Winning Space Race with Data Science

Syifa Afnani Santoso
09/03/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API, Web Scraping
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- Project background and context

  The aim of this project is to predict if the Falcon 9 first stage will successfully land. SpaceX says on its website that the Falcon 9rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By determining if the stage will land, we can determine the cost of a launch. This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.

- Problems you want to find answers
  - What are the main characteristics of a successful or failed landing?
  - What are the effects of each relationship of the rocket variables on the success or failure of a landing?
  - What are the conditions which will allow SpaceX to achieve the best landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- 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

- Collecting data from SpaceX API
  - URL: https://api.spacexdata.com/v4/

- Collecting data from web scrapping Wikipedia
  - URL: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Collection – SpaceX API

How to collect data from SpaceX API:

1. Getting response from API
2. Convert response to JSON format
3. Transform data
4. Create dictionary with data
5. Create data frame
6. Filter data frame
7. Export data to file

[Link to code]

1
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2
```
data = pd.json_normalize(response.json())
```

3
```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4
```
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}
```

5
```
data = pd.DataFrame(launch_dict)
```

6
```
data_falcon9 = data[data['BoosterVersion'] != 'Falcon 1']
```

7
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

How to collect data from web scrapping Wikipedia:

1. Getting response from HTML
2. Find all tables from BeautifulSoup
3. Get all column names
