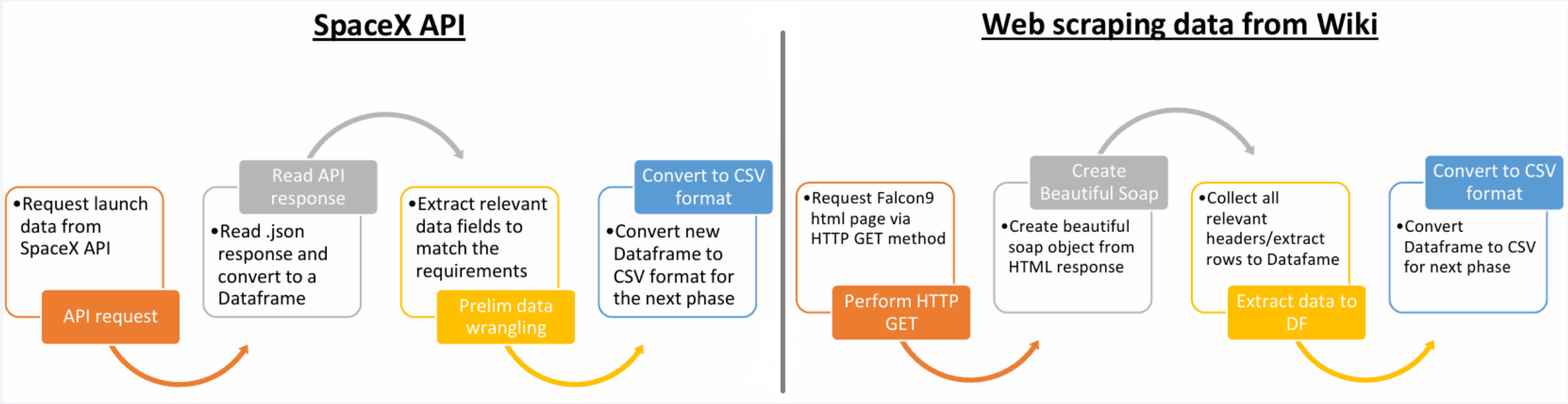
Executive Summary

- Data collection methodology:
  SpaceX API
  Web scrap Falcon 9 and Falcon Heavy launch records from Wikipedia
- Perform data wrangling
  Defined training labels for supervised models by translating mission outcomes into binary classifications: 0 for unsuccessful landings and 1 for successful landings.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

  Generated a 'class' column, applied standardization and data transformation, performed train-test splitting, and evaluated multiple classification algorithms to identify the best-performing model using test data.

Data Collection

Data collection involves gathering information from various sources, which can be structured, unstructured, or semi-structured. In this project, data was obtained through the SpaceX API and web scraping from Wikipedia pages containing relevant launch records.



Data Collection – SpaceX API

1. API Request and read response into DF
2. Declare global variables
3. Call helper functions with API calls to populate global vars
4. Construct data using dictionary
5. Convert Dict to Dataframe, filter for Falcon9 launches, covert to CSV

1. Create API GET request, normalize data and read in to a Dataframe:

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

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

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

2. Declare global variable lists that will store data returned by helper functions with additional API calls to get relevant data

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

3. Call helper functions to get relevant data where columns have IDs (e.g., rocket column is an identification number)
   - getBoosterVersion(data)
   - getLaunchSite(data)
   - getPayloadData(data)
   - getCoreData(data)

4. Construct dataset from received data & combine columns into a dictionary:

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

4. Create Dataframe from dictionary and filter to keep only the Falcon9 launches:

```
# Create a data from launch_dict
df_launch = pd.DataFrame(launch_dict)
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df_launch[df_launch['BoosterVersion']!= 'Falcon 1']
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub

Data Collection - Scraping

1. Perform HTTP GET to request HTML page
2. Create Beautiful Soap object
3. Extract column names from HTML table header
4. Create Dictionary with keys from extracted column names
5. Call helper functions to fill up dict with launch records
6. Convert Dictionary to Dataframe

1. Create API GET method to request Falcon9 launch HTML page

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

```
html_data = requests.get(static_url).text
```

2. Create Beautiful Soap object

```
soup = BeautifulSoup(html_data,"html.parser")
```

3. Find all the tables on the Wiki page and extract relevant column names from the HTML table header

```
html_tables = soup.find_all ('table')
```

```
column_names = []

# Apply find_all() function with `th` element on firs
# Iterate each th element and apply the provided extr
# Append the Non-empty column name (`if name is not N
colnames = soup.find_all('th')
for x in range (len(colnames)):
    name2 = extract_column_from_header(colnames[x])
    if (name2 is not None and len(name2) > 3):
        column_names.append(name2)
```

4. Create an empty Dictionary with keys from extracted column names:

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each vc
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5. Fill up the launch_dict with launch records extracted from table rows.
   - Utilize following helper functions to help parse HTML data

```
def date_time(table_cells):
```

```
def booster_version(table_cells):
```

```
def landing_status(table_cells):
```

```
def get_mass(table_cells):
```

6. Convert launch_dict to Dataframe:

```
df=pd.DataFrame(launch_dict)
```

GitHub

Data Wrangling

- Performed EDA to identify patterns and establish classification labels for supervised learning models.
- Defined Training Labels: Mission outcomes were categorized as 1 for successful first-stage landings and 0 for unsuccessful attempts.
- Landing Scenarios Considered:

  True Ocean: Booster successfully landed in a designated ocean region.

  False Ocean: Booster failed to land in a specific ocean region.

  RTLS: Successful landing on a ground pad.

  False RTLS: Unsuccessful landing on a ground pad.

  True ASDS: Successful landing on a drone ship.

  False ASDS: Unsuccessful landing on a drone ship.

GitHub

Data Wrangling

1. Load dataset in to Dataframe
2. Find patterns in data
3. Create landing outcome label

1. Load SpaceX dataset (csv) in to a Dataframe

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appd
art_1.csv")
```

2. Find data patterns:

i. Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

ii. Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
GEO       1
HEO       1
SO        1
ES-L1     1
```

iii. Calculate number/occurrence of mission outcomes per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
```

3. Create a landing outcome label from Outcome column in the Dataframe

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class = []
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0

GitHub

EDA with Data Visualization

To gain deeper insights into the dataset, various charts were plotted:

- Scatter Plot: Highlights correlations between two variables, making patterns more visible.
  Examined relationships between:
  Flight Number vs. Launch Site
  Payload vs. Launch Site
  Flight Number vs. Orbit Type
  Payload vs. Orbit Type
- Bar Chart: Helps compare variable values at a given time, clearly showing the most common or highest groups.
  Visualized:
  Success rate across different orbit types
- Line Chart: Tracks changes over time, helping to identify trends.
  Observed:
  Yearly trend of average launch success

GitHub

EDA with SQL

SQL Analysis for SpaceX Dataset

- To gain deeper insights into the SpaceX dataset, the following SQL queries were executed on an IBM DB2 cloud instance:
- Retrieve the unique launch site names from the space mission records.
- Extract five entries where the launch site name starts with "CCA".
- Compute the total payload mass transported by boosters associated with NASA (CRS).
- Determine the average payload mass carried by booster version F9 v1.1.
- Identify the date of the first successful landing on a ground pad.
- List booster names that had successful drone ship landings and carried payloads between 4,000 and 6,000 kg.
- Calculate the total number of successful and failed mission outcomes.
- Find the booster versions that transported the maximum payload mass, using a subquery.
- Retrieve records of failed drone ship landings in 2015, along with booster versions and launch site details.
- Rank landing outcomes (e.g., successful ground pad landings or failed drone ship landings) between June 4, 2010, and March 20, 2017, in descending order.

GitHub

Build an Interactive Map with Folium

- Used Folium to visualize launch sites and analyze proximity factors affecting success rates.
- Added markers, labels, and clustered success/failure indicators.
- Calculated distances to coastlines, railroads, highways, and cities.
- Integrated interactive tools for enhanced geospatial analysis.
- Found that launch sites are close to railways, highways, and coastlines but strategically distanced from cities.

GitHub

Build a Dashboard with Plotly Dash

Plotly Dash Interactive Dashboard

Developed a real-time web application using Plotly Dash to visualize SpaceX launch data interactively.

