Winning Space Race with Data Science

BK

November 19, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API
  - Data collection with webscraping
  - Data wrangling
  - Exploratory data analysis with SQL
  - Exploratory data analysis with data visualization
  - Interactive visual analytics with Folium
  - Machine learning prediction
- Summary of all results
  - Exploratory data analysis result
  - Interactive analytics in screenshots
  - Predictive analytics result from machine learning lab

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - The price of each launch.
  - When SpaceX will reuse the first stage.
  - Using machine learning models and public information to answer the questions.

Section 1

Methodology

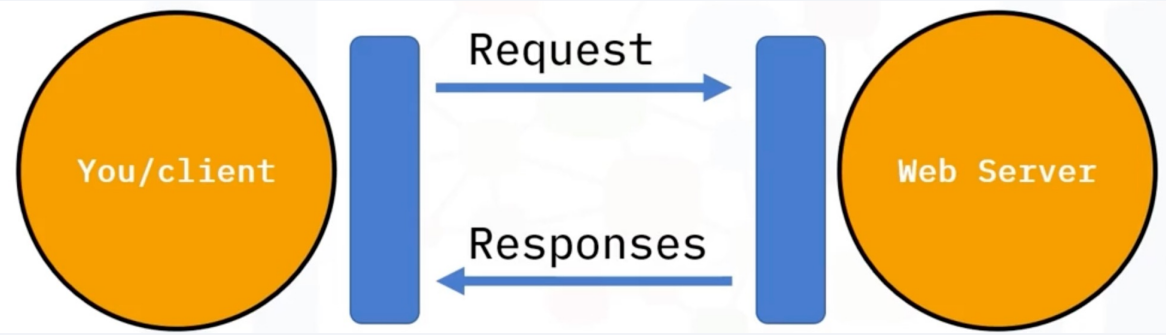
Methodology

Executive Summary

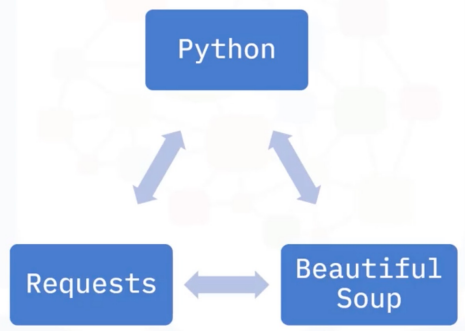
- Data collection methodology:
  - REST API and Webscraping.
- Perform data wrangling
  - One-hot encoding for machine learning algorithm.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The first dataset was colleted by REST API using the Space X API endpoint.



- The second dataset was collectecd by webscraping to a specific page of Wikipedia.



Data Collection – SpaceX API

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

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

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

```
# Lets take a subset of our dataframe keeping only the features we want and the flight
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 ext
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value i
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Request rocket launch data from SpaceX API

Use json_normalize method to convert the json result into a dataframe

Perform data wrangling to the dataframe

Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

```
# use requests.get() method with the provided static_url
# assign the response to a object
html = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a respons
soup = BeautifulSoup(html, "html5lib")
```

