Winning Space Race with Data Science

Pablo Rodríguez Cordovés

12nd February 2024

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- **Summary of all results**
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- **Project background and context**

  Space X showcases its Falcon 9 spaceflights online, setting the price tag at a competitive 62 million dollars, in stark contrast to the 165 million dollars or more charged by other launch service providers. A significant portion of this price advantage stems from Space X's innovative ability to reclaim and repurpose the initial segment of the rocket. If we are capable of predicting the successful retrieval of this segment, we can accurately project the financial outlay required for each spaceflight. This predictive insight would be invaluable for any competing entities considering entering a bidding war with Space X for orbital launch contracts. The end goal of this venture is to engineer a sophisticated machine learning system designed to ascertain the likelihood of the first stage making a successful touchdown.

- **Problems you want to find answers**
  - What variables contribute to the successful touchdown of a rocket?
  - It is the confluence of diverse aspects that orchestrates the achievement of a successful landing.
  - What prerequisites in operational parameters are essential for securing the success of a landing endeavor?

Section 1

Methodology

Methodology

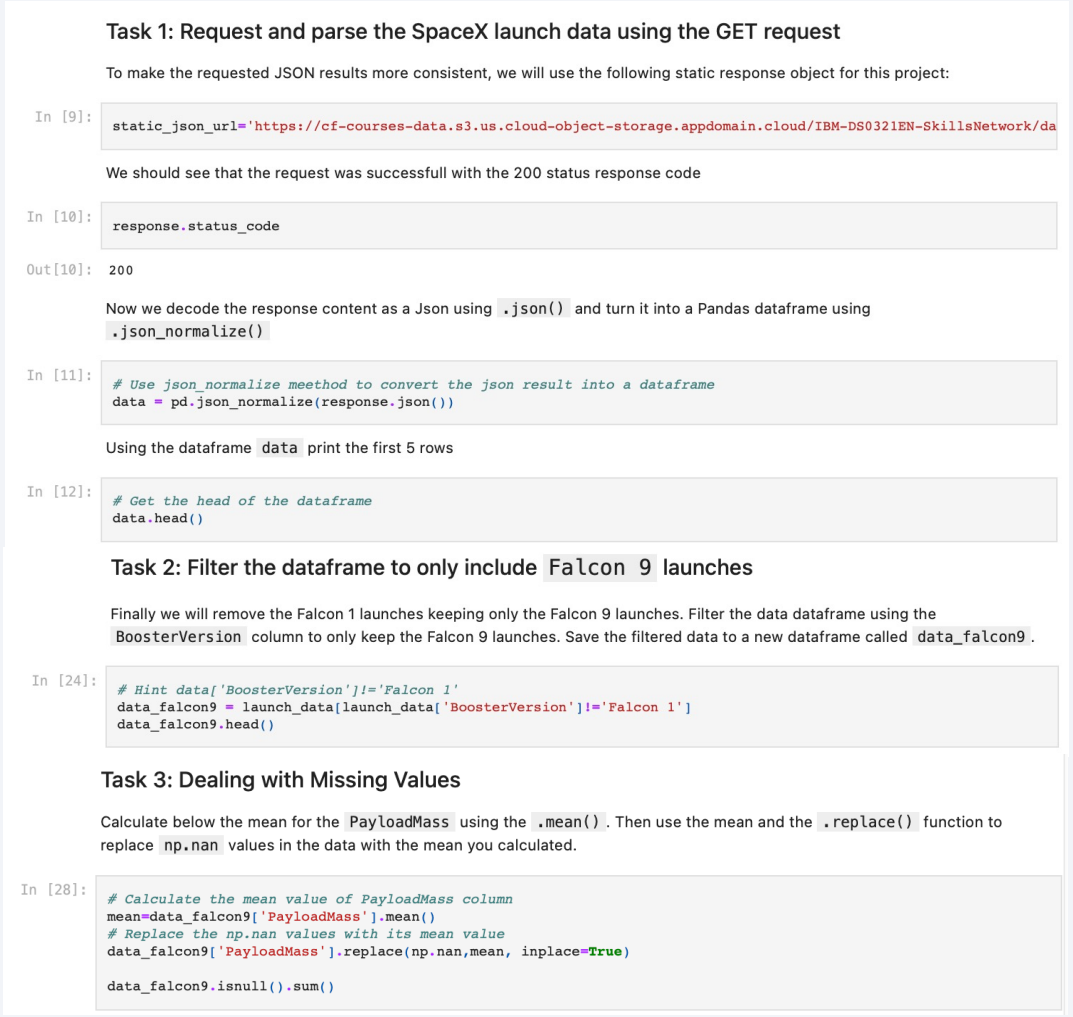
Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- **We gathered information through a range of techniques.**
  - To retrieve the data, we executed GET requests to the SpaceX API.
  - Following this, we transformed the response content into JSON format by invoking the .json() function and then cast it into a Pandas dataframe using .json_normalize().
  - Subsequently, we undertook the cleansing of the dataset, scrutinized it for any absent values, and supplied the missing information where it was deemed necessary.
  - Furthermore, we employed data scraping techniques on Wikipedia to obtain Falcon 9 launch data, utilizing BeautifulSoup.
  - Our aim was to capture the launch details in the form of an HTML table, to parse said table, and to transcribe it into a Pandas dataframe to facilitate subsequent analysis.

