IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jorge Rosa
12/06/2025

Outline

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

Executive Summary

- Summary of methodologies: REST API, data scraping, data wrangling, EDA, interactive visualization, predictive analysis
- Summary of all results:
  - the data shows that mission successes have generally increased over time;
  - some orbits show a greater success than others
  - There are relationships between launch site, payload, and flight number

Introduction

- The goal of this project is to create a prediction model that can accurately predict whether SpaceX will attempt to land a rocket or not.
- The race for space domination has moved large sums of investments from governments and businesses. SpaceX has attempted a unique approach to make the space exploration financially more efficient.
- With that purpose, SpaceX has launched several experimental rockets and a number of commercial missions. In this project, the data from those launches is used to create a predictive model. The model is trained and tested using the data.
- By considering the different aspects of a given mission, the model will be able to predict whether SpaceX should attempt to land a rocket. This helps defining the mission, as well as determining the mission cost.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from SpaceX launches was gathered from an API: the SpaceX REST API
- Perform data wrangling
  - Data was processed from transforming ID codes into meaningful data, by filtering, by dealing with null values, and by applying one hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Several predictive models were trained and tested, using a pipeline implementation method; the best parameters for the models and their accuracy were determined.

Data Collection – SpaceX API

- Example of calling the SpaceX REST API:

```
def getBoosterVersion(data):
    for x in data['rocket']:
      if x:
       response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
       BoosterVersion.append(response['name'])
```

- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

