IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Batool Alghoraibi
8/16/2023

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

Executive Summary

The aim of this research is to analyze data from SpaceX Falcon 9, which has been collected from various sources, and utilize Machine Learning models to predict the success of the first stage landing. This prediction capability provides other space agencies with the ability to determine whether they should compete with SpaceX.

- Methodologies Summary:
  - Data collection, wrangling, and formatting
  - Exploratory data analysis
  - Interactive data visualization
  - Machine learning prediction
- Summary of Results
  - This report will present the results in various formats, including:
  - Data analysis results
  - Data visualization and interactive dashboards
  - Predictive model analysis results

Introduction

- Project background and context
  - With the recent advancements in private space travel, the space industry is becoming increasingly popular and accessible to the general population. However, the high cost of launches remains a significant barrier for new competitors entering the space race.
  - SpaceX, with its first stage reuse capabilities, holds a significant advantage over its competitors. Each SpaceX launch costs approximately $62 million, whereas other competitors spend around $165 million per launch, as SpaceX can reuse the first stage for future missions.
- Problems you want to find answers
  - Determine the likelihood of successful landing for the first stage of SpaceX Falcon 9.
  - Investigate the impact of different parameters/variables on the landing outcomes, such as launch site, payload mass, booster version, and more.
  - Analyze the correlations between launch sites and success rates to understand if certain locations contribute to higher or lower landing success rates.

Section 1

Methodology

Methodology

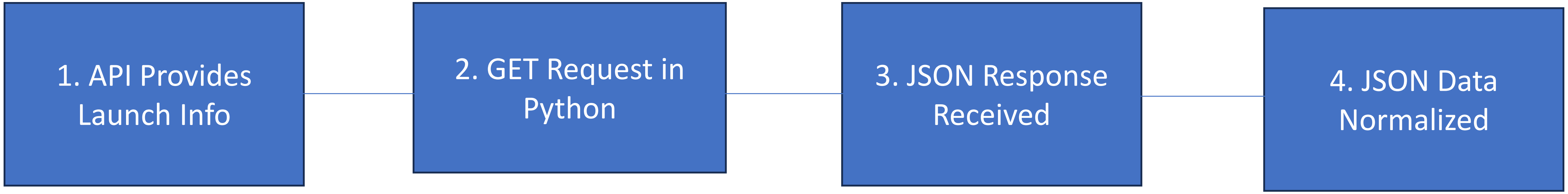
Executive Summary

- Data collection methodology:
  - Data for this research was collected through two methods: accessing the SpaceX REST API and web scraping Wiki pages.
- Perform data wrangling
  - Attribute Identification
  - Attribute Exploration
  - Outcome Conversion
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Created a column for ‘class’; standardized and transformed data; train/test split data; find best classification algorithm (Logistic regression, SVM, decision tree, & KNN) using test data

Data Collection

The data for this research was collected using two main methods: accessing the SpaceX REST API and performing web scraping.

SpaceX REST API:



Web Scraping:

Data Collection – SpaceX API

1. API Request and read response into DF
2. Declare global variables
3. Call helper functions with API calls to populate global vars
4. Construct data using dictionary
5. Convert Dict to Dataframe, filter for Falcon9 launches, covert to CSV

1. Create API GET request, normalize data and read in to a Dataframe:

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

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

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

2. Declare global variable lists that will store data returned by helper functions with additional API calls to get relevant data

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

3. Call helper functions to get relevant data where columns have IDs (e.g., rocket column is an identification number)
   - getBoosterVersion(data)
   - getLaunchSite(data)
   - getPayloadData(data)
   - getCoreData(data)

4. Construct dataset from received data & combine columns into a dictionary:

```
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}
```

4. Create Dataframe from dictionary and filter to keep only the Falcon9 launches:

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

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df_launch[df_launch['BoosterVersion']!= 'Falcon 1']
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub

Data Collection - Scraping

1. Python BeautifulSoup

2. Scrape HTML Tables

3. Parse into Pandas DataFrame

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response tex
from bs4 import BeautifulSoup
response_object = response.content
soup = BeautifulSoup(response_object, 'html.parser')
```

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

```
# Use soup.title attribute
soup.title
```

TASK 2: Extract all column/variable names from the HTML table header

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

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

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

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

```
# 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)
df
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	\n	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1063.2	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1067.1	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

121 rows × 11 columns

[GitHub]

Data Wrangling

The data processing steps involved in this scenario can be summarized as follows:

1. **Attribute Identification**: Review and identify the relevant attributes or columns in the dataset, such as Flight Number, Date, Booster version, Payload mass, Orbit, Launch Site, Outcome, Grid Fins, Reused, Legs, Landing pad, Block, Reused count, Serial, Longitude, and Latitude of launch.

2. **Attribute Exploration**: Examine the specific details and categories within certain attributes. For example, explore the launch sites, categorize orbits into LEO and GTO, and analyze the outcomes of successful and unsuccessful landings.

3. **Outcome Conversion**: Define the desired conversion of landing outcomes to classes. Assign "0" for unsuccessful landings and "1" for successful landings. Designate the variable "Y" as the classification variable representing the outcome of each launch.

By following these steps, the dataset is prepared with identified attributes, explored attribute details, and converted landing outcomes for further analysis and modeling.

Data Wrangling

1. Load dataset in to Dataframe → 2. Find patterns in data → 3. Create landing outcome label

1. Load SpaceX dataset (csv) in to a Dataframe

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appd
art_1.csv")
```

2. Find data patterns:
   i. Calculate the number of launches on each site

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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

   ii. Calculate the number and occurrence of each orbit

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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
GEO       1
HEO       1
SO        1
ES-L1     1
```

   iii. Calculate number/occurrence of mission outcomes per orbit type

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

3. Create a landing outcome label from Outcome

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

This variable will represent the classification variable that of each launch. If the value is zero, the first stage did not means the first stage landed Successfully

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	1
1	1
2	1
3	1
4	1
5	1
6	0
7	0

GitHub

EDA with Data Visualization

The following relationships were represented using pandas, numpy and seaborn libraries:

- The relationship between Flight Number and Launch Site using scatter point
- the relationship between Payload and Launch Site using scatter point
- the relationship between success rate of each orbit type using bar chart
- the relationship between FlightNumber and Orbit type using scatter plot
- the relationship between Payload and Orbit type using scatter point chart
- the launch success yearly trend using line chart

EDA with SQL

- 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.
- 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

GitHub

Build an Interactive Map with Folium

Markers of all Launch Sites:

- A marker with a circle, popup label, and text label of the NASA Johnson Space Center was added using its latitude and longitude coordinates as the starting location.
- Markers with circles, popup labels, and text labels of all Launch Sites were added using their latitude and longitude coordinates to display their geographical locations and proximity to the Equator and coasts.
  Colored markers of the launch outcomes for each Launch Site:
- Colored markers indicating success (green) and failed (red) launches were added using a Marker Cluster to identify launch sites with relatively high success rates.
  Distances between a Launch Site and its proximities:
- Colored lines were added to illustrate the distances between the Launch Site and its proximities such as the railway and coastline.

GitHub

Build a Dashboard with Plotly Dash

Launch Sites Dropdown List:

- Implemented a dropdown list to allow users to select a Launch Site.

Pie Chart showing Success Launches (All Sites/Certain Site):

- Created a pie chart to display the total count of successful launches for all sites, and if a specific Launch Site is selected, the chart shows the success vs. failed counts for that site.