Data Collection - SpaceX API



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/da
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response.status_code
```

Out[10]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [12]:
```
# Get the head of the dataframe
data.head()
```

Task 2: Filter the dataframe to only include `Falcon 9` launches

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

In [24]:
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

Task 3: Dealing with Missing Values

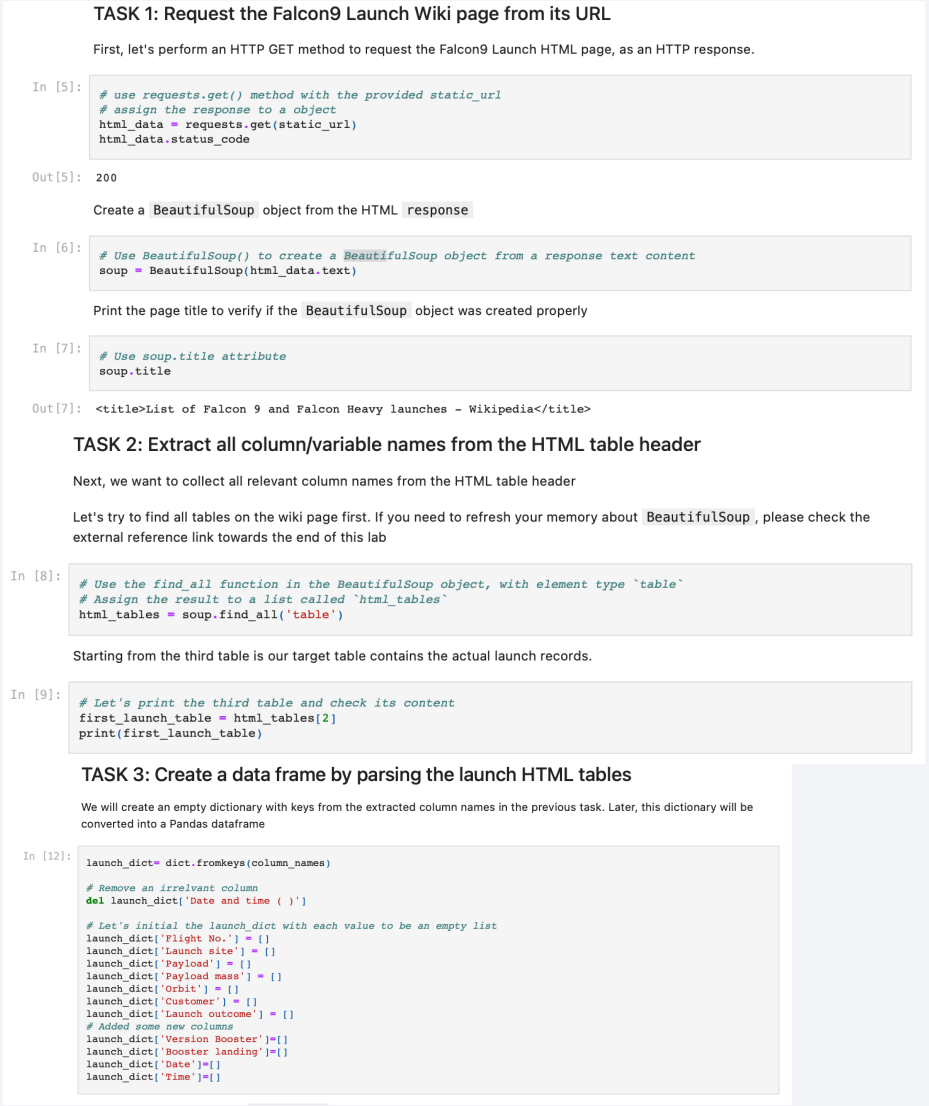
Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

In [28]:
```
# Calculate the mean value of PayloadMass column
mean=data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,mean, inplace=True)

