IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tatarnikov Egor
August 10, 2025

Outline

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

Executive Summary

- **Summary of Methodologies:**
- Data collection via API and web scraping
- Data wrangling and cleaning
- Exploratory data analysis using SQL and visualizations
- Interactive geographic visual analytics with Folium
- Machine learning for prediction

Summary of Results:

- Insights from exploratory data analysis
- Interactive visual analytics demonstrated with screenshots
- Performance and outcomes of predictive models

Introduction

Project Background:

- Launch success depends on factors like launch site, payload, and booster type.
- Analyzing past data helps improve future missions.

Problems to Solve:

- Which sites have the best success rates?
- How does payload affect success?
- Which booster versions perform best?
- Can we predict launch outcomes?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX launch records via APIs and web scraping
- Perform data wrangling
  - Data was cleaned, transformed, and analyzed using SQL and visualization tools to prepare it for modeling and insights
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models were built using Logistic Regression, SVM, Decision Trees, and KNN. Hyperparameters were tuned with GridSearchCV, and models were evaluated using accuracy and confusion matrices on test data

Data Collection

The data was collected from public SpaceX launch records using APIs and web scraping techniques. It includes detailed information on launch sites, payloads, launch outcomes, and booster versions, compiled into structured datasets for analysis.

Sources:

- Space X API (https://api.spacexdata.com/v4/rockets/)
- Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)

Data Collection – SpaceX API

- Data collection with SpaceX REST API

- GitHub: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/11%20Data%20Collection/jupyter-labs-spacex-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

Check the content of the response

In [8]:
```
print(response.content)
```


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://imag
es2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"la
unch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.c
om/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-spacex-inaugural-falcon-1-rocket-los
t-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z","static
_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":3
3,"altitude":null,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew":
[],"ships":[],"capsules":[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":
1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:00","d
ate_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":1,"gridfins":false,"legs":fals
e,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"tb
d":false,"launch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"reco
vered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images
2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":
[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","articl
e":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wikipedia.org/wiki/Demo
Sat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","su
ccess":false,"failures":[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}],"d
etails":"Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+
7 min 30 s, Failed to reach orbit, Failed to recover first stage","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b6b6c
3bb0006eeb1e2"],"launchpad":"5e9e4502f5090995de566f86","flight_number":2,"name":"DemoSat","date_utc":"2007-03-21T01:10:00.000
Z","date_unix":1174439400,"date_local":"2007-03-21T13:10:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"cor
e":"5e9e289ef35918416a3b2624","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_succes
s":null,"landing_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdaffd86e0006
04b32b"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"http
s://images2.imgbox.com/6c/cb/na1tzhHs_o.png","large":"https://images2.imgbox.com/4a/80/k1oAkY0k_o.png"},"reddit":{"campaign":
```

Data Collection - Scraping

- Web scraping process from Wikipedia

- GitHub: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/11%20Data%20Collection/jupyter-labs-webscraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:
```python
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [7]:
```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [8]:
```python
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

In [9]:
```python
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

In [10]:
```python
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

The data processing workflow included:

- Importing raw datasets
- Handling missing and inconsistent values
- Converting data types for analysis compatibility
- Filtering and selecting relevant features
- Aggregating and summarizing data where necessary
- Preparing clean datasets ready for exploratory analysis and modeling
- GitHub: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/12%20Data%20Wrangling/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

During the exploratory data analysis, several types of plots were used to better understand the data:

- **Catplot**: Used to visualize the relationship between categorical variables and numerical outcomes. This helped in comparing success rates across different launch sites and categories.
- **Barplot**: Employed to show the count or frequency of different categories, such as the number of launches per site or booster version. Barplots provided a clear overview of categorical distributions.
- **Lineplot**: Used to observe trends and changes over time or across ordered variables, such as payload mass versus launch success. This helped to identify patterns and correlations in continuous data.

- GitHub URL:https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/22%20EDA%20with%20Visualization/edadataviz.ipynb

EDA with SQL

- Counted launches by site
- Calculated success rates per site
- Filtered launches by payload range Joined launch data with booster info
- Analyzed launch trends over time
- Aggregated failure causes by site Identified top-performing sites
- Retrieved site locations for mapping
- GitHub URL: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/21%20EDA%20with%20SQL/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Markers:** Added to pinpoint exact launch site locations for easy identification on the map.
- **Circles:** Used to highlight launch sites with a visible radius, making them stand out and showing their proximity areas.
- **Lines (Polylines):** Drawn between launch sites and nearby points of interest (coastlines, cities, railways) to visualize distances and spatial relationships. These objects help reveal geographic patterns and assist in analyzing how location factors may affect launch success.
- GitHub URL: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/31%20Interactive%20Visual%20Analytics%20with%20Folium%20(Maps)/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Dropdown for Launch Site Selection:** Allows users to filter data by specific launch sites or view all sites together, enabling focused analysis.
- **Pie Chart:** Shows success vs. failure counts for selected launch sites to quickly visualize success rates.
- **Payload Range Slider:** Lets users select a payload mass range to explore its effect on launch outcomes.
- **Scatter Plot:** Displays correlation between payload mass and launch success, with points color-coded by booster version for deeper insight.
- GitHub URL: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/32%20Plotly/spacex-dash-app.py

Predictive Analysis (Classification)

- **Data Preparation:** Cleaned and standardized input features; split data into training and testing sets.
- **Model Selection:** Tested multiple algorithms — Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
- **Hyperparameter Tuning:** Used GridSearchCV with cross-validation to find optimal hyperparameters for each model.
- **Model Evaluation:** Assessed models on test data using accuracy and confusion matrices to compare performance.
- **Best Model Identification:** Selected the model with the highest test accuracy and balanced classification results.
- **Flowchart:** Data Preprocessing → Train-Test Split → Model Training → Hyperparameter Tuning (GridSearchCV) → Model Evaluation → Best Model Selection
- GitHub URL: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project/blob/main/41%20ML/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