4. Create dictionary with table records
5. Add records to dictionary
6. Create data frame from dictionary
7. Export data to file

[Link to code]

1
```
static_url = "https://en.wikipedia.org/w/ \
    index.php?title=List_of_Falcon_9_and_ \
    Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url).text
```

2
```
soup = BeautifulSoup(response)
html_tables = soup.find_all('table')
```

3
```
column_names = []
for col in first_launch_table.find_all('th'):
    name = extract_column_from_header(col)
    if name != None and len(name)>0:
        column_names.append(name)
```

4
```
launch_dict= dict.fromkeys(column_names)

del launch_dict['Date and time ( )']

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'] = []

launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5
```
extracted_row = 0
for table_number,table in enumerate(soup.find_all \
        ('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

6
```
df=pd.DataFrame(launch_dict)
```

7
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Data wrangling process includes:

1. Calculate launches number for each site
2. Calculate launches number for each orbit
3. Calculate number of mission outcomes per orbit type
4. Create landing outcome label
5. Export data to file

[Link to code]

1

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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2

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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3

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

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

4

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
landing_class = []
for element in df['Outcome']:
    if element in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5

```
df.to_csv("dataset_part\_2.csv", index=False)
```

EDA with Data Visualization

- Scatter Plots
  show relationship between two variables called correlation