Key Features:

- Launch Site Filter: Dropdown menu allows selection of either all launch sites or a specific site.
- Success Rate Visualization: Pie chart displays total successful launches, with detailed success/failure counts per site.
- Payload Analysis: Range slider enables exploration of payload variations and their impact on launch success.
- Correlation Insights: Scatter plot shows the relationship between payload mass and mission outcomes, with booster versions color-coded for clarity.

This dashboard enhances data-driven decision-making by providing an intuitive way to analyze launch patterns and trends.

Dashboard Insights Summary

Most Successful Launch Site: KSC LC-39A with 10 successful launches.

Highest Success Rate: KSC LC-39A, achieving 76.9% success.

Optimal Payload Range: 2000–5000 kg has the highest success rate.

Least Successful Payload Ranges: 0–2000 kg and 5500–7000 kg.

Best Performing Booster Version: FT has the highest launch success rate.

GitHub

Predictive Analysis (Classification)

1. Read dataset into Dataframe and create a 'Class' array
2. Standardize the data
3. Train/Test/Split data in to training and test data sets
4. Create and Refine Models
5. Find the best performing Model

1. Load SpaceX dataset (csv) in to a Dataframe and create NumPy array from the column class in data

```
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object
et_part_2.csv")
```

```
Y = data['Class'].to_numpy()
```

2. Standardize data in X then reassign to variable X using transform

```
X= preprocessing.StandardScaler().fit(X).transform(X)
```

3. Train/test/split X and Y in to training and test data sets.

```
# Split data for training and testing data sets
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split
( X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

4. Create and refine Models based on following classification Algorithms: (below is LR example)
   i. Create Logistic Regression object and then create a GridSearchCV object
   ii. Fit train data set in to the GridSearchCV object and train the Model

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
LR = LogisticRegression()
logreg_cv = GridSearchCV(LR, parameters,cv=10)
logreg_cv.fit(X_train, Y_train)
```

   iii. Find and display best hyperparameters and accuracy score

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

   iv. Check the accuracy on the test data by creating a confusion matrix

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

   v. Repeat above steps for Decision Tree, KNN, and SVM algorithms

3. Find the best performing model

```
Model_Performance_df = pd.DataFrame({'Algo Type': ['Logistic Regression', 'SVM','Decision Tree','KNN'],
'Accuracy Score': [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_],
'Test Data Accuracy Score': [logreg_cv.score(X_test, Y_test), svm_cv.score(X_test, Y_test),
tree_cv.score(X_test, Y_test), knn_cv.score(X_test, Y_test)]})
```

```
i = Model_Performance_df['Accuracy Score'].idxmax()
print('The best performing alogrithm is '+ Model_Performance_df['Algo Type'][i]
+ ' with score ' + str(Model_Performance_df['Accuracy Score'][i]))
```

```
The best performing alogrithm is Decision Tree with score 0.875
```

	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

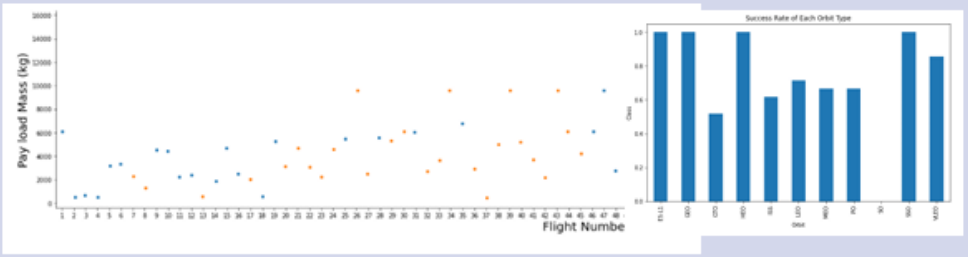
GitHub

Results

Following sections and slides explain results for:

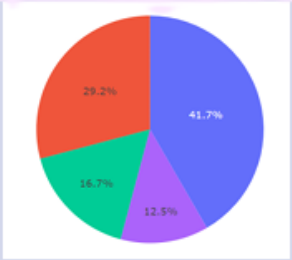
Exploratory data analysis results

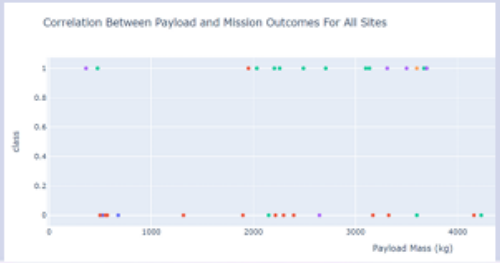
- Samples:



Interactive analytics demo in screenshots

- Samples

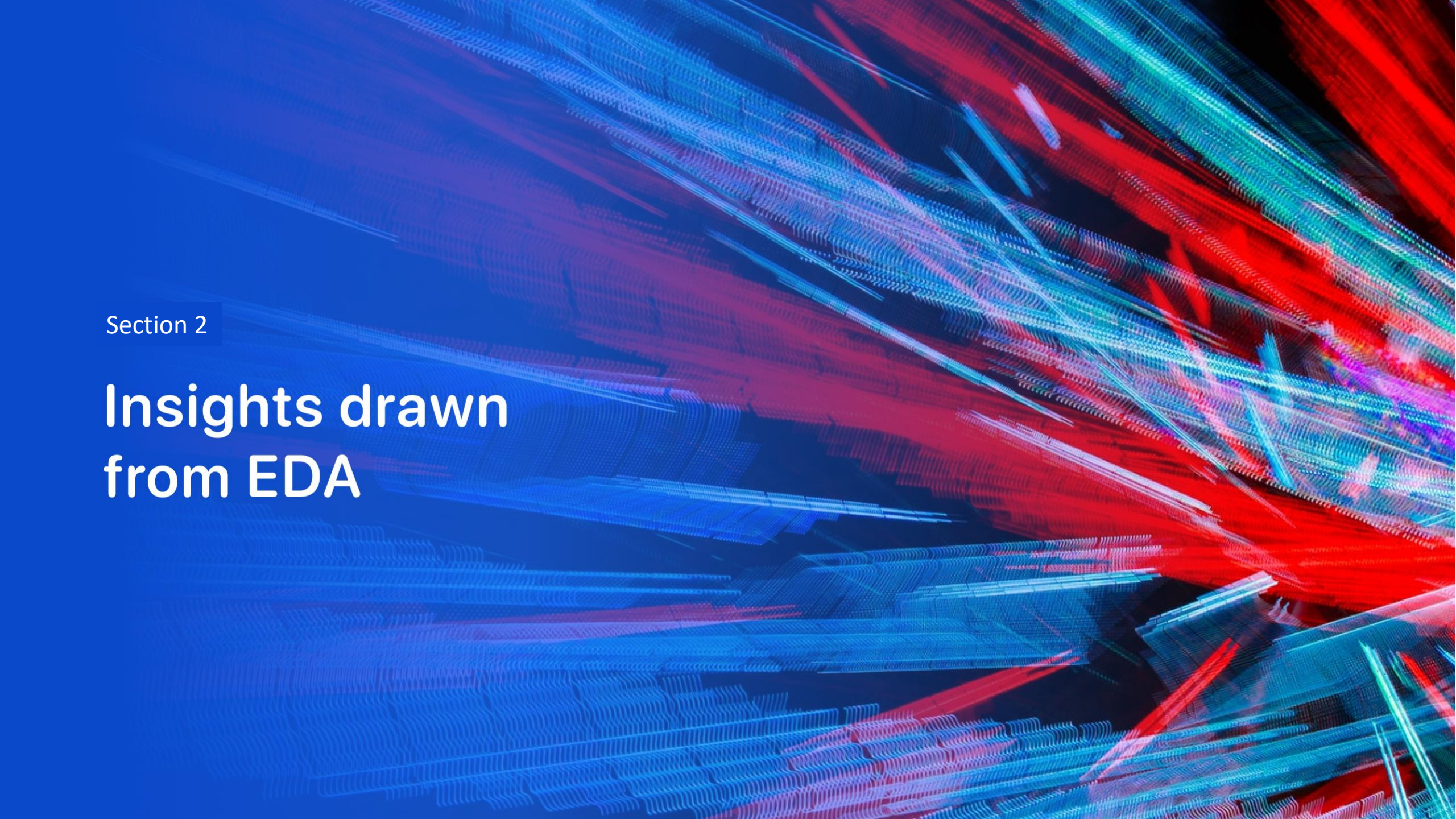




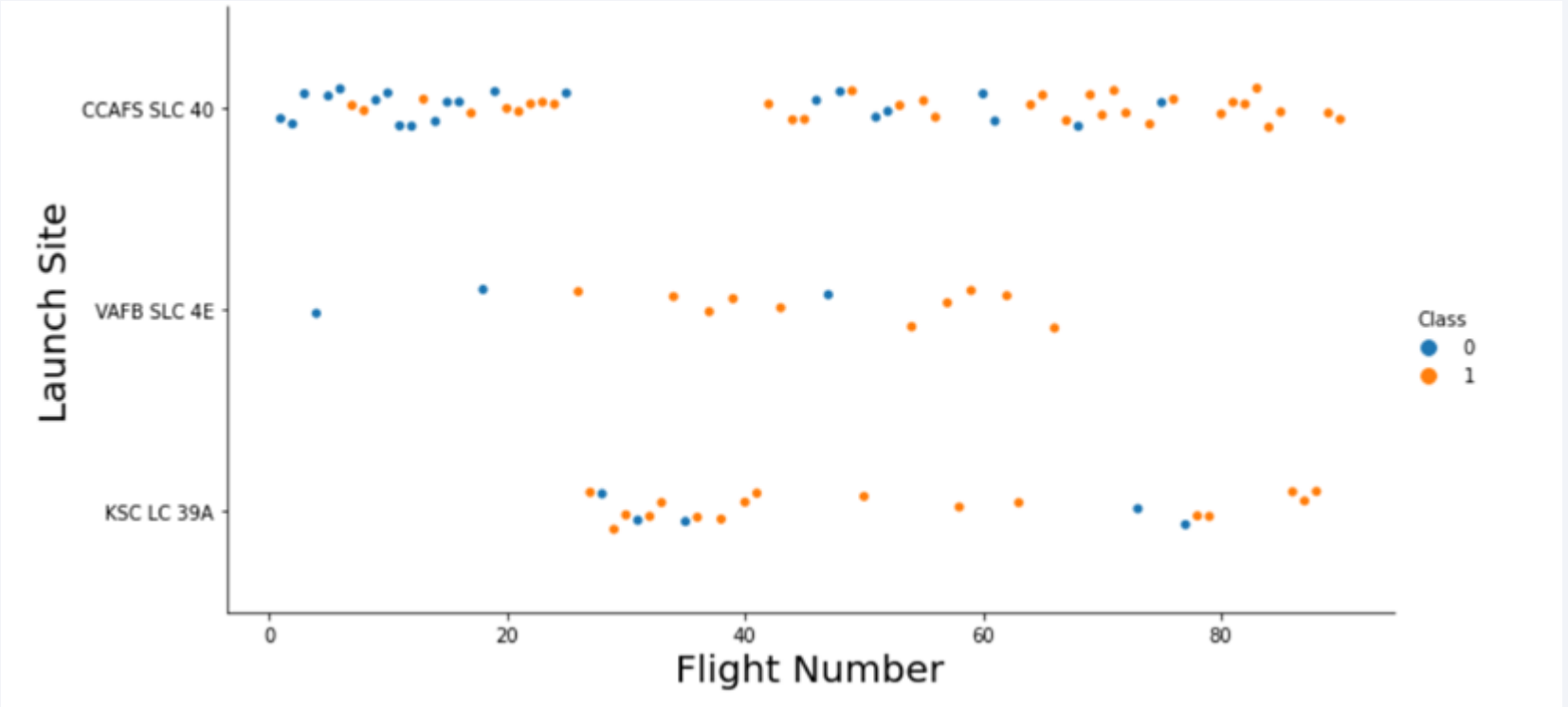
Predictive analysis results

- Samples

	Algo Type	Accuracy Score
2	Decision Tree	0.903571
3	KNN	0.848214
1	SVM	0.848214
0	Logistic Regression	0.846429


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



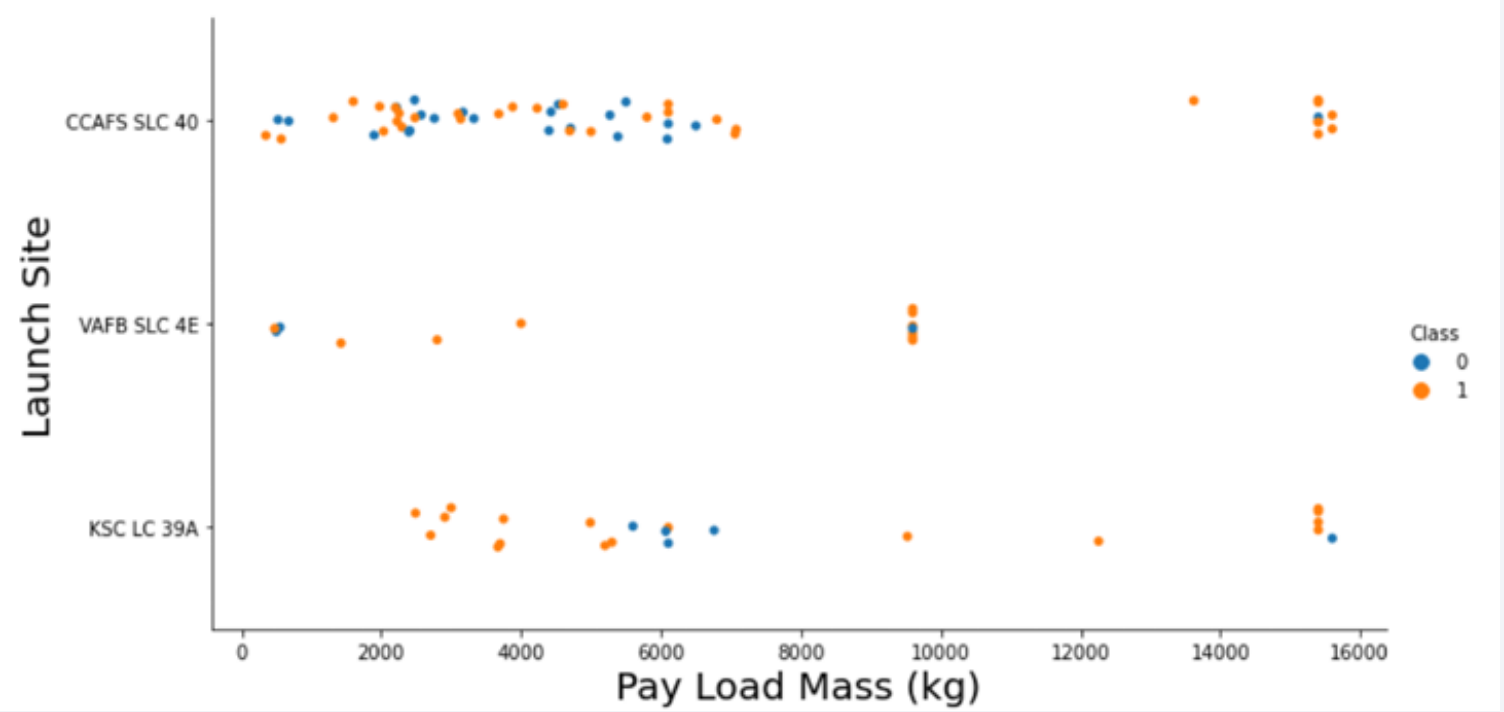
Success rates (Class=1) increases as the number of flights increase

For launch site ‘KSC LC 39A’, it takes at least around 25 launches before a first successful launch