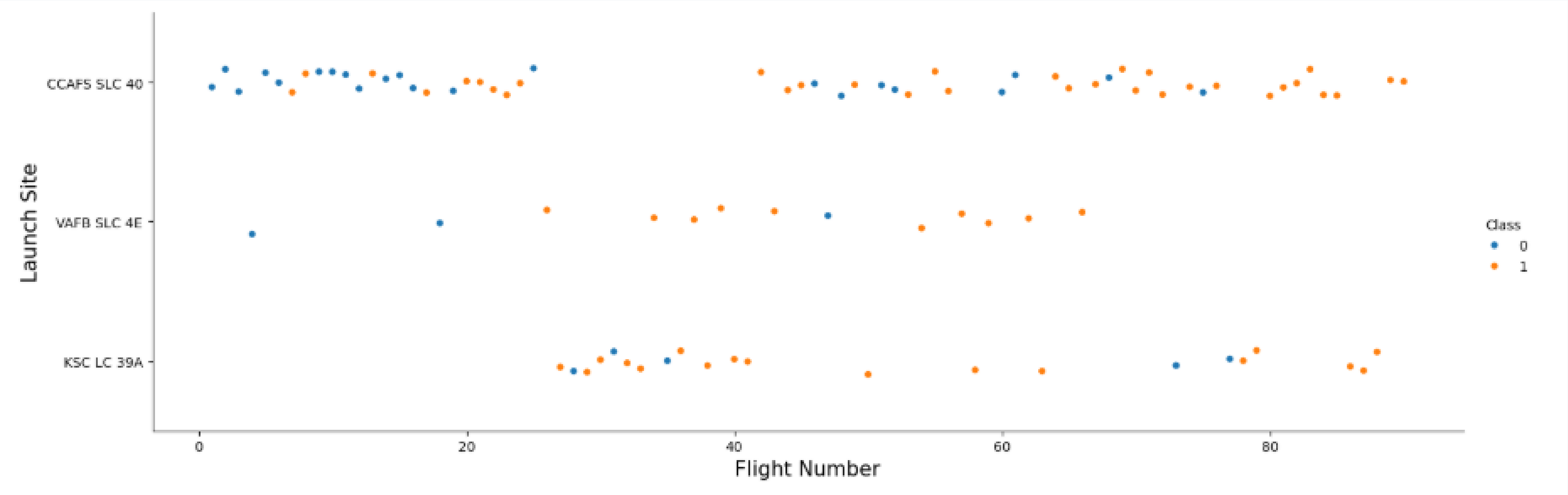
Results

- SpaceX operates from four distinct launch sites.
- Initial launches were conducted for SpaceX and NASA missions.
- The average payload capacity of the Falcon 9 v1.1 booster is approximately 2,928 kg.
- The first successful booster landing occurred in 2015, five years after the inaugural launch.
- Several Falcon 9 booster versions successfully landed on drone ships, often carrying payloads above the average.
- Nearly all missions were successful overall.
- Two booster versions, F9 v1.1 B1012 and B1015, failed drone ship landings in 2015.
- Landing success rates improved steadily over time.

Section 2

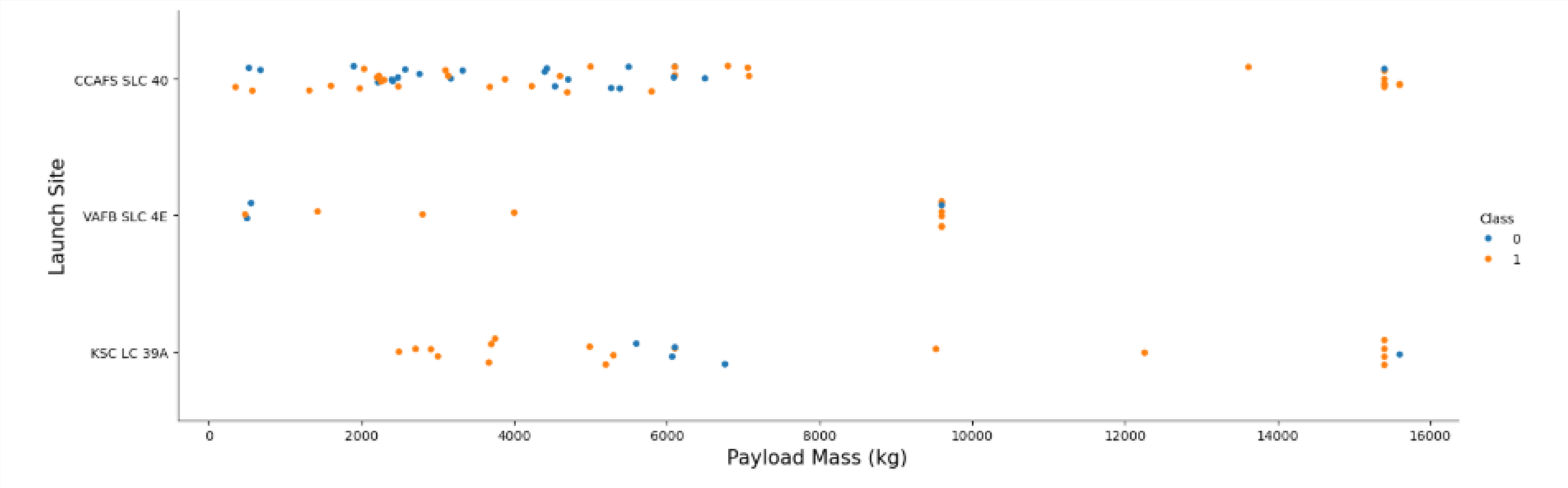
Insights drawn from EDA

Flight Number vs. Launch Site



- The most successful launch site at present is CCAF5 SLC-40, with the highest number of recent successful launches.
- VAFB SLC-4E ranks second, followed by KSC LC-39A in third place.
- Overall, the success rate has shown consistent improvement over time.