Slider of Payload Mass Range:

- Added a slider that allows users to select a range for the Payload mass.

Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:

- Generated a scatter chart to visualize the relationship between Payload mass and Launch Success rate for different Booster versions.

GitHub

Predictive Analysis (Classification)

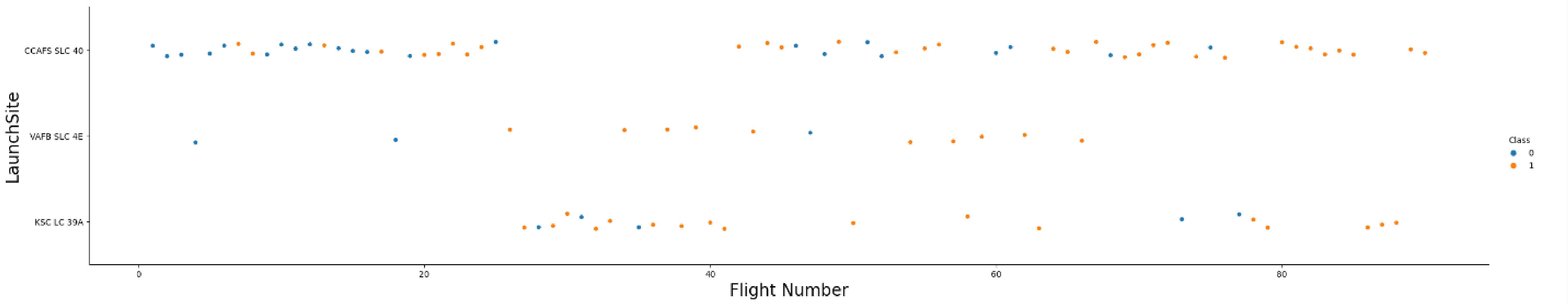
1. Creating a NumPy array from the "Class" column in the data.
2. Standardizing the data using StandardScaler, and then fitting and transforming it.
3. Splitting the data into training and testing sets using the train_test_split function.
4. Creating a GridSearchCV object with cv=10 to find the best parameters.
5. Applying GridSearchCV on LogReg, SVM, Decision Tree, and KNN models.
6. Calculating the accuracy on the test data using the .score() method for all models.
7. Examining the confusion matrix for all models.
8. Finding the method that performs best by examining the Jaccard score and F1 score metrics.

Results

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

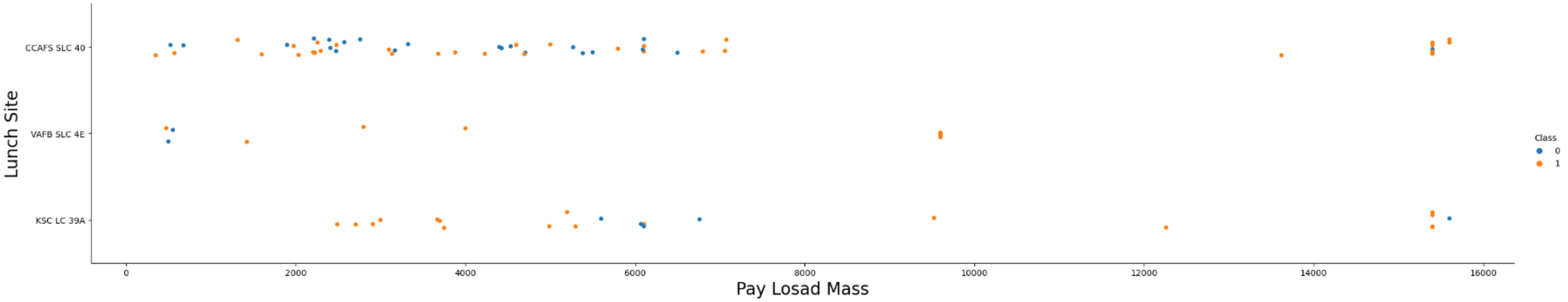
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- The earliest flights in the dataset were unsuccessful, while the latest flights were all successful. This indicates a potential improvement in launch success over time.
- The CCAFS SLC 40 launch site accounts for approximately half of all launches in the dataset, suggesting it is a commonly used site.
- The VAFB SLC 4E and KSC LC 39A launch sites have shown higher success rates compared to other sites. This implies that these sites may have better launch conditions or operational procedures.
- Based on the observed data, it can be inferred that newer launches tend to have a higher rate of success. This suggests that advancements in technology or improved launch processes contribute to increased success rates over time.