  - Flight number vs Payload mass
  - Flight number vs Launch site
  - Payload vs Launch site
  - Orbit vs Flight number
  - Payload vs Orbit type
  - Orbit vs Payload mass

- Bar Graphs
  show relationship between numeric and categoric variables
  - Success rate vs Orbit

- Line Graphs
  show trend of data variables that can help to show global behavior and make prediction for unseen data
  - Success rate vs Year

[Link to code]

EDA with SQL

- Display the names of unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

[Link to code]

Build an Interactive Map with Folium

- Map objects that are created and added to a folium map:
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name (*folium.Circle, folium.map.Marker*)
  - Red circles at each launch site coordinates with label showing launch site name (*folium.Circle, folium.map.Marker, folium.features.DivIcon*)
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (*folium.plugins.MarkerCluster*)
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing (*folium.map.Marker, folium.Icon*)
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them (*folium.map.Marker, folium.PolyLine, folium.features.DivIcon*)
- These objects are created in order to understand the problem and data better. We can easily show all launch sites, their surroundings and the number of successful and unsuccessful landings

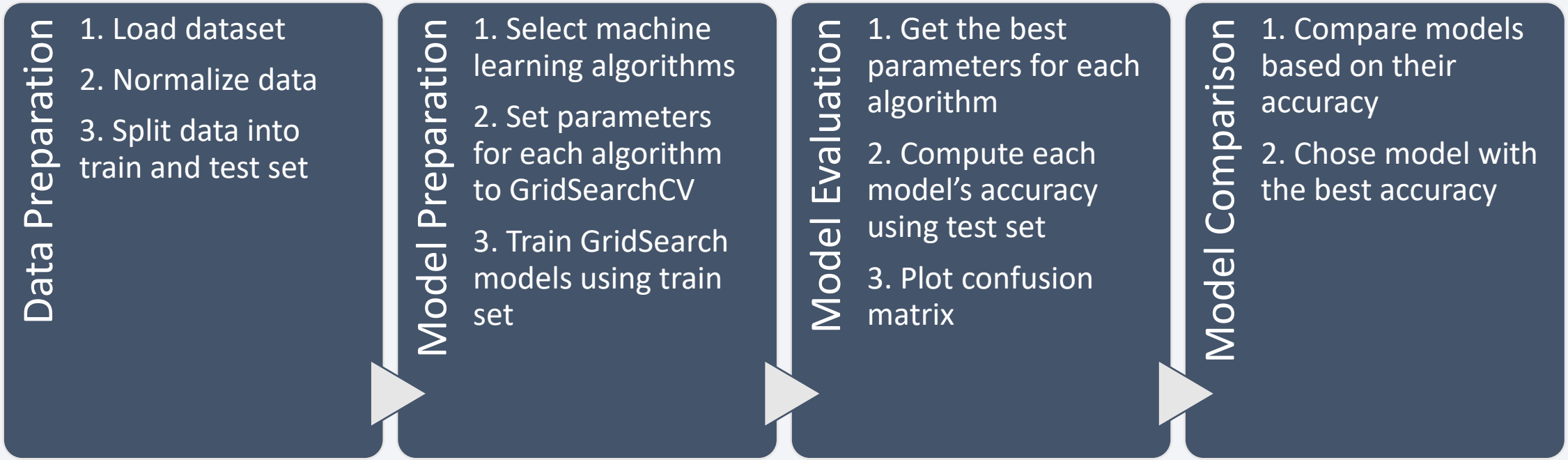
[Link to code]

Build a Dashboard with Plotly Dash

- The dashboard includes:
  - Dropdown allows a user to choose the launch site or all launch sites (*dash_core_components.Dropdown*)
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (*plotly.express.pie*)
  - Rangeslider allows a user to select a payload mass in a fixed range (*dash_core_components.RangeSlider*)
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (*plotly.express.scatter*)
- These objects are added to the dashboard so that user can easily go through and see the visualizations of the data from any site and time frame