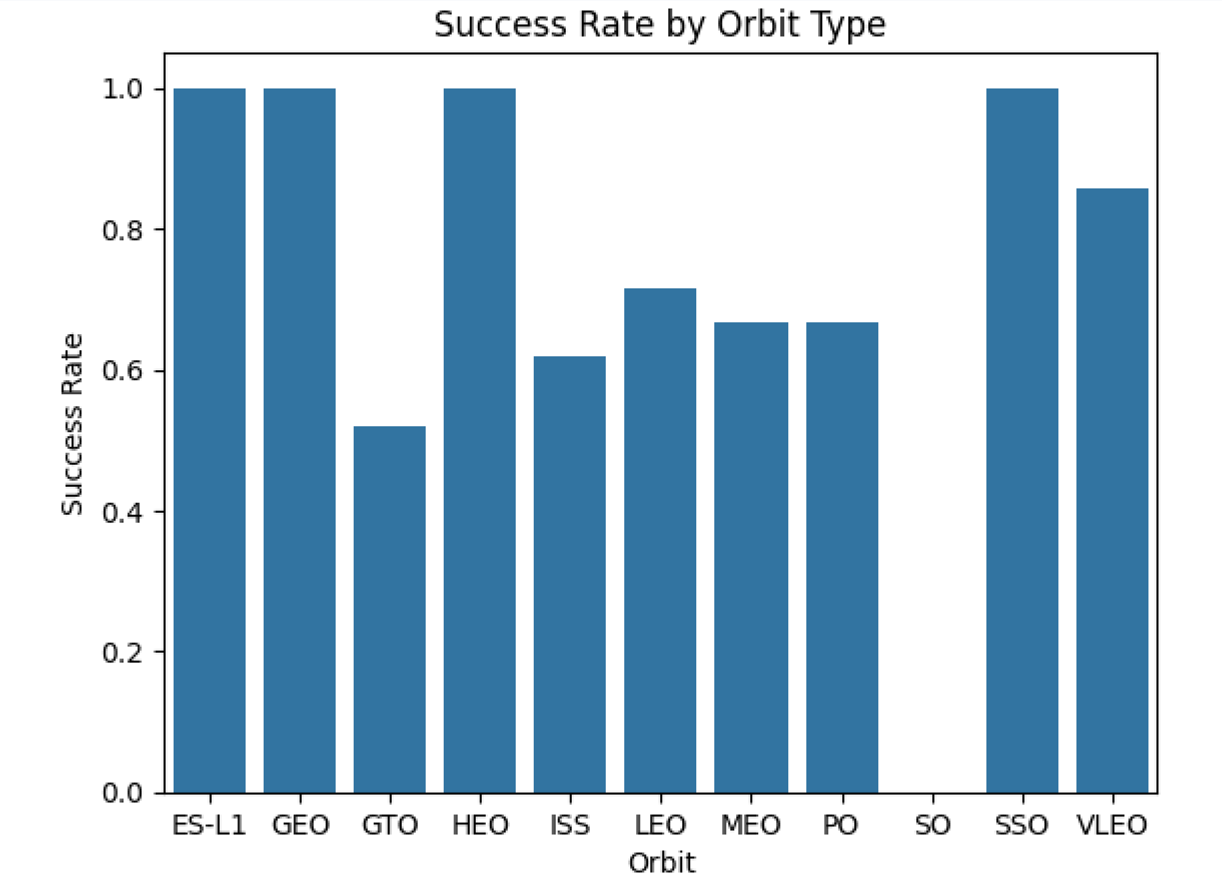
Payload vs. Launch Site



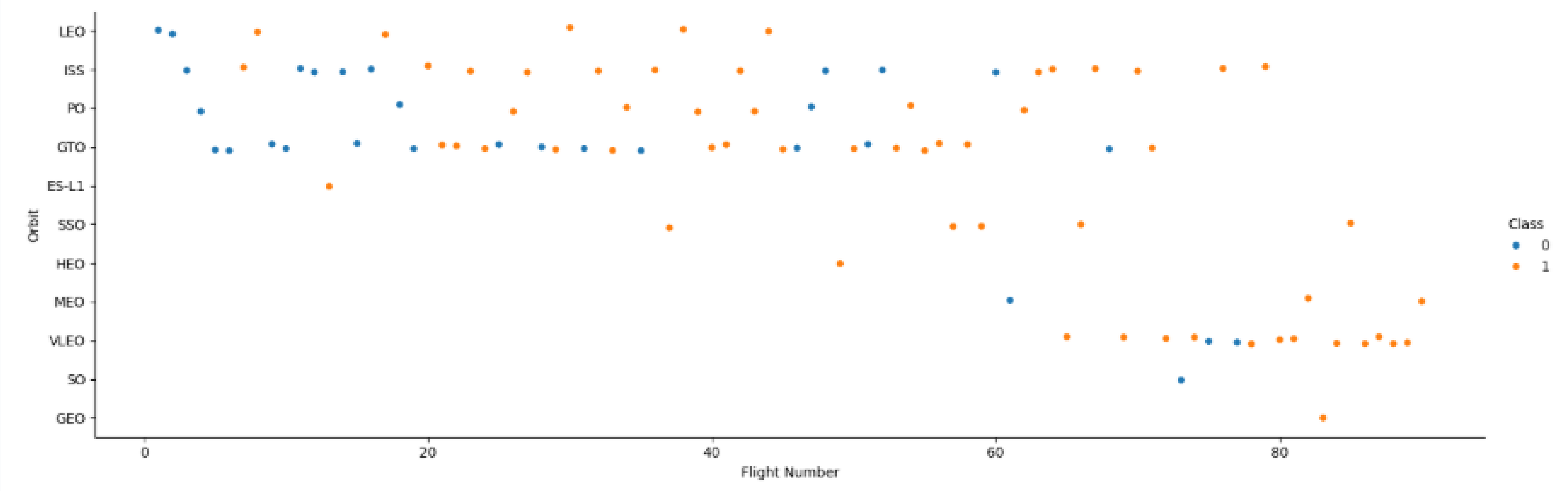
- Payloads exceeding 9,000 kg (roughly the weight of a school bus) show a very high success rate.
- Payloads above 12,000 kg appear to be launched exclusively from CCAFS SLC-40 and KSC LC-39A.

