IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Francisco Javier Gomez Zaldivar

August 20, 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- In this Capstone project, we will tackle the task of predicting whether the first stage of SpaceX's Falcon 9 rocket will land successfully. The ability to reuse the first stage of its rockets is one of the key factors that allows SpaceX to offer launches at a significantly lower cost than its competitors.
- To achieve this, data was first collected from various sources and processed to improve its quality by performing data wrangling. Subsequently, exploratory data analysis was performed using tools such as SQL, data visualization, and basic statistical analysis. Interactive data visualization was also used, creating a dashboard and geospatial visualization. Finally, predictive models were built, evaluated, and refined to determine the likelihood of the Falcon 9 first stage landing successfully.

Introduction

The ability to reuse the first stage of its rockets is one of the key factors that allows SpaceX to offer launches at a significantly lower cost than its competitors. While SpaceX advertises Falcon 9 rocket launches at a price of $62 million, other providers charge more than $165 million for each launch. Much of this savings is attributed to the reuse of the first stage.

The purpose of this analysis is to develop a predictive model to determine the probability of the Falcon 9 first stage landing successfully. This prediction is critical as it directly influences the overall cost of the launch. The information obtained can be of great value to competing companies wishing to bid against SpaceX in the space launch market.

Section 1

Methodology

Methodology

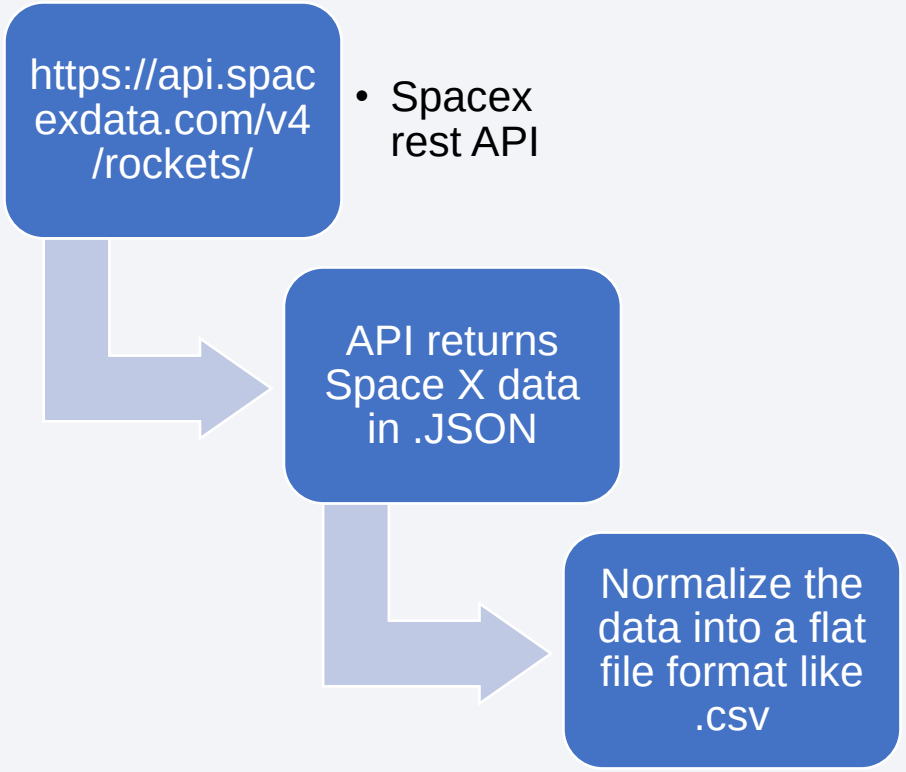
The general methodology consisted of the following:

- Data collection methodology:
  - Data collection was done using the Space X API and WebScraping from the Wikipedia site.
- Perform data wrangling
  - The data was processed with the Python libraries Pandas and Numpy
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The Machine Learning algorithms that were used to build the models were: Logistic regression, Support vector machine (SVM), Decision tree, K-nearest neighbors (KNN)

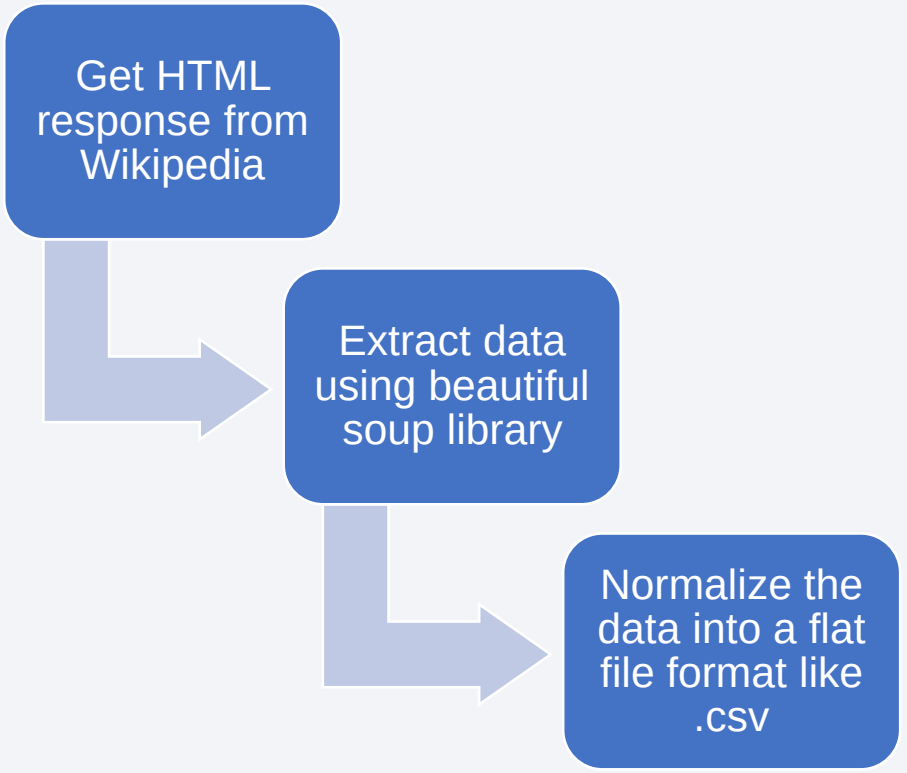
1 Data Collection

The data sets were collected using two different approaches

SPACE X API



WEB SCRAPING



Data Collection – SpaceX API

- Request rocket launch data from SpaceX API

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

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

- Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

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

GitHub URL to Notebook

- Apply custom functions to clean data

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

Data Collection – SpaceX API

- Create a Pandas data frame from the dictionary launch_dict.

```
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}
```

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

GitHub URL to Notebook

- Filter the dataframe to only include Falcon 9 launches and export data file (CSV)

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL

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

- Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.text, 'html.parser')
```

- Find all tables on the wiki page and Getting column names

```
# 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')
```

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

GitHub URL to Notebook

Data Collection - Scraping

- Create an empty dictionary with keys from the extracted column names

```
launch_dict= dict.fromkeys(column_names)

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

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

GitHub URL to Notebook

- Appending data to keys (see the Notebook)

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible"))
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Data Collection - Scraping