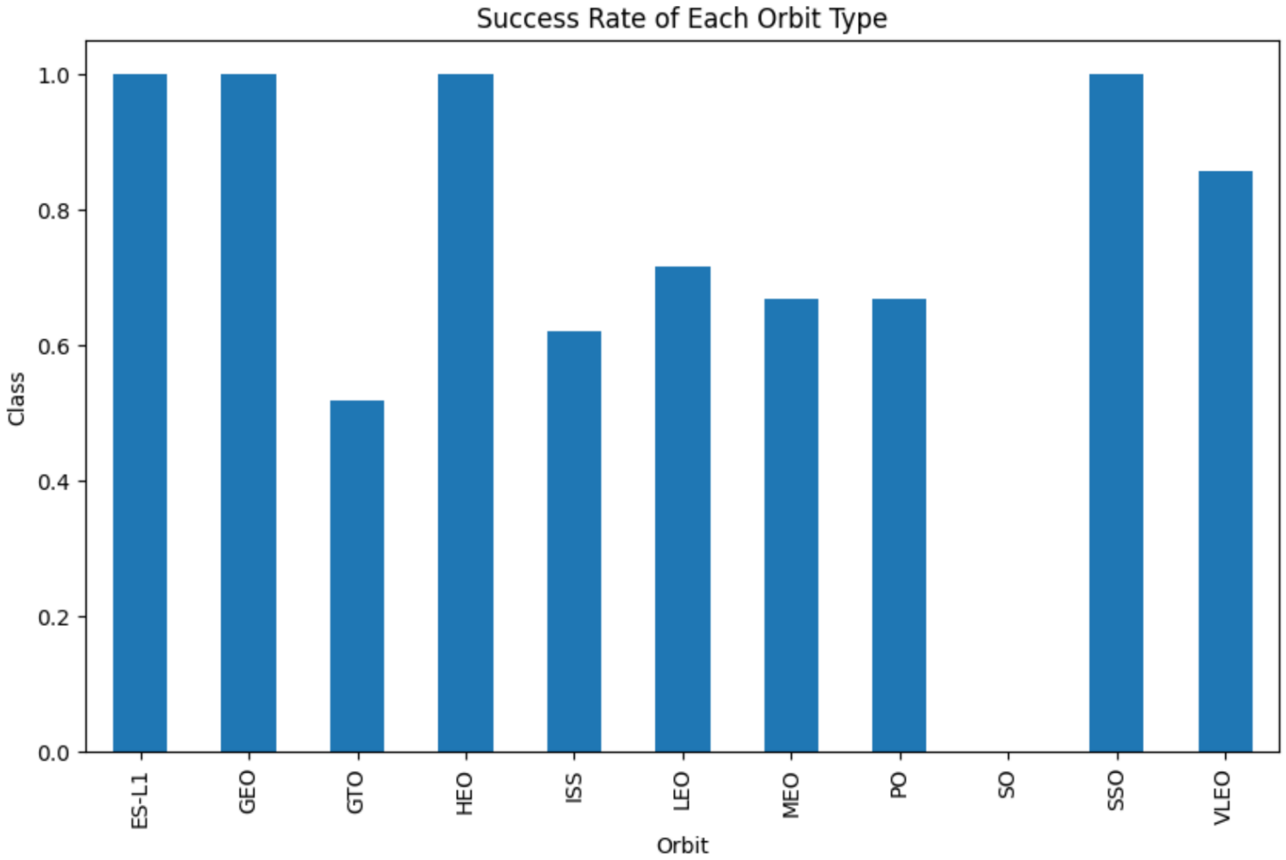


Payload vs. Launch Site

- The data analysis reveals that there is a positive correlation between the payload mass and the success rate for each launch site. In other words, as the payload mass increases, the likelihood of a successful launch also increases.
- It is observed that a majority of launches with a payload mass exceeding 7000 kg have been successful. This suggests that the launch sites have the capability and infrastructure to handle larger payloads effectively.
- Moreover, the specific launch site KSC LC 39A stands out with a 100% success rate for payload masses under 5500 kg as well. This indicates that the site has demonstrated consistent success even for relatively lower payload masses, showcasing its reliability and efficiency.

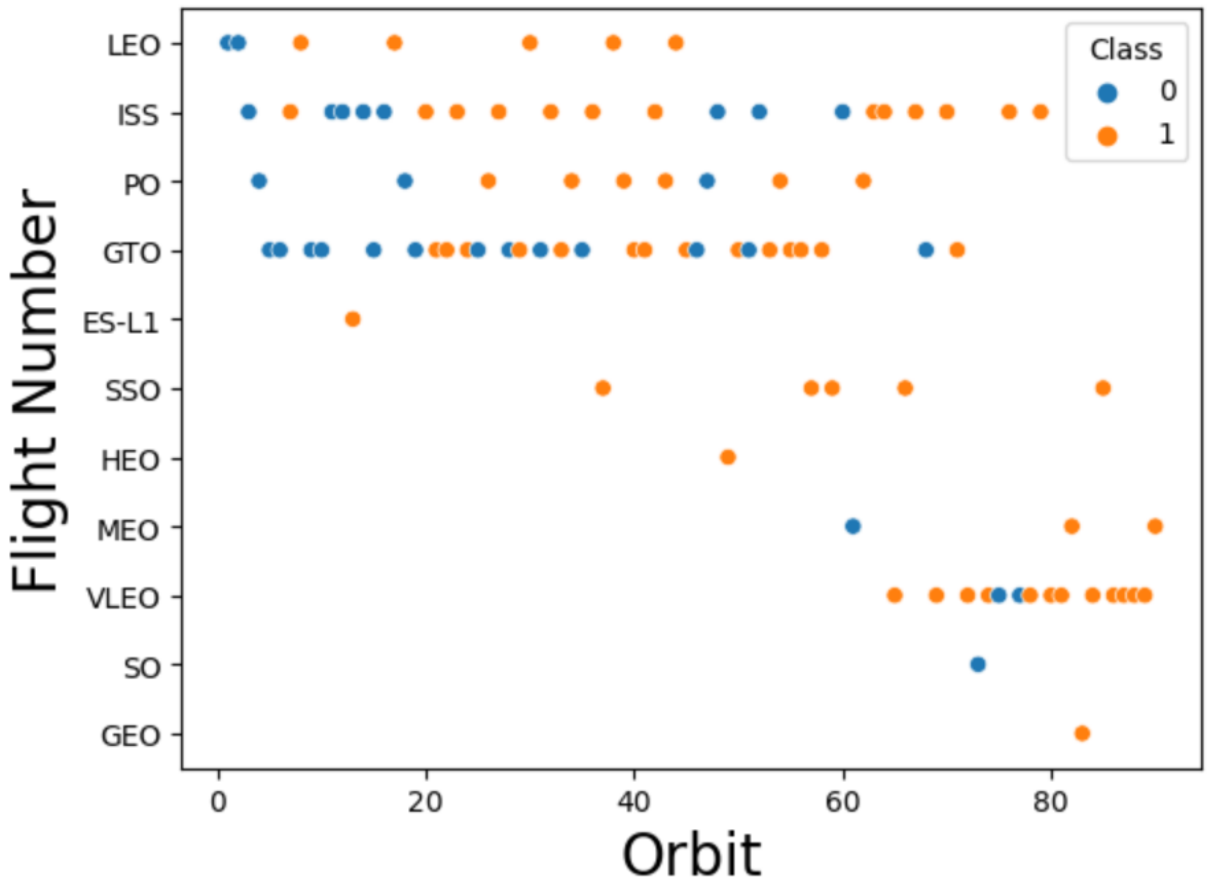
Success Rate vs. Orbit Type

- Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate: - SO
- Orbits with success rate between 50% and 85%: - GTO, ISS, LEO, MEO, PO