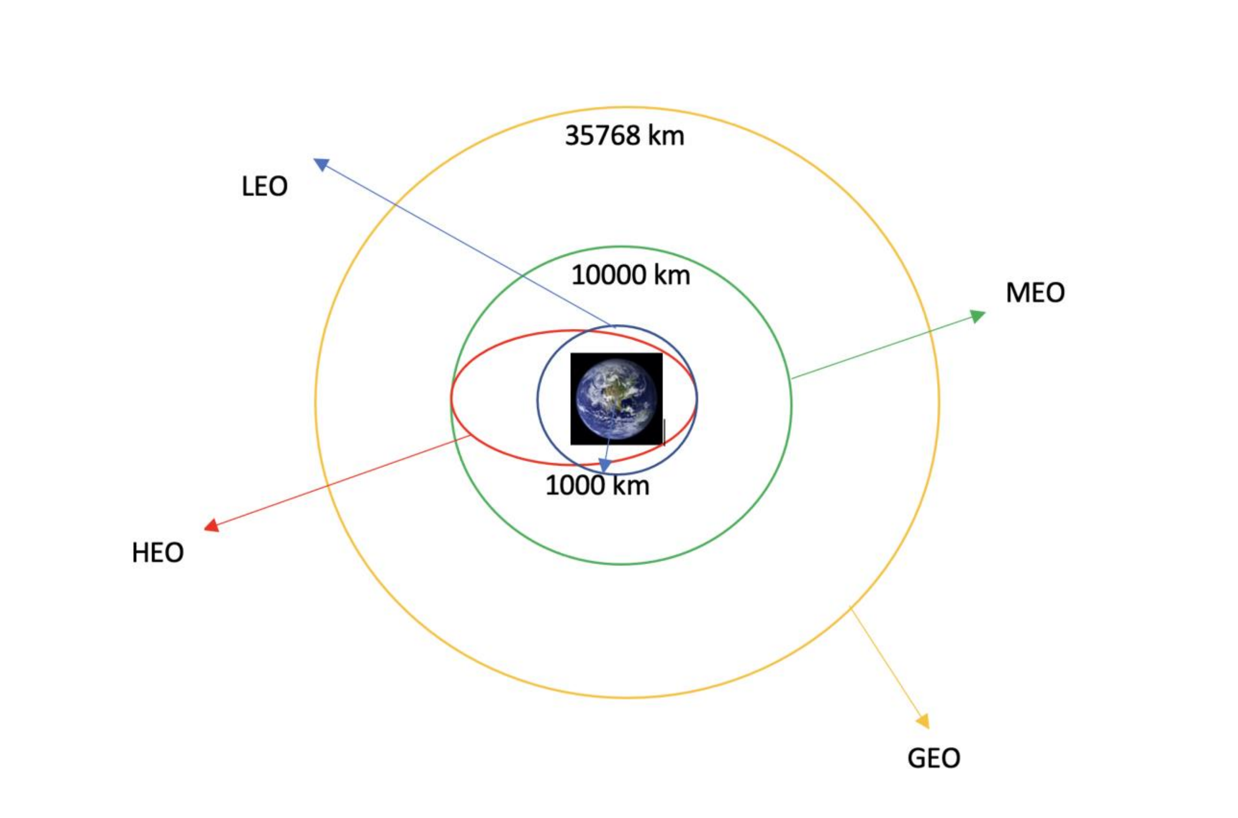
- Web scraping process: BeautifulSoup
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/jupyter-labs-webscraping.ipynb

Example:

```
soup = BeautifulSoup(response,
'html.parser')
html_tables = soup.find_all('table')
column_names = []
for table in first_launch_table.find_all('th'):
    name =
extract_column_from_header(table)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

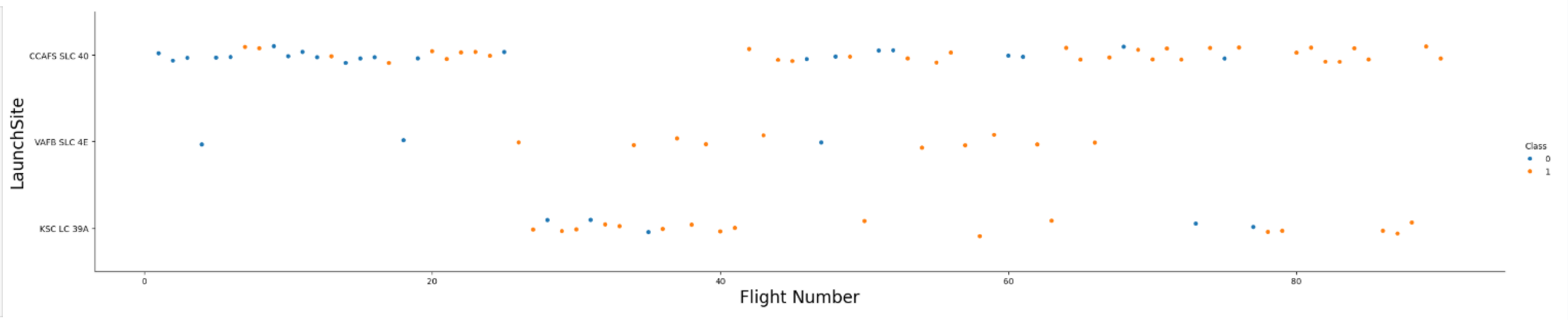
Data Wrangling

- Data was processed by classification, feature engineering and calculation
- Ex: landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df["Outcome"]]
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/edadataviz.ipynb
- Example: relationship between launch site and flight number: shows how during a certain period the main launch site changed
- See github for more examples

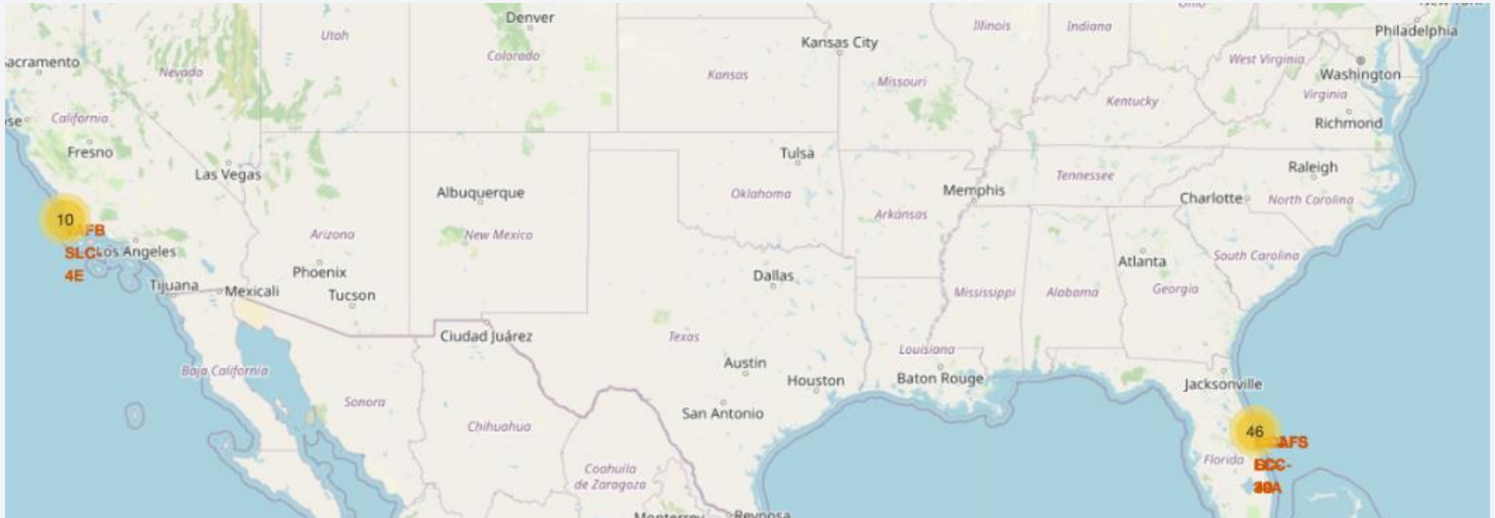


EDA with SQL

- Examples of SQL queries performed:
  - %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