- Create a data frame

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df
```

- Dataframe to .CSV

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

GitHub URL to Notebook

2 Data Wrangling

- Determine the number of launches on each site:

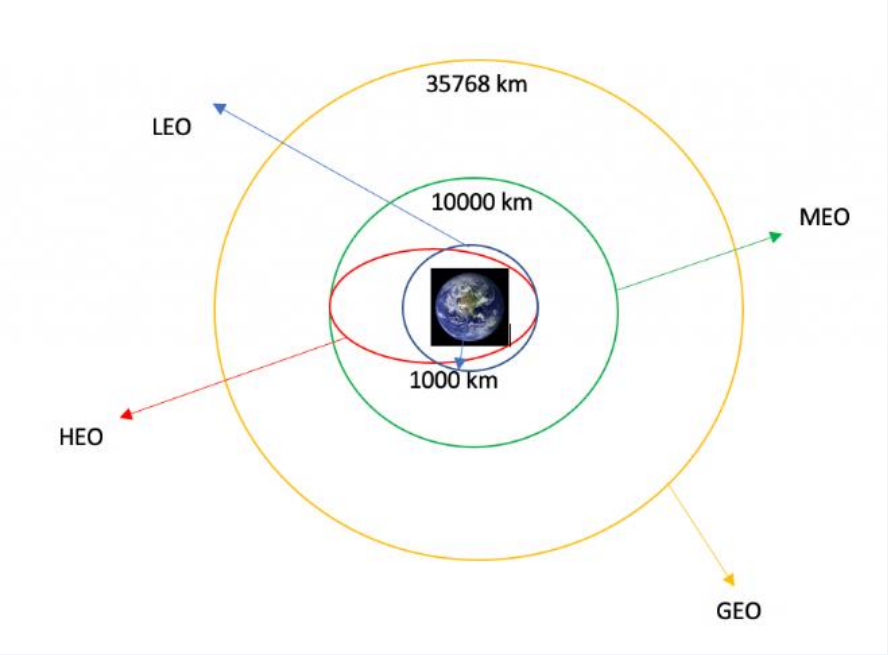
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

- Each launch aims at a specific orbit (the types of orbits are shown in the image). So the number and occurrence of each orbit was calculated

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

[GitHub URL to Notebook]

Diagram showing the type of orbits



Data Wrangling

- Calculate the number and occurrence of mission outcome of the orbits

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

- Create a landing outcome label from Outcome column

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

- Export data set as .CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```

GitHub URL to Notebook

3 EDA with Data Visualization

The graphs that were plotted were the following:

- *Scatter Plot: Scatter plots show how much one variable is affected by another*.

Flight Number VS. Payload Mass

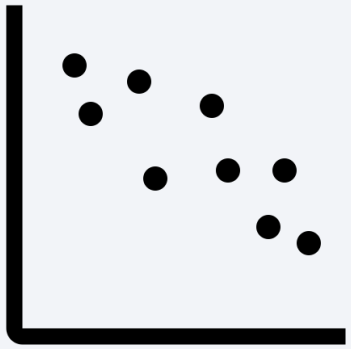
Flight Number VS. Launch Site

Payload VS. Launch Site

Orbit VS. Flight Number

Payload VS. Orbit Type

Orbit VS. Payload Mass



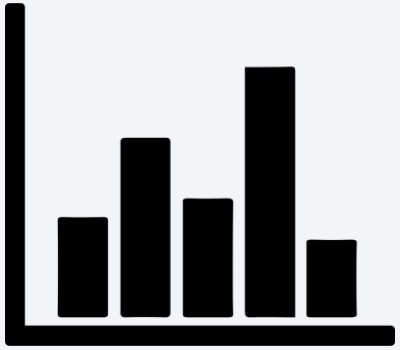
- *Line Plot: Line graphs are useful in that they show data variables and trends*

Success Rate VS. Year

GitHub URL to Notebook

- *Bar plot: The graph represents categories on one axis and a discrete value in the other*

Success Rate VS Orbit Type



4 EDA with SQL

The SQL queries that were performed for the analysis are listed below:

- Display the names of the 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 succesful landing outcome in ground pad was acheived.
- 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. Use a subquery
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months 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.

GitHub URL to Notebook

5 Build an Interactive Map with Folium