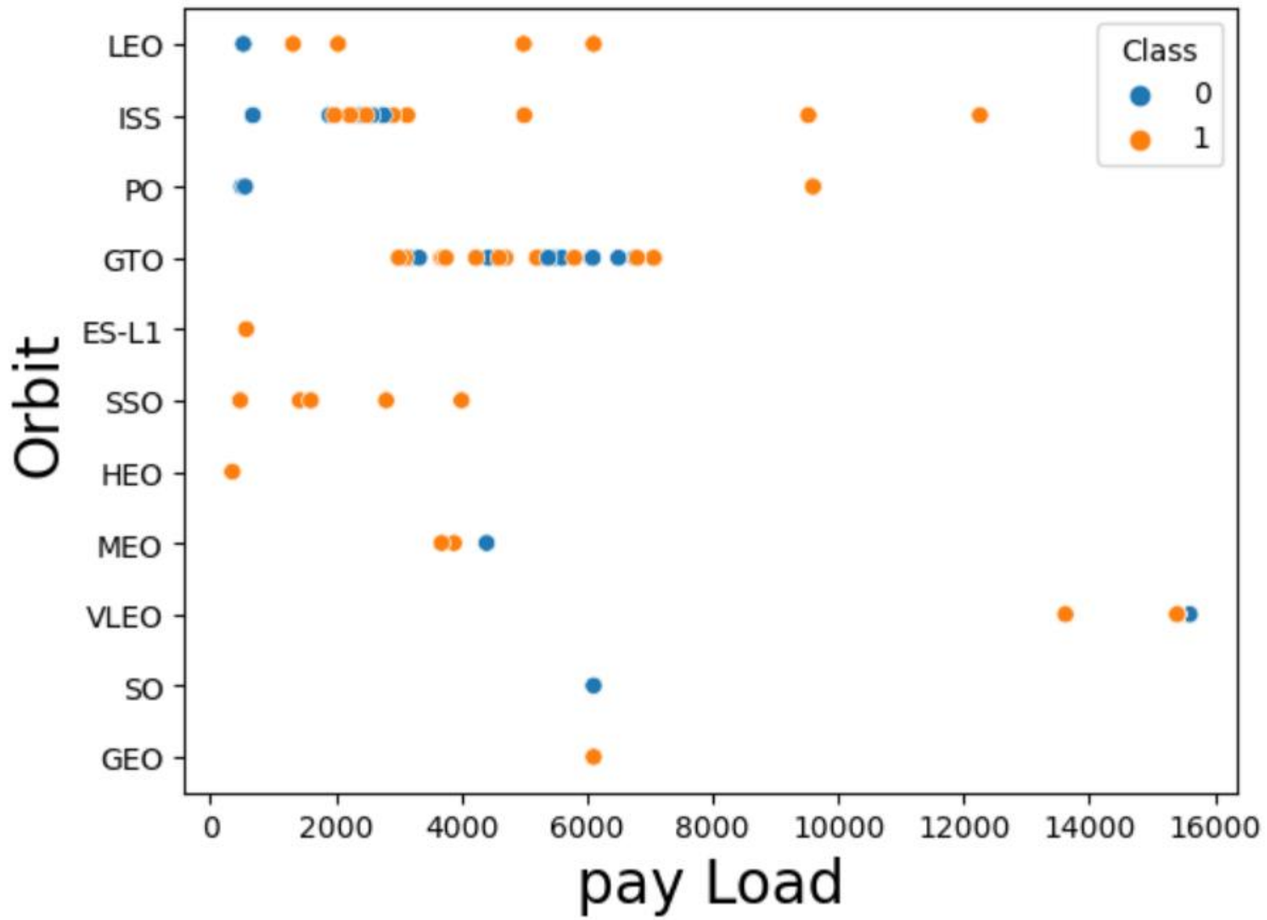
Flight Number vs. Orbit Type

• In LEO orbit, there is a positive correlation between the success rate and the number of flights, indicating improved success with more launches.
• However, in GTO orbit, there is no apparent relationship between the flight number and the success rate, suggesting other factors influence success in this orbit.



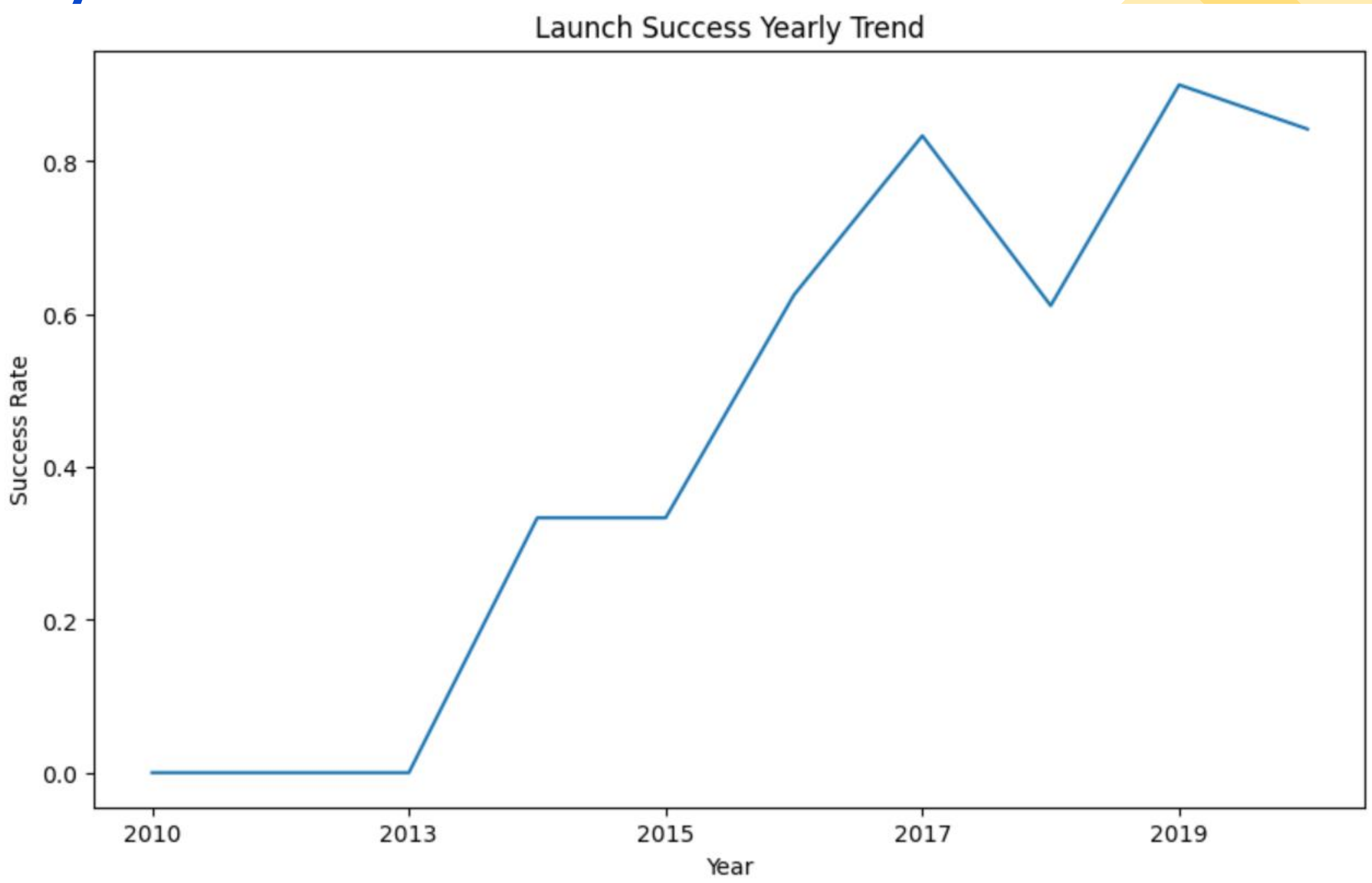
Payload vs. Orbit Type

- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits



Launch Success Yearly Trend

- The success rate since 2013 kept increasir



All Launch Site Names

Displaying the names of the unique launch sites in the space mission

```
]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

]:

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

Launch Site Names Begin with 'CCA'

Displaying 5 records where launch sites begin with the string 'CCA'.

```
%%sql SELECT *From SPACEXTABLE
WHERE Launch_Site  LIKE 'CCA%'
;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Displaying the total payload mass carried by boosters launched by NASA (CRS).

```
%%sql SELECT COUNT(PAYLOAD_MASS__KG_) AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

Total_Payload_Mass
20

Total Payload Mass

Displaying average payload mass carried by booster version F9 v1.1

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
```

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

Average_Payload_Mass
2928.4

Average Payload Mass by F9 v1.1

First Successful Ground Landing Date

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

```
%%sql SELECT MIN(Date) AS First_Successful_Landing_Date
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';
```

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

First_Successful_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listing the names of the boosters which have success in drone ship and have 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_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000;
```

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

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

Listing the total number of successful and failure mission outcomes.