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

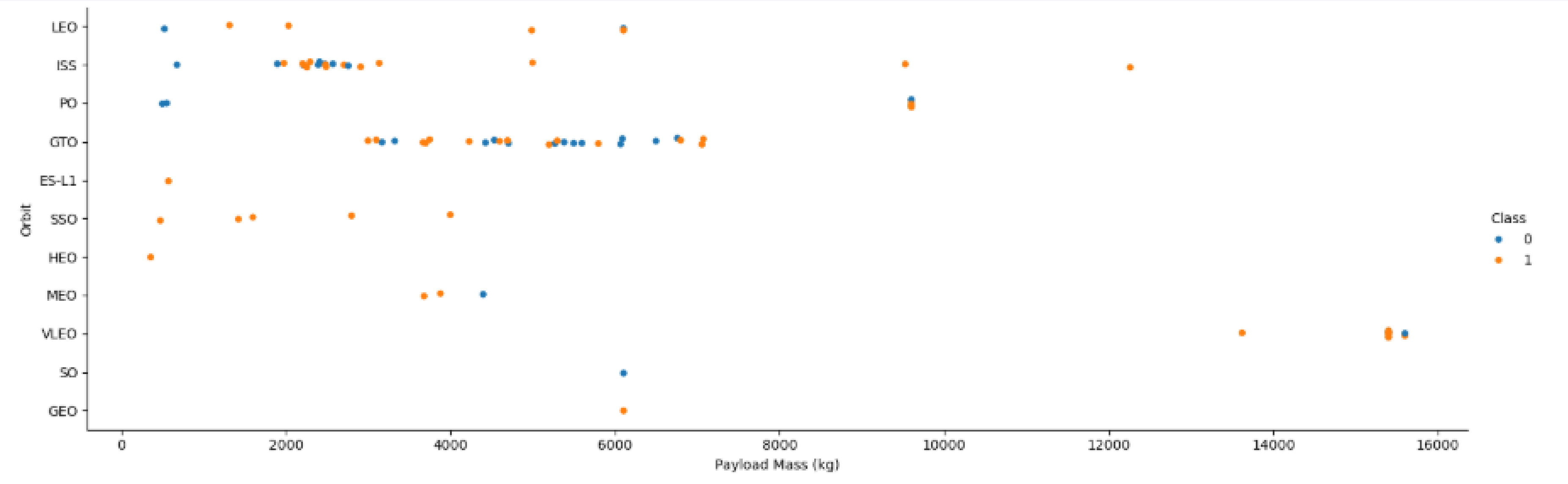


Flight Number vs. Orbit Type



- For LEO orbits, success rates are linked to the number of flights,
- For GTO orbits, there is no apparent correlation between flight number and success.

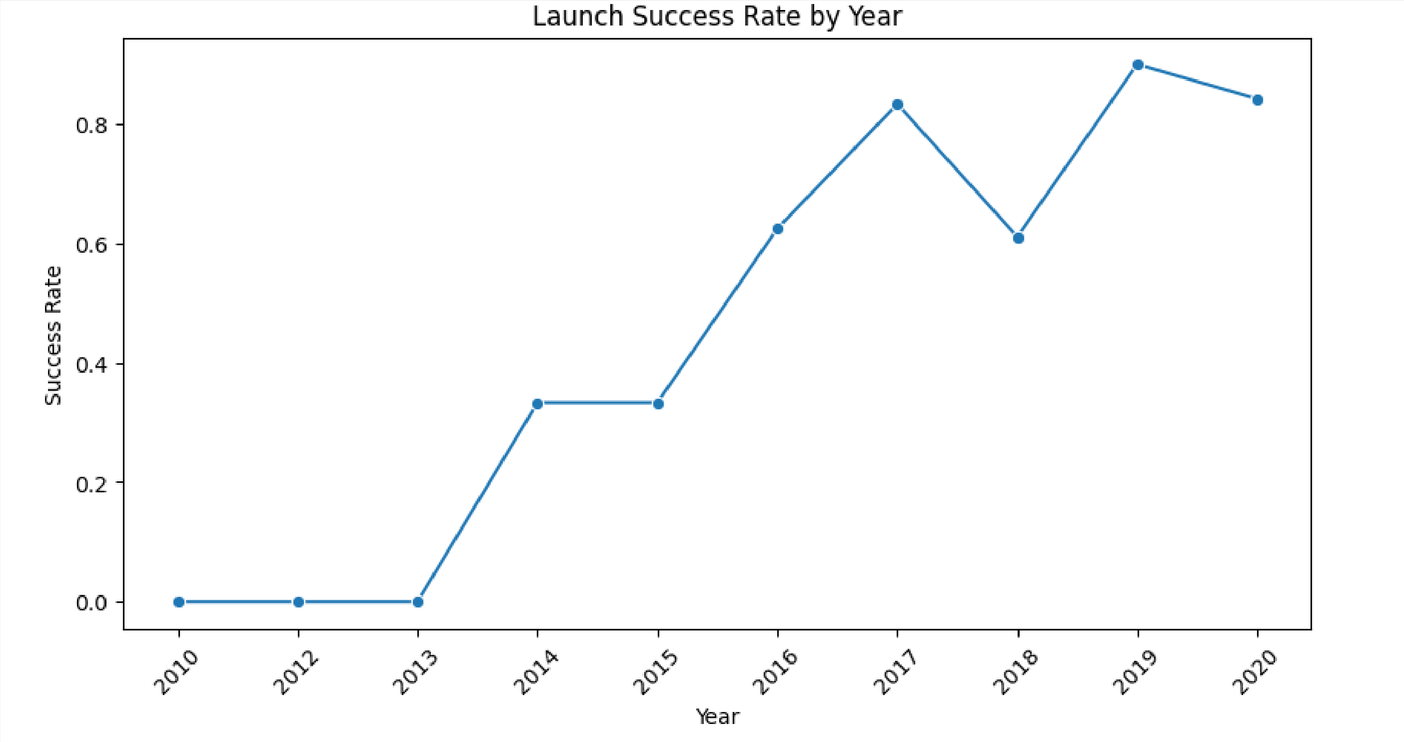
Payload vs. Orbit Type



- There is no clear relationship between payload and success rate for GTO orbits.
- The ISS orbit accommodates the widest payload range and maintains a high success rate.
- Launches to SO and GEO orbits are relatively rare.