Payload vs. Launch Site

Launch Site VAFB SLC 4E - Key Observations

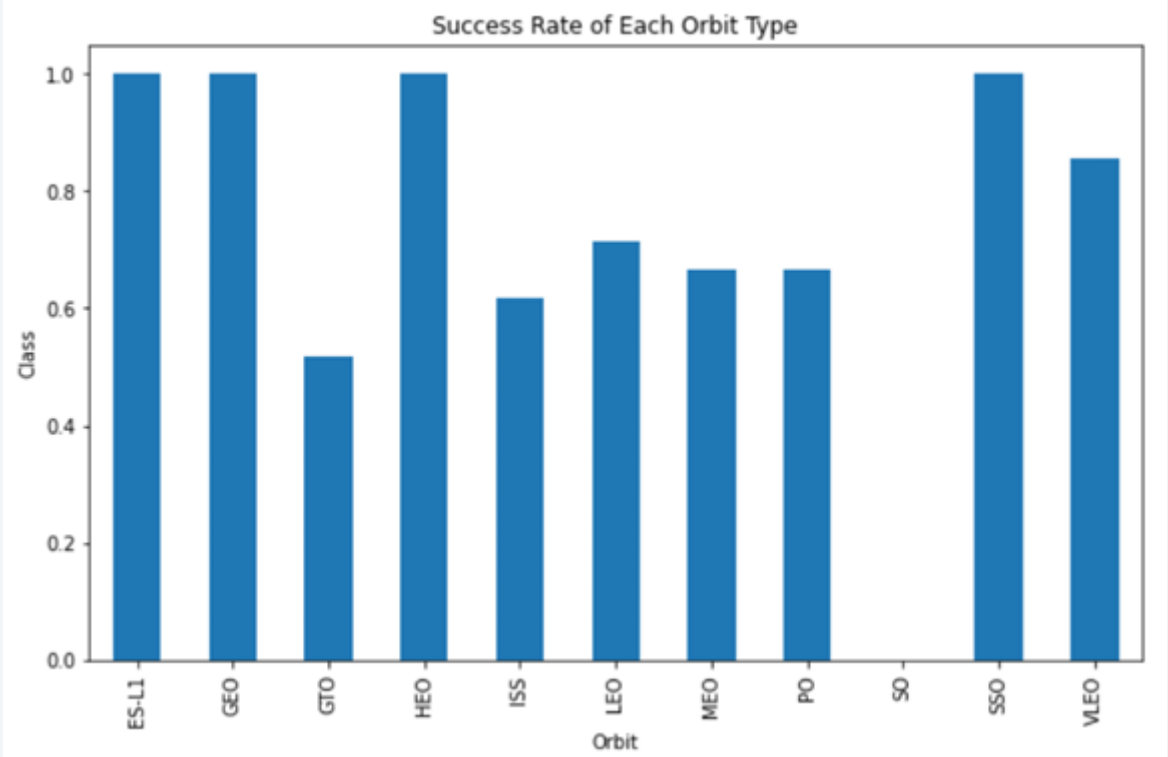
- No rockets have been launched from VAFB SLC 4E carrying a payload exceeding 10,000 kg.
- Success rate at this site tends to increase as payload mass increases.
- No distinct correlation or pattern is observed between launch site and payload mass.



Success Rate vs. Orbit Type

The highest success rates:
Orbits ES-LI, GEO, HEO, and SSO

The lowest success rate:
GTO orbit



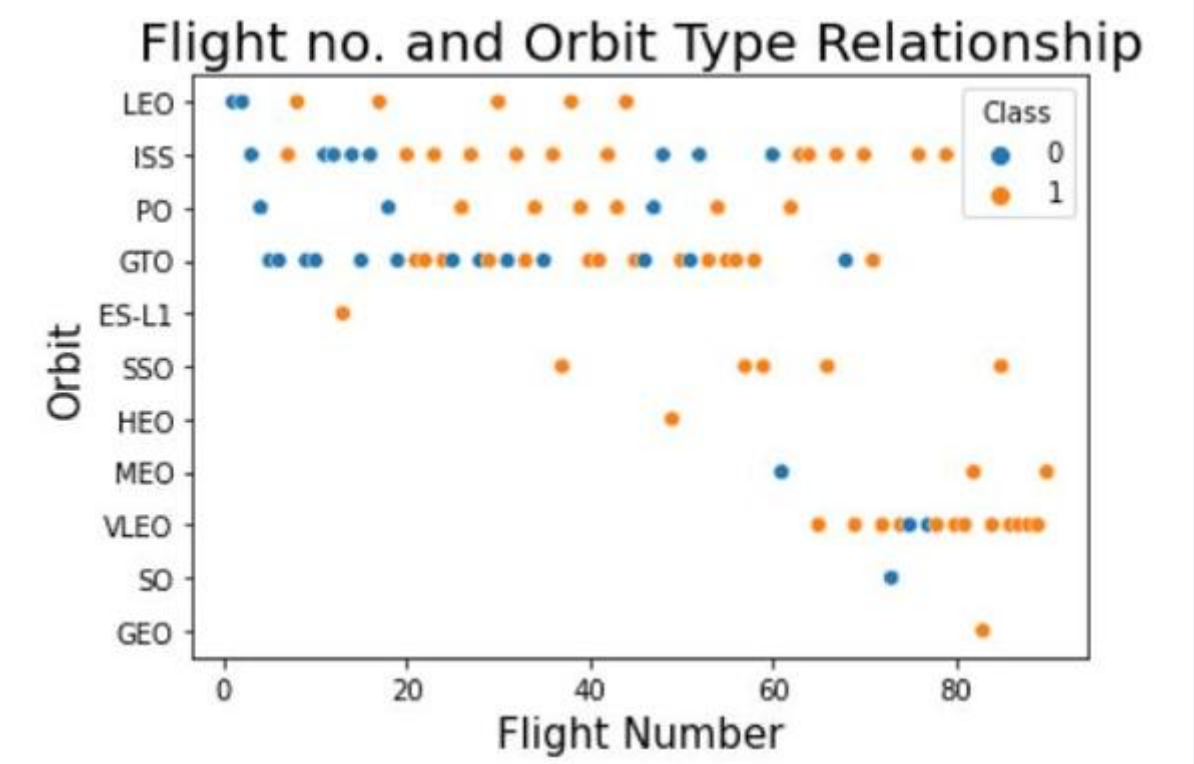
Flight Number vs. Orbit Type

Orbit-Based Landing Trends

VLEO Orbit: The first successful landing does not occur until after 60+ flights.

General Trend: For most orbits (LEO, ISS, PO, SSO, MEO, VLEO), successful landing rates tend to improve with higher flight numbers.

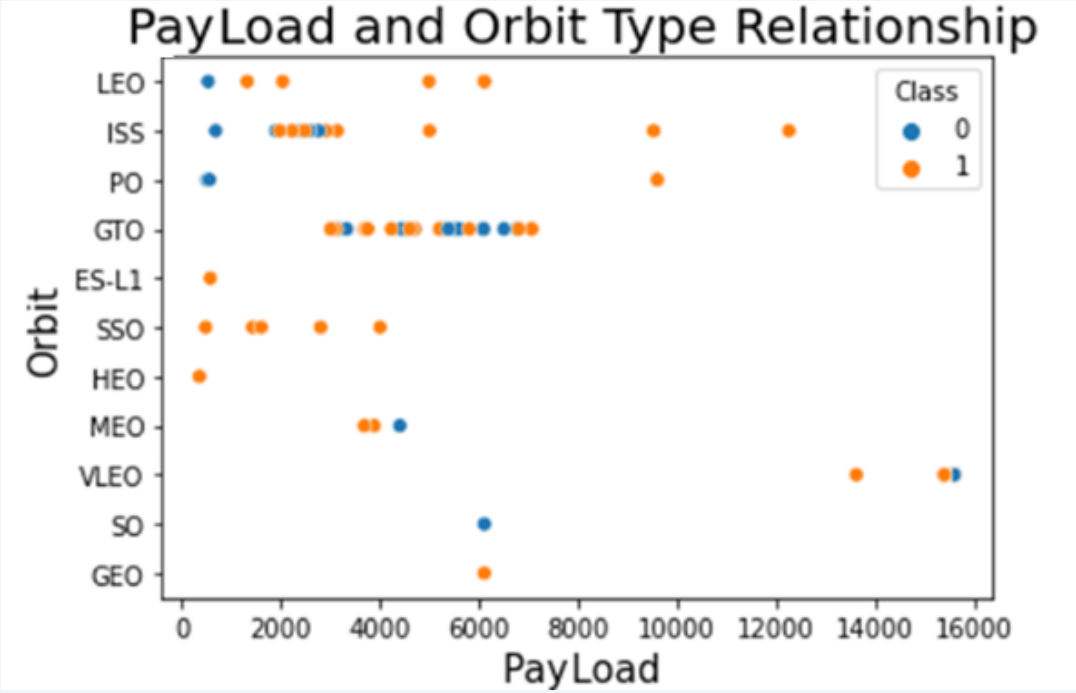
GTO Orbit: No clear correlation exists between flight number and landing success.