  - %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5
  - %sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)“
  - %sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1%"
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/jupyter-labs-eda-sql-coursera_sqllite.ipynb

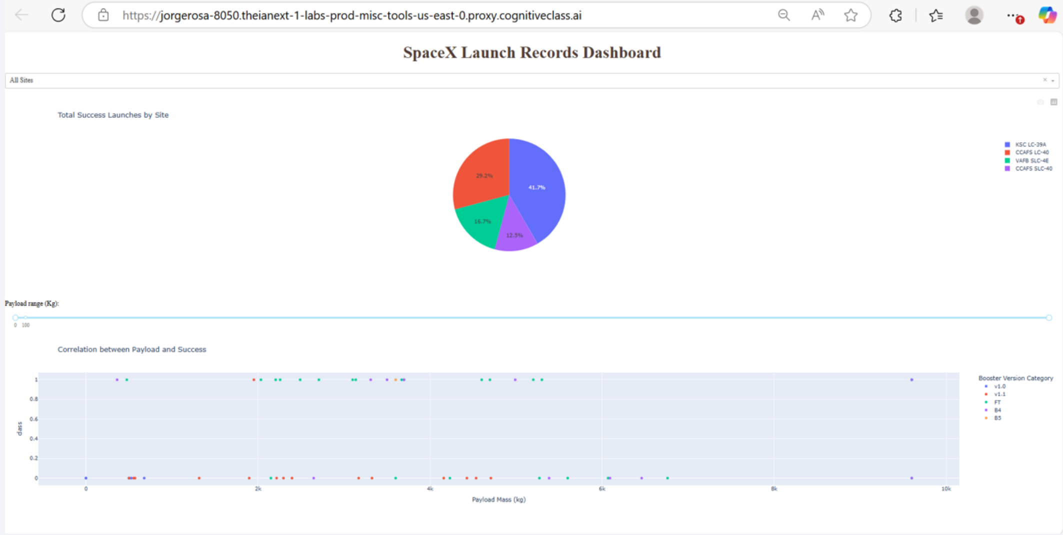
Build an Interactive Map with Folium

- Several map objects were added: number of launches, launch success, cluster of launches
- Those objects were added for exploratory visual information
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/lab_jupyter_launch_site_location.ipynb



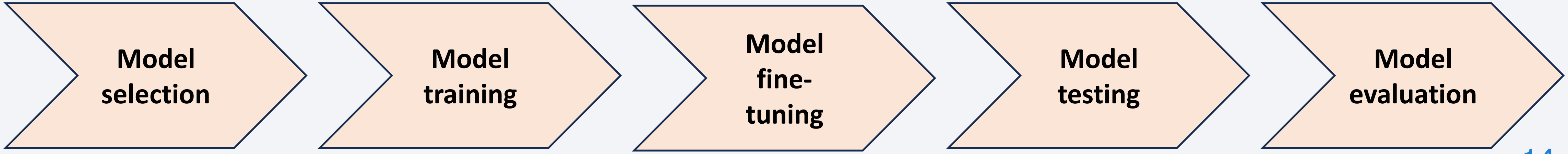
Build a Dashboard with Plotly Dash

- A pie chart shows the number of successful launched per site; a specific site can be selected;
- A histogram with the success class vs payload was created, showing the relationship between the two, for each booster category
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/de844c65f1202e4c0e38e3120cb71a9d1b2fd100/spacex-dash-app.py



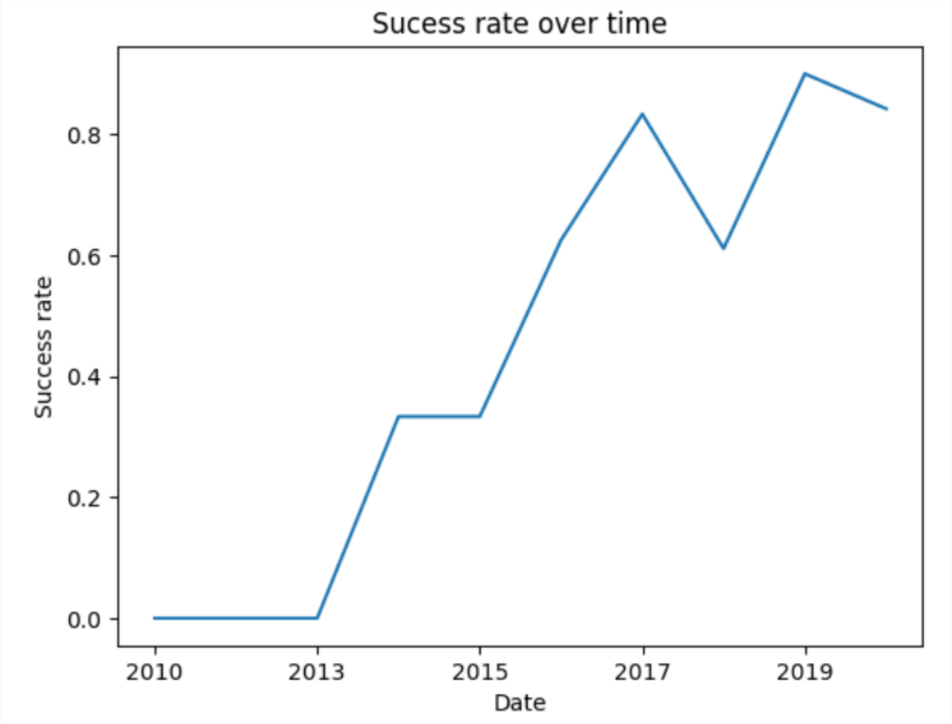