[Link to code]

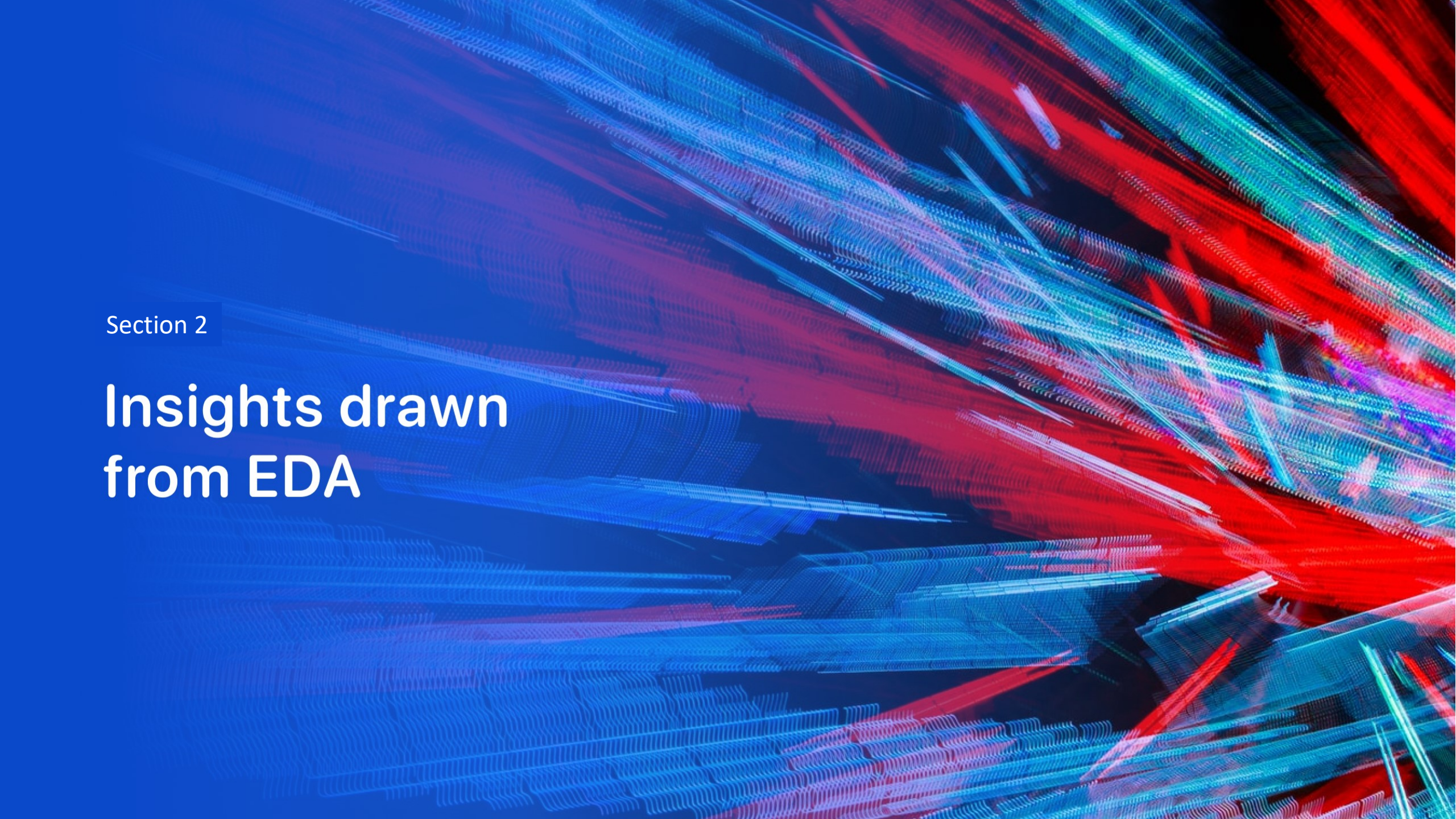
Predictive Analysis (Classification)



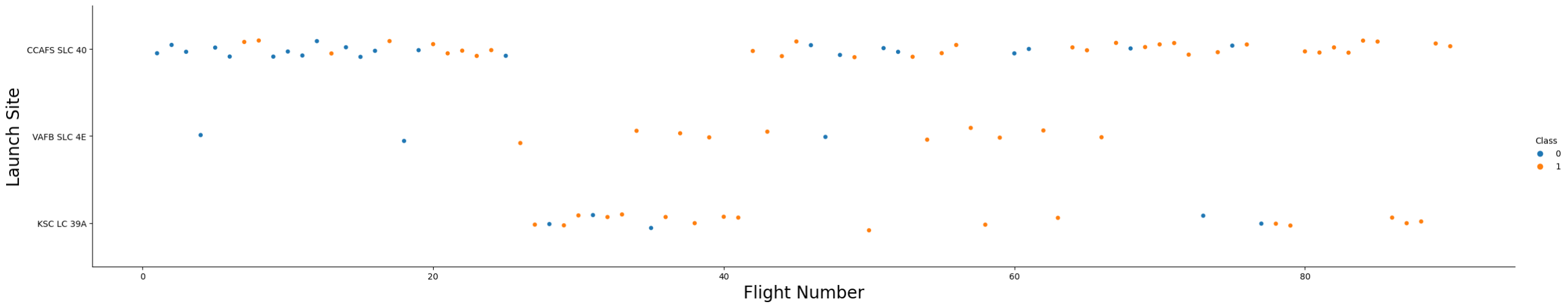
[Link to code]

Results

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

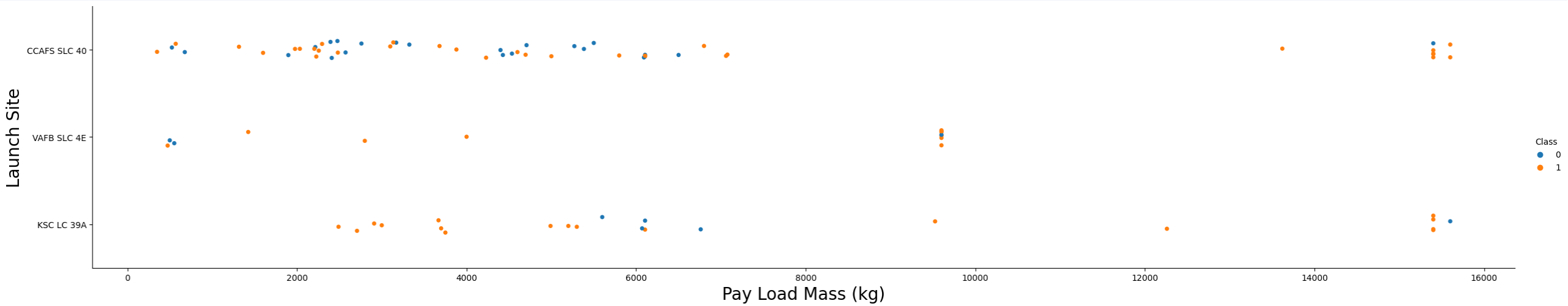

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



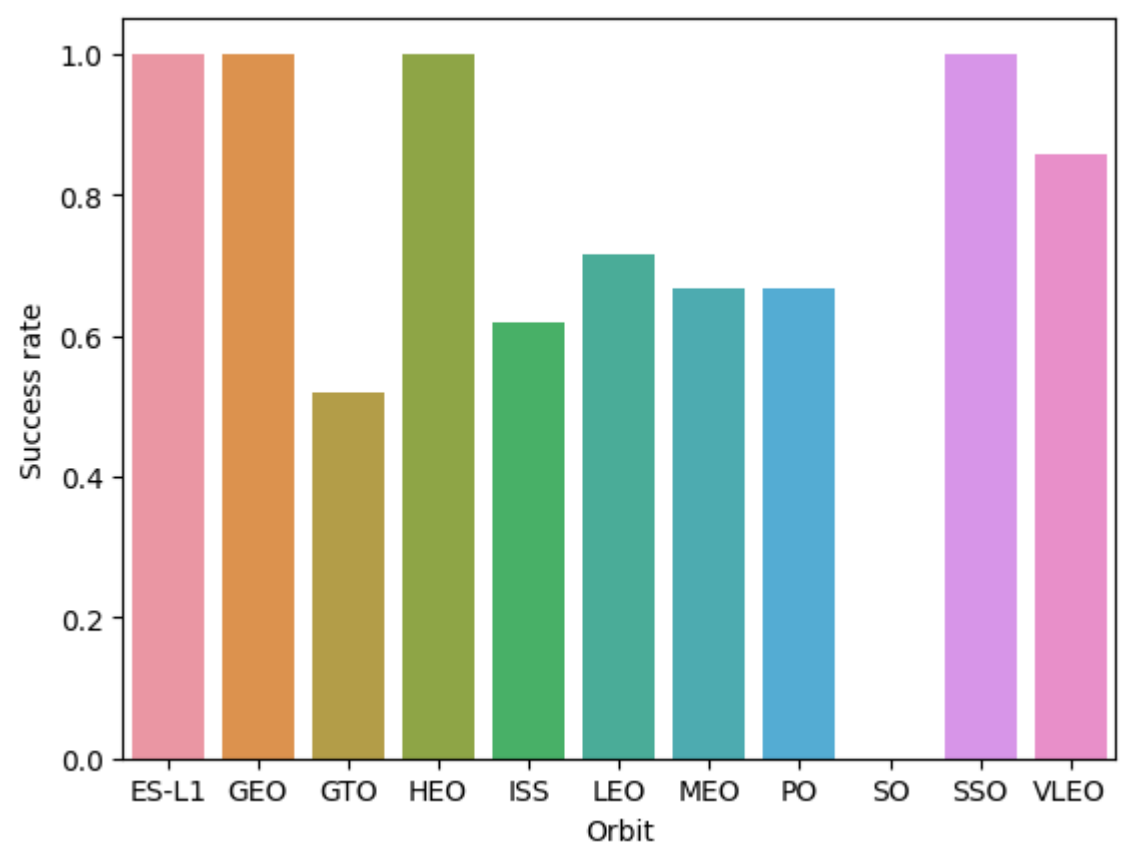
From the plot above, we can observe that in general for every site, the success rate is increasing along with flight number

Payload vs. Launch Site



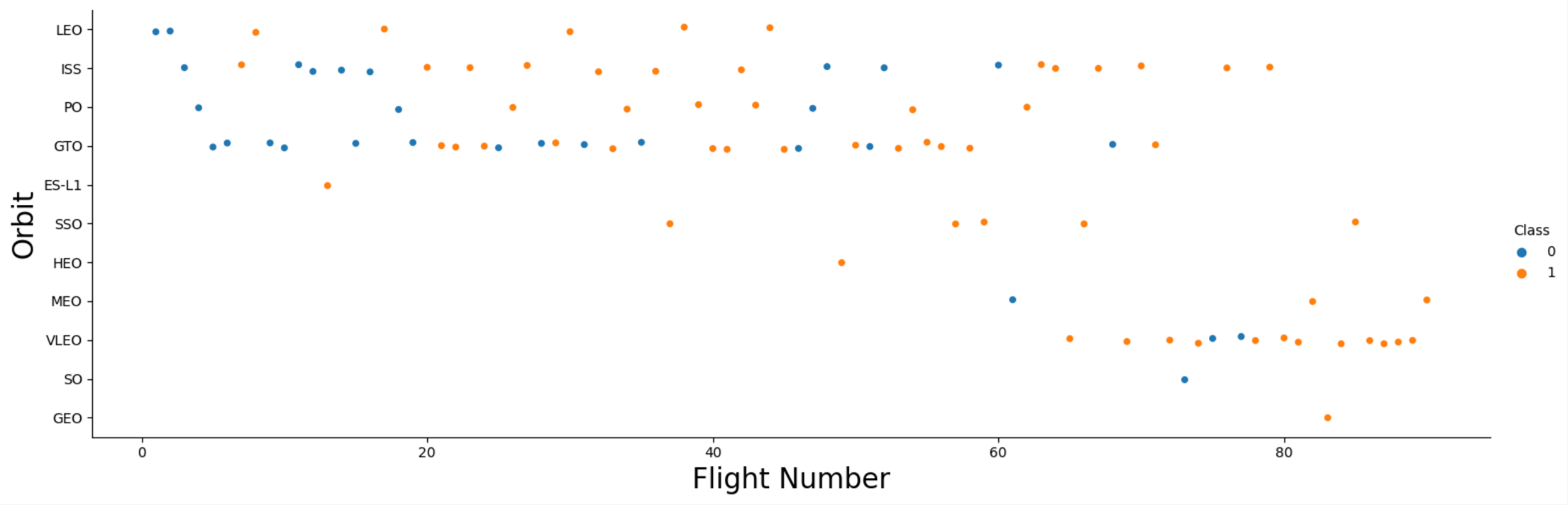
From the plot above, there seems to be no relationship between payload mass and launch site; with a side note that for VAFB-SLC launch site there are no rockets launched for heavy payload mass (> 10,000 kg)

Success Rate vs. Orbit Type



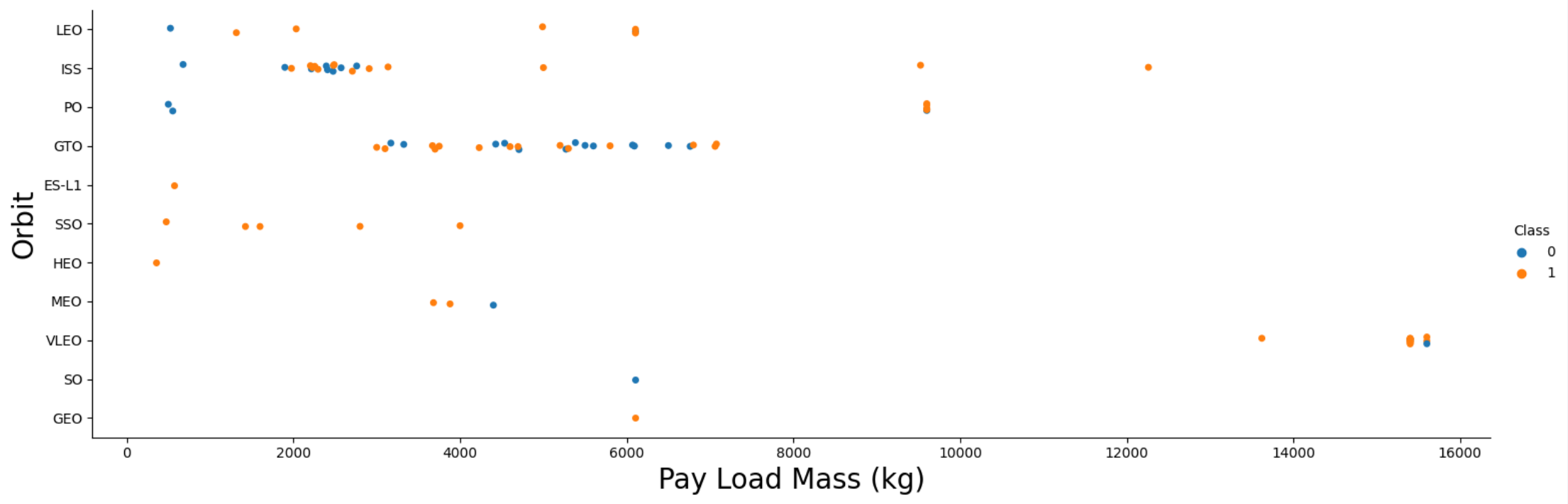
This bar plot displays the success rate for different orbit type. We note that orbit ES-L1, GEO, HEO, and SSO have the best success rate

Flight Number vs. Orbit Type



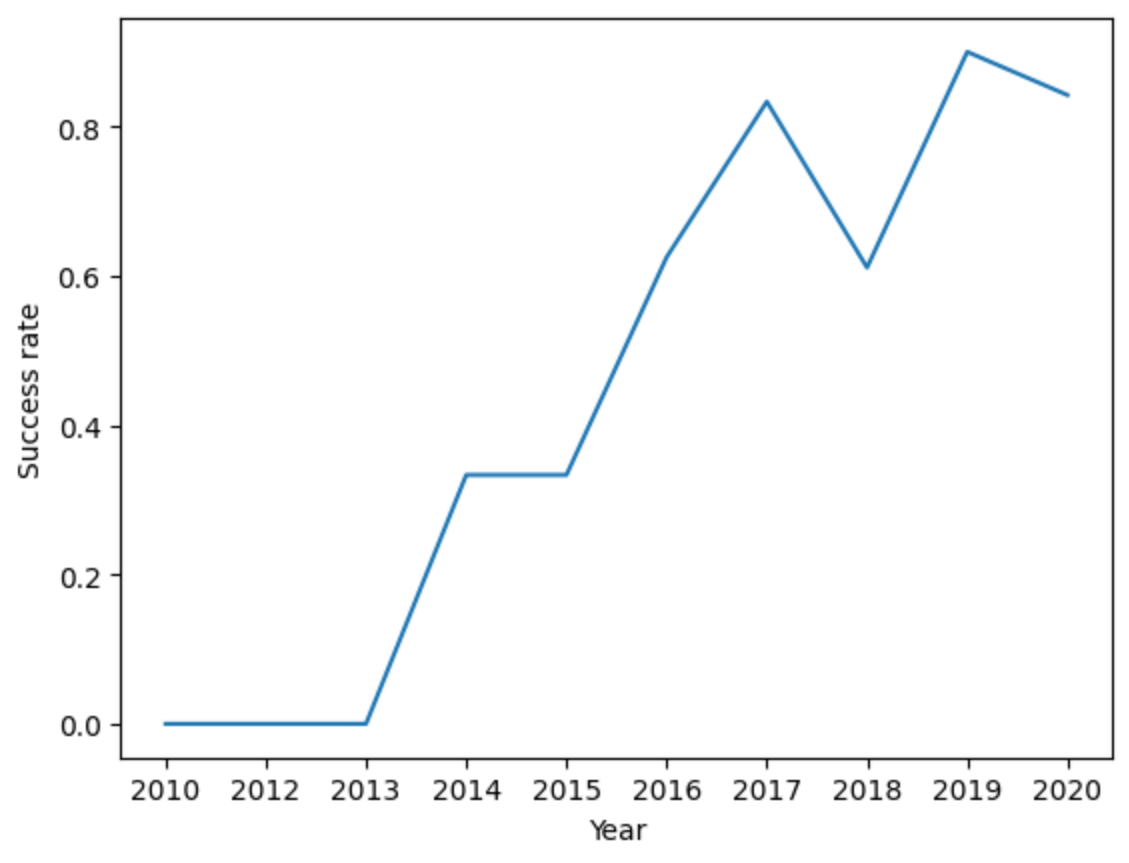
From the plot above, we can observe that in general for each orbit type, the success rate is increasing along with flight number; with an exception for orbit GTO, there seems to be no relationship between these two variables for this particular orbit

Payload vs. Orbit Type



From the plot above, we can observe the successful landing rate for orbit Polar, LEO, and ISS are increasing with heavy payloads; however for GTO, both failed and successful landings are both exist here, so we can't draw a conclusion for this particular orbit based on this data

Launch Success Yearly Trend



This bar plot displays the yearly trend of success rate from 2010-2020. We can observe that the success rate kept increasing since 2013

All Launch Site Names

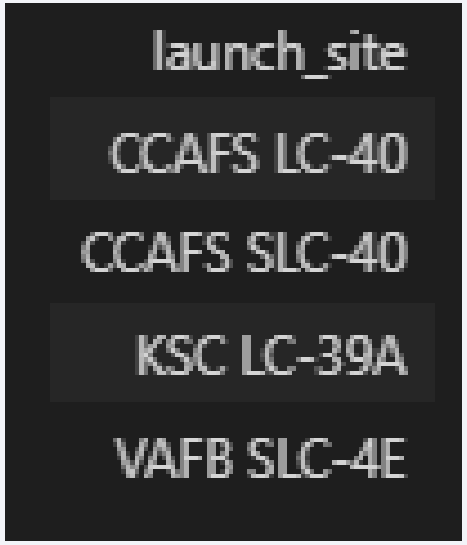
Query

%sql select distinct(LAUNCH_SITE) from SPACEX