Payload vs. Orbit Type

Payload Impact on Landing Success

- **LEO, ISS, PO, and SSO Orbits:** Higher payload mass is generally associated with increased landing success.
- **GEO Orbit:** No distinct pattern is observed between payload mass and landing success rates.



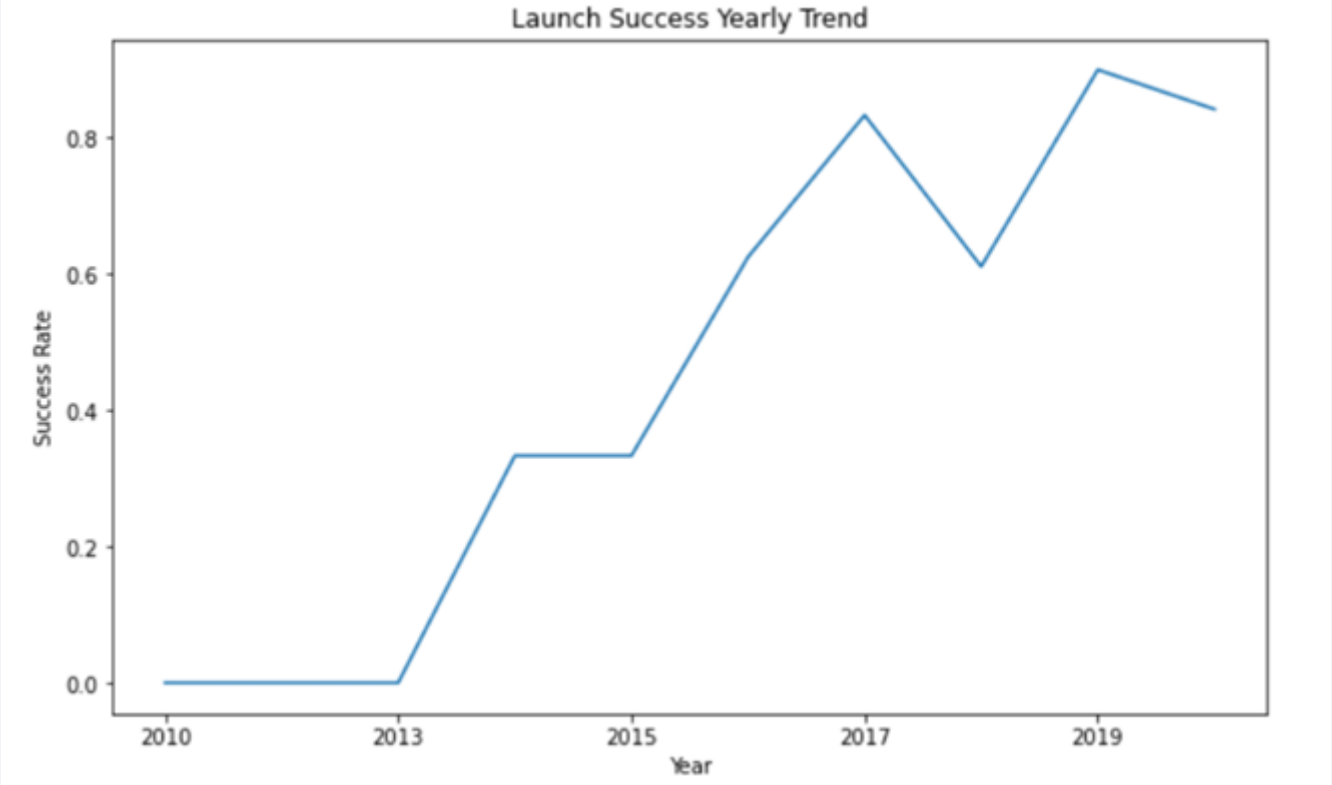
Launch Success Yearly Trend

Launch Success Rate Trends

2013–2020: Success rates saw an 80% increase.

2010–2013 & 2014–2015: Success rates remained stable.

2017–2018 & 2019–2020: Success rates declined.



All Launch Site Names

Query :

```
select distinct Launch_Site from spacextbl
```

Description :

- ‘distinct’ returns only unique values from the queries column (Launch_Site)
- There are 4 unique launch sites

Result :

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

Launch Site Names Begin with 'CCA'

Query :

```
select * from spacextbl where Launch_Site LIKE 'CCA%' limit 5;
```

Description :

- Using keyword ‘Like’ and format ‘CCA%’, returns records where ‘Launch_Site’ column starts with “CCA”.
- Limit 5, limits the number of returned records to 5

Result :

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Query :

```
select sum(PAYLOAD_MASS__KG_) from spacextbl where Customer = 'NASA (CRS)'
```

Description :

'sum' adds column 'PAYLOAD_MASS_KG' and returns total payload mass for customers named 'NASA (CRS)'

Result :

45596

Average Payload Mass by F9 v1.1

Query :

```
select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version LIKE 'F9 v1.1'
```

Description :

'avg' keyword returns the average of payload mass in 'PAYLOAD_MASS_KG' column where booster version is 'F9 v1.1' .

Result :

2928

First Successful Ground Landing Date

Query :

```
select min(Date) as min_date from spacextbl where Landing__Outcome = 'Success (ground pad)'
```

Description :

- ‘min(Date)’ selects the first or the oldest date from the ‘Date’ column where first successful landing on group pad was achieved
- Where clause defines the criteria to return date for scenarios where ‘Landing_Outcome’ value is equal to ‘Success (ground pad)’

Result :

min_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Query :

```
select Booster_Version from spacextbl where (PAYLOAD_MASS__KG_> 4000 and PAYLOAD_MASS__KG_ < 6000)
and (Landing__Outcome = 'Success (drone ship)');
```

Description :

- The query identifies booster versions where the payload mass falls between 4,000 and 6,000 kg and the landing outcome is successful on a drone ship.
- The 'AND' operator ensures that only records meeting both conditions are retrieved.

Result :

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

Total Number of Successful and Failure Mission Outcomes

Query :

```
select Mission_Outcome, count(Mission_Outcome) as counts from spacextbl group by Mission_Outcome
```

Description :

- The ‘group by’ keyword arranges identical data in a column in to group
- In this case, number of mission outcomes by types of outcomes are grouped in column ‘counts’

Result :

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

Boosters Carried Maximum Payload

Query :

```
select Booster_Version, PAYLOAD_MASS__KG_ from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

Description :

- Subquery: Uses the MAX function to identify the highest payload mass value.
- Main Query: Retrieves booster versions along with their corresponding payload mass, filtering for the maximum payload value of 15,600 kg.

Result :

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

Query :

```
select Landing__Outcome, Booster_Version, Launch_Site from spacextbl where Landing__Outcome = 'Failure (drone ship)' and year(Date) = '2015'
```

Description :

- Retrieves failed drone ship landings in 2015, listing landing outcome, booster version, and launch site.
- Uses 'AND' operator to filter records where both conditions are met.
- Extracts the year from the Date column using the YEAR function.
- Results show CCAFS LC-40 as the launch site, with booster versions F9 v1.1 B1012 and B1015 experiencing failed landings.

Result :

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

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

Query :