Predictive Analysis (Classification)

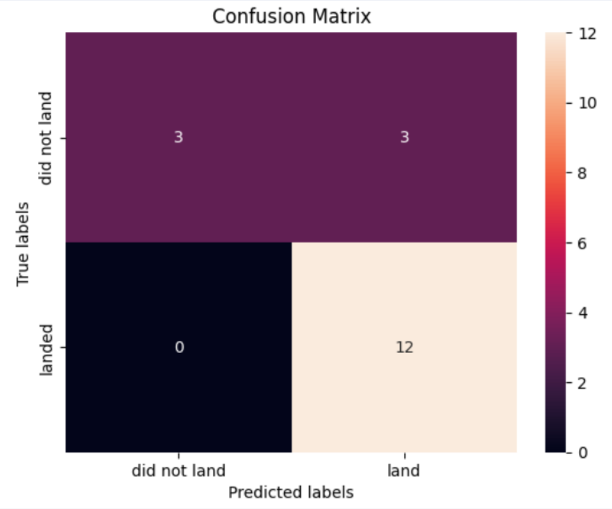
- Several predictive models were trained and then tested
- The optimal model parameters were found
- The models accuracy was measured
- GitHub URL: https://github.com/jorgerosa-galp/testrepo/blob/07572053469fbba098388dedb87dae9081507c9f/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

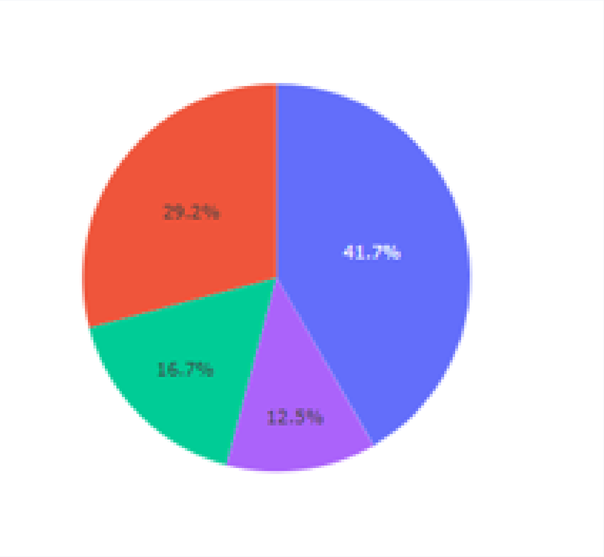


Results

- Exploratory data analysis results
  - The data shows that the success rate improved over time, after the initial misses



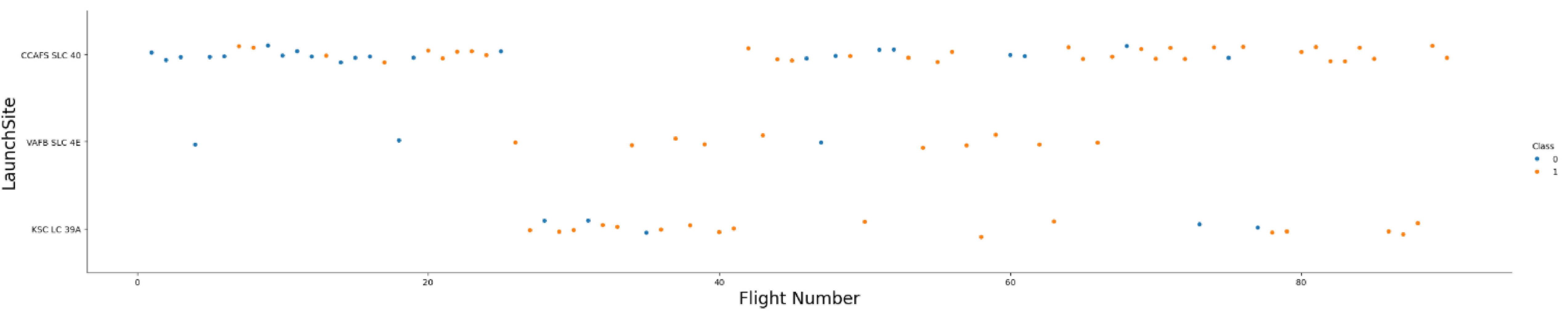




Section 2

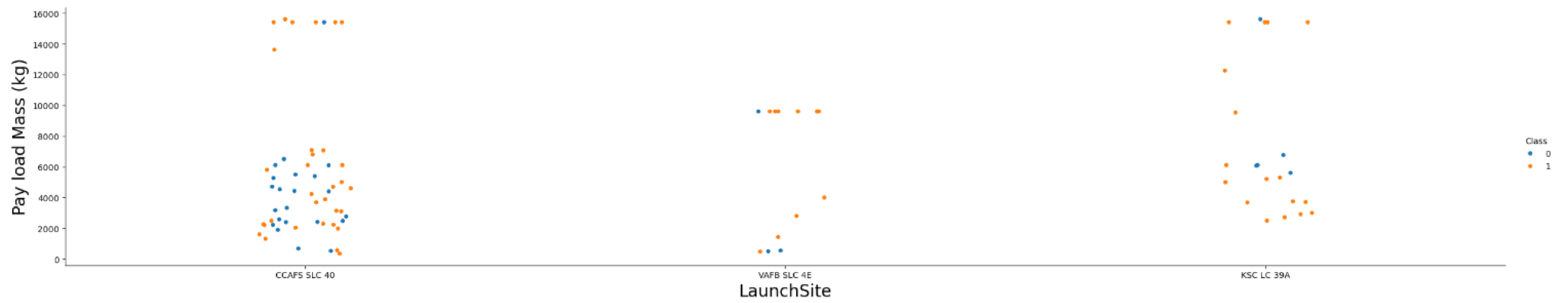
Insights drawn from EDA

Flight Number vs. Launch Site



- The scatter plot of Flight Number vs. Launch Site shows how the launch site changed across the flight number
- This might be temporary changes in site, possibly due to weather conditions
- The launch site VAFB SLC 4E shows a behaviour of a back up site, being used sparingly

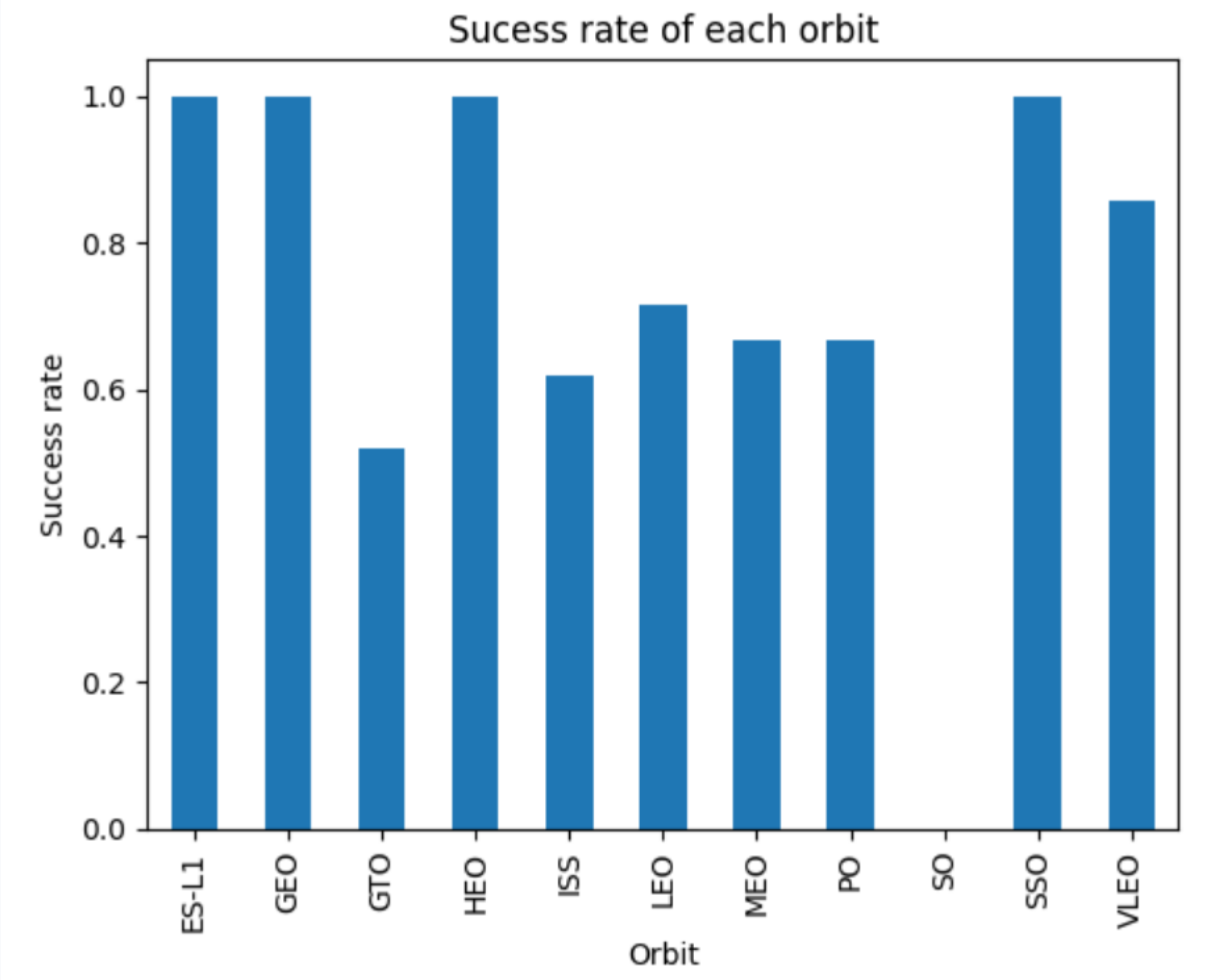
Payload vs. Launch Site