- To visualize the launch data on an interactive map, we used the latitude and longitude coordinates of each launch site and added a circle marker around each site, labeled with the name of the launch site.
- We assigned the launch outcomes (failures, successes) to classes 0 and 1, using green and red markers on the map within a MarkerCluster.
- Using Haversine's formula, we calculated the distance from the launch site to various landmarks to identify trends related to the surroundings of the launch site. Lines were drawn on the map to measure the distance to these landmarks.

GitHub URL to Notebook

6 Build a Dashboard with Plotly Dash

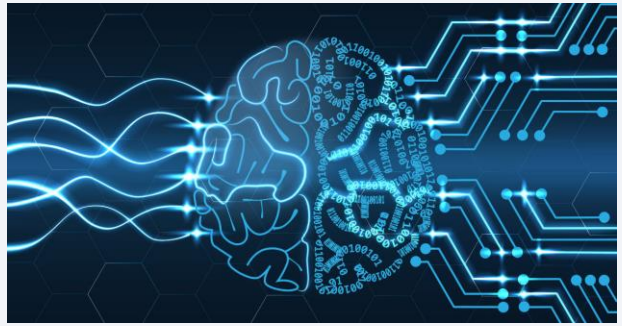
Dash functions are used to build an interactive platform where you can adjust inputs with a dropdown menu and a range slider.

- Using a pie chart and a scatterplot, the interactive site shows:
- The total number of successful launches at each site.
- The relationship between payload mass and mission outcome (success or failure) for each launch site.

GitHub URL to code

7 Predictive Analysis (Classification)

The machine learning prediction phase include the following steps:



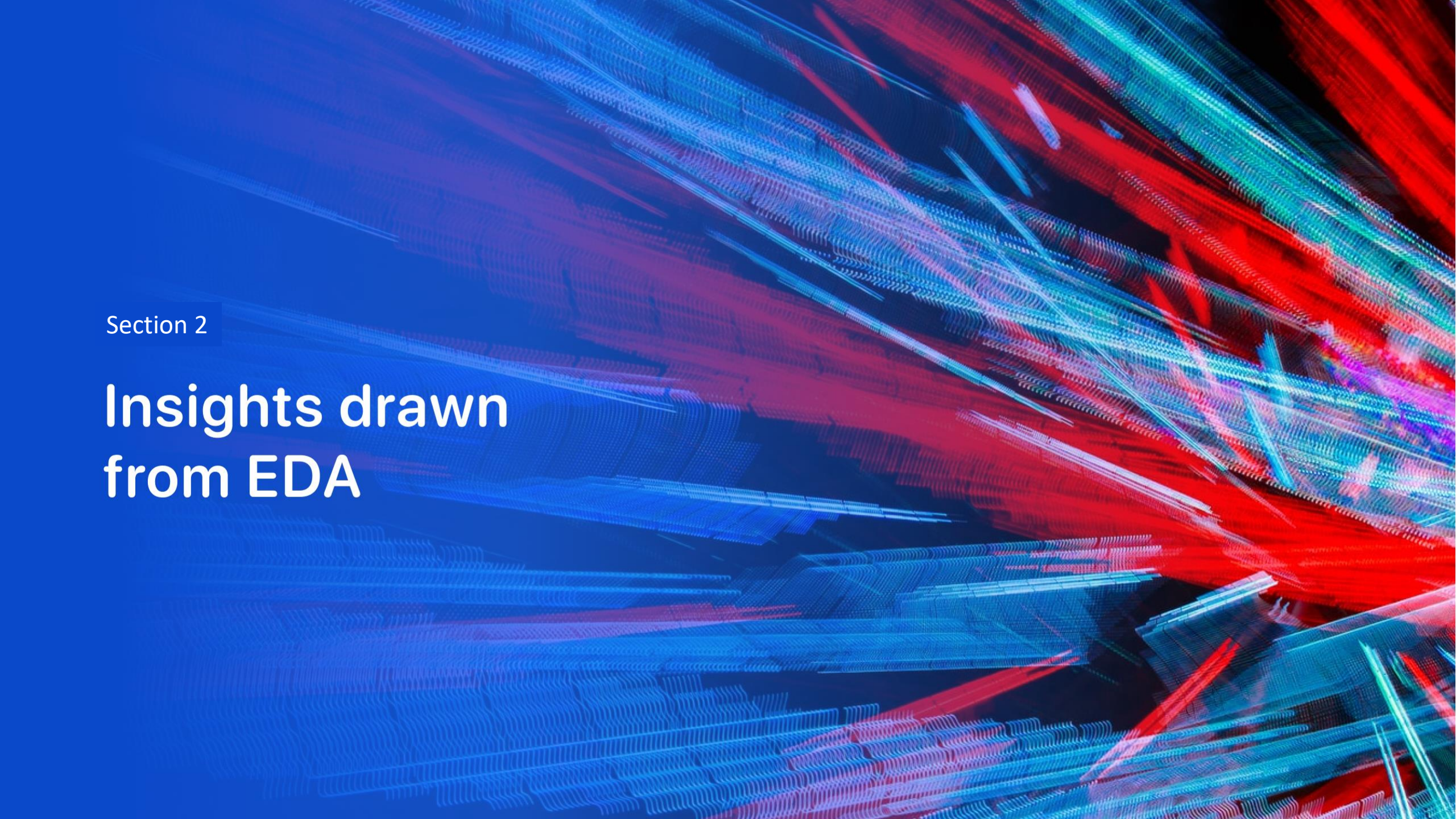
- Standardizing the data
- Splitting the data into training and test data
- Creating machine learning models, which include:
- Logistic regression
- Support vector machine (SVM)
- Decision tree
- K nearest neighbors (KNN)
- Fit the models on the training set
- Find the best combination of hyperparameters for each model
- Evaluate the models based on their accuracy scores and confusion matrix

The model with the best accuracy score wins the best performing model

GitHub URL to Notebook

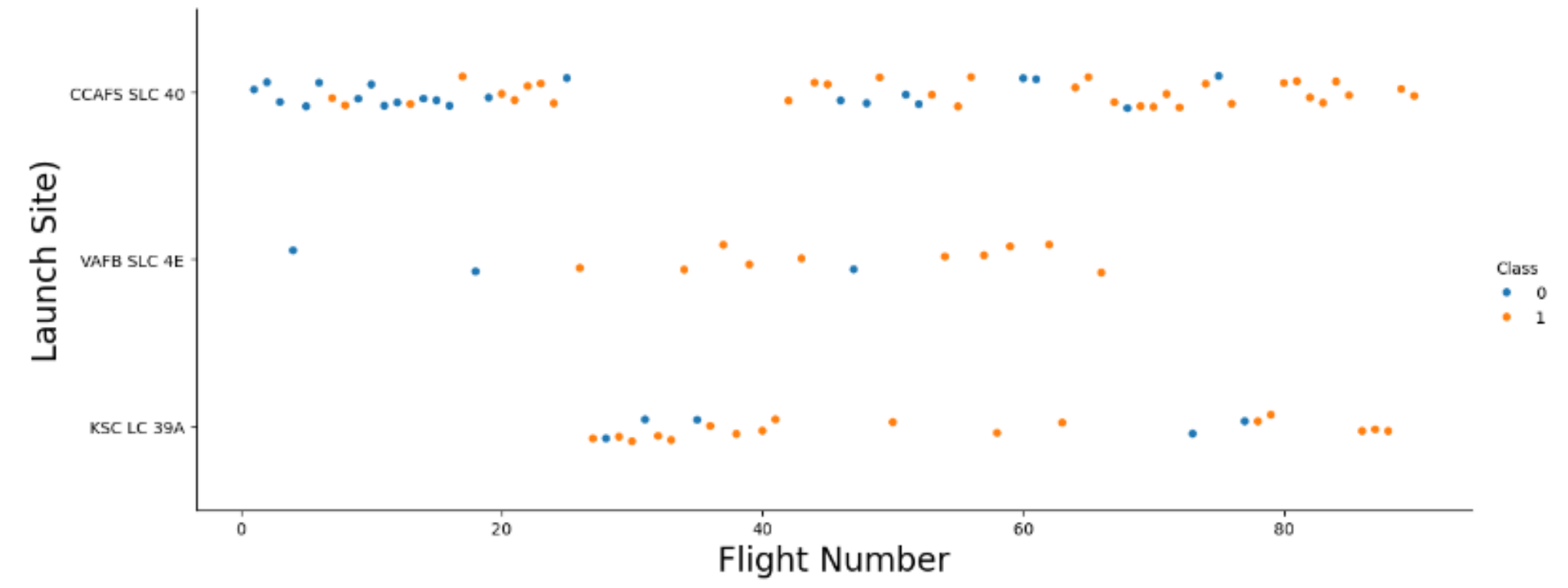
Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

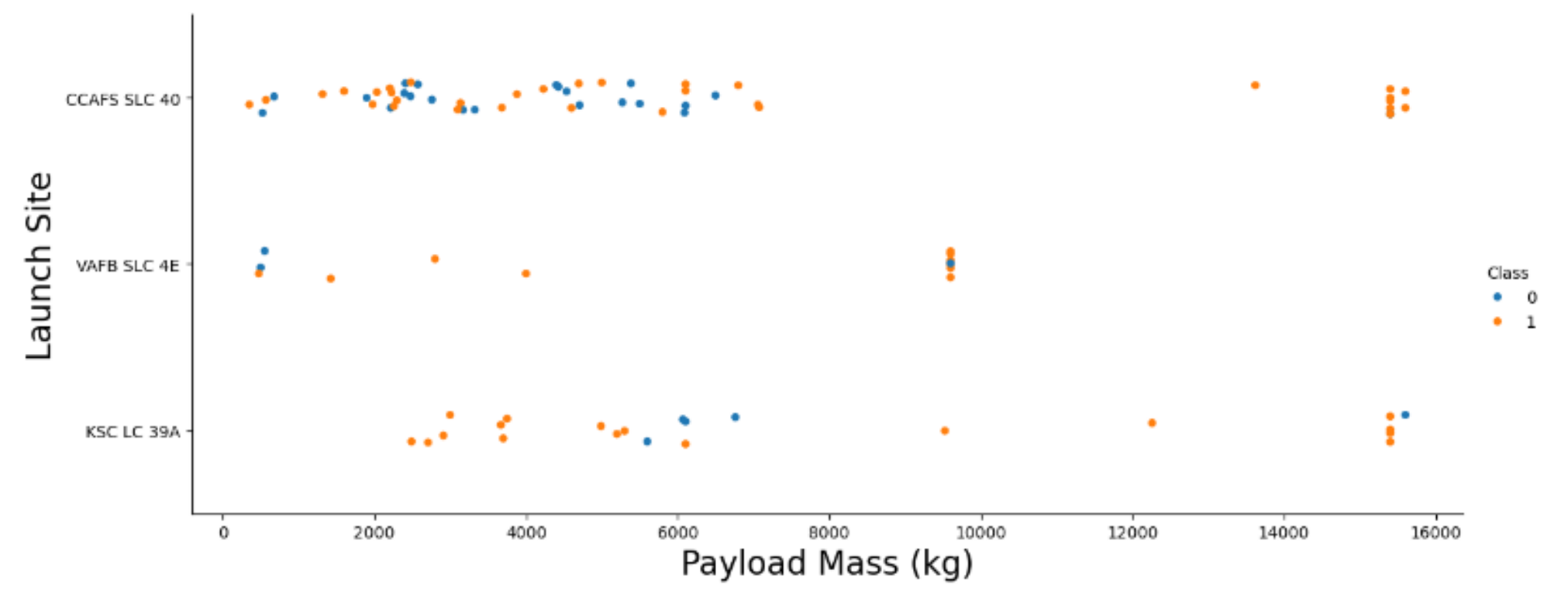
The relationship between flight number and launch site



The success rate at a launch site increases as the number of flights from that site rises.

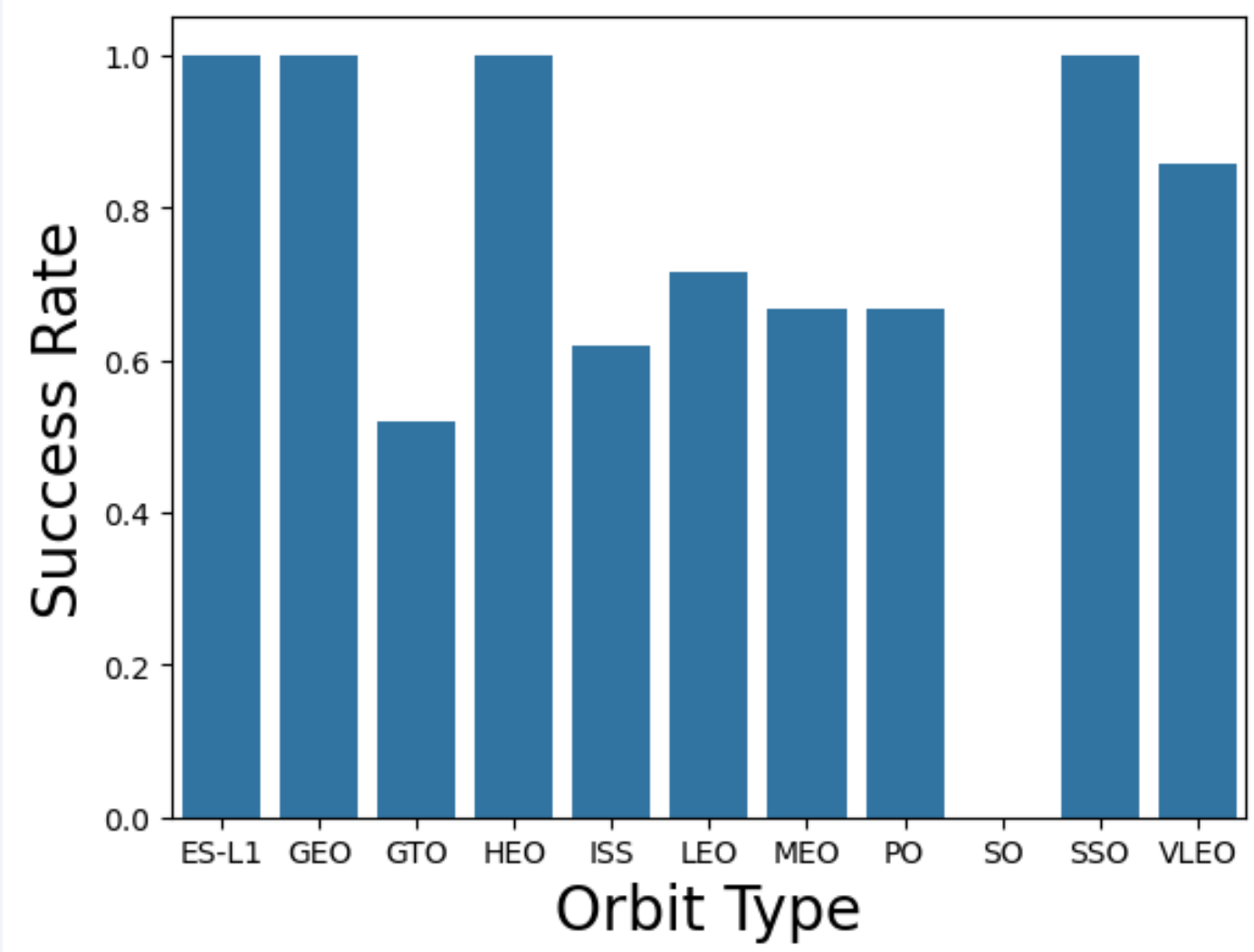
Payload vs. Launch Site

The relationship between payload mass and launch site



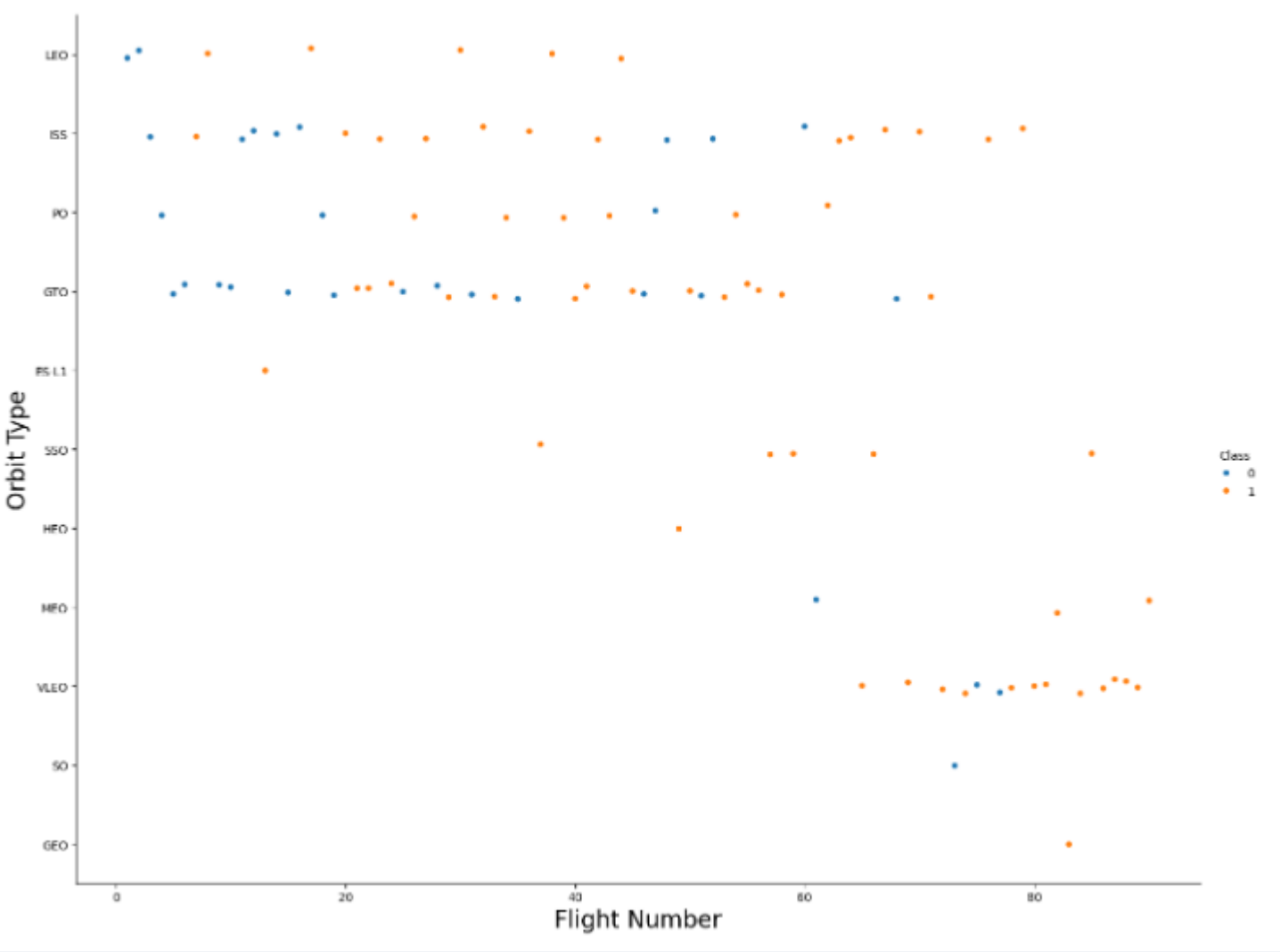