Launch Success Yearly Trend

- The success rate began rising in 2013 and continued to improve through 2020.
- The initial three years appear to have been a period of adjustments and technological advancements.



All Launch Site Names

- There are four launch sites from which SpaceX launches rockets.

```
In [11]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

In [13]:
```
%%sql
SELECT * FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

Out[13]:

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

- The total payload carried by boosters from NASA

In [14]:
```
%%sql
SELECT SUM(Payload_Mass__kg_) AS TotalPayloadMass
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

Out[14]:

TotalPayloadMass
45596

Average Payload Mass by F9 v1.1

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

In [15]:
```
%%sql
SELECT AVG(Payload_Mass__kg_) AS AvgPayloadMass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
```

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

Out[15]:

AvgPayloadMass
2928.4

First Successful Ground Landing Date

- The first successful landing outcome on ground pad

```
In [16]: %%sql
         SELECT MIN(Date) AS FirstSuccessLandingDate
         FROM SPACEXTABLE
         WHERE Landing_Outcome LIKE '%Success%' AND Landing_Outcome LIKE '%ground pad%';

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

Out[16]:

FirstSuccessLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [17]:

```
%%sql
SELECT DISTINCT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%Success%' AND Landing_Outcome LIKE '%drone ship%'
AND Payload_Mass__kg_ > 4000 AND Payload_Mass__kg_ < 6000;
```

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

Out[17]:

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

Total Number of Successful and Failure Mission Outcomes

- Total number of landing outcomes

In [18]:
```
%%sql
SELECT Landing_Outcome, COUNT(*) AS Count
FROM SPACEXTABLE
GROUP BY Landing_Outcome;
```

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

Out[18]:

Landing_Outcome	Count
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- The names of the boosters which have carried the maximum payload mass

```
In [19]:
%%sql
SELECT Booster_Version, Payload_Mass__kg_
FROM SPACEXTABLE
WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTABLE);

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

Out[19]:

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

- Failed landing outcomes in drone ship in 2015

In [20]:
```
%%sql
SELECT
    substr(Date, 6, 2) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 1, 4) = '2015'
AND Landing_Outcome LIKE '%Failure%'
AND Landing_Outcome LIKE '%drone ship%';
```

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

Out[20]:

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

In [21]:

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

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

Out[21]:

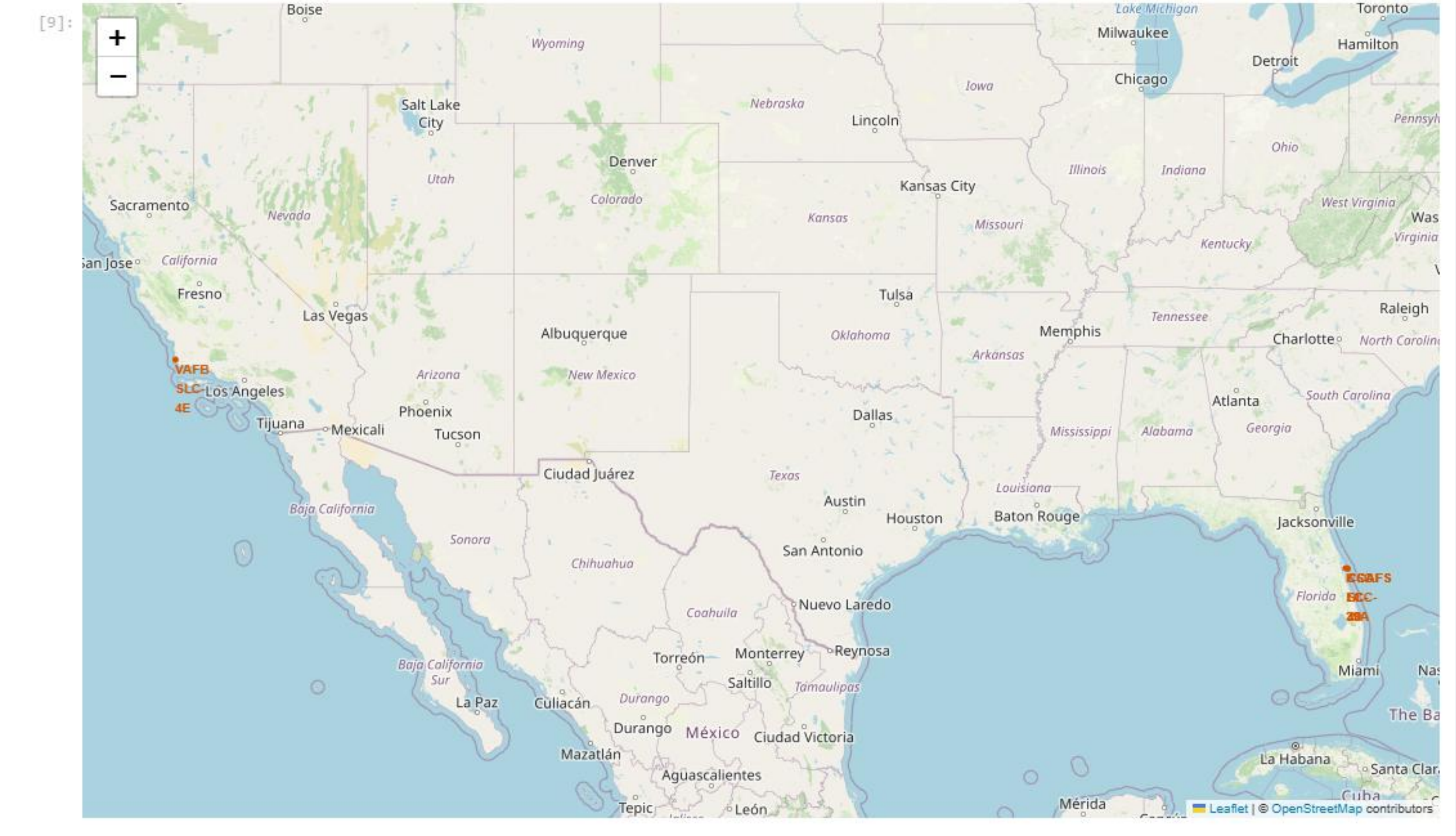
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