```
# Use the find_all function in the BeautifulSoup object, with element typ
# Assign the result to a list called `html_tables`
html_tables = soup.find_all(name = 'table')
```

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

...

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

Perform a HTTP GET method to request the Falcon 9 launch page

Create an object from the HTML response

Perform data wrangling to the dataframe

Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-webscraping.ipynb

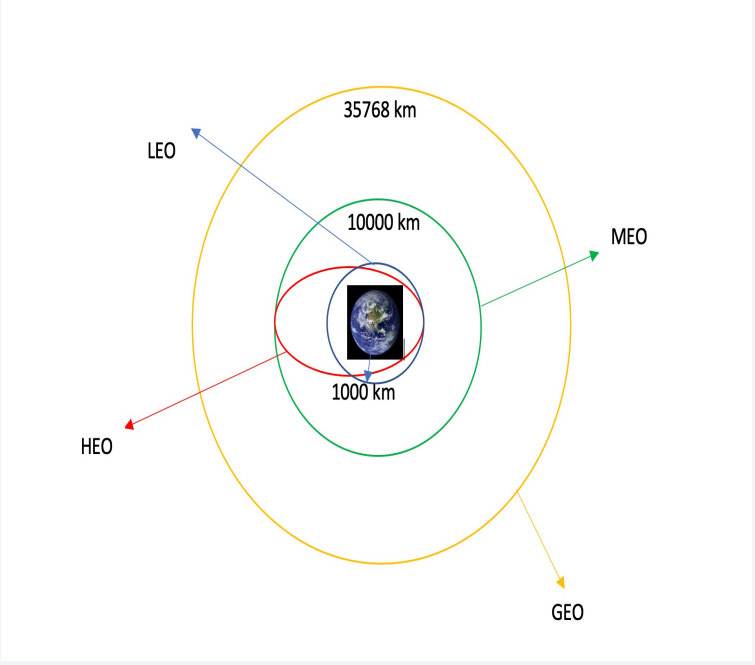
Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident.

For example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

We will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

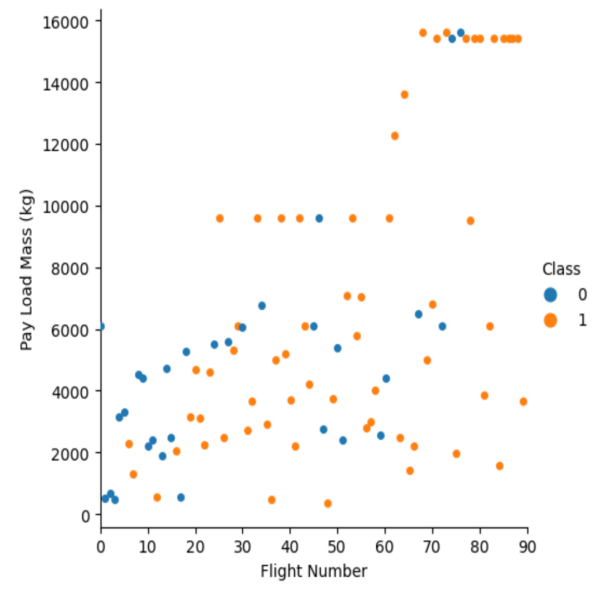
Each launch aims to an dedicated orbit, and here are some common orbit types:

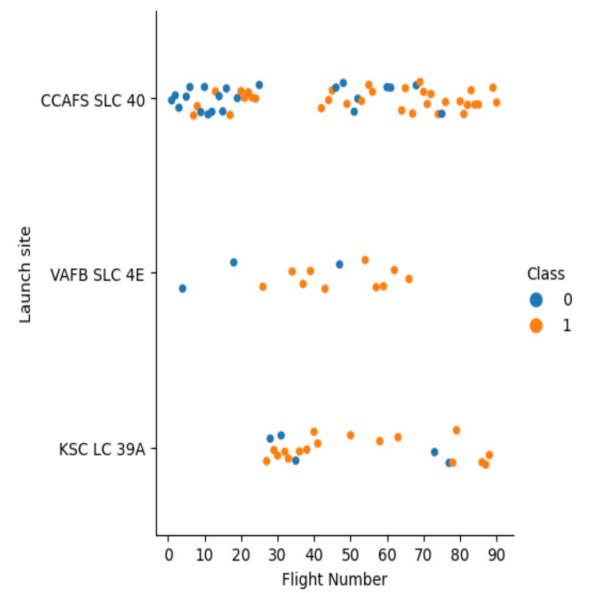


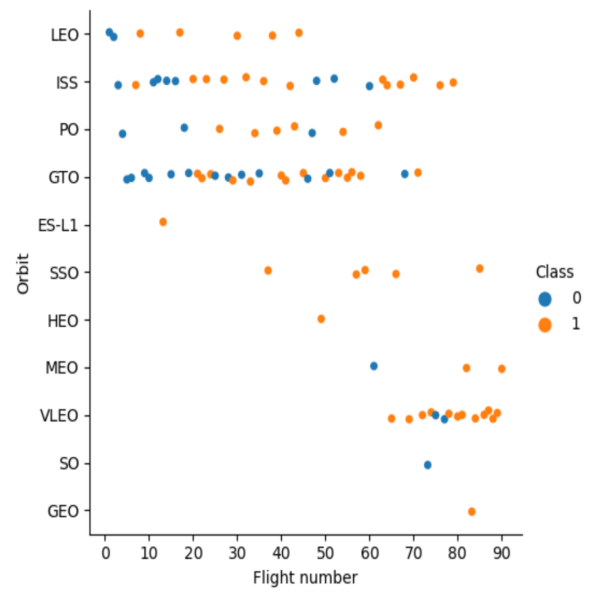
Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

The scatter plots show the relation between two variables.



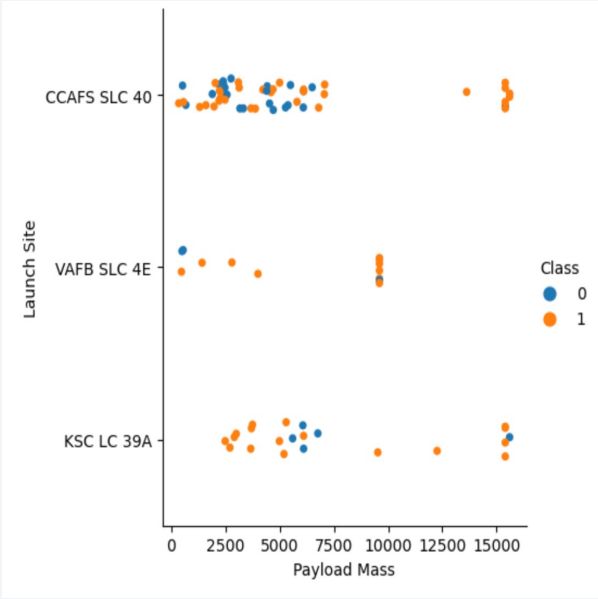


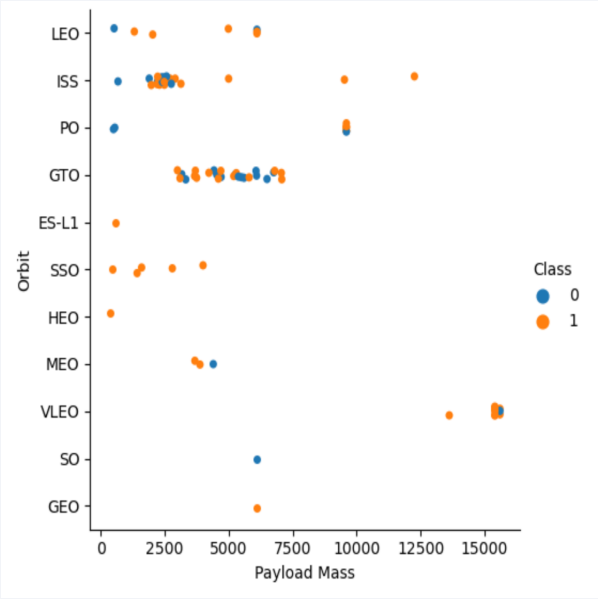


Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization

The scatter plots shows if there is any correlation between variables.

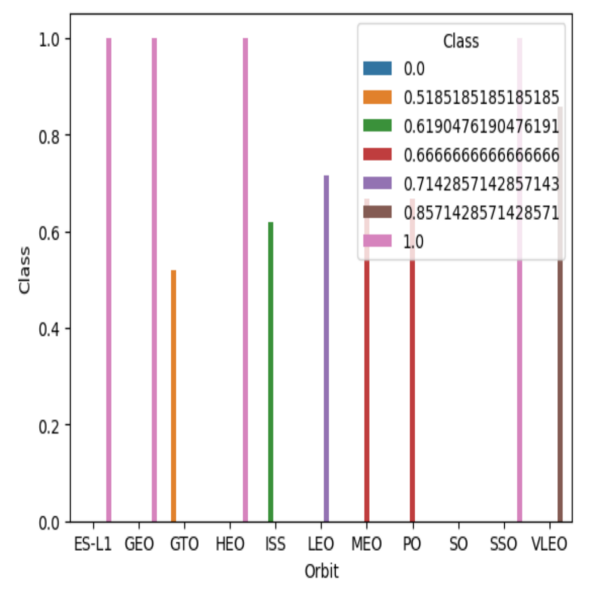


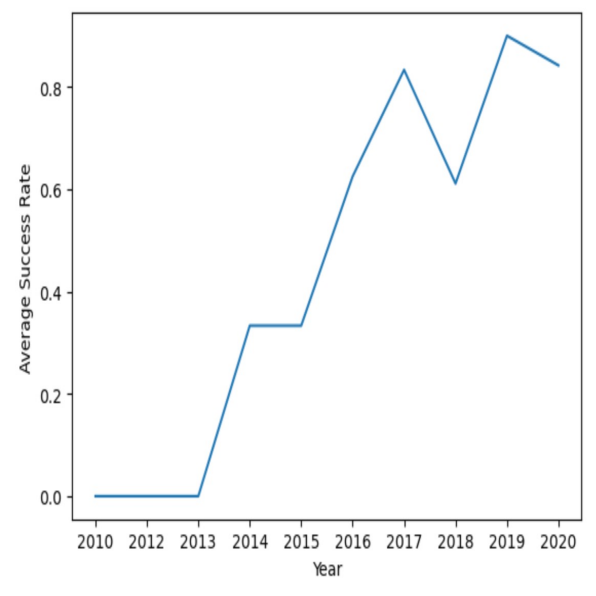


Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization

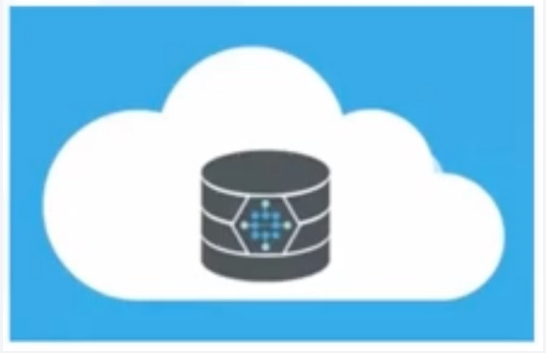
The bar plot shows the success rate in percentage of each orbit type.





Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-eda-dataviz.ipynb

EDA with SQL



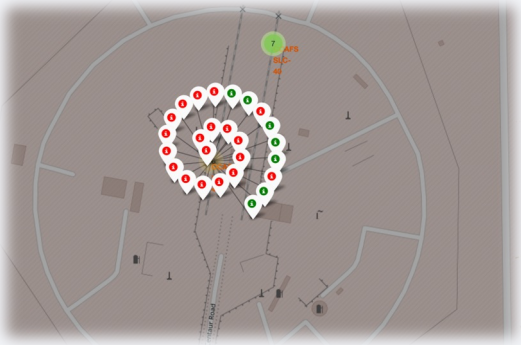
SQL queries performed:

- Displaying the names of the unique launch sites in the space mission.
- Displaying five records where launch sites begin with the string: CCA.
- Displaying the total payload mass carried by boosters launched by NASA (CRS).
- Displaying the average payload mass carried by booster version: F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was acheived.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- 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.

Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- First, the locations of the launch site were marked on the interactive map by circles to find them easy. The coordinates were the latitude and the longitude and each circle was labeled by the name of the the associated launch site.
- Next, markers for all launch records were created. If a launch was successful (class = 1), then we use a green marker and if a launch was failed, we use a red marker (class = 0),
- Last, we calculated the distance between a launch site to its proximities using a line to:
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
  - Do launch sites keep certain distance away from cities?



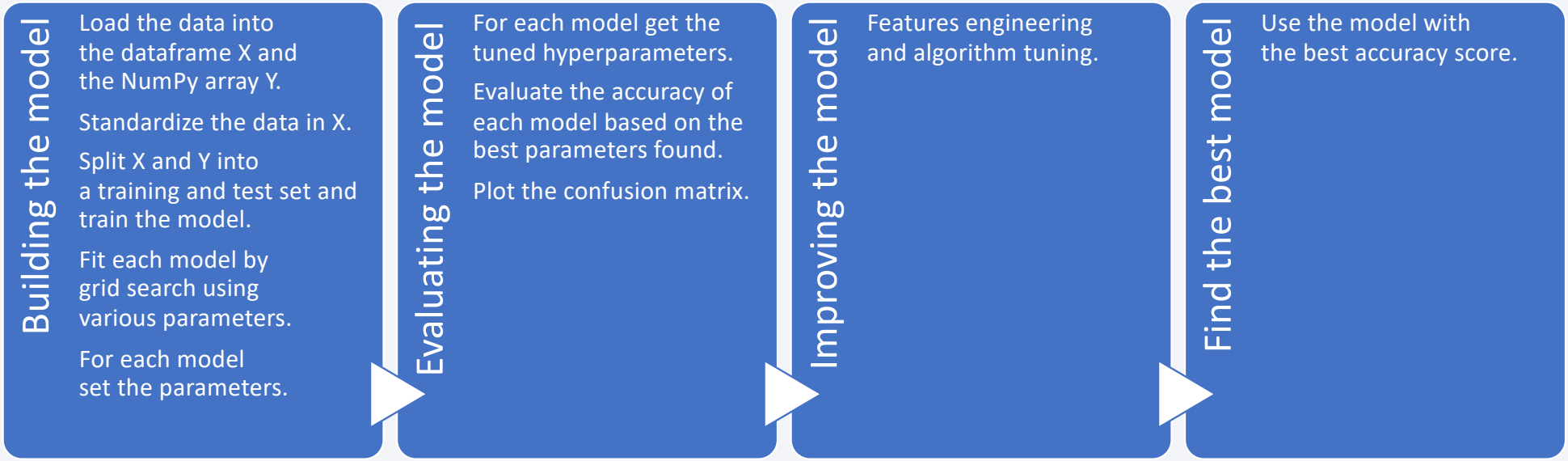
Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard provides on demand data analysis of the launch records.
- A pie plot shows the successful launches and can be filtered to plot the total of all sites or to plot an individual site. The colors represents the launch site which can be taken from the legend on the chart.
- A scatter plot shows the relationship between the payload mass (kg) and success rate where one means success and zero means failure. A range slider provides the configuration of the payload mass (kg) to zoom closer into the graph. The colors represents the booster version which can be taken from the legend on the graph.