data_falcon9.isnull().sum()
```

- Data was acquired by initiating GET requests to the SpaceX API, followed by the purification and preliminary organization and structuring of the collected information.
- The notebook containing these processes is available at the following URL: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/01_Data_Collection.ipynb

Data Collection -Scraping

- Web scraping techniques were employed to extract Falcon 9 launch records using BeautifulSoup.
- The extracted table was then parsed and formatted into a Pandas dataframe.
- For detailed procedures, refer to the notebook available at: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/02_Web_Scraping.ipynb



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.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]: 200

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text)
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</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

In [8]:
```
# 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')
```

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

In [9]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

TASK 3: Create a data frame by parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

In [12]:
```
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']=[]
```

Data Wrangling

TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

In [7]:
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [8]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [9]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

TASK 4: Create a landing outcome label from Outcome column

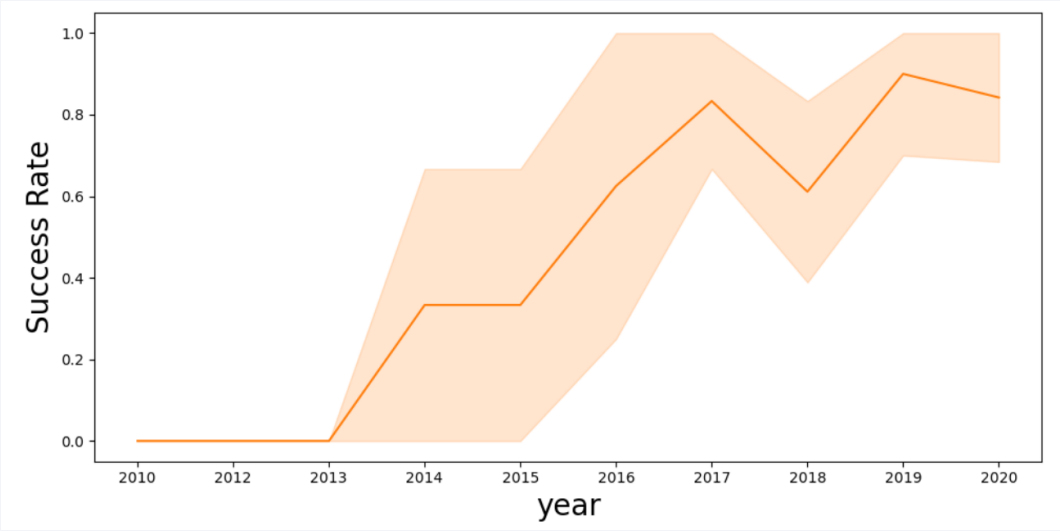
Using the `Outcome` , create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome` ; otherwise, it's one. Then assign it to the variable `landing_class` :

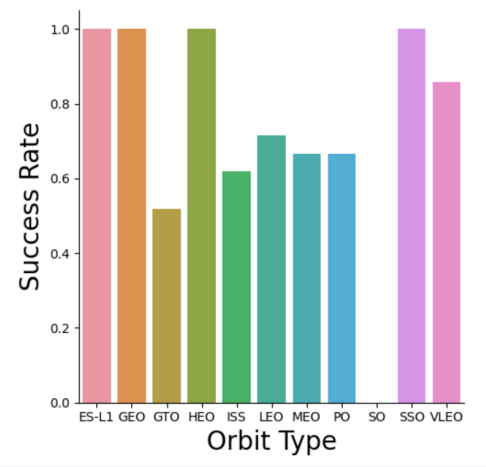
In [12]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

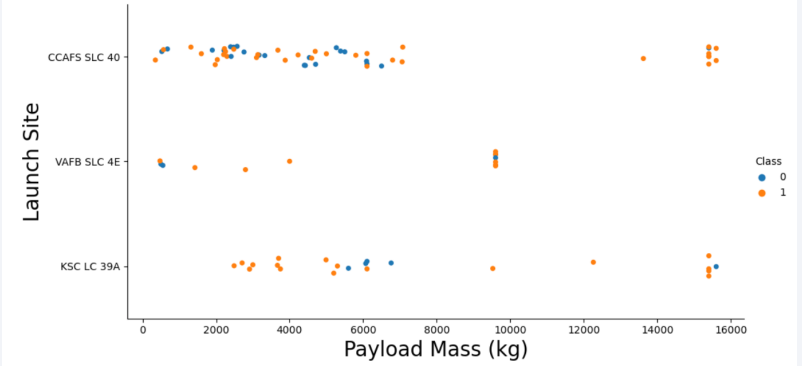
- Exploratory data analysis was conducted to establish the training labels.