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

Section 3

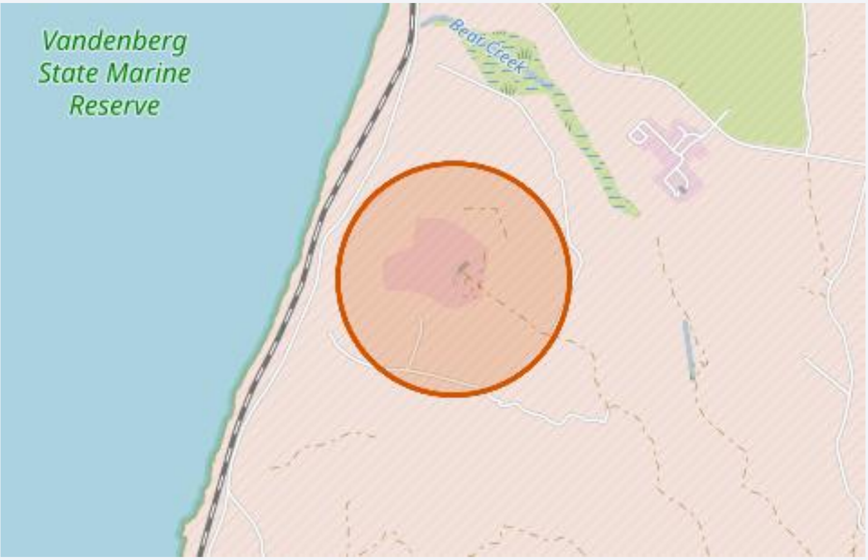
Launch Sites Proximities Analysis

SpaceX launch sites



- Location of four launch sites from which SpaceX launches rockets.

Markers of launch sites

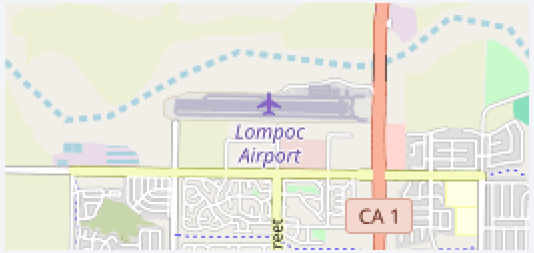


West coast

East coast

Landmarks of launch sites

Vandenberg Space Launch Complex



Lompoc Airport

Cape Canaveral Air Force Station

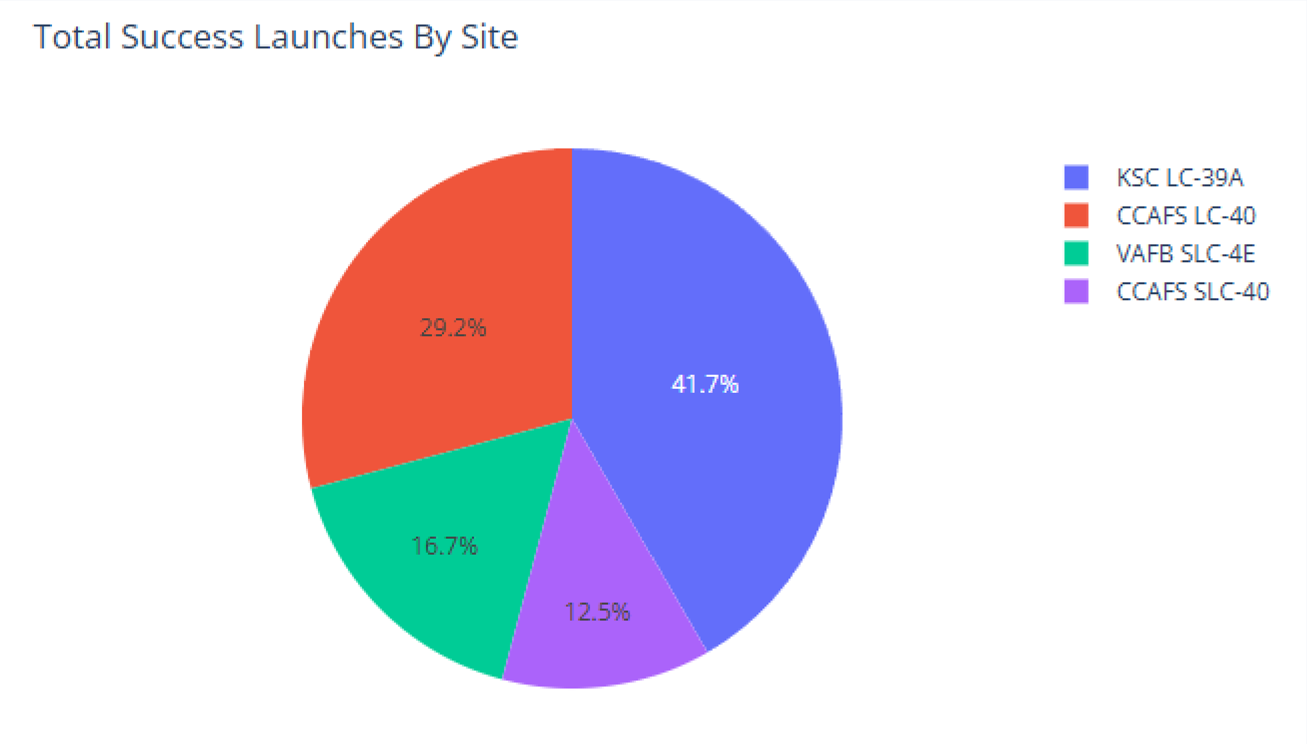
Cape Canaveral SFS Skid Strip

Section 4

Build a Dashboard with Plotly Dash

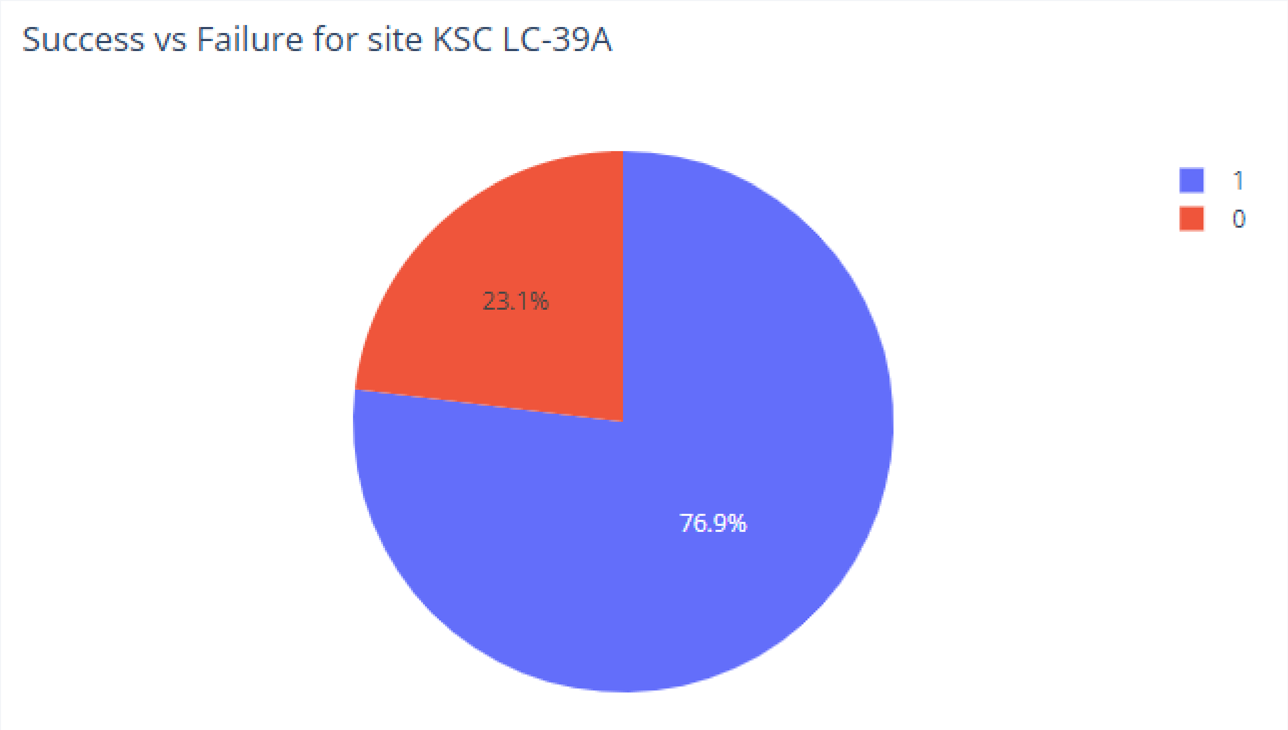
Successful Launches by Site

- The largest number of successful launches from KSC LC-39A

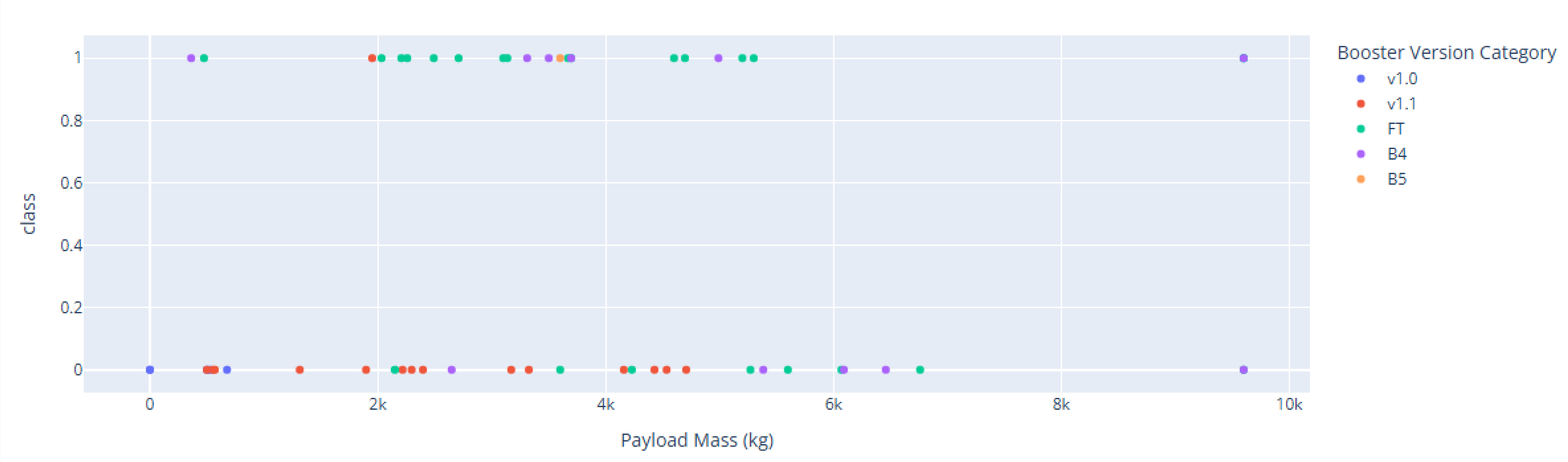


Launches from KSC LC-39A

- KSC LC-39A has 76.9% launch success rate



Payload and Success Correlation



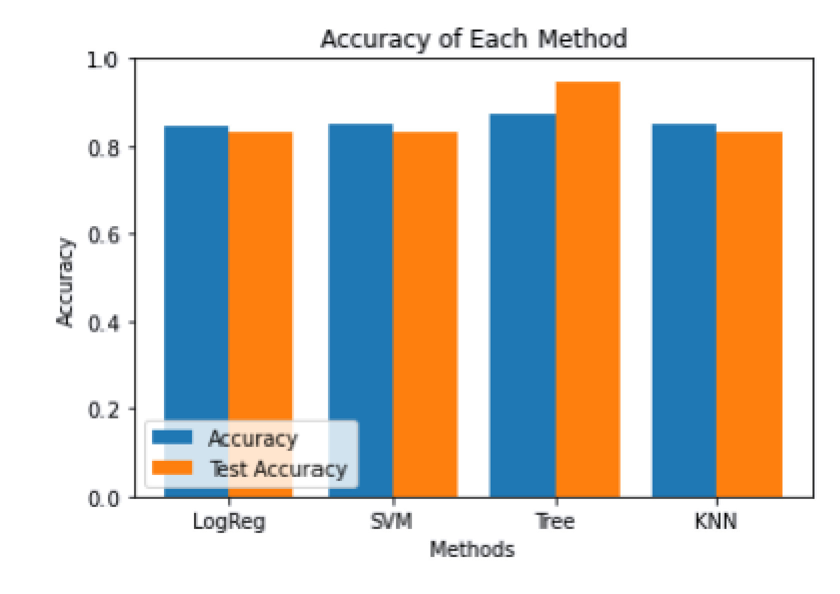
Most successful: payloads < 6,000 kg with FT boosters

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Four models were trained.