Source: https://github.com/starfile/applied-ds-capstone/blob/main/dash_app/spacex_dash_app.py

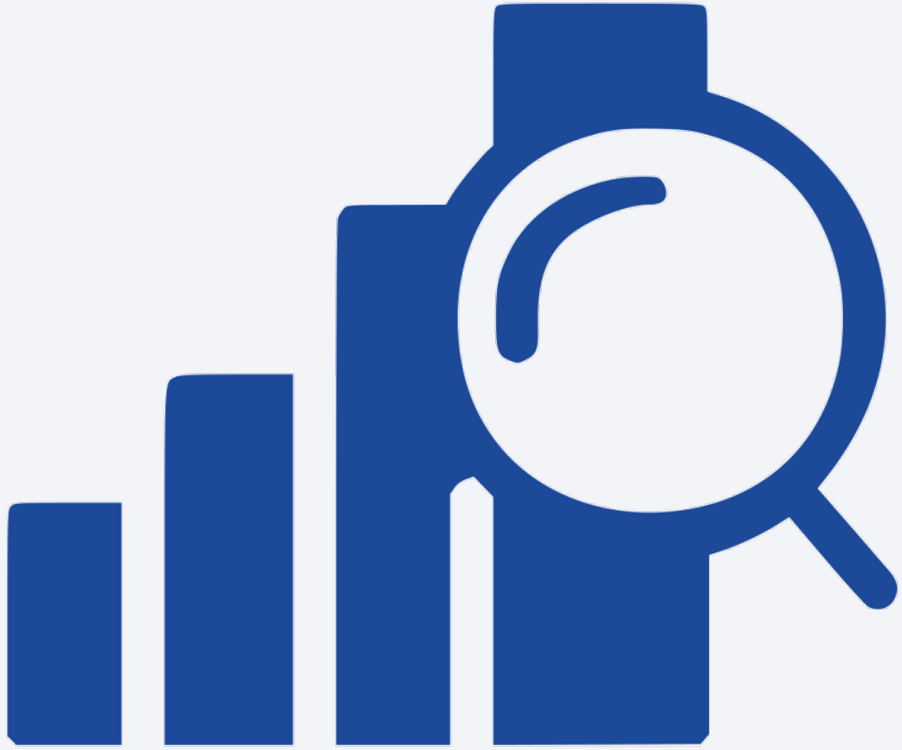
Predictive Analysis (Classification)

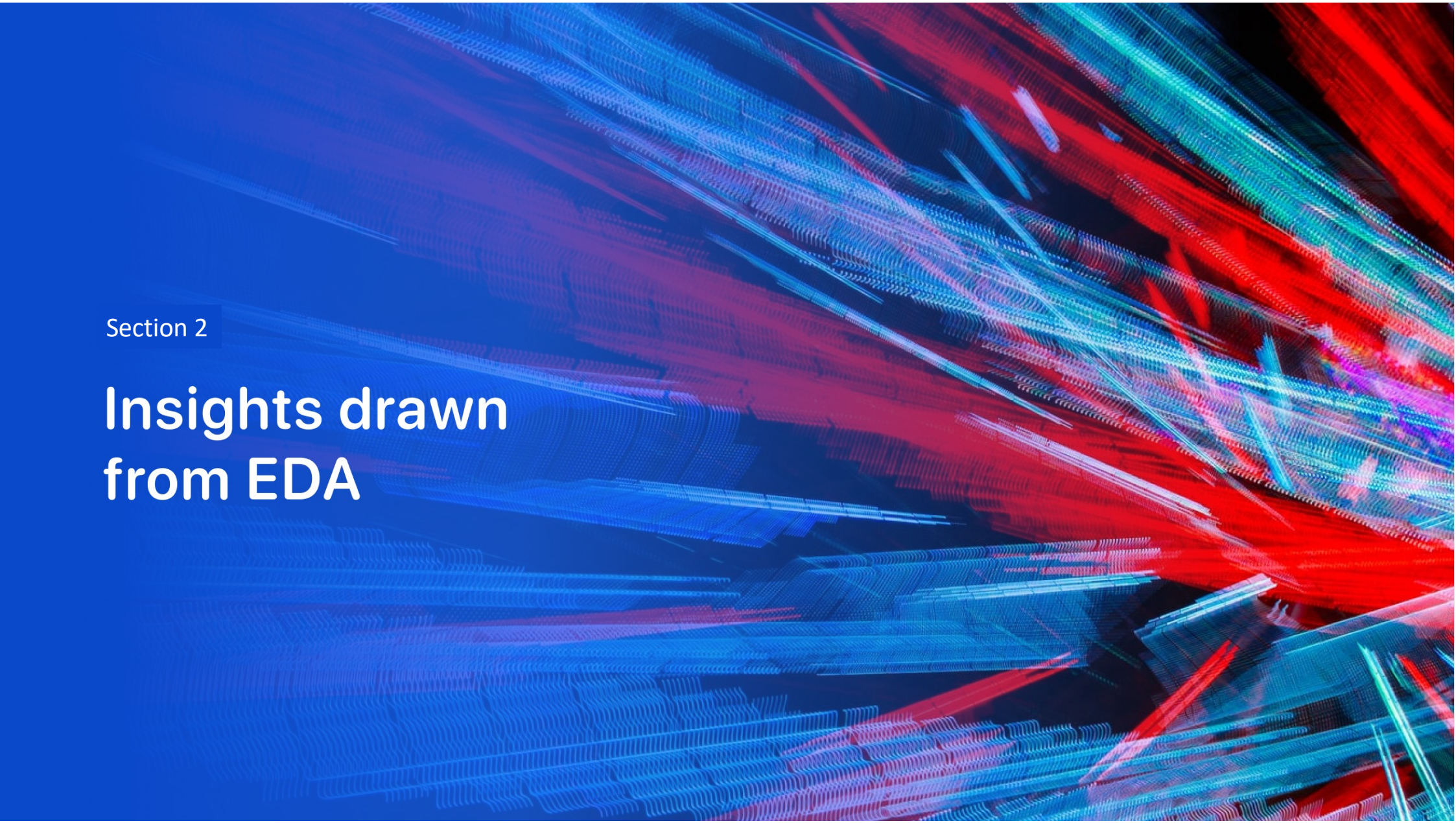


Source: https://github.com/starfile/applied-ds-capstone/blob/main/notebooks/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



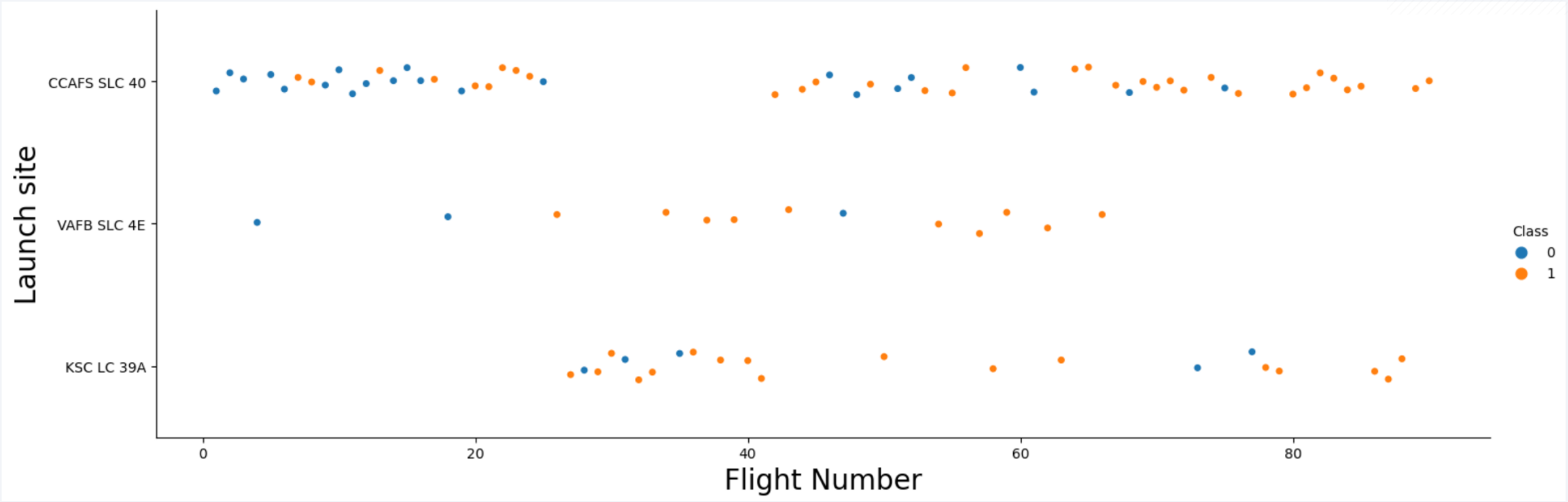


Section 2

Insights drawn from EDA

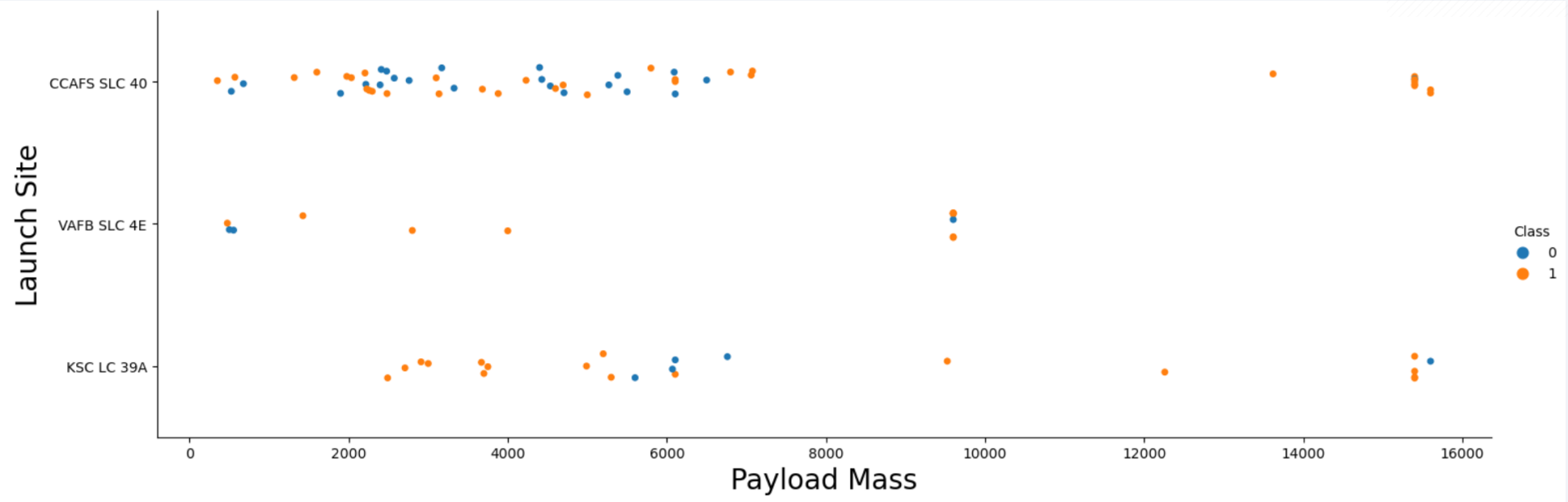
Flight Number vs. Launch Site

- The success rate increases if the number of flights increase.
- Especially the launch site KSC LC 39A has a high success rate.



Payload vs. Launch Site

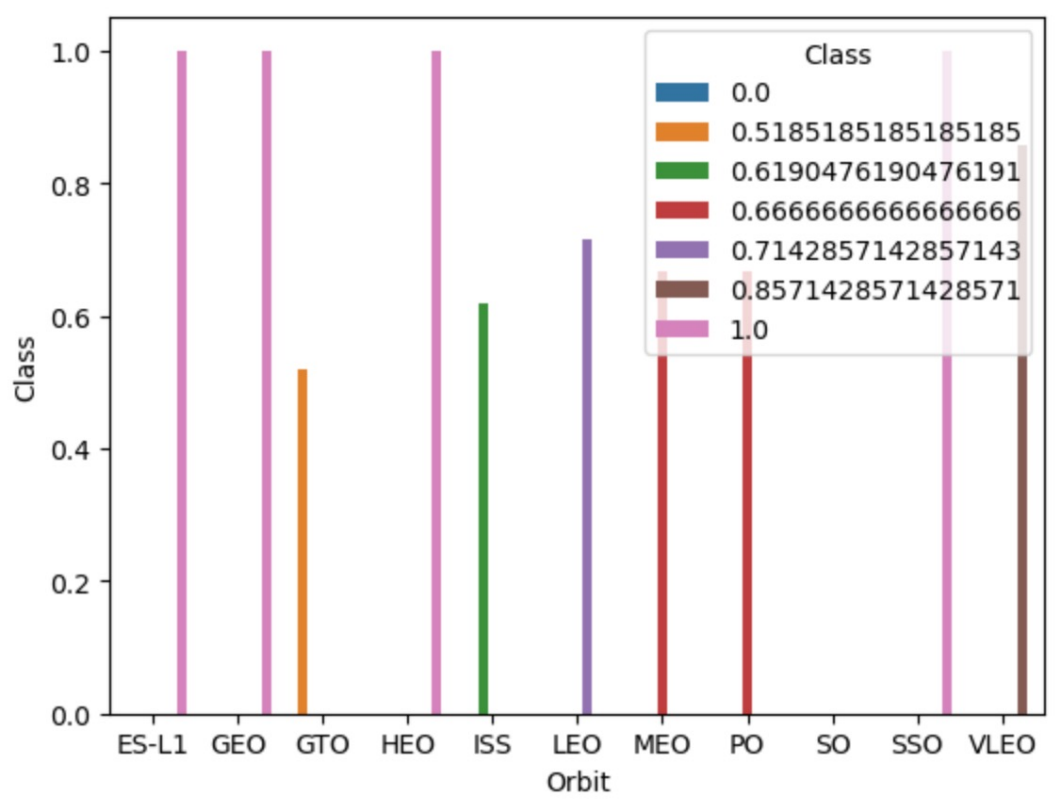
- Higher payload of the rocket indicates higher success rate.
- The reason for this assumption cannot be determined from the plot.



Success Rate vs. Orbit Type

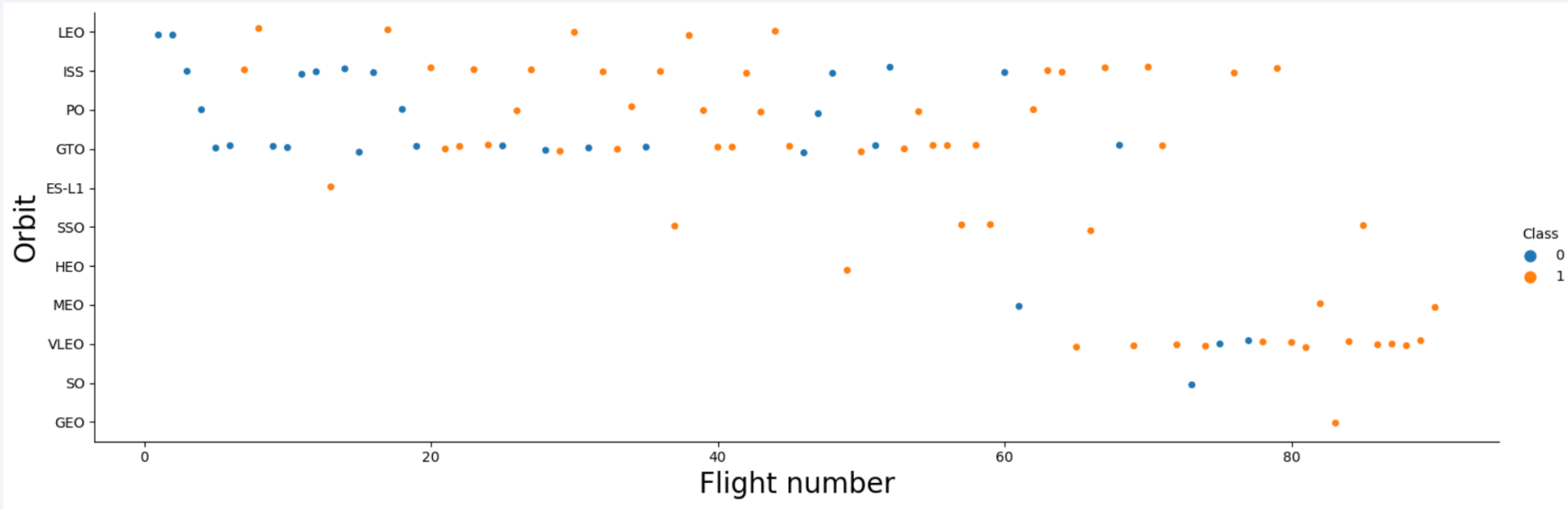
Orbits with best success:

- GEO
- ES-L1
- HEO
- SSO



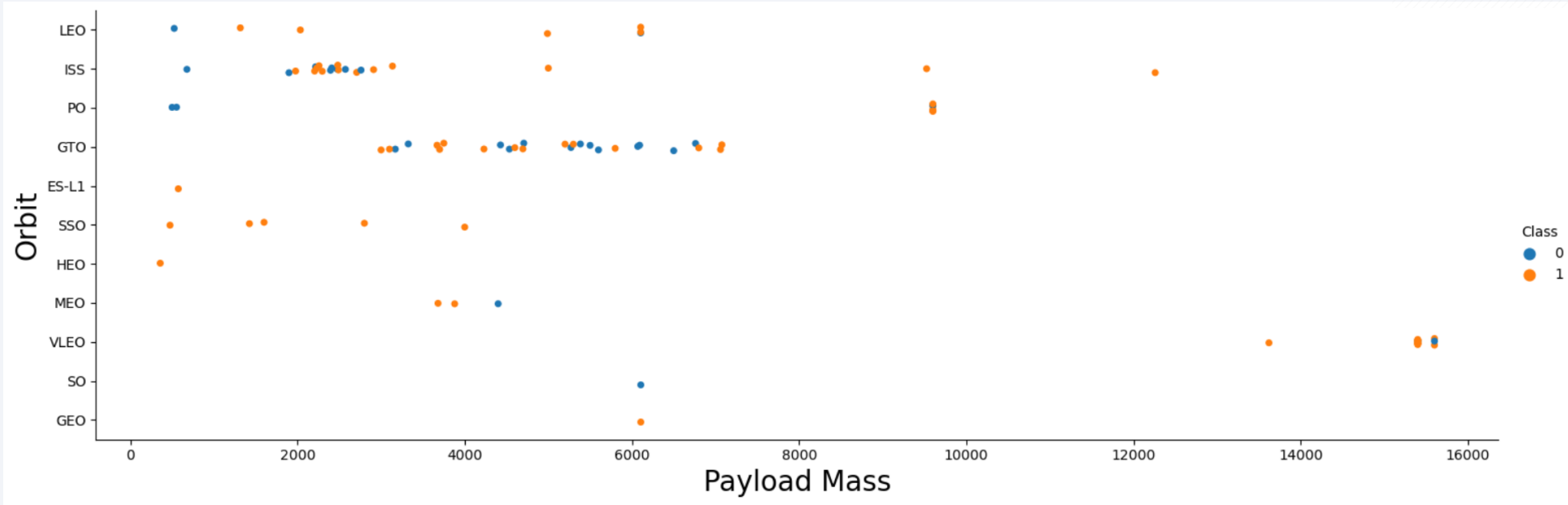
Flight Number vs. Orbit Type

- The SSO orbit had 100% success rate.
- The success rate increases related to the flight numbers for orbit: LEO & PO.



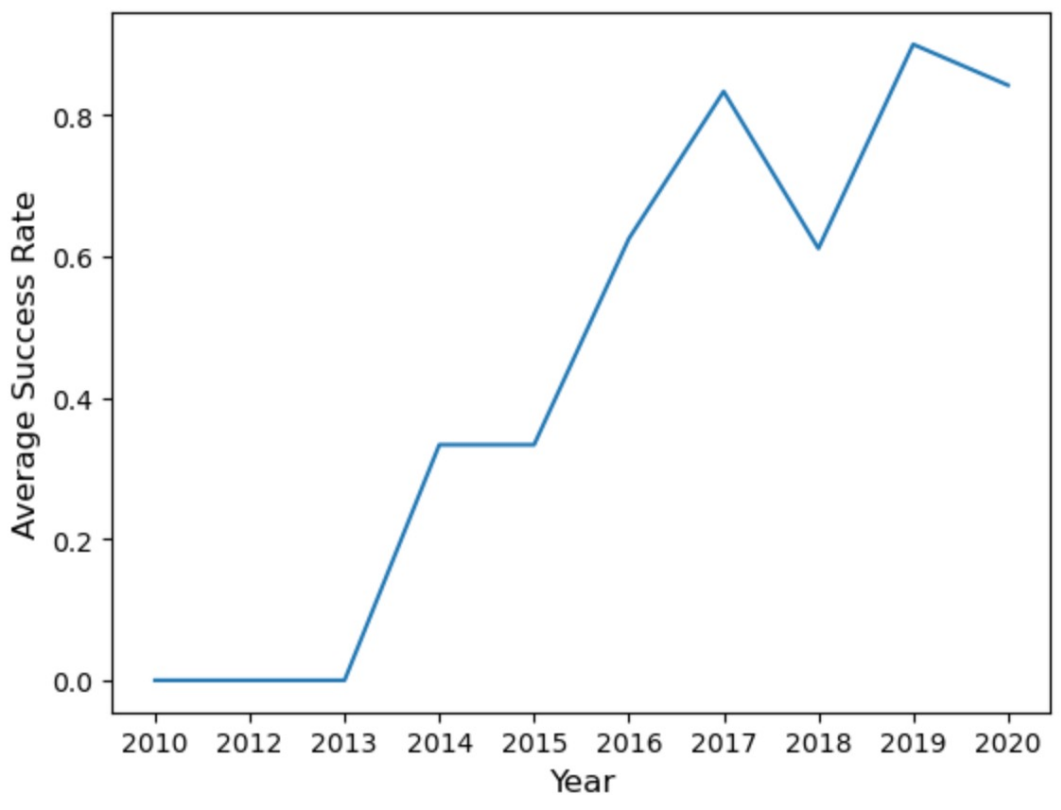
Payload vs. Orbit Type

- Higher payload has lower success rate on GTO orbit.
- Higher payload has better success rate on orbit: LEO & ISS.



Launch Success Yearly Trend

Since 2013 the average success rate increases. From 2017 and 2018 the average success rate dropped but could recover from 2018 to 2019.



All Launch Site Names

Display the names of the unique launch sites in the space mission:

```
%sql select DISTINCT LAUNCH_SITE as UNIQUE_LAUNCH_SITE from SPACEX;
```

The SQL query selects the LAUNCH_SITE column from the database. The keyword DISTINCT ensures that only unique values are in the column. The keyword AS gives the name UNIQUE_LAUNCH_SITE to the column.

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

Launch Site Names Begin with 'CCA'

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

```
%sql select * from SPACEX where LAUNCH_SITE like 'CCA%' limit 5;
```

The SQL query selects all columns from the database. The predicate LIKE in the WHERE clause ensures that the result set contains only those rows if the values of the column LAUNCH_SITE starts with the word 'CCA'. The keyword LIMIT restricting the number of rows in the result set to 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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Display the total payload mass carried by boosters launched by NASA (CRS):

```
%sql select sum(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD_MASS from SPACEX;
```

The SQL query tells the database to calculate the sum of the column PAYLOAD_MASS__KG_ and return only the result. The keyword AS gives the name TOTAL_PAYLOAD_MASS to the column.

total_payload_mass
619967

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1:

```
%sql select avg(PAYLOAD_MASS__KG_) as "AVERAGE_PAYLOAD_MASS_F9_V1.1" from SPACEX where BOOSTER_VERSION LIKE 'F9 v1.1%';
```

AVERAGE_PAYLOAD_MASS_F9_V1.1
2534

The SQL query tells the database to calculate the average of the column PAYLOAD_MASS__KG_ and return only the result. The predicate LIKE in the WHERE clause ensures that the calculation is only done for those rows if the values of the column BOOSTER_VERSION starts with the word 'F9 v1.1'. The keyword AS gives the name 'AVERAGE_PAYLOAD_MASS_F9_V1.1' to the column.

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was achieved:

```
%sql select min(DATE) as FIRST_SUCCESSFUL_LANDING from SPACEX where LANDING__OUTCOME = 'Success (ground pad)';
```

- The SQL query tells the database to get the minimum value of the column DATE. The WHERE clause ensures that the minimum value is only taken from those rows if the column LANDING__OUTCOME contains 'Success (ground pad)'. The keyword AS gives the name 'FIRST_SUCCESSFUL_LANDING' to the column.

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters (success in drone ship and payload mass 4000 ≤ x ≤ 6000):

```
%sql select BOOSTER_VERSION from SPACEX where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

The SQL query selects the BOOSTER_VERSION column. The result set has only those values if the column LANDING__OUTCOME is 'Success (drone ship)' and the column PAYLOAD_MASS__KG is between 4000 and 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

List the total number of successful and failure mission outcomes:

```
%sql select MISSION_OUTCOME, count(MISSION_OUTCOME) as TOTAL from SPACEX group by MISSION_OUTCOME;
```

The SQL query selects the MISSION_OUTCOME column and counts how often each value occurs using a second column named 'TOTAL'. The keyword GROUP BY ensures that the result set is properly summarized.