```
select Landing__Outcome, count(*) as LandingCounts from spacextbl where Date between '2010-06-04' and '2017-03-20'
group by Landing__Outcome
order by count(*) desc;
```

Description :

Grouping Data: The GROUP BY keyword categorizes records by Landing_Outcome.

Filtering by Date: The BETWEEN and AND keywords extract data from June 4, 2010, to March 20, 2017.

Sorting Results: The ORDER BY keyword arranges landing outcome counts in descending order.

Final Output: A ranked list showing landing outcome frequencies within the specified date range.

Result :

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

Section 3

Launch Sites Proximities Analysis

Falcon 9 Launch Sites Overview

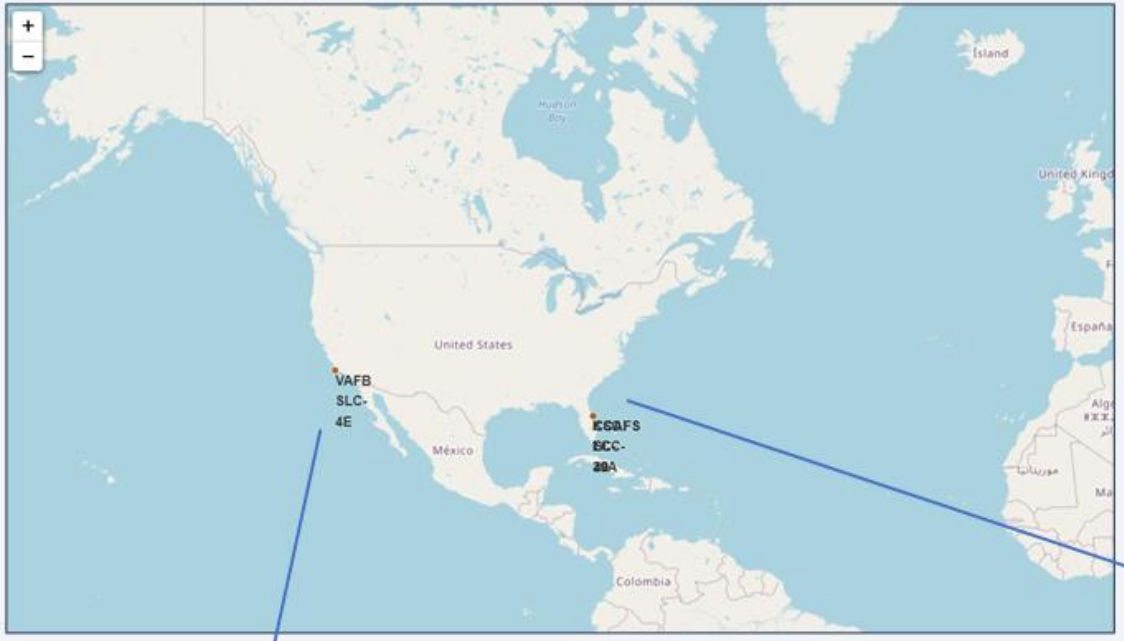


Fig 1 – Global Map

Fig 1: Global map showcasing Falcon 9 launch sites in the United States (California & Florida), each marked with a circle, label, and popup.
Observation: All launch sites are positioned near the coast.
VAFB SLC-4E (CA)
CCAFS LC-40 (FL)
KSC LC-39A (FL)

Fig 2 – Zoom 1

Fig 2 & 3: Provide a zoomed-in view of launch locations, highlighting four sites, including CCAFS SLC-40 (FL).

Fig 3 – Zoom 2

Launch Site Visualization Summary

Fig 1: Displays a U.S. map marking all launch sites, with numbers indicating total successes and failures.
Fig 2–5: Provide zoomed-in views, using green markers for successful launches and red markers for failed attempts.
Key Observation: KSC LC-39A has the highest number of successful launches among all sites.

Fig 1 – US map with all Launch Sites

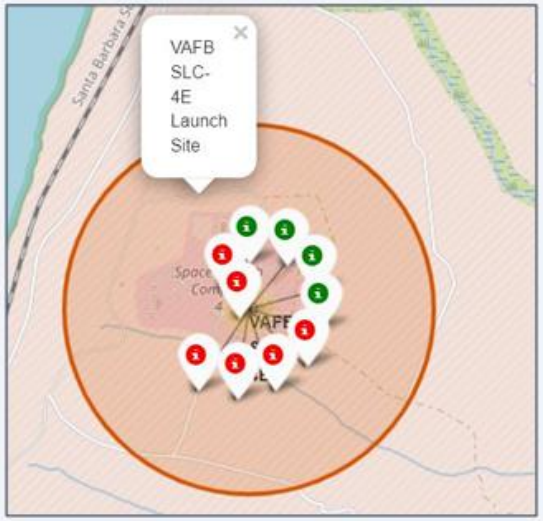


Fig 2 – VAFB Launch Site with success/failed markers

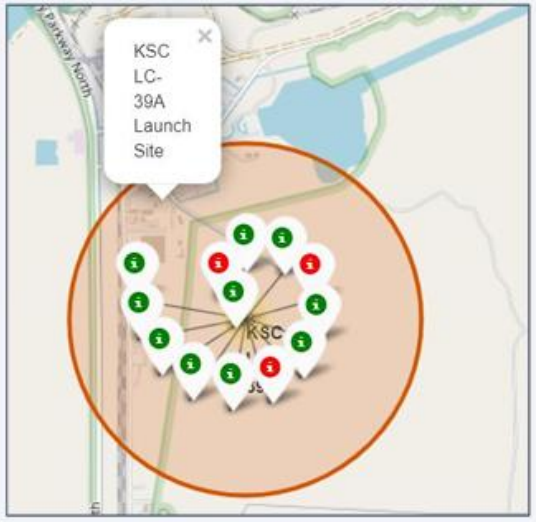


Fig 3 – KSC LC-39A success/failed markers

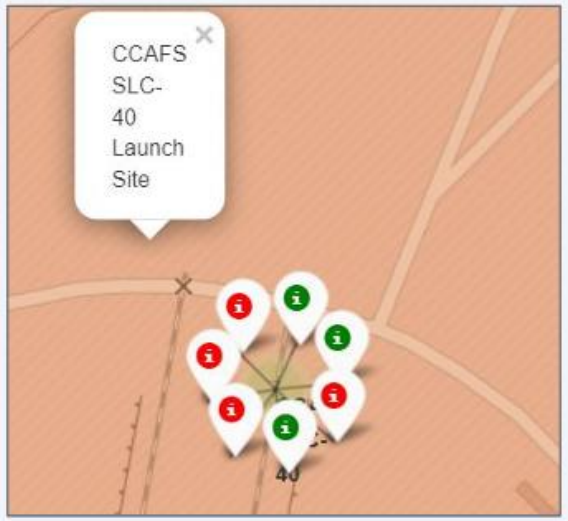