- Decision Tree classifier showed the highest accuracy.

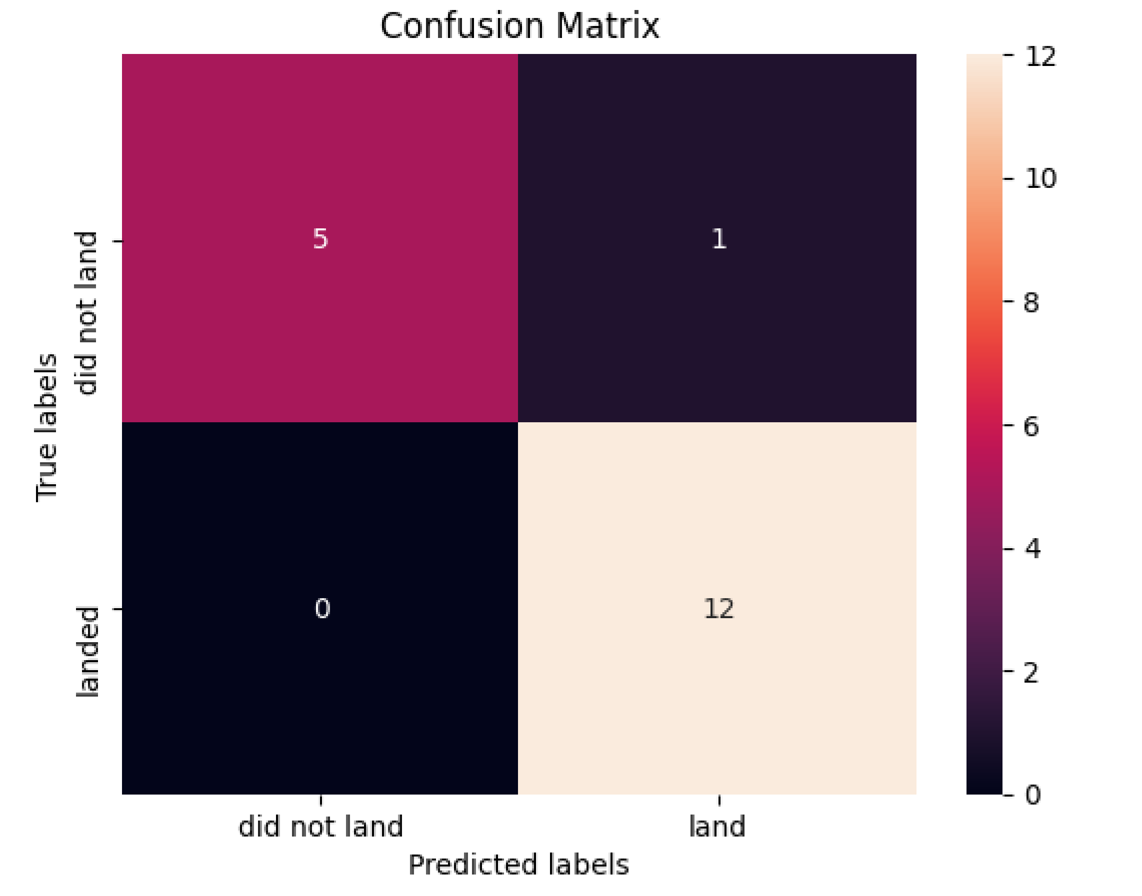


Confusion Matrix

There are:

- 12 TP
- 5 TN
- 1 FP
- 0 FN

Good result



Conclusions

- Multiple data sources were analyzed, enabling progressive refinement of insights.
- The most reliable launch site is CCAFS SLC-40, with the highest recent success rates.
- Payloads under 6,000 kg combined with FT boosters show the greatest success.
- Launch success rates have steadily improved since 2013, reflecting advancements in technology and processes.
- Certain orbits, such as ISS and LEO, show higher success rates compared to less common orbits like SO and GEO.
- Among the tested models, the Decision Tree classifier demonstrated the best accuracy for predicting successful landings, making it a valuable tool for operational planning.

Appendix

- Additional materials on GitHub: https://github.com/EgorTatarnikov/Coursera_IBM_DS_Final_Project

Thank you!