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

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass:

```
%sql select BOOSTER_VERSION, PAYLOAD_MASS__KG_ as MAX_PAYLOAD_MASS from SPACEX where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX);
```

The SQL query selects the BOOSTER_VERSION and PAYLOAD_MASS__KG columns. The WHERE clause uses a SQL subquery to get the maximum value of the PAYLOAD_MASS__KG column. The result set has only those rows if the PAYLOAD_MASS__KG column equals to this maximum value. The keyword AS gives the name 'MAX_PAYLOAD_MASS' to the column PAYLOAD_MASS__KG.

booster_version	max_payload_mass
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

List the failed landing_outcomes for in year 2015:

```
%sql select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, DATE from SPACEX where LANDING__OUTCOME = 'Failure (drone ship)' and YEAR(DATE) = 2015;
```

The SQL query selects the LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE and DATE columns. The result set has only those rows if the column LANDING__OUCOME equals to 'Failure (drone ship)' and the column DATE equals to 2015.

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

Rank the count of landing outcomes from 2010-06-04 to 2017-03-20:

```
%%sql
select LANDING__OUTCOME as LANDING_OUTCOME, COUNT(LANDING__OUTCOME) as TOTAL from SPACEX
where DATE between '2010-06-04' and '2017-03-20'
group by LANDING__OUTCOME
order by COUNT(LANDING__OUTCOME) DESC;
```

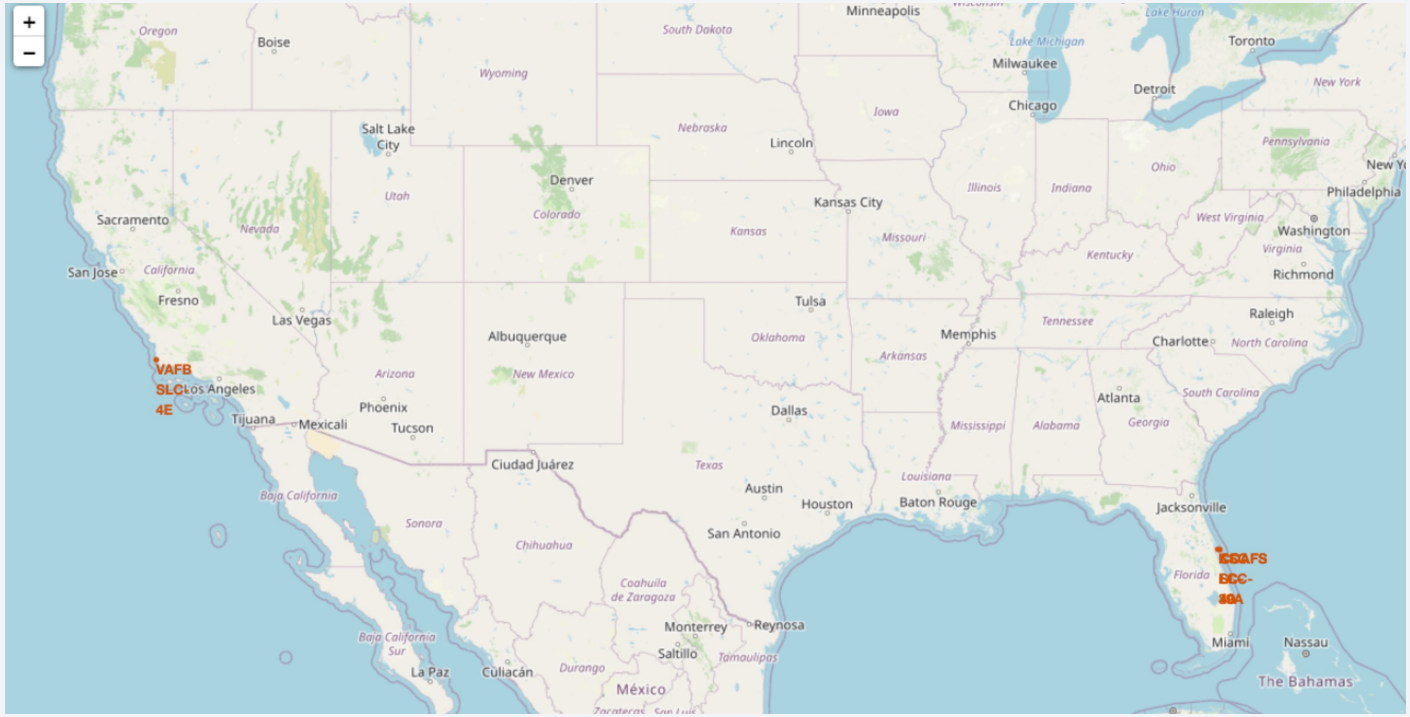
The SQL query selects the LANDING__OUTCOME column and counts how often each value occurs using a second column named 'TOTAL'. The keyword GROUP BY ensures that the result set is properly summarized. The keyword ORDER BY ensures that the result set is ordered descending using the keyword DESC.

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

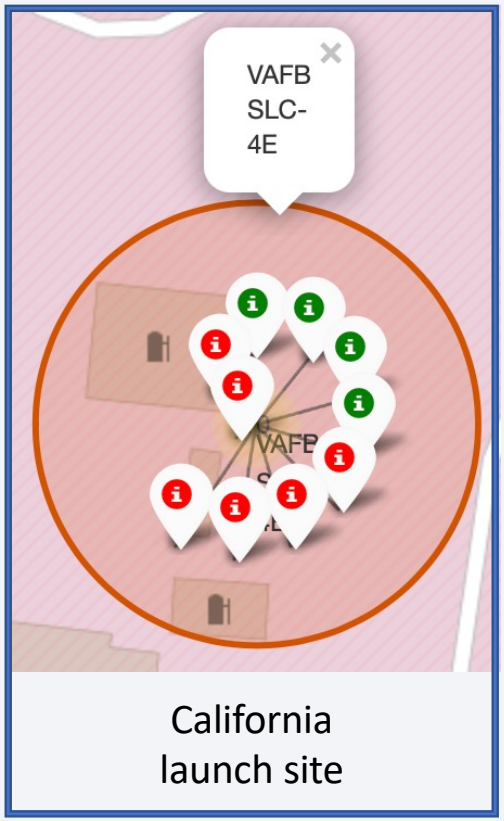

Section 3
Launch Sites
Proximities Analysis

Launch sites on a map

All launch sites are within the United States.