```
%%sql SELECT Mission_Outcome, COUNT(*) AS Total_Count
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

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

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

Total Number of Successful and Failure Mission Outcomes

Boosters Carried Maximum Payload

Listing the names of the booster versions which have carried the maximum payload mass

```
%%sql SELECT DISTINCT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
);
```

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

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

- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015

```
%%sql SELECT strftime('%Y-%m', Date) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE strftime('%Y', Date) = '2015'
    AND Landing_Outcome = 'Failure (drone ship)';
```

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

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

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

- Ranking 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(*) AS Outcome_Count
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC;
```

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

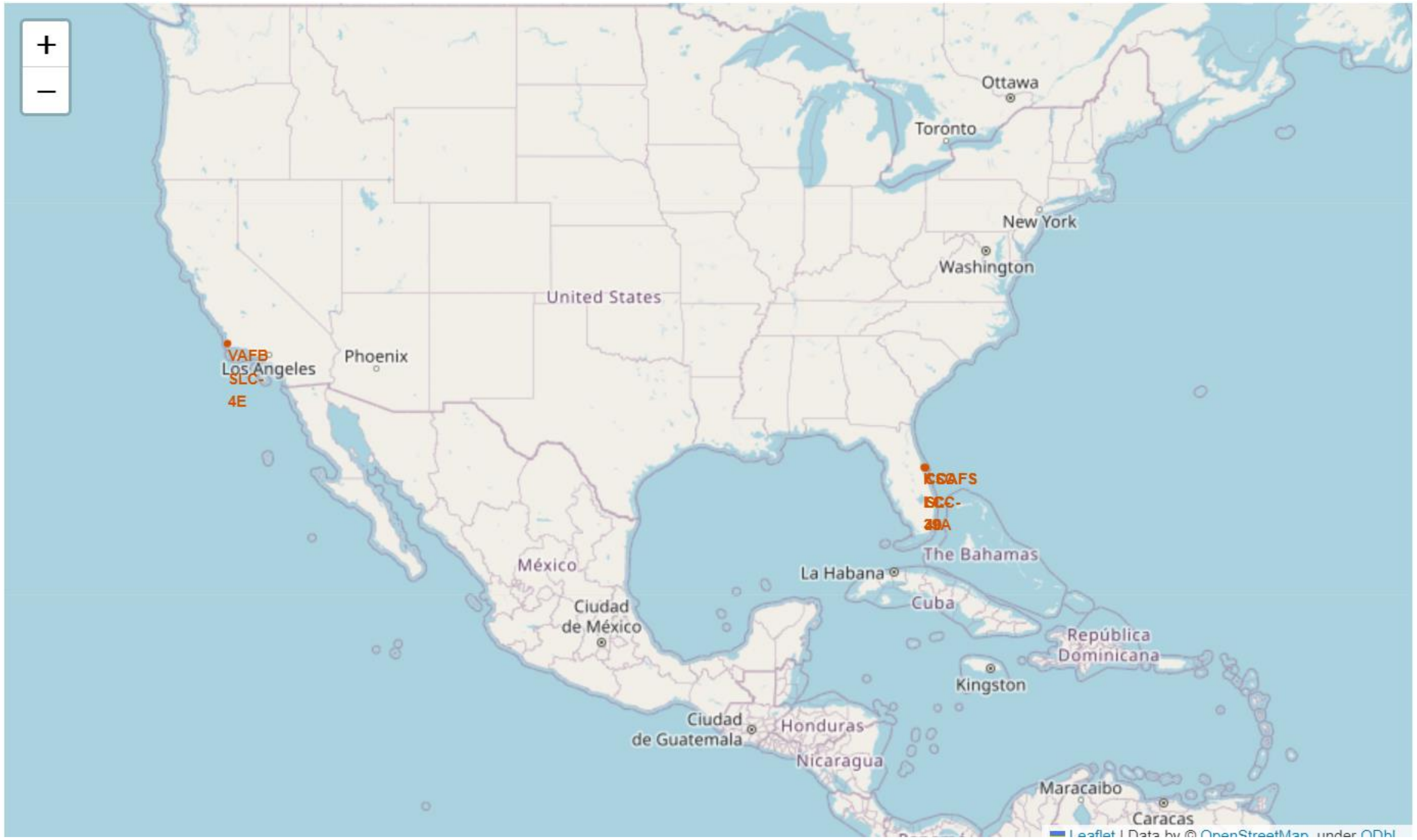