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.

Success Rate vs. Orbit Type



The ES-L1, GEO, HEO and SSO orbits have the highest success rate

Flight Number vs. Orbit Type

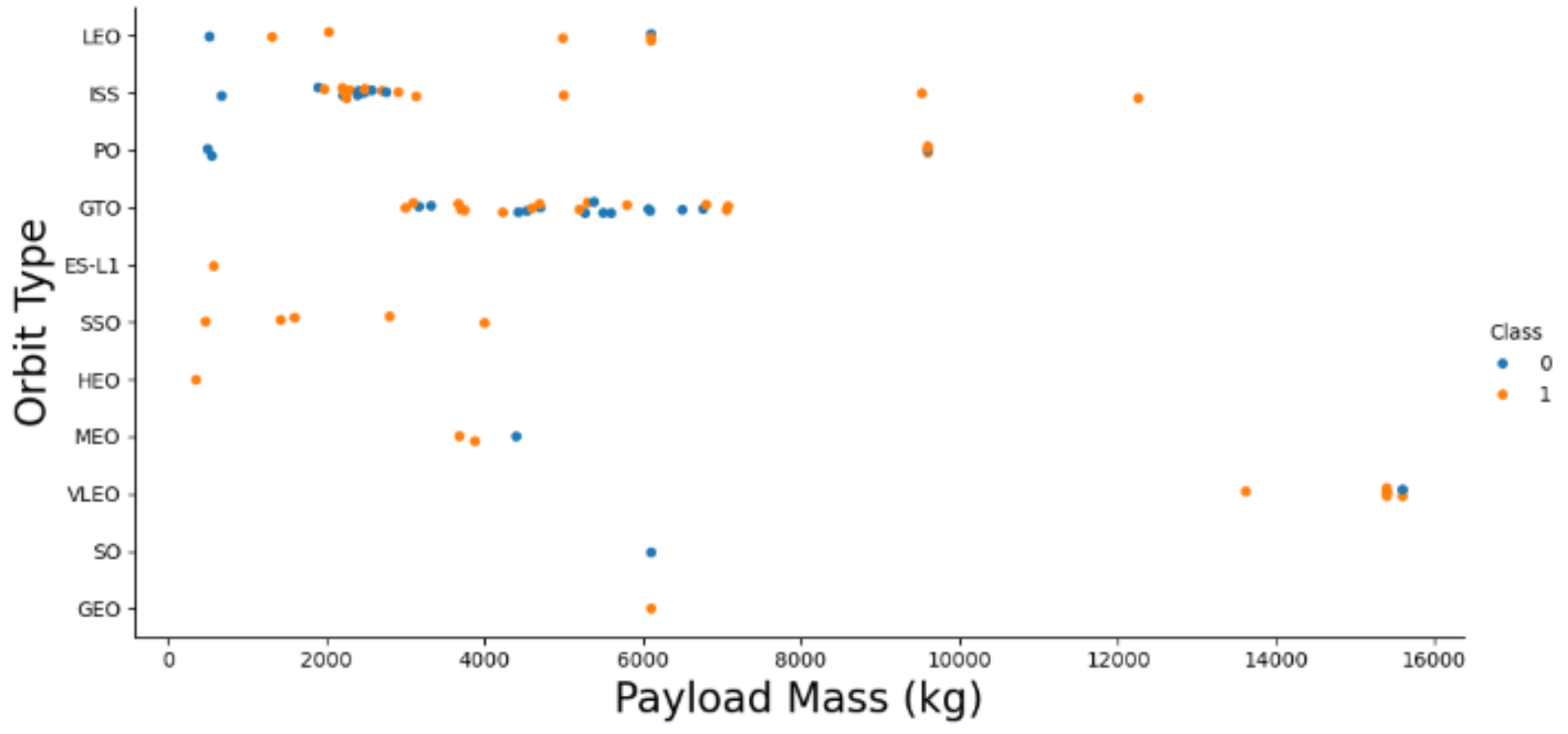


In general, for popular orbit types like LEO and GTO, as the number of flights increases, there appears to be a trend toward a higher number of successes (more orange dots) compared to failures.

This trend reinforces the idea that more experience (more flights) in certain orbit types can lead to greater success.

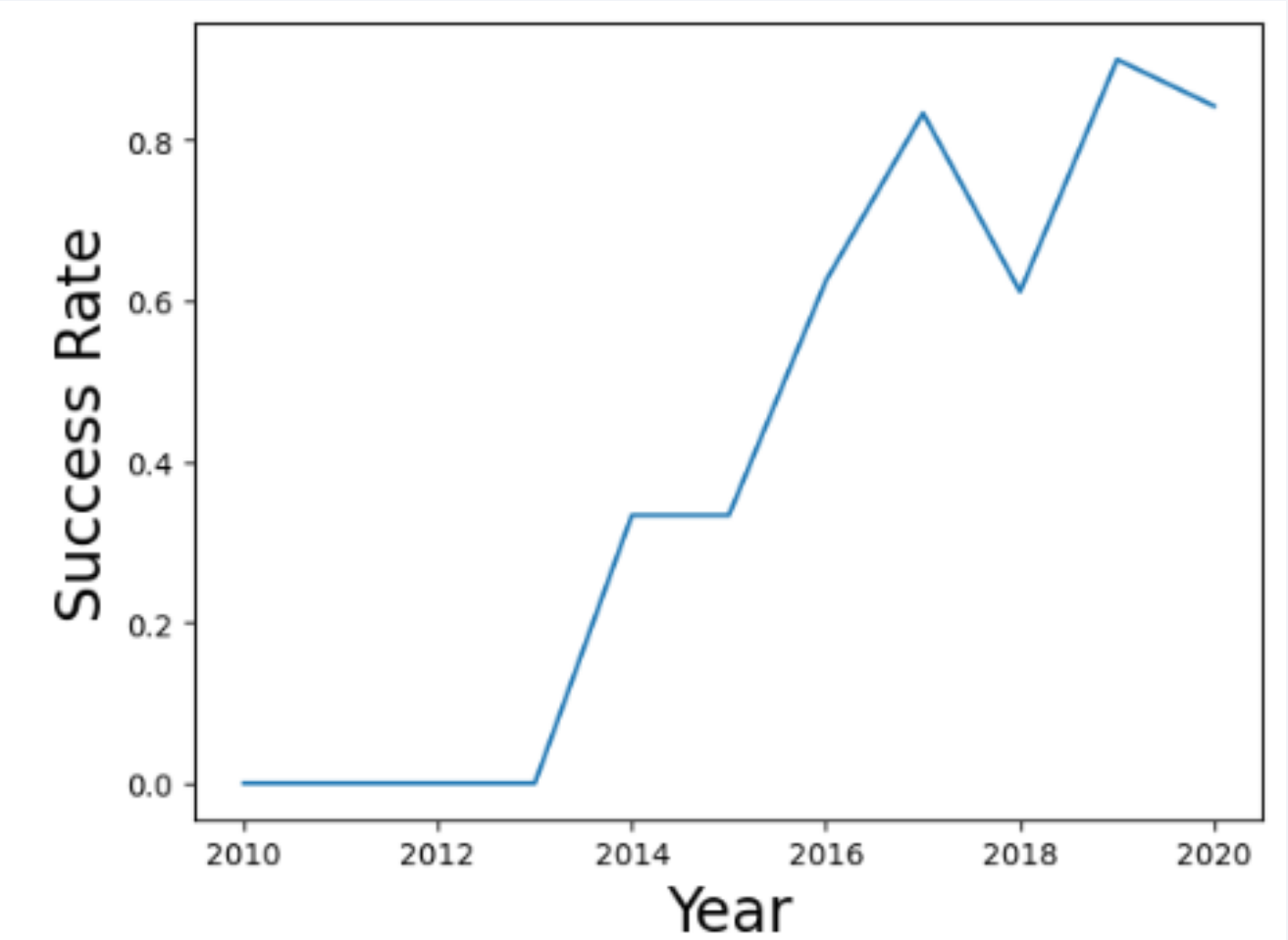
Payload vs. Orbit Type

The relationship between payload mass and orbit type



There does not appear to be a clear relationship between payload mass and launch success. Both light and heavy payloads have examples of successful and failed launches, suggesting that payload mass is not the sole determinant of success.

Launch Success Yearly Trend

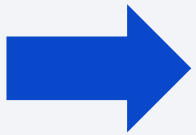


There is a clear and sustained increase in the success rate from about 2013 to 2020. For 2019 and 2020, the success rate reaches very high levels, around 80-90%. This suggests that for those years, most launches were successful, reflecting a high level of maturity in the launch capability.

All Launch Site Names

- Find the names of the unique launch sites

SQL Query
%sql select distinct (LAUNCH_SITE) from SPACEXTBL;



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

Explanation

Selects all unique values from the LAUNCH_SITE column in the SPACEXTBL table. The result will be a list of launch sites without duplicates that are present in the table.

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

SQL Query

%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 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	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

Explanation

Selects all columns (*) from the SPACEXTBL table, but only for rows where the LAUNCH_SITE value starts with "CCA". Also, the query returns only the first 5 rows that meet that condition.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

SQL Query