Explanation

Using select to display data in column **launch_site** from table **SPACEX** and distinct method to avoid duplications

Result



Launch Site Names Begin with 'CCA'

Query

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

Explanation

Using where method followed by like to filter data from **LAUNCH_SITE** that begin with '**CCA**' and limit 5 to only display 5 rows

Result

DATE	time_utc	booster_version	launch_site	payload
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

Total Payload Mass

Query

%sql select sum(PAYLOAD_MASS_KG_) from SPACEX where CUSTOMER='NASA (CRS)' group by CUSTOMER

Result

Explanation

Using sum method followed by group by to return the sum of payload mass for each customer and filter it where the customer is '**NASA (CRS)**'

Average Payload Mass by F9 v1.1

Query

%sql select avg(PAYLOAD_MASS_KG_) from SPACEX where BOOSTER_VERSION='F9 v1.1' group by BOOSTER_VERSION

Result

Explanation

Using avg and group by to return the average of payload mass for each booster version and filter it where the booster version is '**F9 v1.1**'

First Successful Ground Landing Date

Query

%sql select min(DATE) from SPACEX where LANDING_OUTCOME='Success (ground pad)'

Result

Explanation

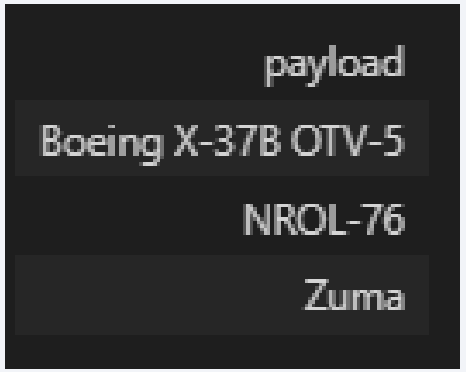
Using min method to get the minimum value from **DATE** column and filter it where landing outcome is '**Success (ground pad)**'

Successful Drone Ship Landing with Payload between 4000 and 6000

Query

%sql select distinct(PAYLOAD) from SPACEX where LANDING_OUTCOME='Success (ground pad)' and (PAYLOAD_MASS_KG_ between 4000 and 6000)

Result



Explanation

Using distinct method to get the unique boosters name and filter it where landing outcome is '**Success (ground pad)**' and payload mass between **4000 and 6000**

Total Number of Successful and Failure Mission Outcomes

Query

%sql select MISSION_OUTCOME, count(*) from SPACEX group by MISSION_OUTCOME