- The scatter plot of Payload vs. Launch Site shows how payload is distributed for each site
- In the VAFB-SLC launch site there are no rockets launched for heavypayload mass (greater than 10000) and most launched have a medium payload

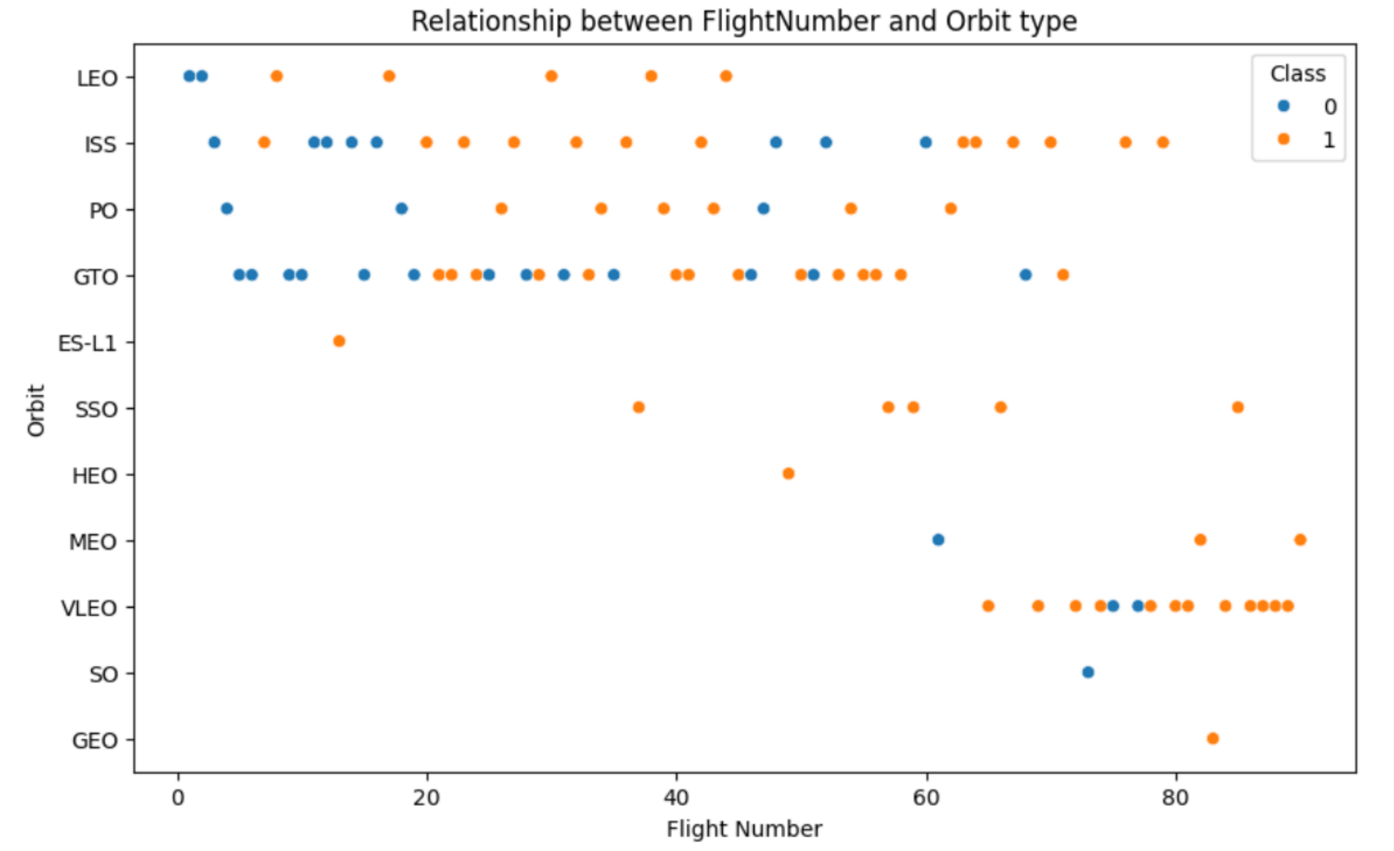
Success Rate vs. Orbit Type

- The bar chart for the success rate of each orbit type shows the most successful orbits
- The least successful orbits should be avoided in future launches



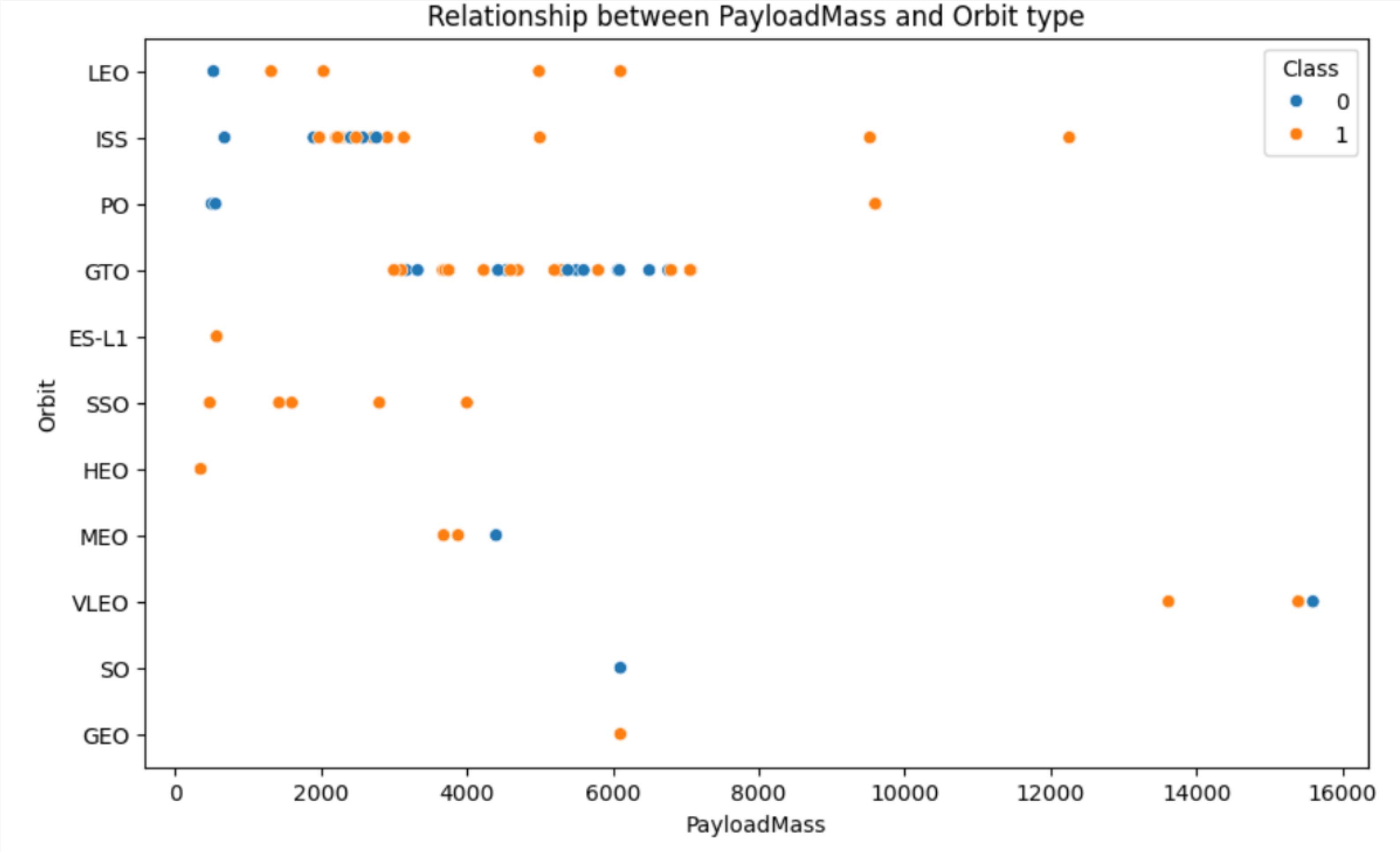
Flight Number vs. Orbit Type

- The scatter point of Flight number vs. Orbit type shows how the orbit type progressed
- In the LEO orbit, success seems to be related to the number of flights
- In the GTO orbit, there appears to be no relationship between flight number and success



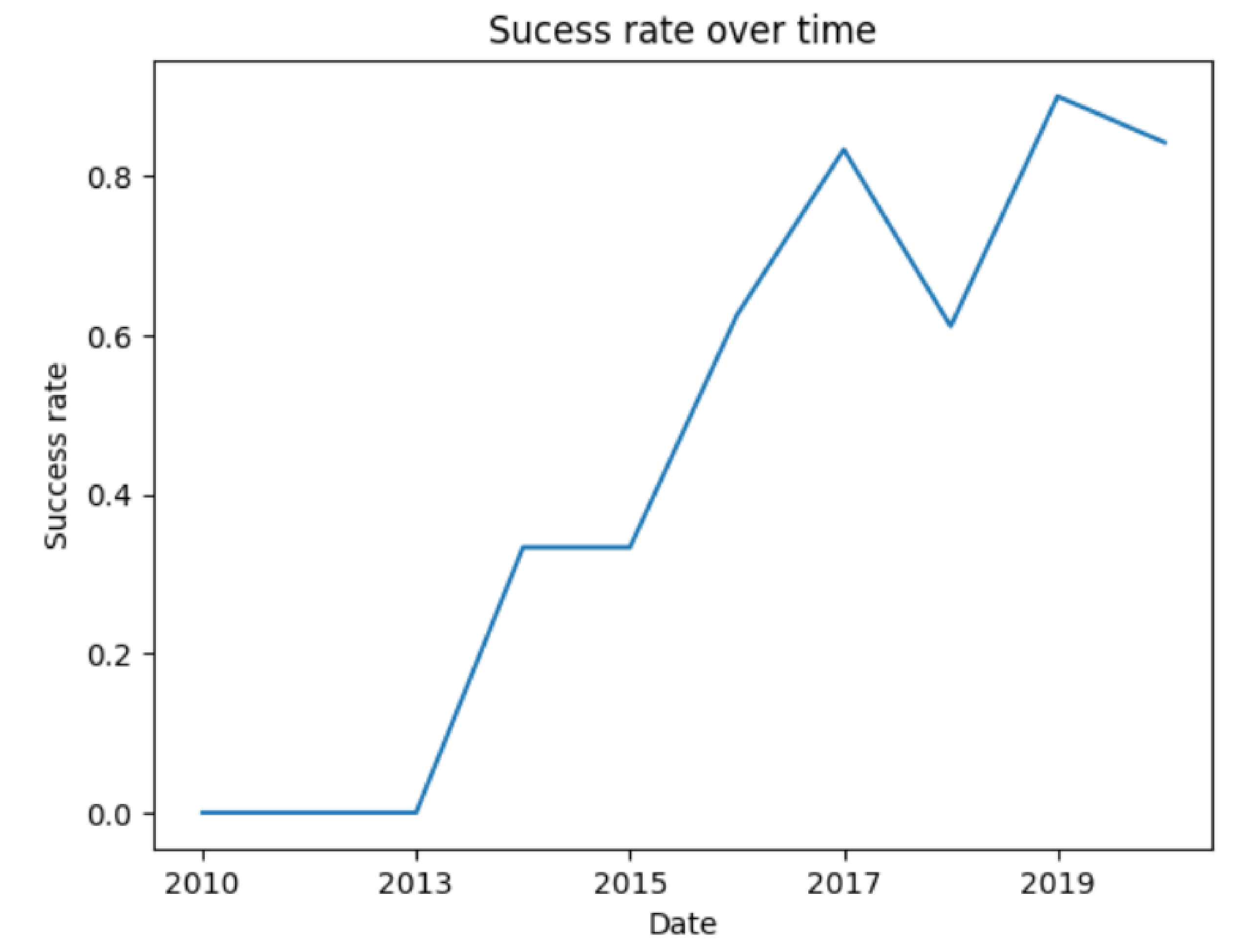
Payload vs. Orbit Type

- The scatter point of payload vs. orbit type shows how payload is distributed for each orbit
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS
- For GTO, both successful and unsuccessful landings are present.