```
%sql SELECT SUM
(PAYLOAD_MASS__KG_) FROM
SPACEXTBL WHERE
CUSTOMER='NASA (CRS)'
```

SUM (PAYLOAD_MASS__KG_)
45596

Explanation

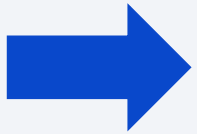
Calculates the total sum of payload mass (in kilograms) of all launches flown by SpaceX for NASA under the CRS contract, as recorded in the SPACEXTBL table.

Average Payload Mass by F9 v1.1

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

SQL Query

```
%sql SELECT
AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL WHERE
booster_version LIKE 'F9 v1.1%'
```



AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

Explanation

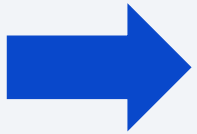
Calculates the average value of the payload mass (in kilograms) of launches performed with the booster rocket version starting with 'F9 v1.1', using the data stored in the SPACEXTBL table.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

SQL Query

```
%sql SELECT MIN(DATE) FROM
SPACEXTBL WHERE
Landing_Outcome = 'Success
(ground pad)';
```



MIN(DATE)
2015-12-22

Explanation

This SQL query returns the earliest (first) date on which a SpaceX rocket successfully landed on a ground platform, based on data stored in the SPACEXTBL table.

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 Query

```
%sql SELECT DISTINCT
BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_
BETWEEN 4000 AND 6000 AND
LANDING_OUTCOME = 'Success (drone
ship)';
```

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

Explanation

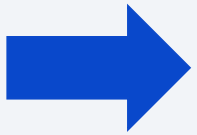
This SQL query selects all unique booster rocket versions (BOOSTER_VERSION) that were used in launches where the payload mass was between 4000 and 6000 kilograms and the rocket achieved a successful landing on an unmanned spacecraft in the ocean, using data stored in the SPACEXTBL table.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

SQL Query

```
%sql SELECT MISSION_OUTCOME,
COUNT(*) FROM SPACEXTBL GROUP
BY MISSION_OUTCOME
```



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

Explanation

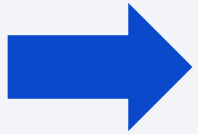
This SQL query groups missions by their outcome (MISSION_OUTCOME) and counts how many missions are in each group. The result will be a list of each type of mission outcome (e.g. "Success", "Failure") along with the number of missions that had that outcome, using data stored in the SPACEXTBL table.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

SQL Query

```
%sql SELECT
BOOSTER_VERSION,PAYLOAD_MASS_
_KG_ FROM SPACEXTBL WHERE
PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM
SPACEXTBL)
```



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

Explanation

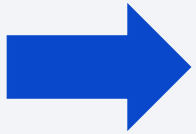
This SQL query selects the booster rocket versions (BOOSTER_VERSION) and payload masses (PAYLOAD_MASS__KG_) from all launches where the payload had the largest mass recorded in the SPACEXTBL table. That is, the query returns the details of the launches with the largest payload mass.

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

SQL Query

```
%sql SELECT substr(Date,6,2) as month,
DATE,BOOSTER_VERSION,
LAUNCH_SITE, [Landing_Outcome] \
FROM SPACEXTBL \
where [Landing_Outcome] = 'Failure
(drone ship)' and substr(Date,0,5)='2015';
```



month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Explanation

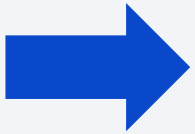
This SQL query extracts specific information about SpaceX launches that occurred in 2015 and ended in a failed landing on a drone ship. The query returns the month of the launch (in numeric format), the full launch date, the booster rocket version, the launch site, and the landing result.

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 Query

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS qty FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY qty DESC;
```



Landing_Outcome	qty
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

Explanation: This SQL query counts the number of times each type of landing outcome (LANDING_OUTCOME) occurred on SpaceX launches that occurred between June 4, 2010, and March 20, 2017. It then groups the results by landing type and sorts them in descending order of occurrence, with the most frequent results displayed first.