Landing_Outcome	Outcome_Count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

All launch sites location markers on a global map

All launch sites are situated near coastlines. Launching rockets towards the ocean reduces the risk of debris falling or exploding near populated areas, enhancing safety measures during launches.

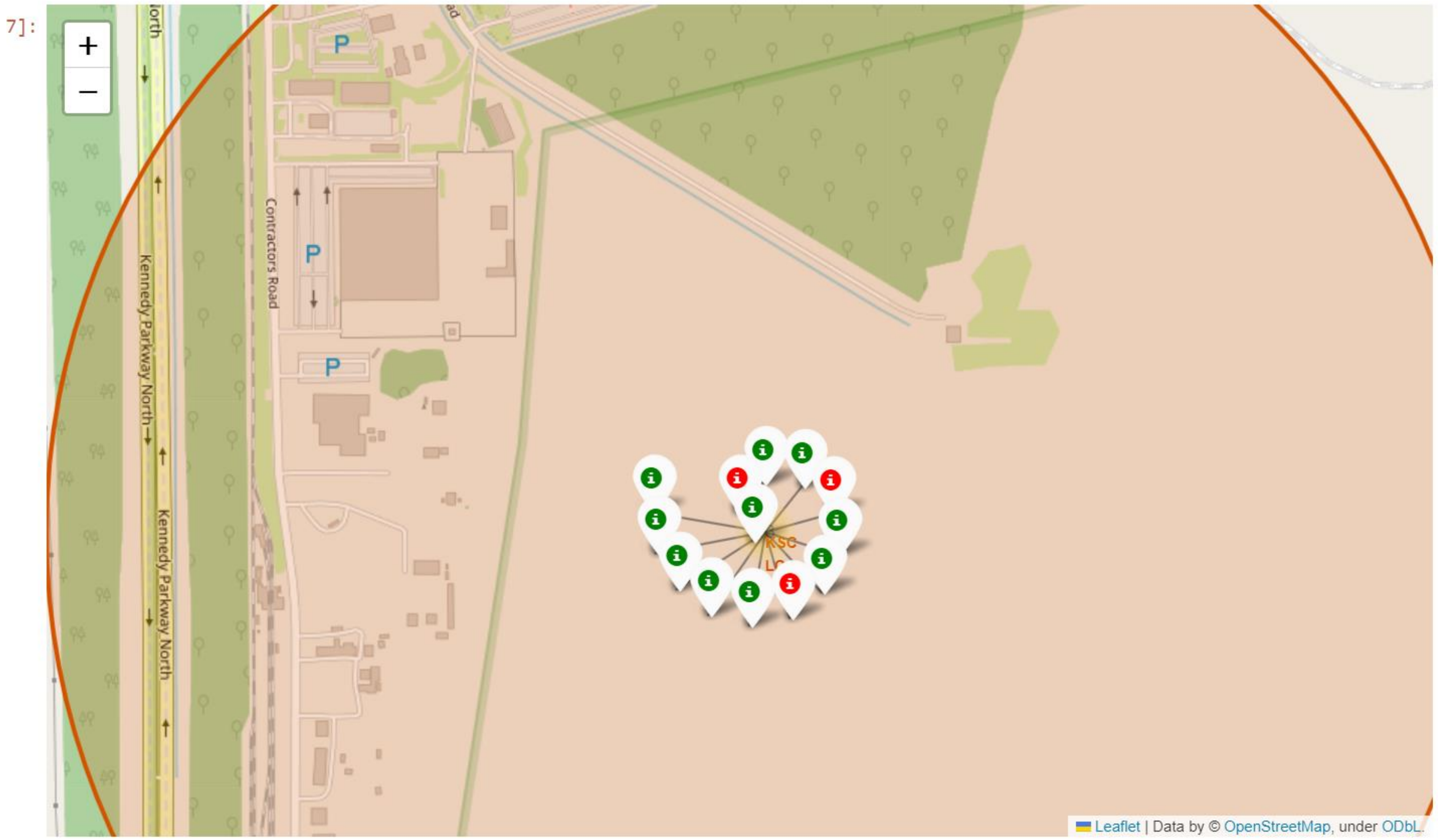


Colour-labeled launch records on the map

- From the colour-labeled markers we should be able to easily identify which launch sites have relatively high success rates.

- Green Marker = Successful Launch - Red Marker = Failed Launch

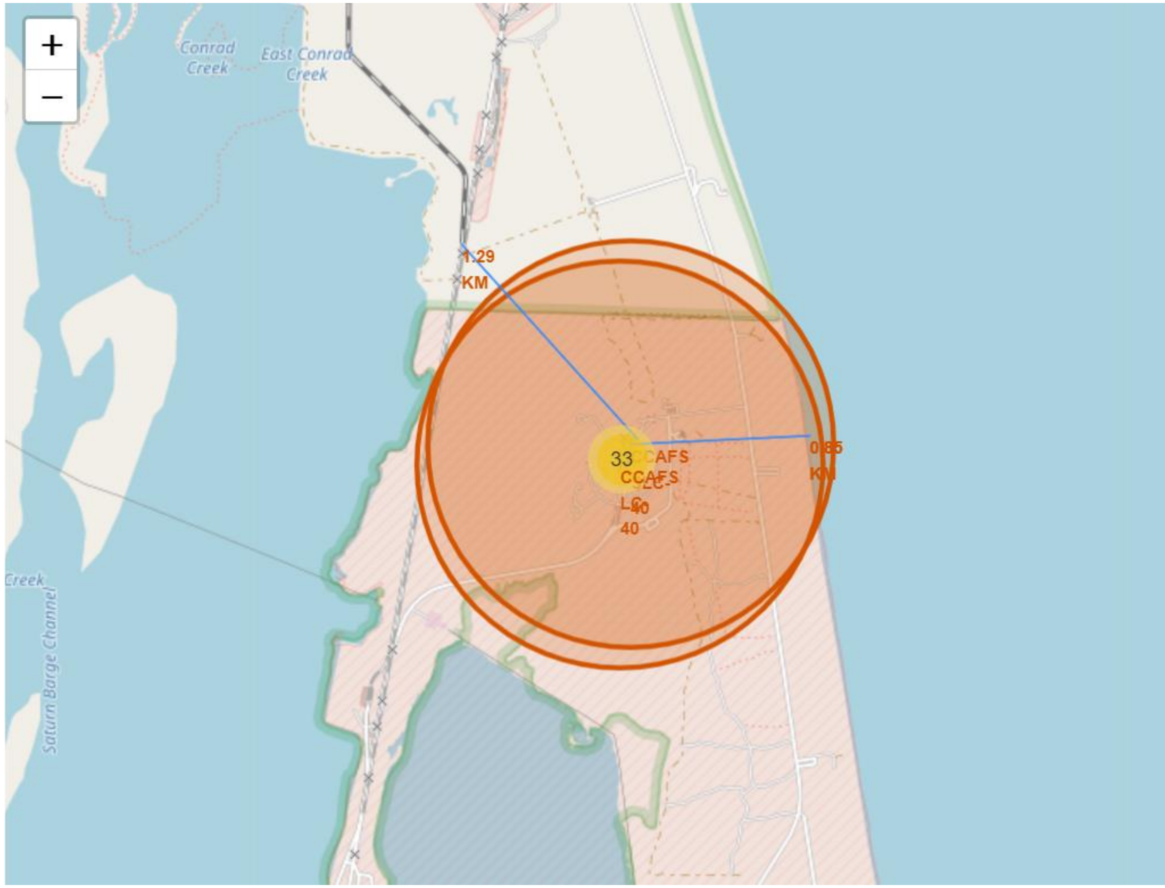
- Launch Site KSC LC-39A has a very high Success Rate



Distance from the launch site CCAFS SLC-40 to its proximities

• Upon visual analysis of the launch site CCAFS SLC-40 , it is evident that it is in close proximity to various features:

- It is relatively close to a railway, approximately 1.29 km away.
- It is relatively close to the coastline, approximately 0.85km away.

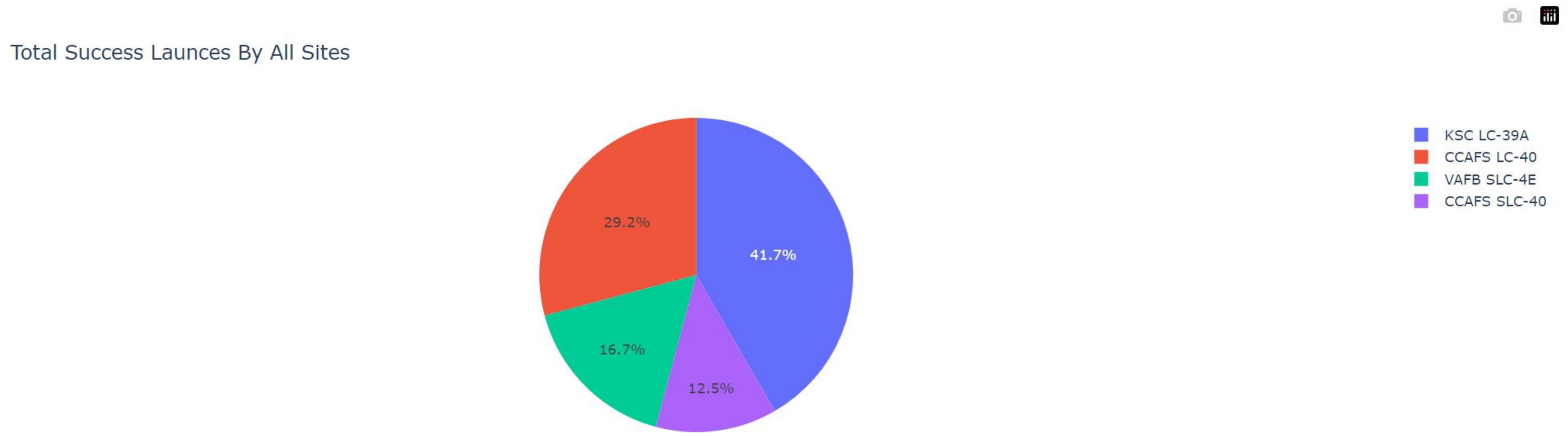


Section 4

Build a Dashboard with Plotly Dash

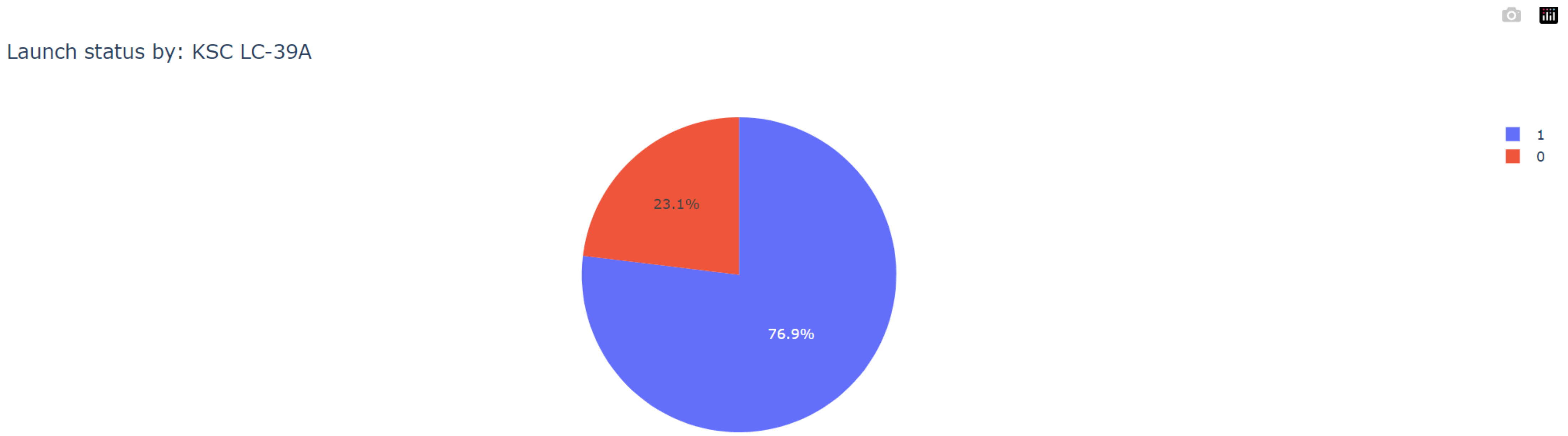
Launch success count for all sites

The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.



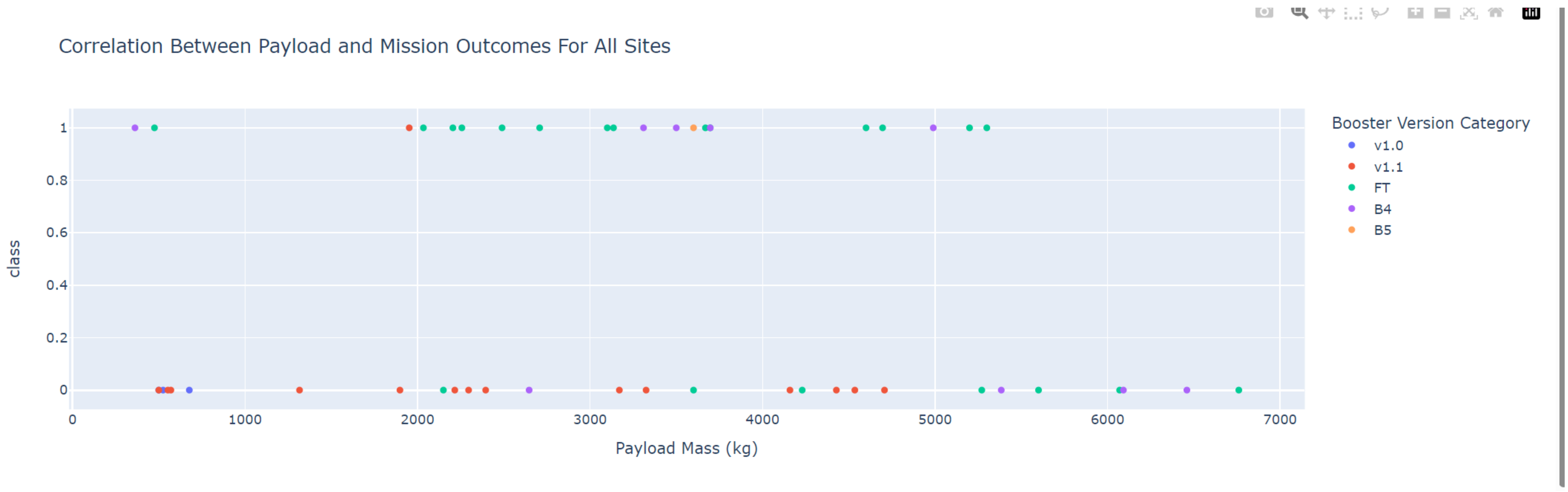