- We assessed the frequency of launches per site and quantified the variety and frequency of orbits.
- A label for landing outcomes was generated from the 'outcome' column and the findings were saved in CSV format.
- The processes are documented in the notebook accessible at: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/03_Data_Wrangling.ipynb

EDA with Data Visualization

- Data exploration involved visual analysis of correlations among flight number and launch site, payload versus launch site, success rates across different orbit types, relationships between flight number and orbit type, and annual trends in launch success.
- For a detailed exploration, consult the notebook at: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/05_Data_Visualization.ipynb







EDA with SQL

- The SpaceX dataset was imported into a PostgreSQL database directly through a Jupyter notebook interface. Through SQL-based exploratory data analysis, we derived insights from the dataset. Our queries included investigations such as:
  - Listing all distinct launch site names involved in the space missions.
  - Calculating the cumulative payload mass deployed by NASA (CRS) missions.
  - Computing the mean payload mass delivered by the F9 v1.1 booster model.
  - Enumerating the total counts of missions with successful and unsuccessful outcomes.
  - Identifying unsuccessful landing attempts on drone ships, including specific booster versions and the names of the launch sites involved.
- For a comprehensive guide on these processes, refer to the notebook located at: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/04_SQL.ipynb

EDA with SQL

Task 1

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

In [8]:
```
%sql ibm_db_sa://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.ap
%sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;
```

Task 2

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

In [9]:
```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