Fig 4 – CCAFS SLC-40 success/failed markers

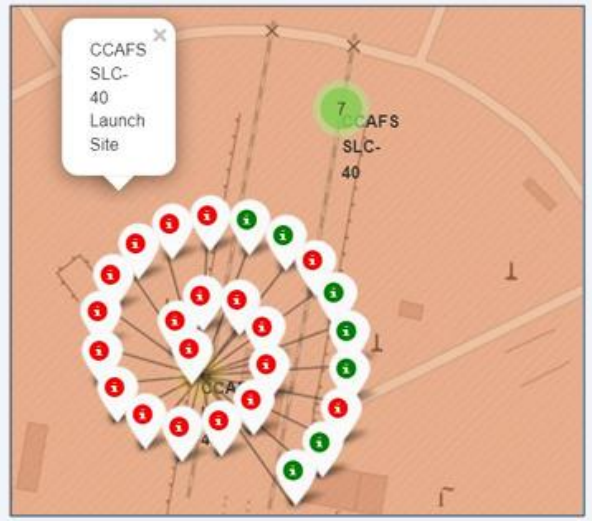


Fig 5 – CCAFS SLC-40 success/failed markers

Launch Site Proximity Analysis

Fig 1: Maps all proximity sites near VAFB SLC-4E, highlighting distances. City Lompoc is farther from the launch site compared to the coastline, railroad, and highway (14.09 km away).

Fig 2: Provides a zoomed-in view, detailing distances of coastline, railroad, and highway from the launch site.

General Observation: Cities are intentionally positioned farther from launch sites to minimize risks, while launch sites are strategically located near coastlines, railroads, and highways for efficient access to resources.

Fig 1 – Proximity site map for VAFB SLC-4E

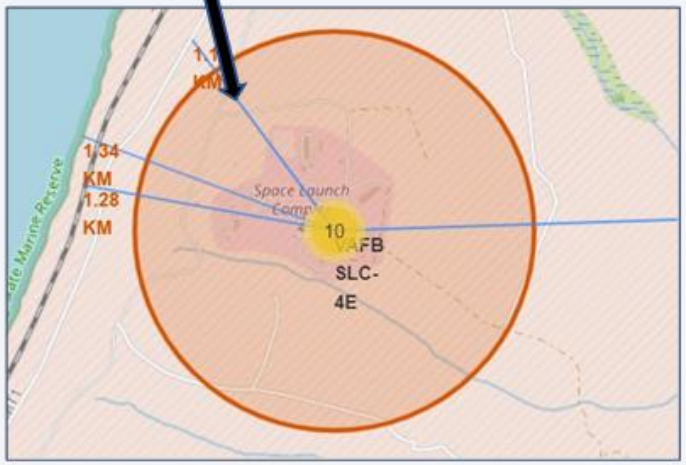
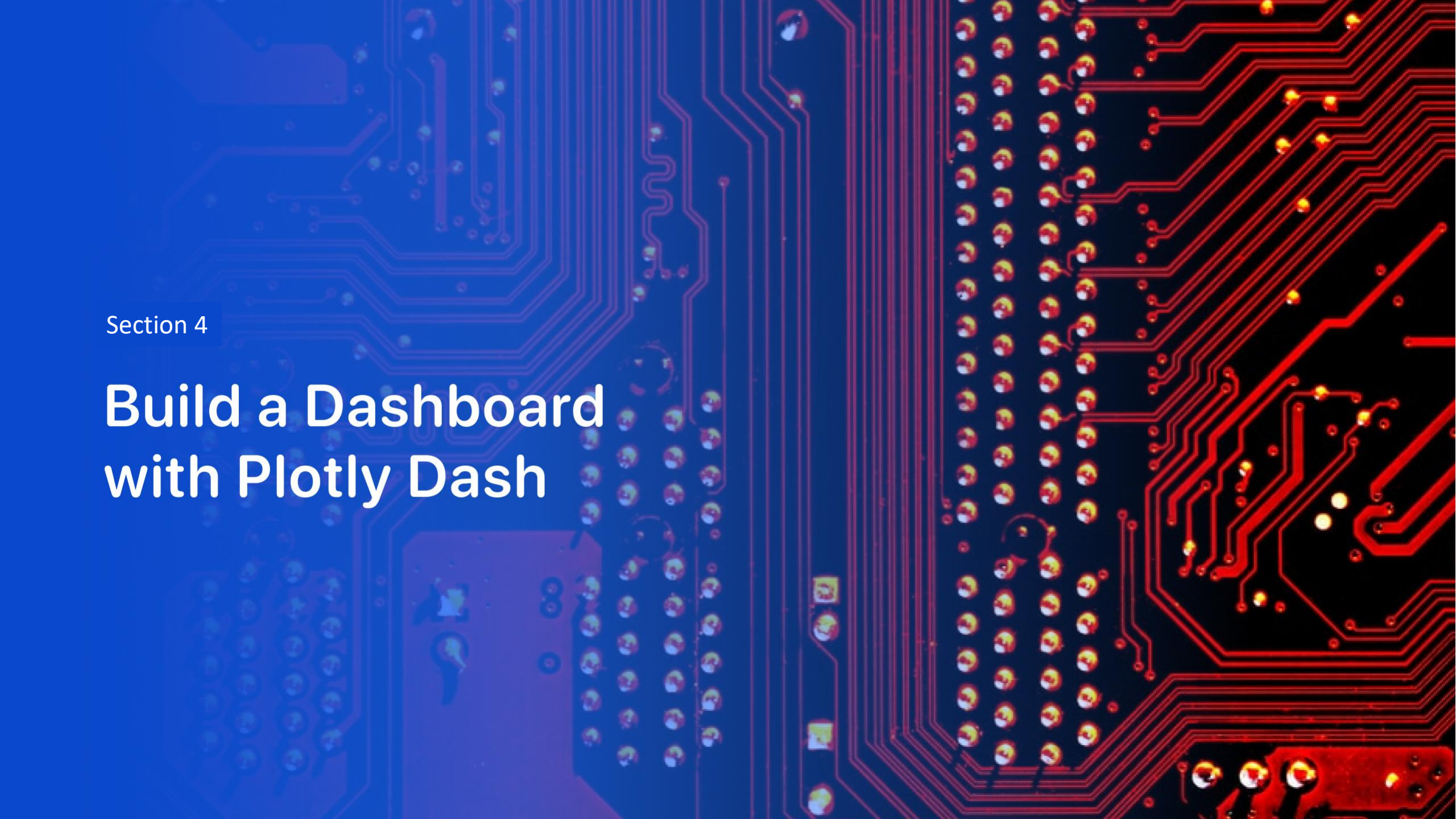


Fig 2 – Zoom in for sites – coastline, railroad, and highway


Section 4
Build a Dashboard with Plotly Dash

Launch Success Rates at All Sites

KSC LC-39A holds the highest launch success rate.

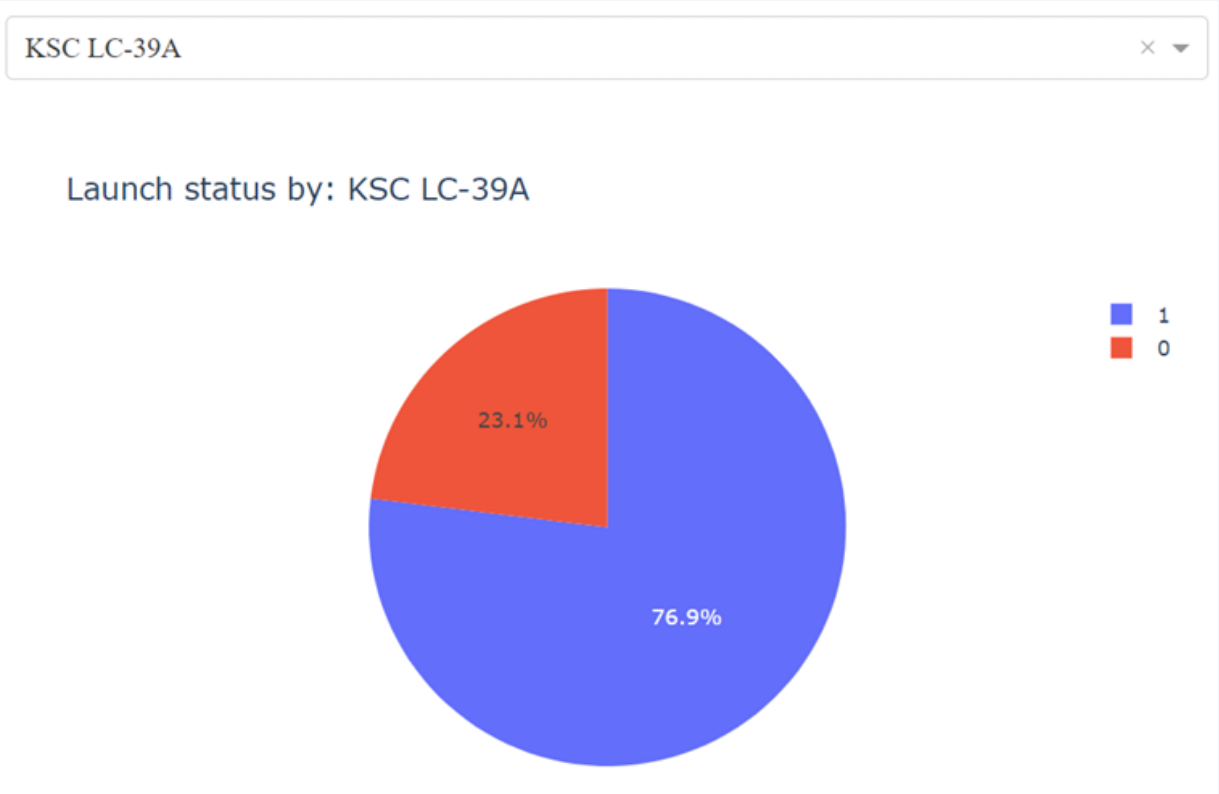
CCAFS SLC-40 has the lowest launch success rate.

KSC LC-39A: Leading Launch Success

KSC LC-39A boasts the highest launch success rate and count.

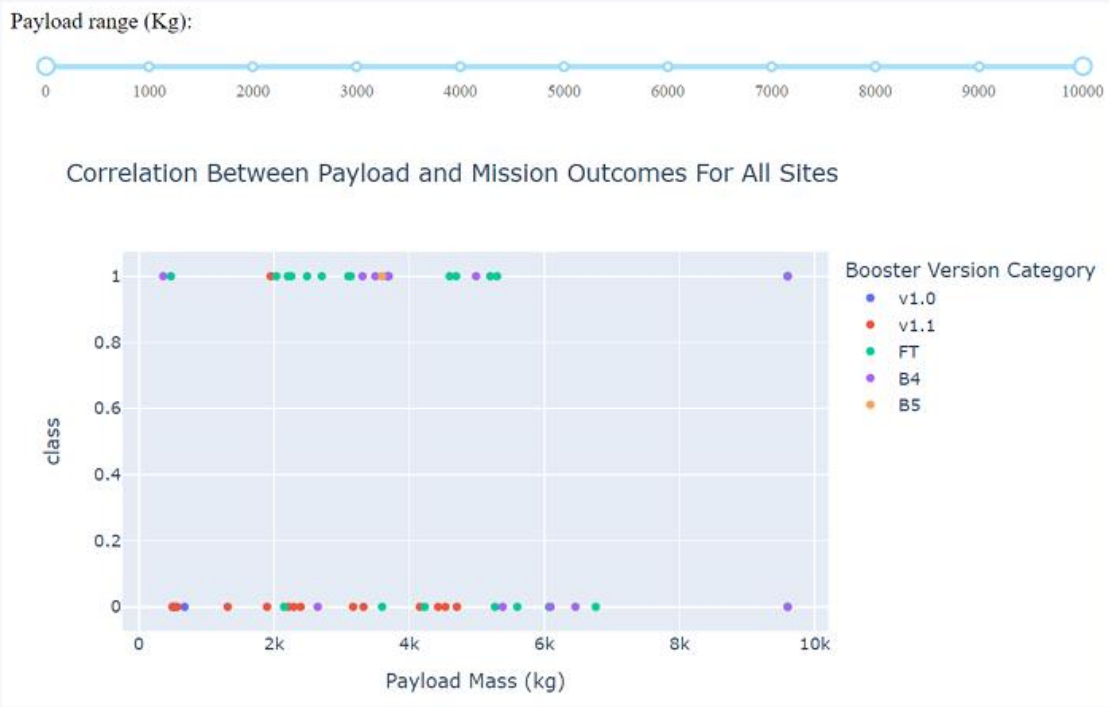
Success Rate: 76.9%

Failure Rate: 23.1%



Scatter Plot: Payload vs. Launch Outcome Across All Sites

- The highest number of successful launches occurred within the 2000–5500 kg payload range.
- Booster version ‘FT’ recorded the most successful launches overall.
- Booster ‘B4’ was the only one to achieve a successful launch with a payload exceeding 6,000 kg.



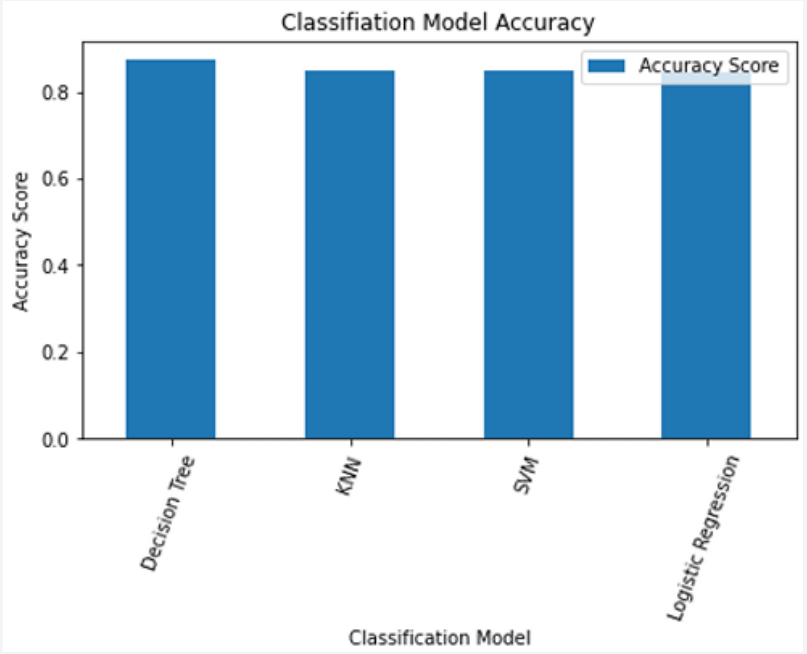
Section 5

Predictive Analysis (Classification)

Classification Accuracy