Section 3

Launch Sites Proximities Analysis

All launch sites' location markers on a global map

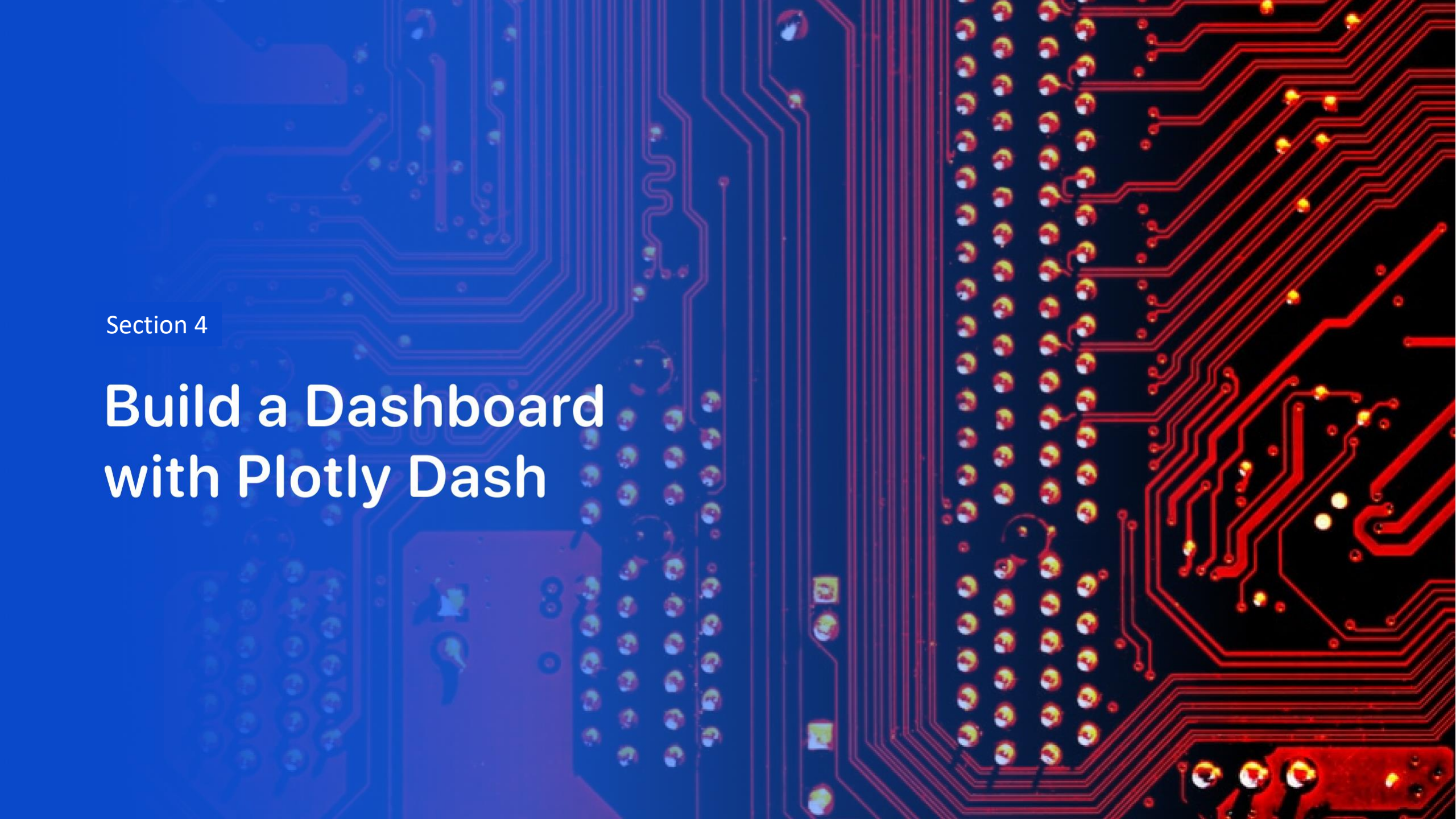
We can see that SpaceX launch sites are situated along the coasts of the United States, namely in Florida and California.

The succeeded launches and failed launches for each site on map

If we focus on one of the launch sites, we can see green and red markers. Each green marker indicates a successful launch, while each red marker signifies a failed launch.

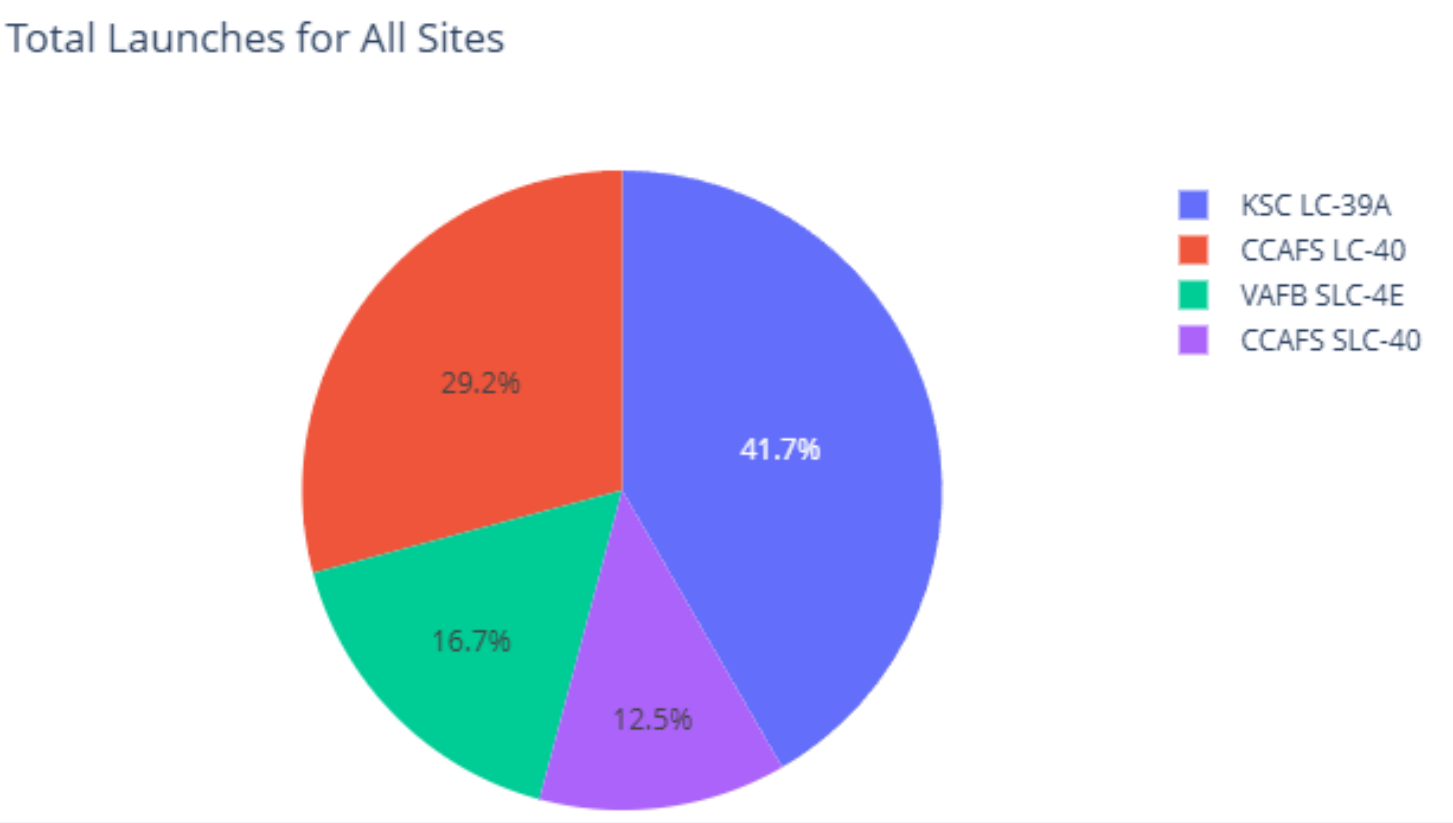
Distances between a launch site to its proximities such as the nearest city, railway, or highway