Result

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Explanation

Using count and group by to count the total number of record for each mission outcome

Boosters Carried Maximum Payload

Query

%sql select BOOSTER_VERSION, PAYLOAD_MASS_KG_ from SPACEX where PAYLOAD_MASS_KG_=(select max(PAYLOAD_MASS_KG_) from SPACEX)

Explanation

Using max method inside a subquery to get the maximum value from payload mass column, then using this value to filter the data

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

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

Result

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

Explanation

Using year method to get the year value from **DATE** column and filter the data where the landing outcome is ‘**Failure (drone ship)’** and the year is ‘**2015**’

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

Query

%sql select LANDING_OUTCOME, count(*) as landing_count from SPACEX where DATE between '2010-06-04' and '2017-03-20' group by LANDING_OUTCOME order by count(*) desc

Explanation

Using count and group by to count the total number of record for each landing outcome, filter the data where the date is between '**2010-06-04**' and '**2017-03-20**' and order by the total number in descending order

Result

landing_outcome	landing_count
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

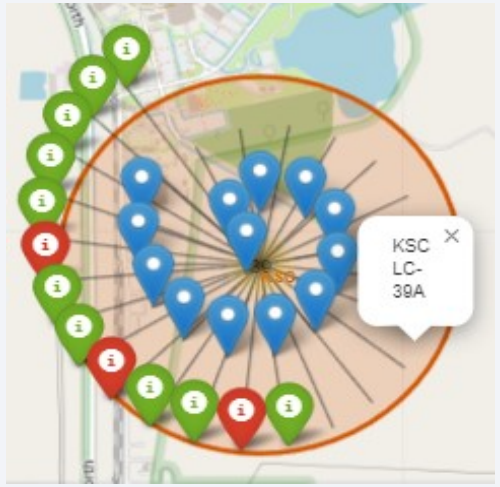
Section 3

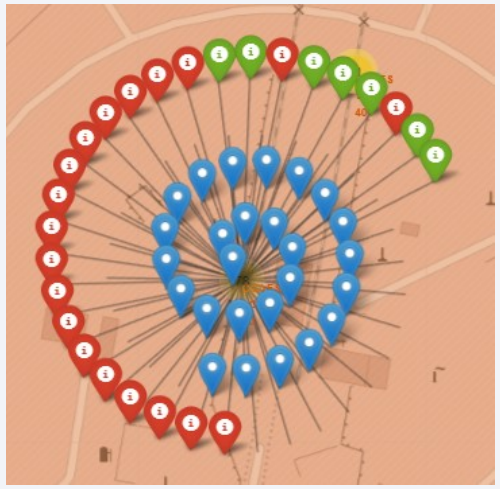
Launch Sites Proximities Analysis

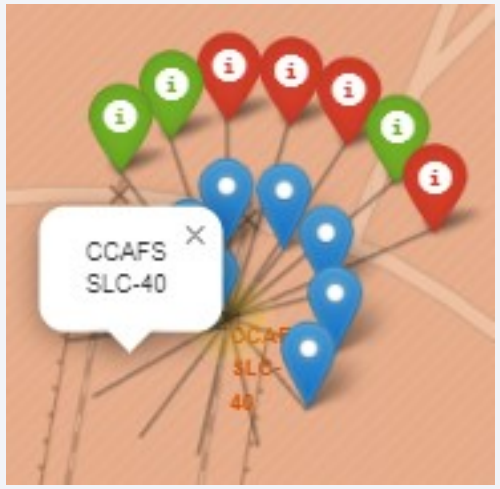
Launch Site Locations

The SpaceX launch sites are located at coastal area in California and Florida

Successful Launches for Each Site



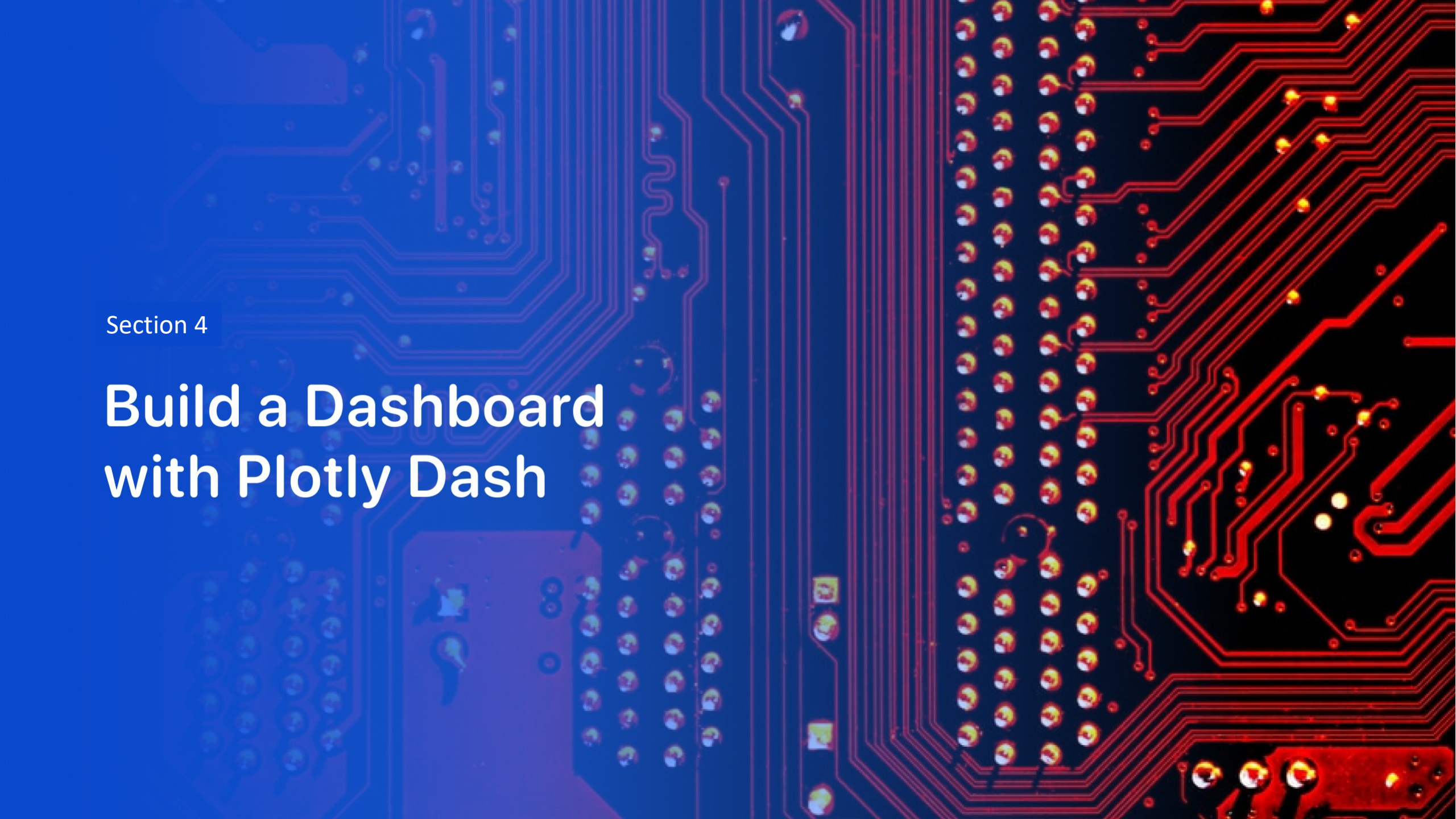




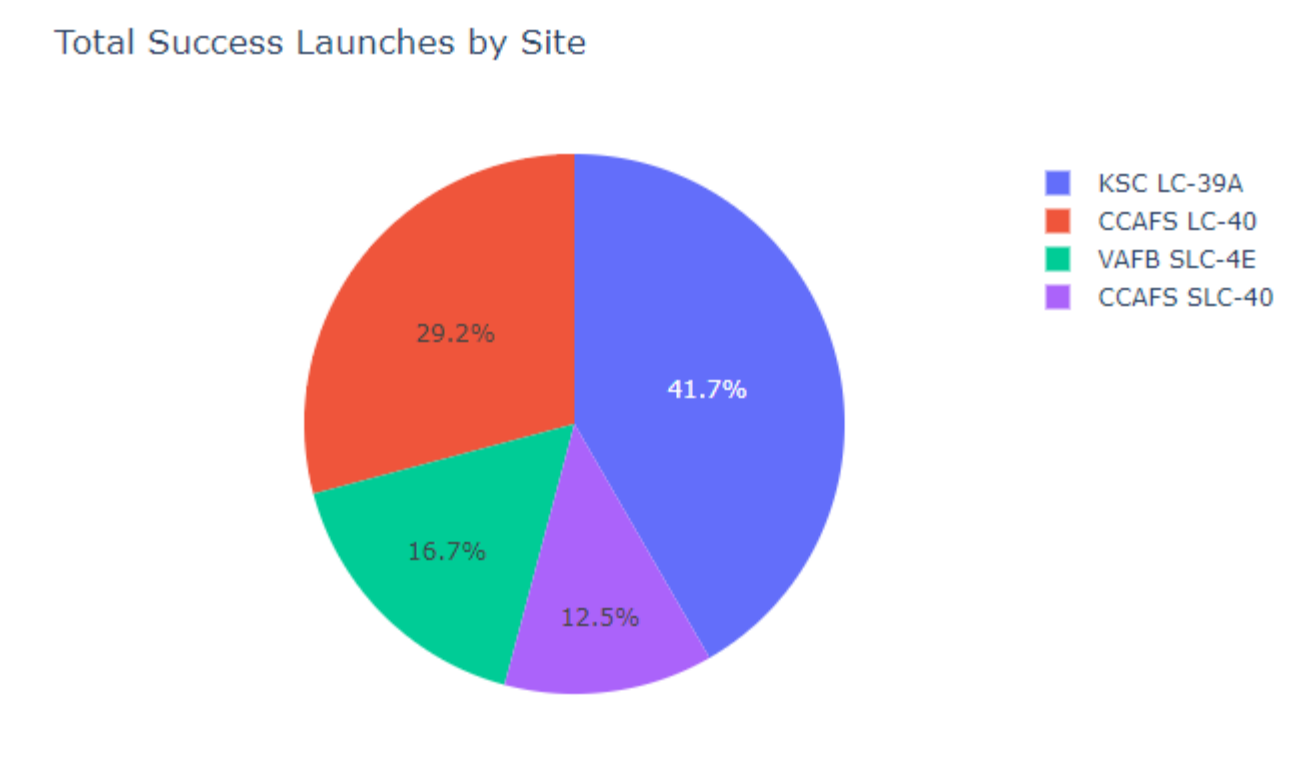
Green marker represents successful launch, Red marker represents unsuccessful launch. We can observe that site KSC LC-39A has the highest launch success rate.