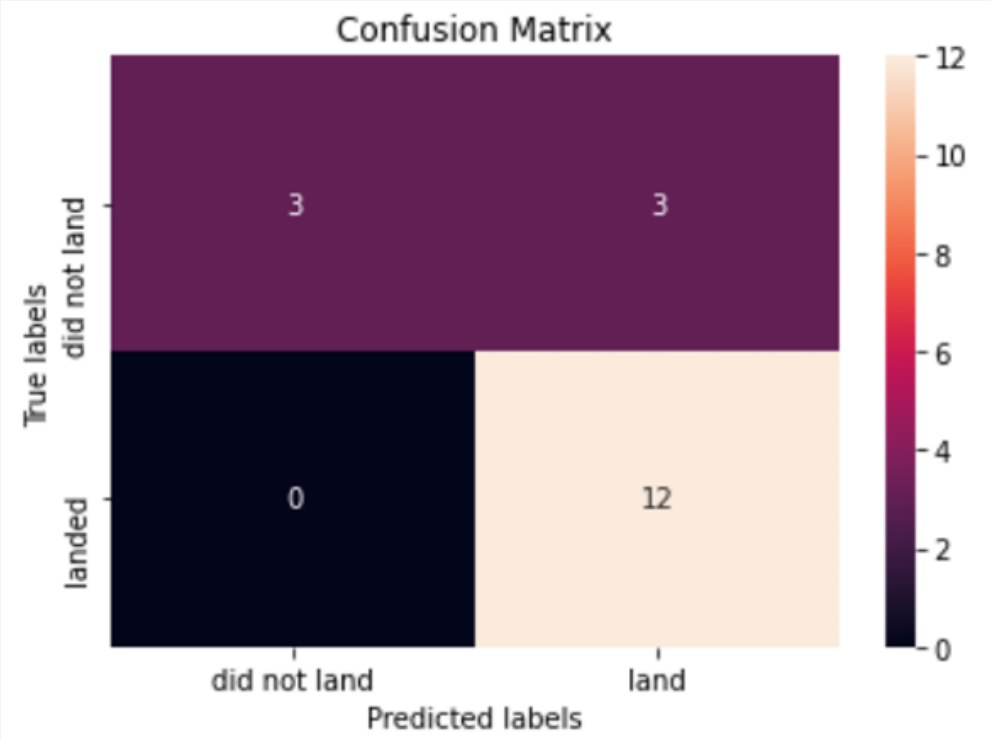
- The Decision Tree algorithm achieves the highest classification score of 0.8750, as shown in the bar chart.
- The accuracy score for all classification algorithms on the test data remains consistent at 0.8333.
- Since accuracy scores are closely matched and test results are identical, a larger dataset may be required for further model refinement.

	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333



Confusion Matrix

- The confusion matrix remains identical across all models (LR, SVM, Decision Tree, KNN).
- The classifier made 18 predictions in total.
- 12 cases were correctly predicted as successful landings (True Positives).
- 3 cases were correctly predicted as failed landings (True Negatives).
- 3 cases were incorrectly predicted as successful landings (False Positives).
- Overall accuracy: 83% with a misclassification/error rate of 16.5%.



Conclusions

- Increasing flight numbers improve the likelihood of successful first-stage landings.
- Higher payload mass is generally associated with better success rates, though no clear correlation exists.
- Launch success rate saw a significant 80% increase from 2013 to 2020.
- KSC LC-39A holds the highest success rate, while CCAFS SLC-40 has the lowest.
- Orbits ES-L1, GEO, HEO, and SSO have higher launch success rates, while GTO has the lowest.
- Launch sites are strategically positioned near coastlines, railroads, and highways while maintaining a safe distance from cities.
- Decision Tree is the best-performing machine learning classification model, achieving 87.5% accuracy. All models scored 83% accuracy on test data, suggesting that more data may be needed for better model tuning.

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!