Distance from a launch site to a selected point on the coast


Section 4
Build a Dashboard with Plotly Dash

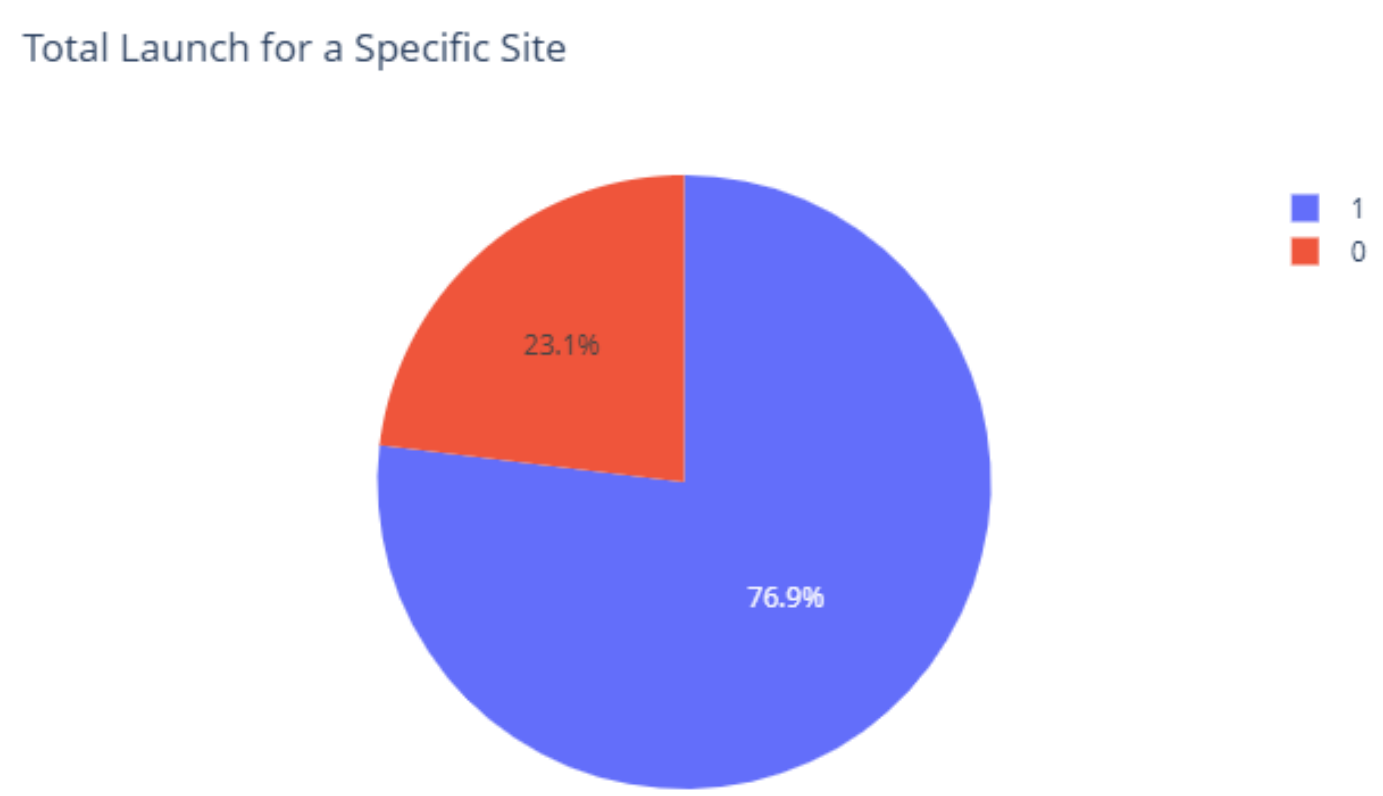
Launch success count for all sites

In the pie chart it can be seen that the KSC LC-40 site has the highest % of success in launches (41.7%), followed by the CCAFS LC-40 site with 29.2% success.



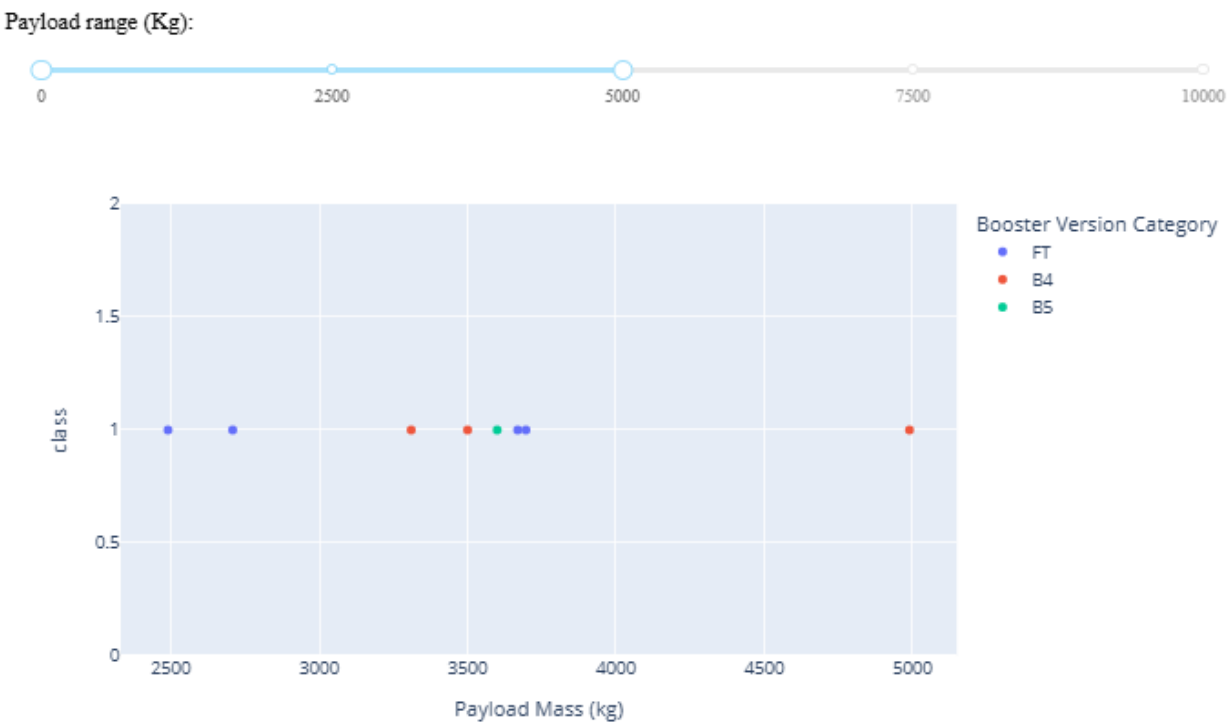
Launch site with highest launch success ratio

The KSC LC-39A site achieved a success rate of 76.9% while obtaining a failure rate of 23.1%.

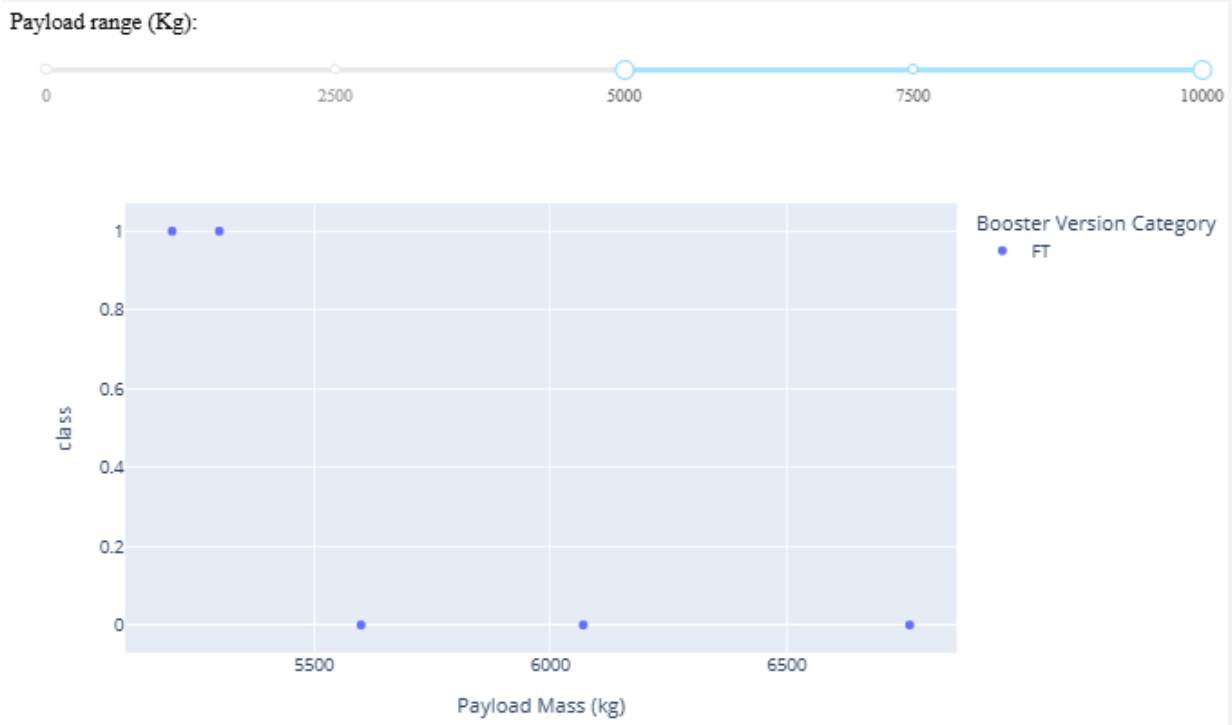


Payload vs. Launch Outcome scatter plot for all sites

Payload range:0-5000 Kg



Payload range:5000-10000 Kg



We can state that the success rates of low weight payloads are higher than those of heavy weight payloads (Class 0 represents failed launches while class 1 represents successful launches)