Success/Failed launches for each site on the map

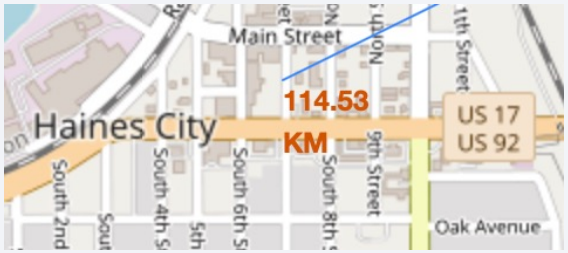


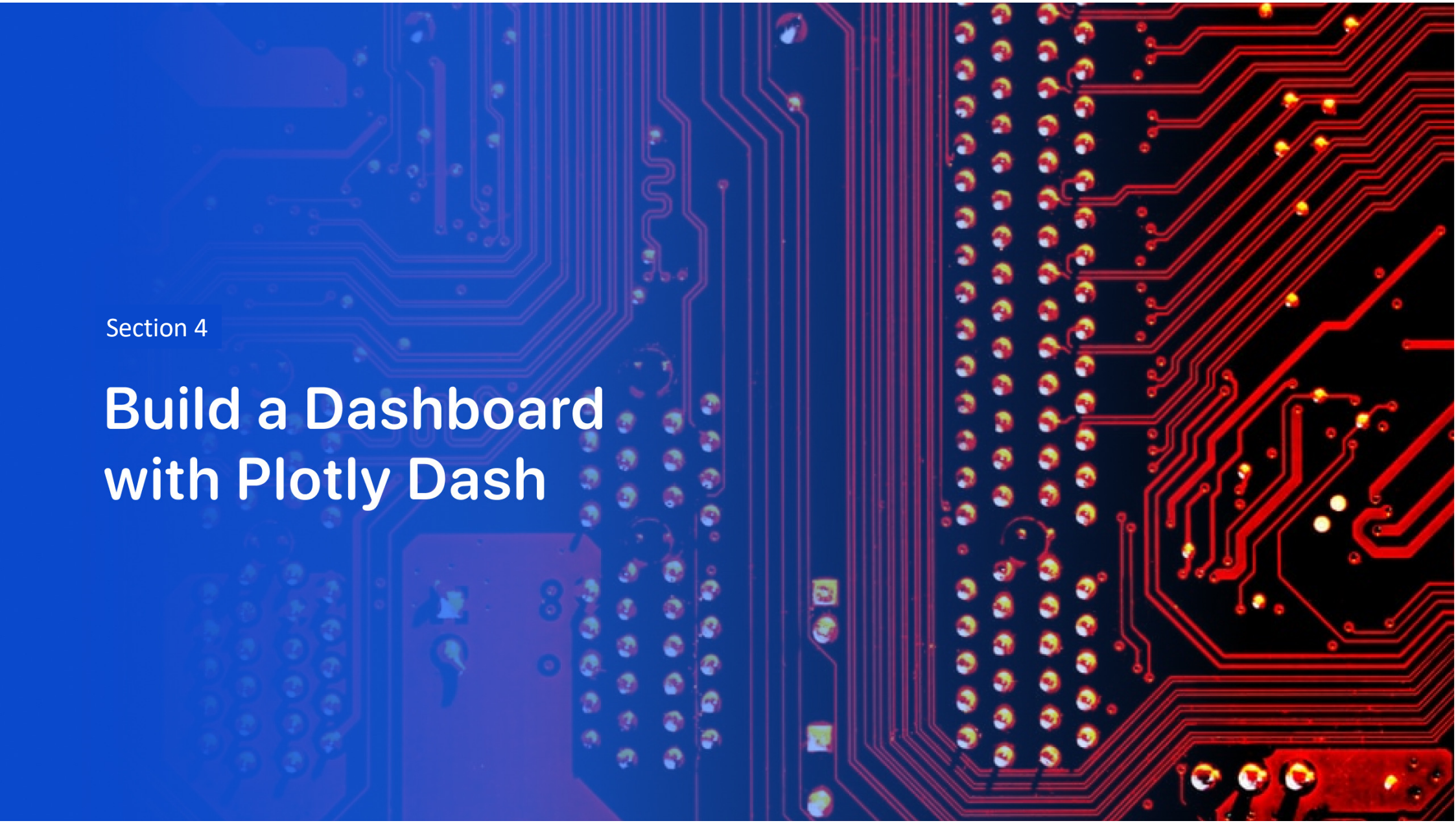
California
launch site

Florida
launch site

Distances between a launch site to its proximities

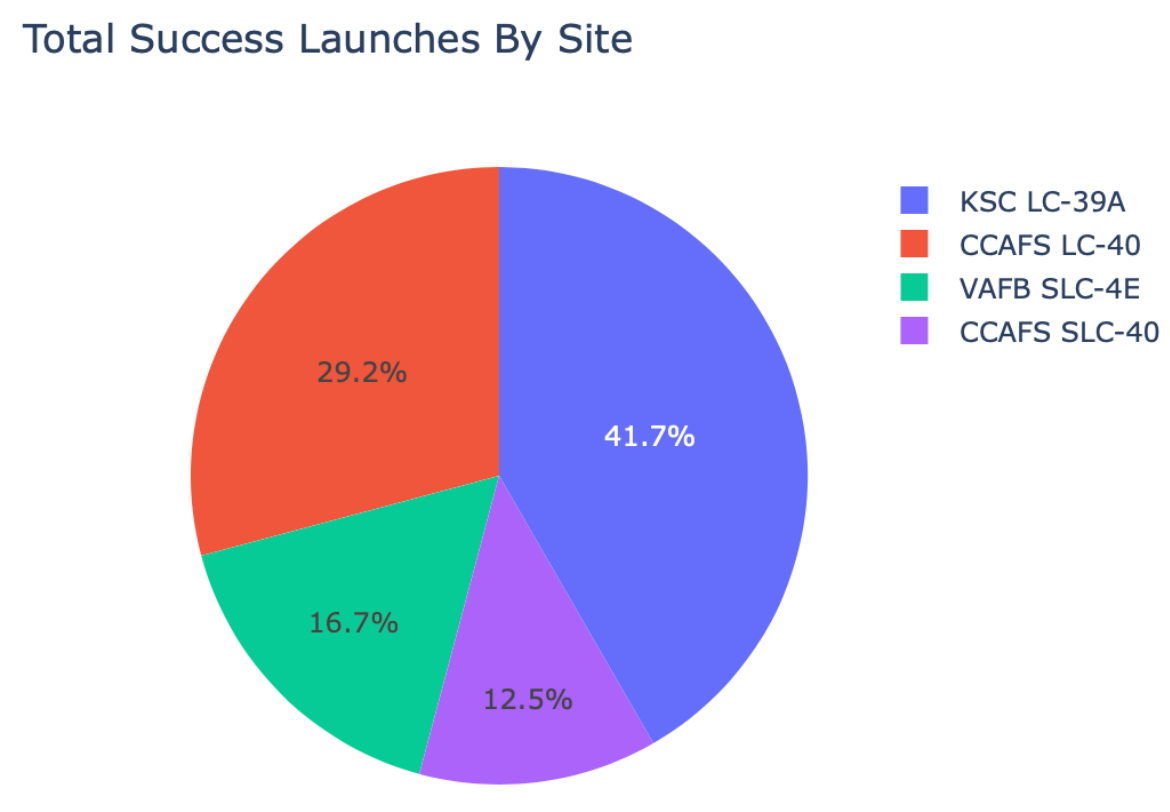
- Are launch sites in close proximity to railways? No.
- Are launch sites in close proximity to highways? No.
- Are launch sites in close proximity to coastline? No.
- Do launch sites keep certain distance away from cities? Yes.




Section 4
Build a Dashboard with Plotly Dash

Launch success count for all sites

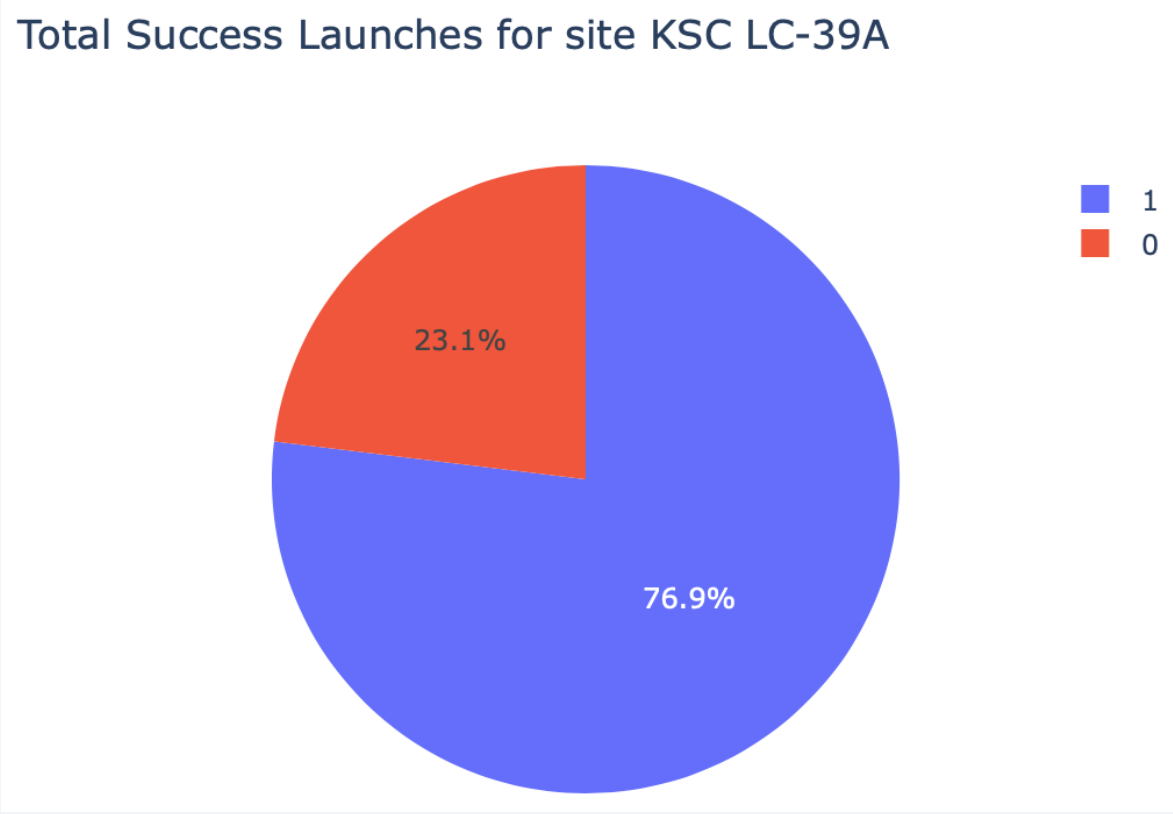
The launch site KSC LC-39A had the most success launches.