Distance between CCAFS SLC-40 and nearest coast

The distance between site CCAFS SLC-40 and the nearest coast is 0.86 km

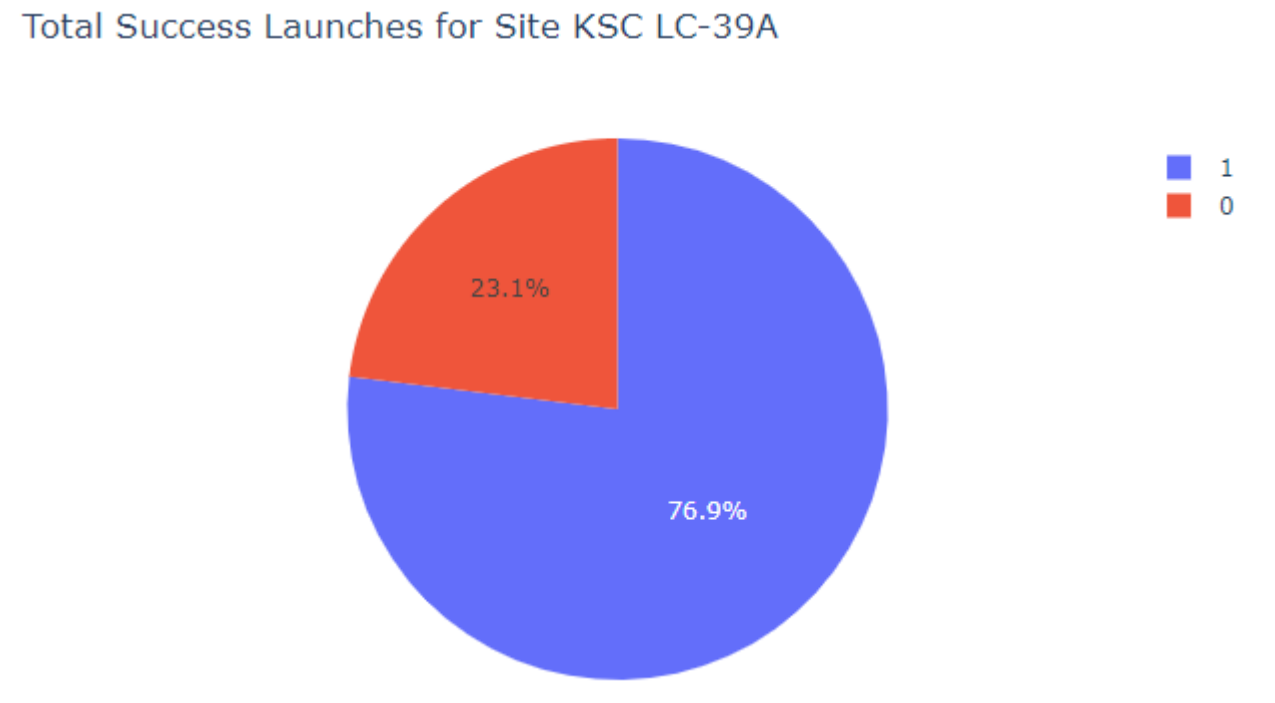

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches by Site



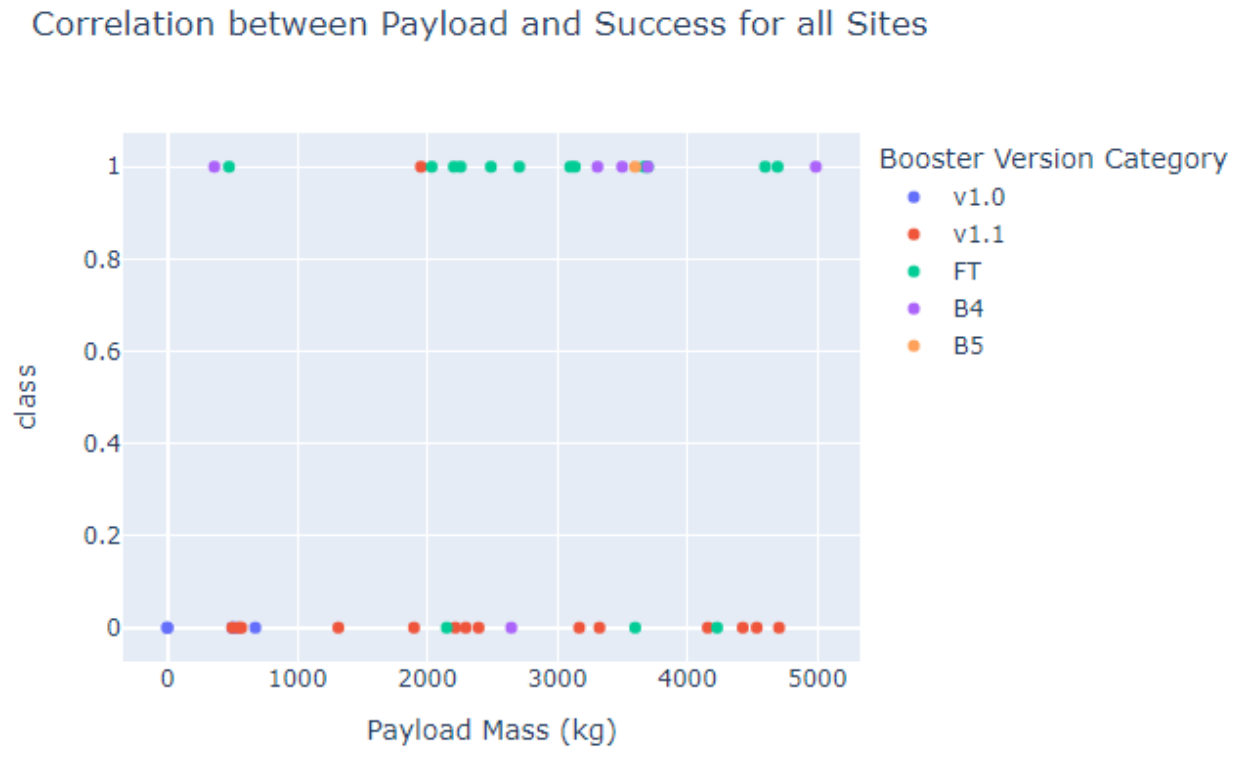
We can observe that KSC LC-39A has the highest launch success ratio, followed by CCAFS LC-40, VAFB SLC-4E, and the lowest is CCAFS SLC-40

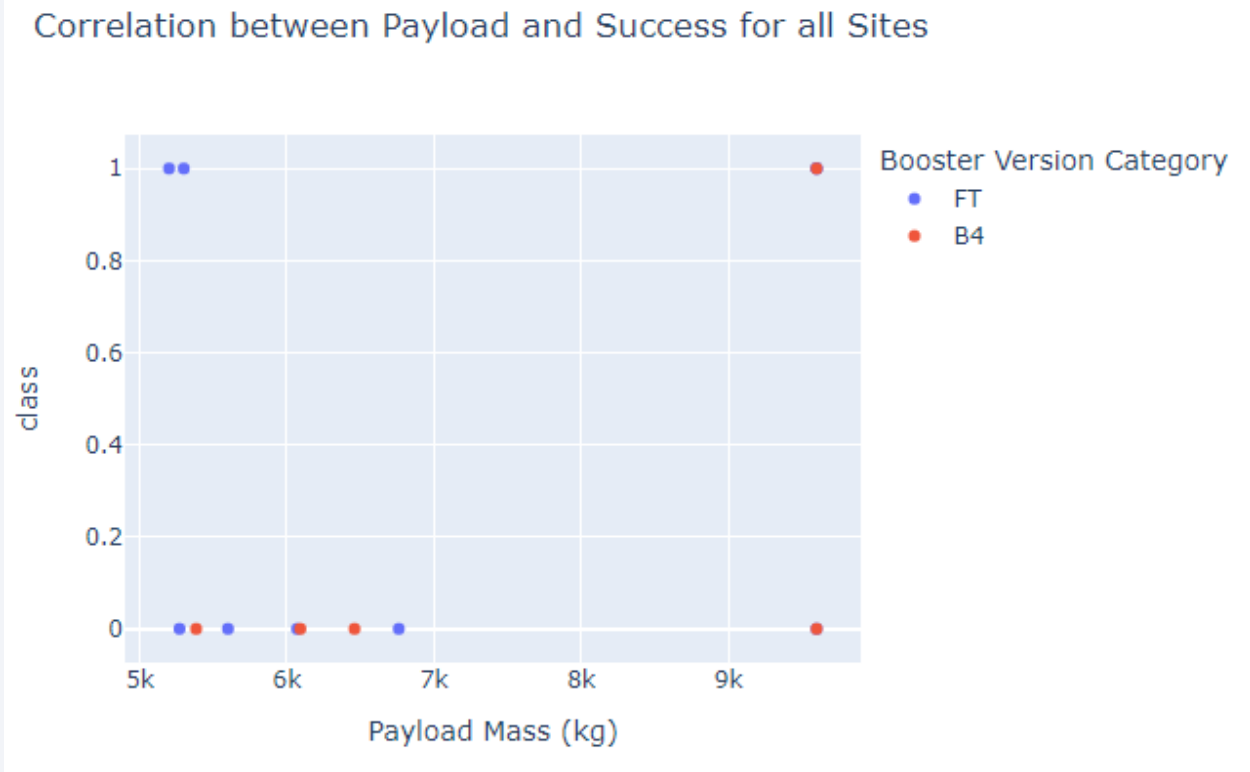
Total Success Launches for Site KSC LC-39A



We can observe that KSC LC-39A has 76.9% success rate and 23.1% failure rate

Correlation between Payloads and Success





Low weighted payloads (<5000 kg) have better success rate than heavy weighted payloads (>5000 kg)

Section 5

Predictive Analysis (Classification)

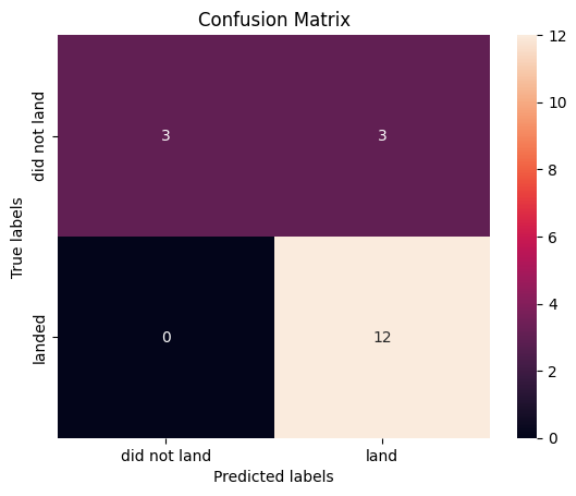
Classification Accuracy

Model	Best Accuracy	Test Accuracy
Logistic Regression	0.84643	0.83333
SVM	0.84821	0.83333
Decision Tree	0.875	0.66667
KNN	0.84821	0.83333

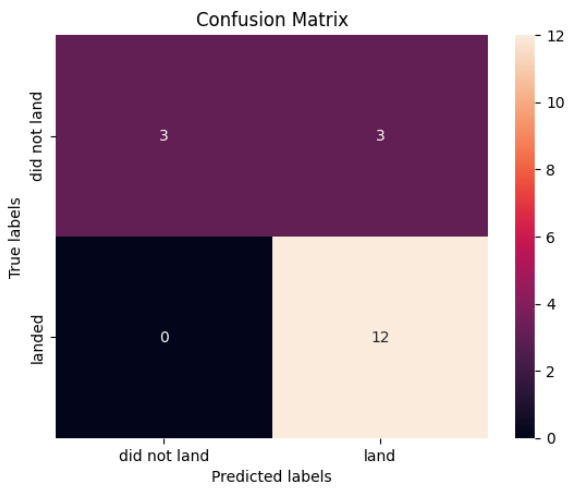
- Best accuracy is an accuracy using validation data, while test accuracy using test data.
- All models have similar performance in best accuracy, model with the highest best accuracy is Decision tree
- Logistic regression, SVM, and KNN all have test accuracy of 0.83, while Decision Tree has the lowest one with 0.67
- To choose the best model, we can calculate the average of both accuracies, which means **SVM** and **KNN** are tied as the best models.