Launch Success Yearly Trend

- The line chart of yearly average success rate shows how the rate has improved over the years
- The sucess rate since 2013 kept increasing till 2020
- There was a decline in 2018, which was recovered in 2019



All Launch Site Names

- Find the names of the unique launch sites:
- %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
- This query selects only the distinct/unique launch site names

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

Launch Site Names Begin with 'CCA'

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

- Find 5 records where launch sites begin with `CCA`
- %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5
- This query selects the first 5 results of all columns that have a launch site name starting with “CCA”

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- %sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)“
- This query selects the sum of the payload mass for the launches that had NASA as customer

total_payload
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- %sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1%“
- This query calculates the average of the payload mass for the booster version F9 v1.1

average_payload
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %sql SELECT MIN(Date) AS first_success FROM SPACEXTABLE WHERE "Mission_Outcome" = "Success“
- This query calculates the earliest date of the successful missions

first_success
2010-06-04

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
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
- This query selects the booster versions for the outcome "Success (drone ship)“ and which have a payload mass between 4000 and 6000 kg

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- %sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome“
- This query calculates the total number of outcomes
- The duplication of “Success” is possibly due to a space (“ “) in one of the data values

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
- This query selects the booster versions that carried the max payload mass

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- %sql SELECT substr(Date, 6,2) as month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Failure (drone ship)" AND substr(Date,0,5)='2015’
- This query shows the booster version, launch site and month for failed missions in 2015

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

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

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

Section 3

Launch Sites Proximities Analysis

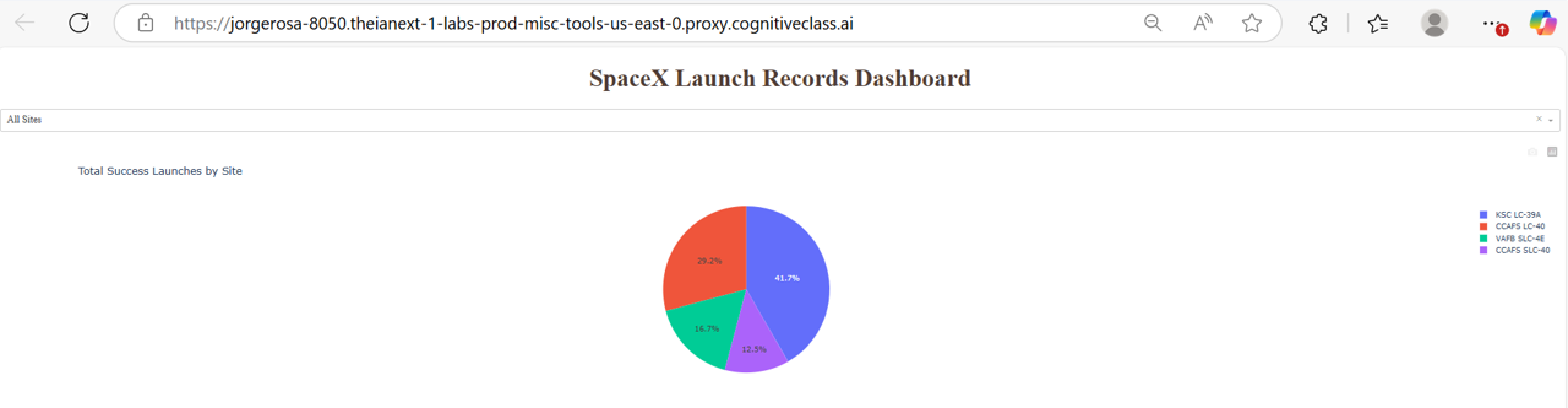
Distance measurement

- In folium, distance can be measured to get insights on the data
- An example is show here, where the distance to the coastline is measured
- Similarly, the distance to railroads, cities, streets and other map features can be measured

Section 4

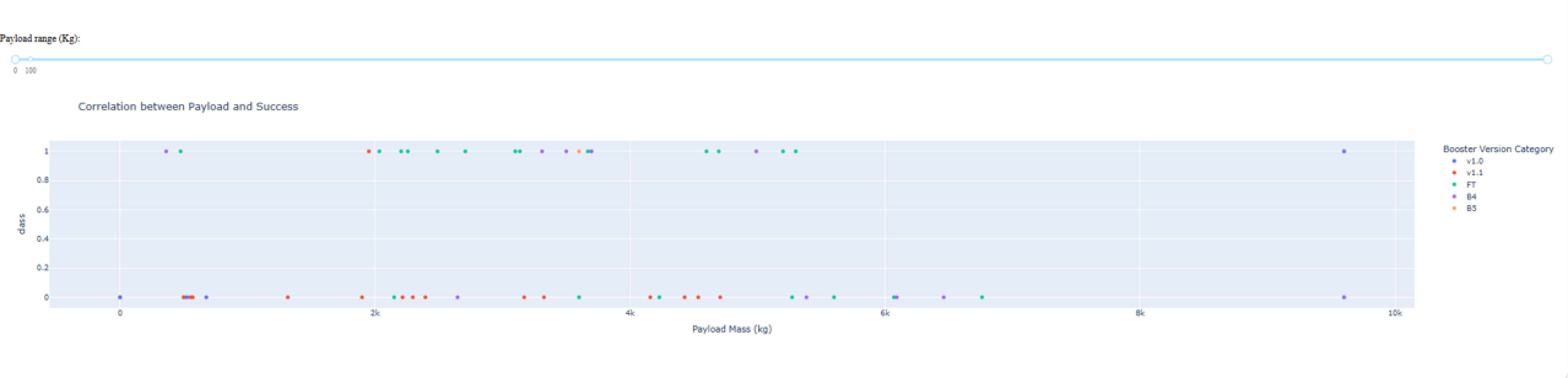
Build a Dashboard with Plotly Dash

Successes by site dashboard



- In this dashboard, the launch site can be selected
- The pie chart shows the success records according to the selection

Success rate vs Payload



- In this dashboard, the success class is shown across the payload mass
- The booster version is shown in different colors for a careful analysis

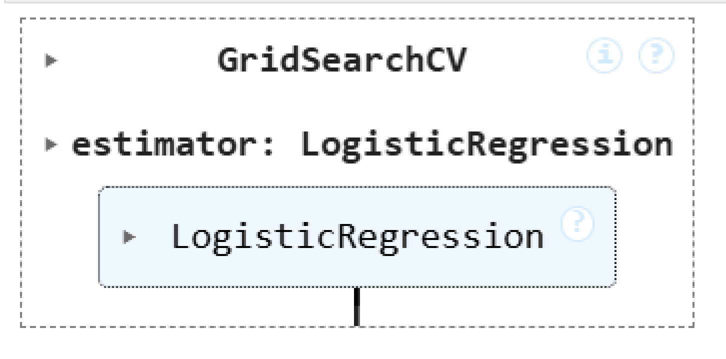
Section 5

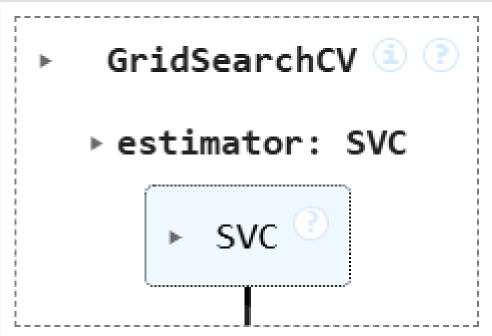
Predictive Analysis (Classification)

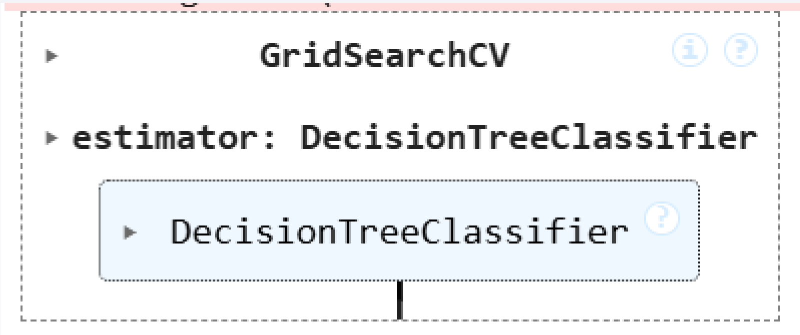
Classification Accuracy

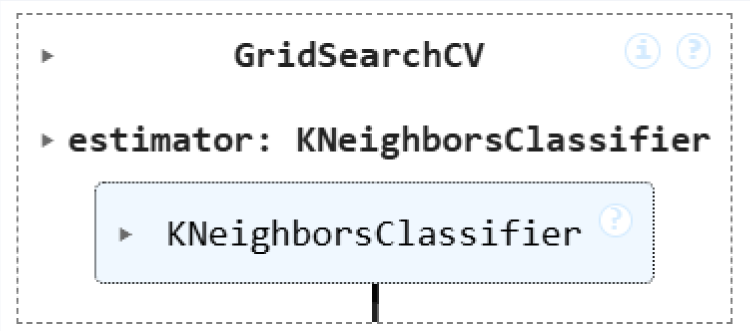
- Several models were tested, all with identical accuracies

- The results from the models were similar



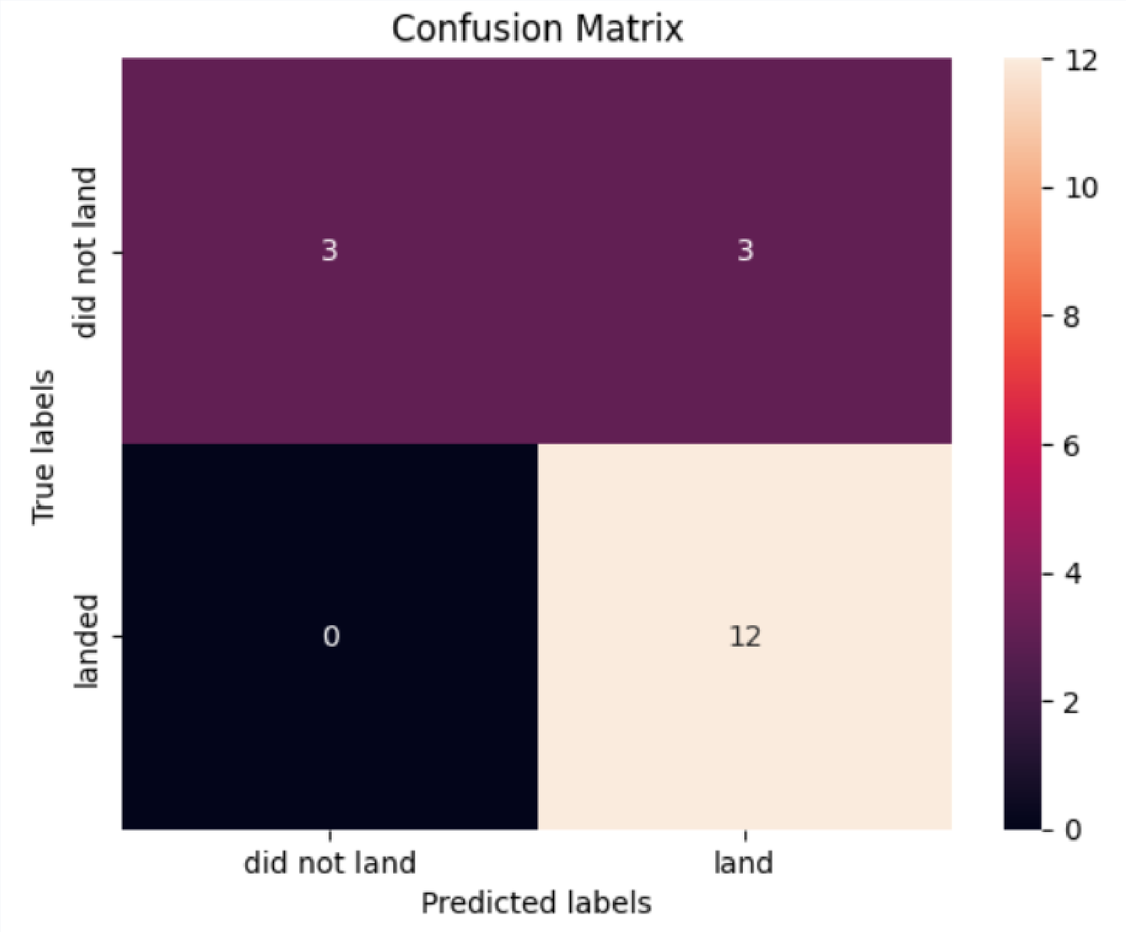






Confusion Matrix

- The confusion matrix shows 12 positive results correctly predicted
- It also shows 3 negative results correctly predicted
- However, 3 missions were predicted to land but actually failed
- There is zero failed cases for unsuccessful missions



Conclusions

- REST API allows data to be efficiently collected for analysis
- Through data wrangling, a thorough data analysis can be achieved
- EDA allows insights to be identified from data
- SQL is an effective language for querying datasets
- In Plotly Dash, dashboards can be created for interactive data visualization
- Predictive analysis capitalizes on data, offering insights on future events

Thank you!