Task 3

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

In [10]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
```

Task 4

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

In [11]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [12]:
```
%sql SELECT MIN(DATE) \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [13]:
```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Task 7

List the total number of successful and failure mission outcomes

In [14]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [15]:
```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [38]:
```
%sql SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
FROM SPACEXTBL \
where [Landing _Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [37]:
```
%sql SELECT [Landing _Outcome], count(*) as count_outcomes \
FROM SPACEXTBL \
WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;
```

Build an Interactive Map with Folium

- We highlighted all rocket launch locations on a folium map, incorporating various map elements such as markers, circles, and lines to denote the outcomes of launches (successful or unsuccessful) at each site.
- Launch results were categorized into two classes: class 0 for failures and class 1 for successes.
- By utilizing markers with distinct colors for each category, we were able to determine which launch locations boasted higher rates of success.
- We also measured the distances from each launch site to nearby features, tackling questions like:
  - The proximity of launch sites to railways, highways, and coastlines.
  - The adherence of launch sites to maintain a specific minimum distance from populated areas.
- For a closer look at these visualizations, access the project through the notebook at https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/06_Interactive_Visual_Analytics.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was developed using Plotly Dash, featuring various visual representations.
- Pie charts were created to illustrate the distribution of total rocket launches from specific locations.
- Additionally, scatter plots were utilized to explore the correlation between launch outcomes and payload weights (in kilograms) across various rocket models.