- Best parameters for SVM: *{'C': 1.0, 'gamma': 0.0316227660168379, 'kernel': 'sigmoid'}*
- Best parameters for KNN: *{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}*

Confusion Matrix

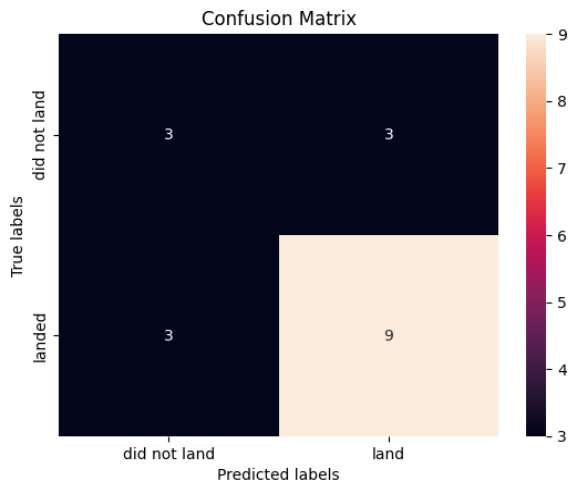
Logistic regression



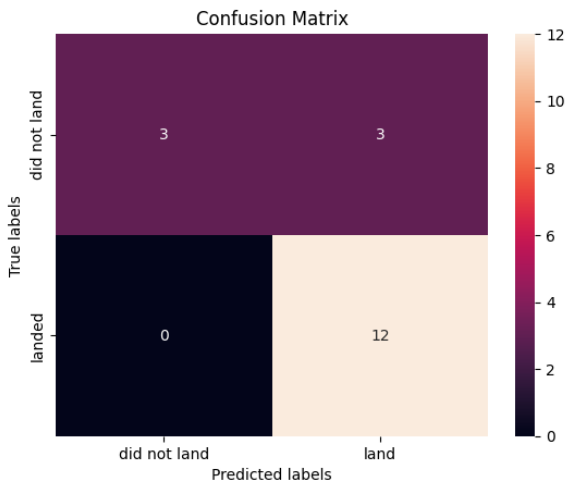
SVM



Decision Tree



KNN



- Confusion matrix is a 2x2 matrix, consisting of true positive (TP), false positive (FP), false negative (FN) and true negative (TN)
- For example, the best models (SVM and KNN) are able to predict 12 positive labels correctly (TP), 3 negative labels correctly (TN), 3 positive labels wrong (FP), and 0 negative labels wrong (FN)
- Examining the confusion matrix, we note that all models can distinguish between different classes
- The main problem is false positive

Conclusions

- There are several factors contributing to the success of a mission, such as flight number, payload mass, orbit type, etc.
- Orbits with the best success rates are ES-L1, GEO, HEO, and SSO
- The trend of launch success rate kept increasing since 2013
- Out of 4 launch sites, KSC LC-39A has the highest launch success rate
- Low weighted payloads have better success rate than heavy weighted payloads
- SVM and KNN are considered the best models compared to Logistic regression and Decision tree based on their average accuracy

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!