Launch site with highest launch success ratio

KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings



Payload Mass vs. Launch Outcome for all sites

The charts show that payloads between 2000 and 5500 kg have the highest success rate.

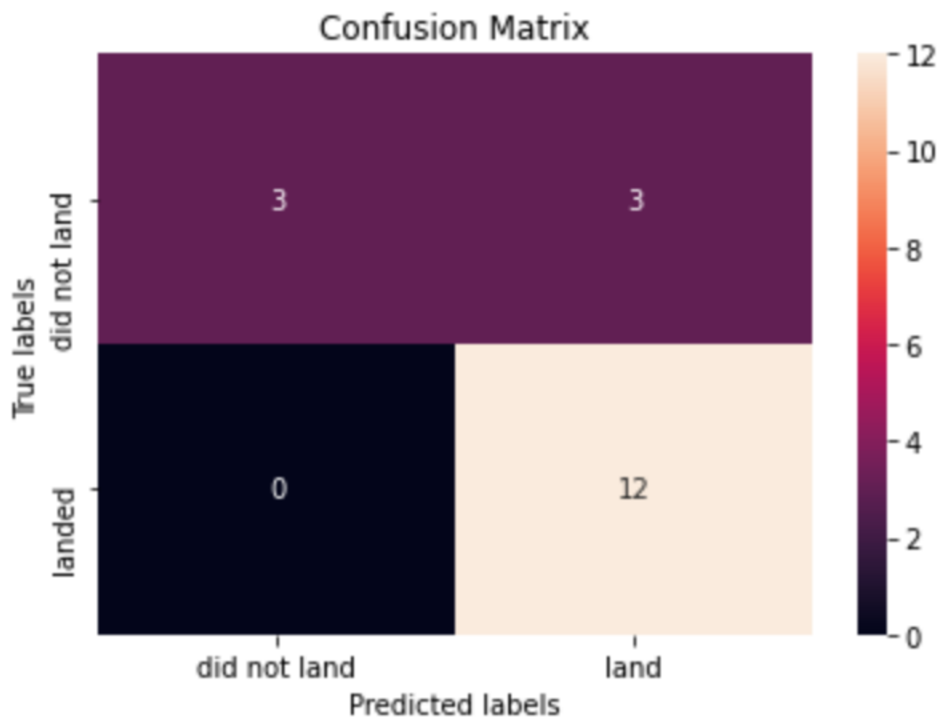


Section 5

Predictive Analysis (Classification)

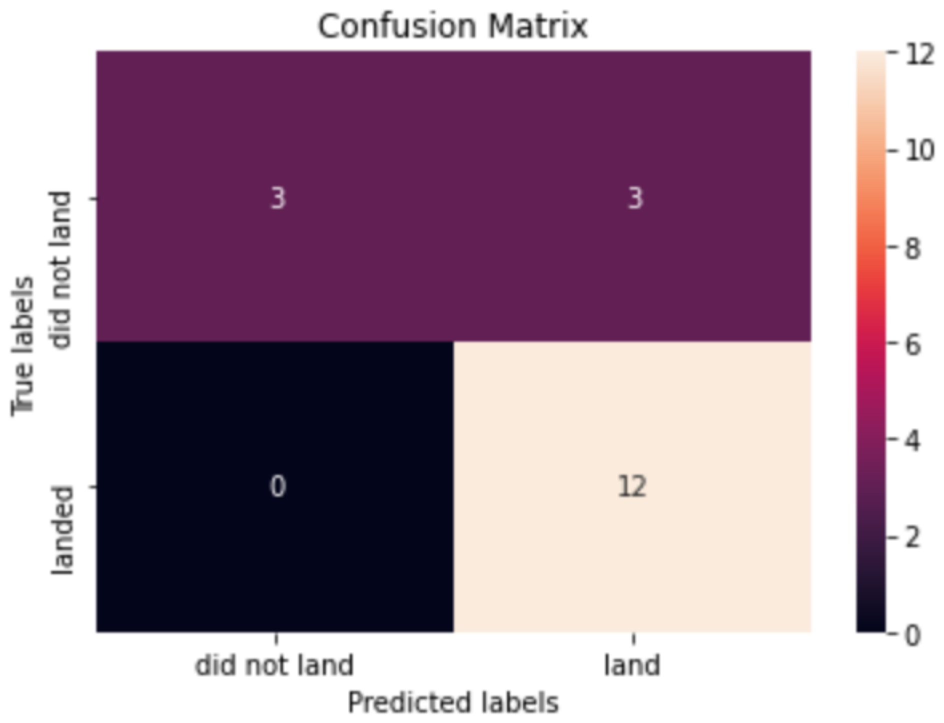
Predictive Analysis

- Logistic regression
  - GridSearchCV best score: 0.8464285714285713
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



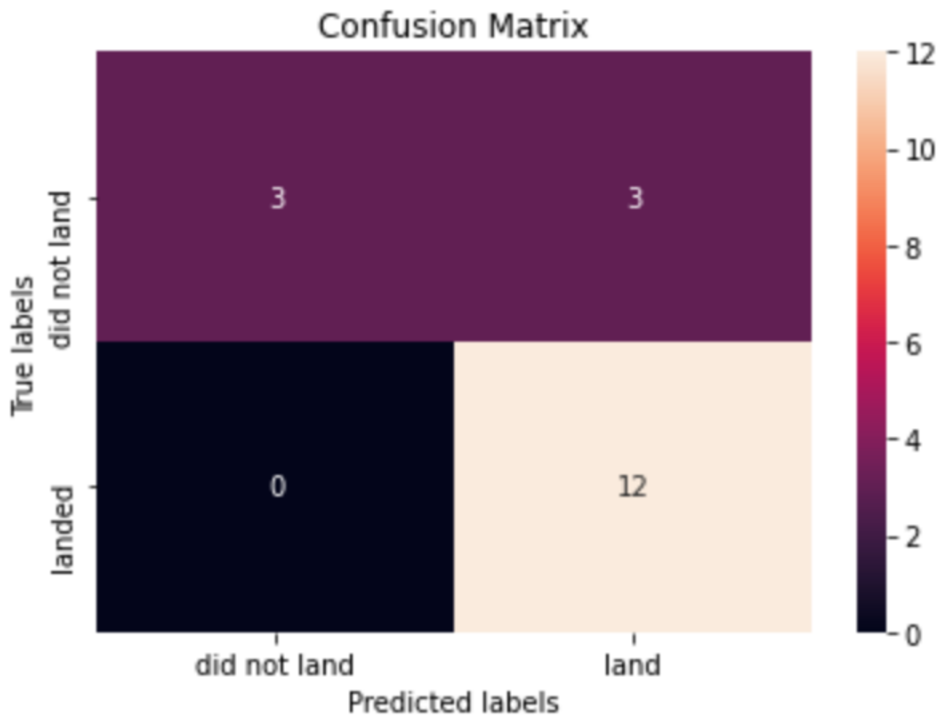
Predictive Analysis

- Support vector machine (SVM)
  - GridSearchCV best score: 0.8482142857142856
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



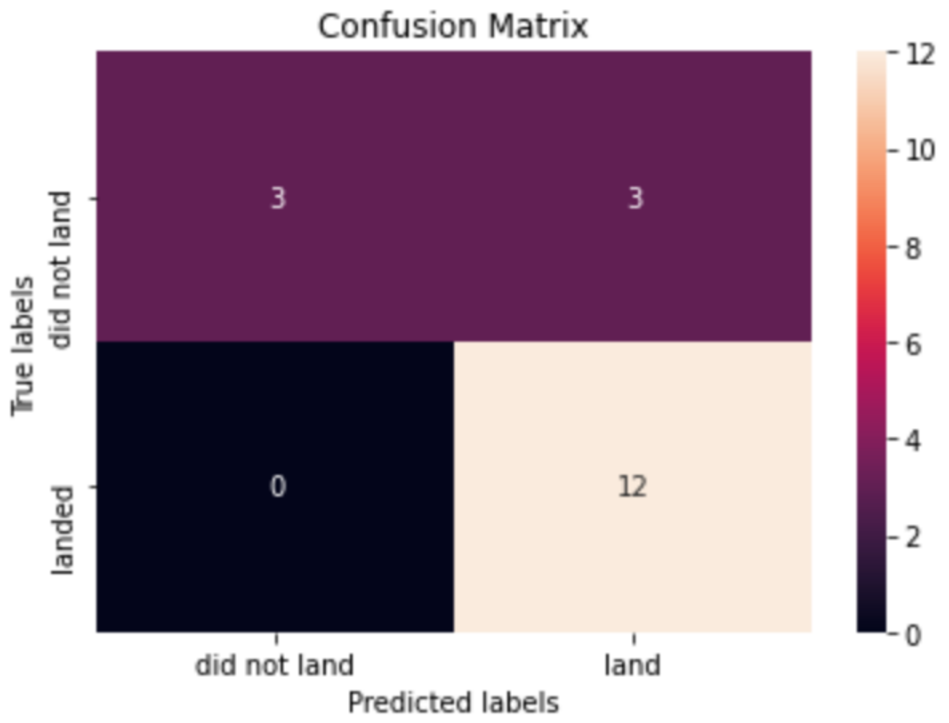
Predictive Analysis

- Decision tree
  - GridSearchCV best score:0.875
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



Predictive Analysis

- K nearest neighbors (KNN)
  - GridSearchCV best score: 0.8482142857142858
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



Predictive Analysis

- Comparing the results of all four models, it is evident that they have identical accuracy scores and confusion matrices when evaluated on the test set. Therefore, their rankings are determined based on their GridSearchCV best scores. The models are ranked as follows, with the first being the best and the last being the worst:
- Decision tree (GridSearchCV best score: 0.8892857142857142)
- K nearest neighbors, KNN (GridSearchCV best score: 0.8482142857142858)
- Support vector machine, SVM (GridSearchCV best score: 0.8482142857142856)
- Logistic regression (GridSearchCV best score: 0.8464285714285713)

Conclusions

- In this project, we try to predict if the first stage of a given Falcon 9 launch will land in order to determine the cost of a launch
- Launches with a lower payload mass tend to have higher success rates compared to launches with larger payload masses.
- The majority of launch sites are located near the Equator line, and all launch sites are situated in close proximity to coastlines.
- The success rate of launches has shown an upward trend over the years, indicating improvements in launch capabilities and processes.
- Among all the launch sites, KSC LC-39A stands out with the highest success rate for its launches.
- Launches into ES-L1, GEO, HEO, and SSO orbits have achieved a 100% success rate, indicating a high level of reliability and precision in reaching these specific orbits.

Appendix

- https://github.com/Batool1227/Apllied-DataScince.git

Thank you!