Launch site with highest launch success ratio

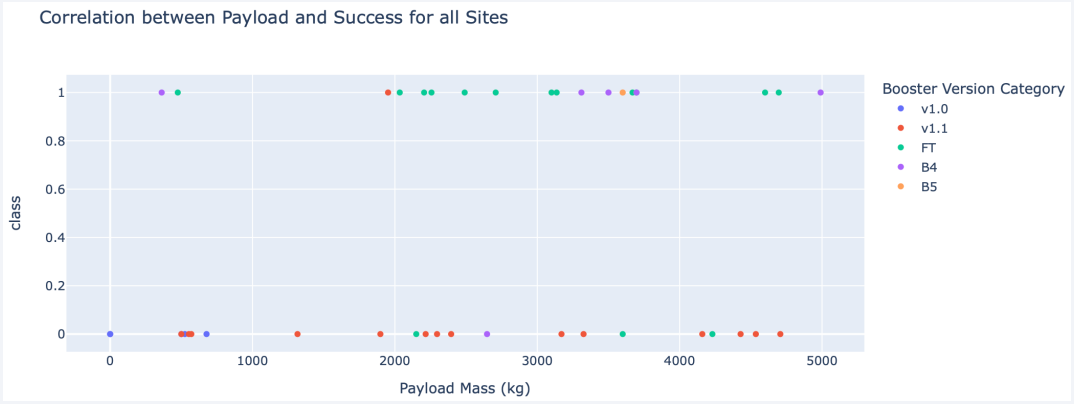
KSC LC-39A statistics:

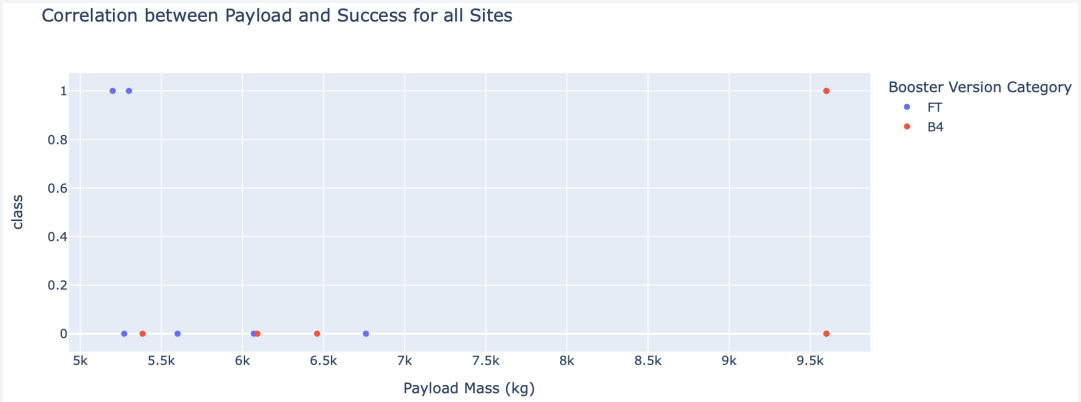
- 76.9% success rate
- 23.1% failure rate



Payload vs. Launch Outcome scatter plot for all sites

The success rate of the lower payloads is higher than the success rate of the higher payloads.





Section 5

Predictive Analysis (Classification)

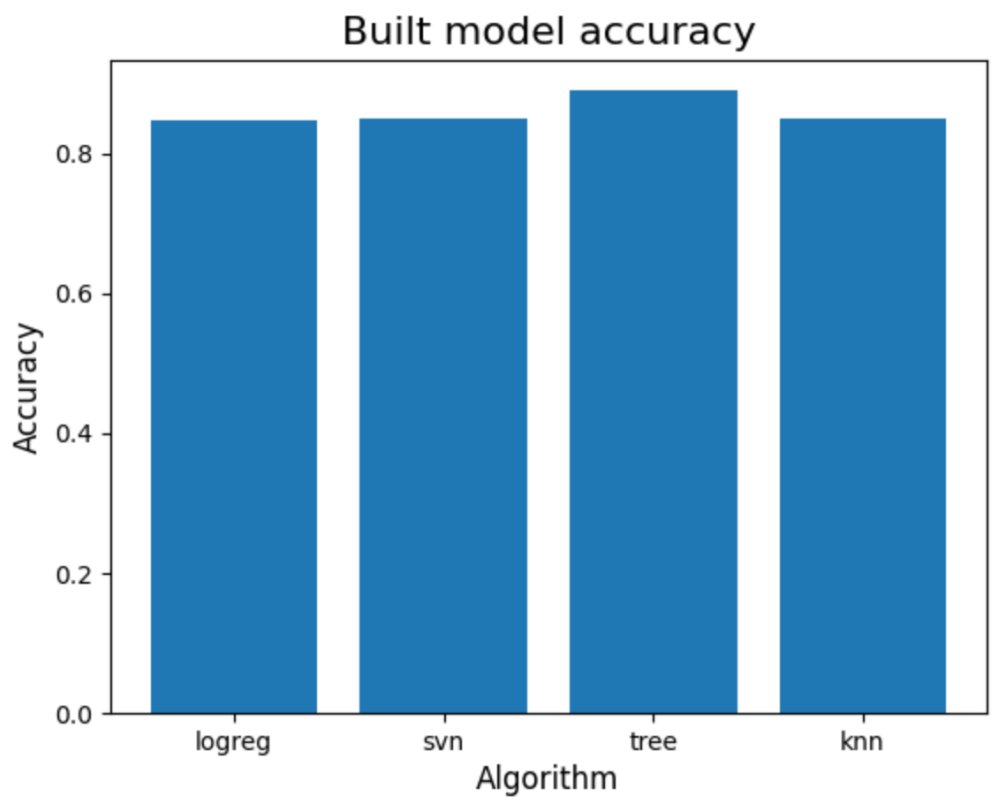
Classification Accuracy

The decision tree model (tree) has the highest classification accuracy of

0.8928571428571145

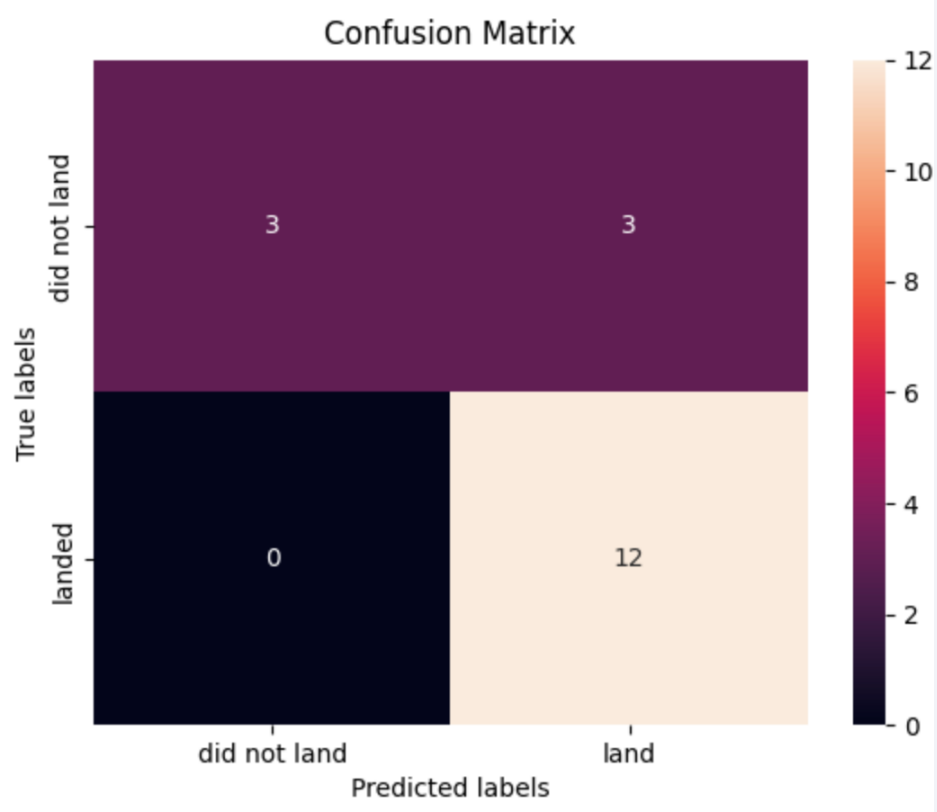
The parameters are:

- criterion: entropy
- max_depth: 4
- max_features: sqrt
- min_samples_leaf: 2
- min_samples_split: 10
- splitter: best



Confusion Matrix

The confusion matrix of the best mode named decision tree. The major issue is the false positive (FP).



Conclusions

- The decision tree algorithm has the best result.
- Rockets with low weighted payloads x $\leq$ 4000 perform better than heavy weighted.
- Since 2013 the success rate of Space X launches increased related to the time in years. There was a temporary drop from 2017 to 2018 but recovered in 2019.
- The most successful launches has the launch site KSC LC-39A with 76.9%.
- The most success rate has the orbit SSO with 100% and the success rate of the orbits LEO and PO increases related to the flight numbers.

Appendix

Links about Space X:

- https://www.spacex.com
- https://en.wikipedia.org/wiki/SpaceX

Thank you!