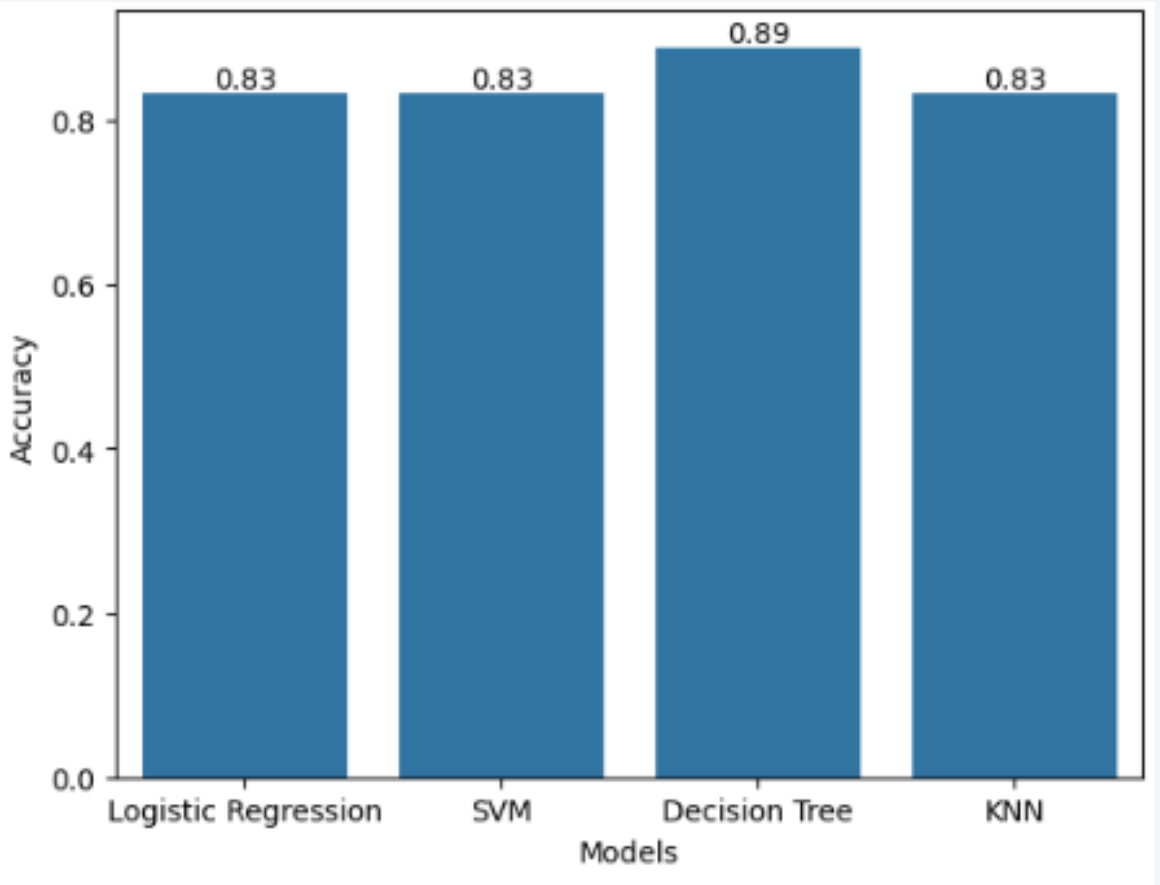
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Of the four models presented, the decision tree was the one that showed the highest accuracy score, which indicates that for this specific data set and problem it is the appropriate predictive analysis model.

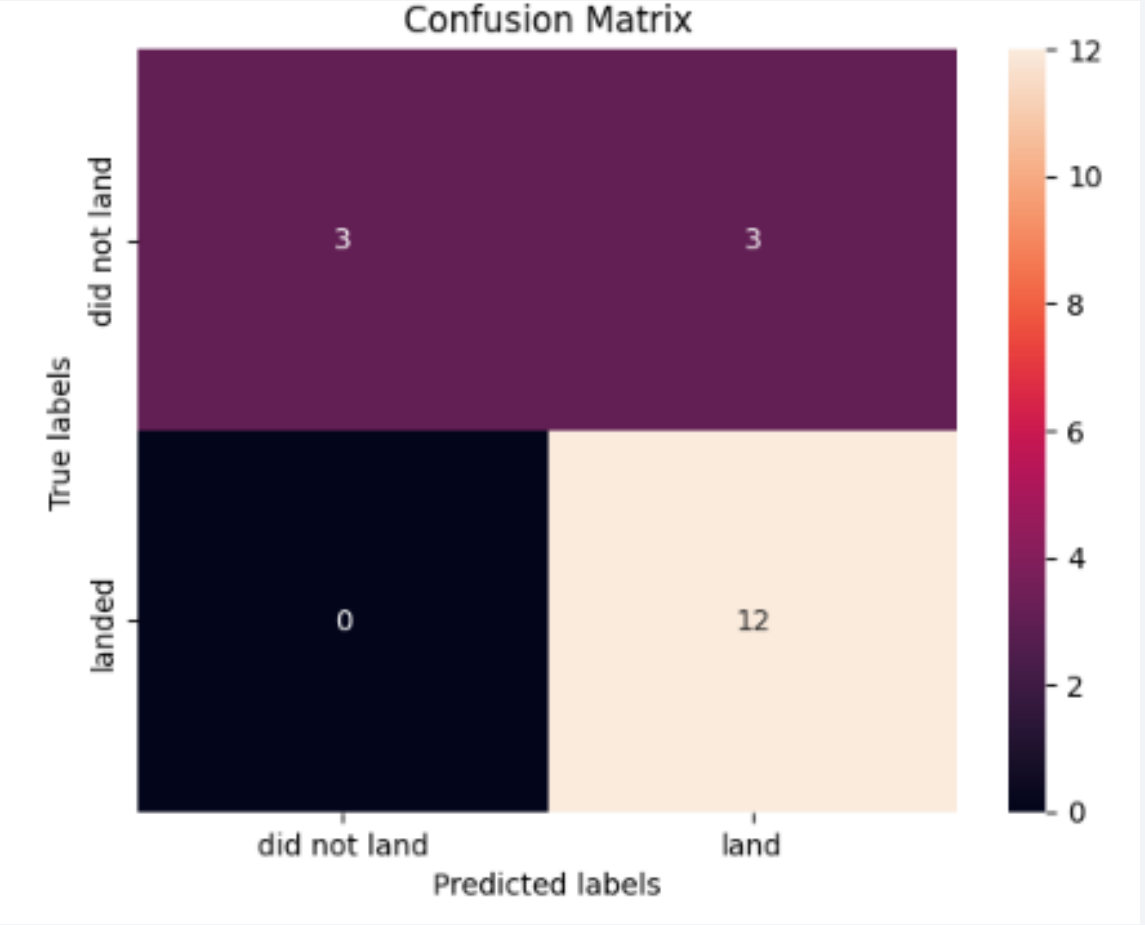
	Models	Accuracy
0	Logistic Regression	0.833333
1	SVM	0.833333
2	Decision Tree	0.888889
3	KNN	0.833333



Confusion Matrix

Confusion matrix of the decision tree model

This decision tree model seems to perform well, especially in predicting the land class, as it has made no errors in that class (0 false negatives). However, it has some errors in predicting the did not land class (3 false positives). Therefore the landings are predicted with good accuracy.



Conclusions

- SpaceX's launch success rate shows a trend of improving over time, reflecting that over the years, SpaceX is constantly refining its launch procedures and technology.
- GEO, HEO, SSO, and ES-L1 orbits have been shown to have the best launch success rates, suggesting that missions targeting these orbits have a higher chance of being completed successfully.
- Lighter payloads tend to perform better in terms of launch success rates, implying that lighter weight missions have a higher chance of success.
- The KSC LC-39A launch site has consistently shown the most successful results, making it the best location for SpaceX launches in terms of reliability.
- The Tree Classification Algorithm has proven to be the most effective for this dataset when compared to other machine learning models, suggesting its ability to capture relevant patterns in Falcon 9 launches.

Thank you!