- For a closer look at these visualizations, access the project through the notebook at https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/07_Interactive_Visual_Analytics_Plotly.py

Predictive Analysis (Classification)

- Data was ingested and processed utilizing numpy and pandas, after which it was divided into training and testing subsets.
- We developed a variety of machine learning algorithms and adjusted their hyperparameters with the assistance of GridSearchCV.
- Model performance was gauged using accuracy as the key indicator, and further enhancements were achieved through feature engineering and fine-tuning of algorithms.
- This process led to the identification of the most effective classification algorithm.
- For detailed methodologies and insights, refer to the notebook at: https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/08_Predictive_Analytics.ipynb

Results

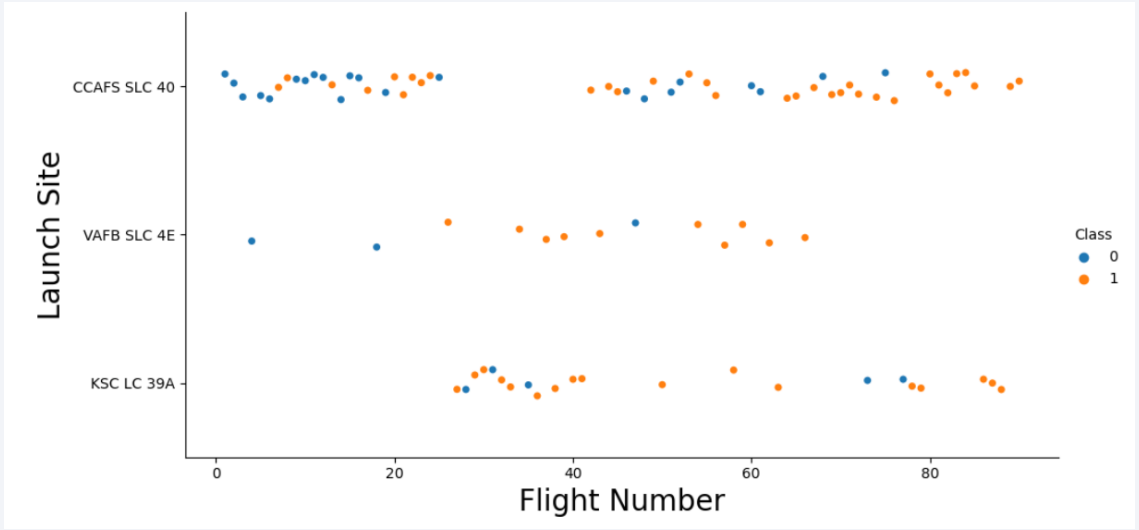
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

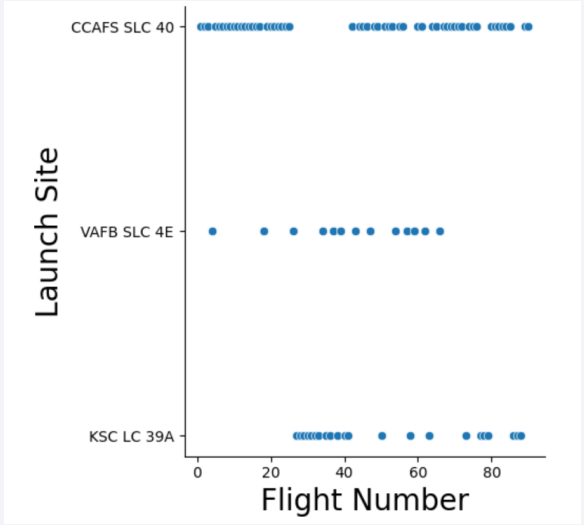
Section 2

Insights drawn from EDA

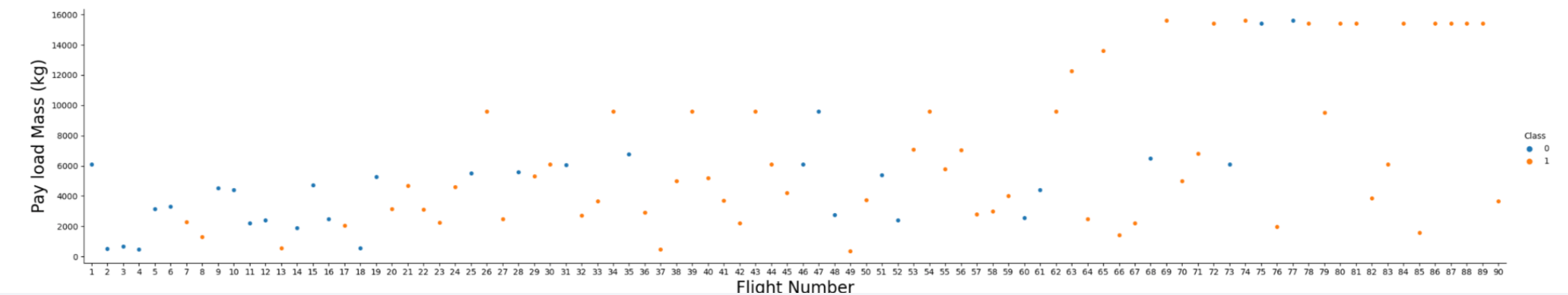
Flight Number vs. Launch Site

→ The analysis of the graphical data revealed a direct correlation between the volume of flights conducted at a particular launch facility and the likelihood of mission success at that site. This observation suggests that increased operational activity at a launch site is associated with a higher probability of successful outcomes, indicating that experience and frequency of launches could play a significant role in enhancing the effectiveness of space missions.





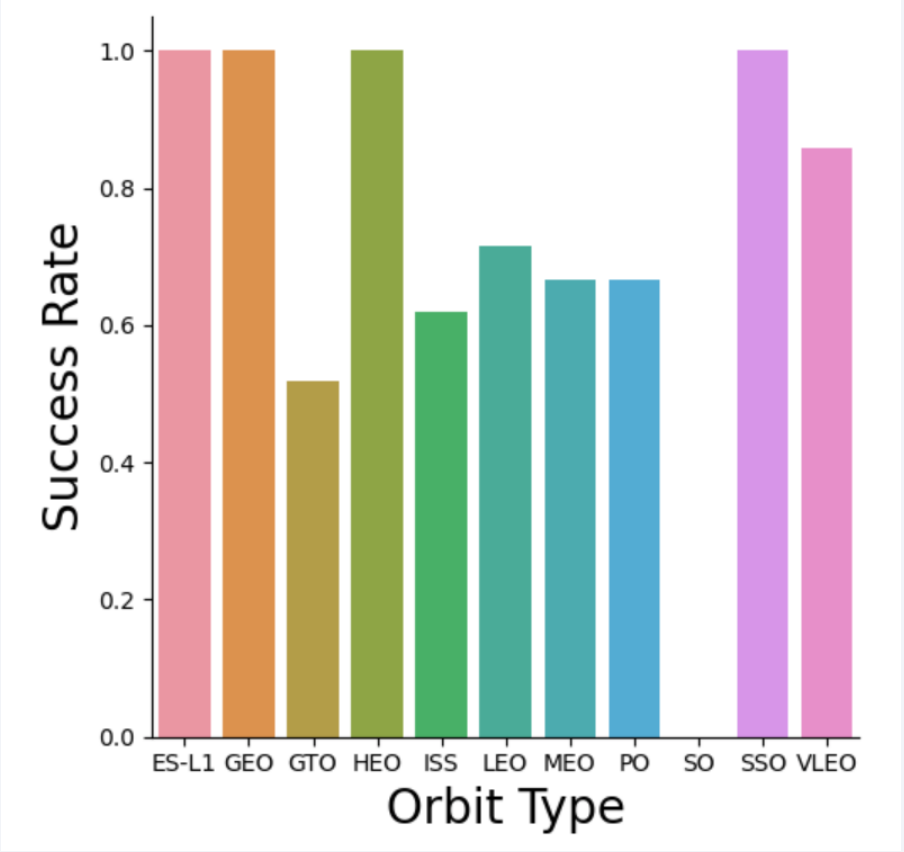
Payload vs. Launch Site



→ The analysis indicates that for launches from the CCAFS SLC 40 site, there is a positive correlation between the mass of the payload and the rocket's success rate. This finding suggests that the ability to handle heavier payloads effectively at this launch site is associated with an increased likelihood of mission success. It implies that the infrastructure, operational expertise, and technological capabilities at CCAFS SLC 40 are particularly adept at managing significant payload weights, which in turn, may contribute to the overall reliability and success of the launches conducted from this location.

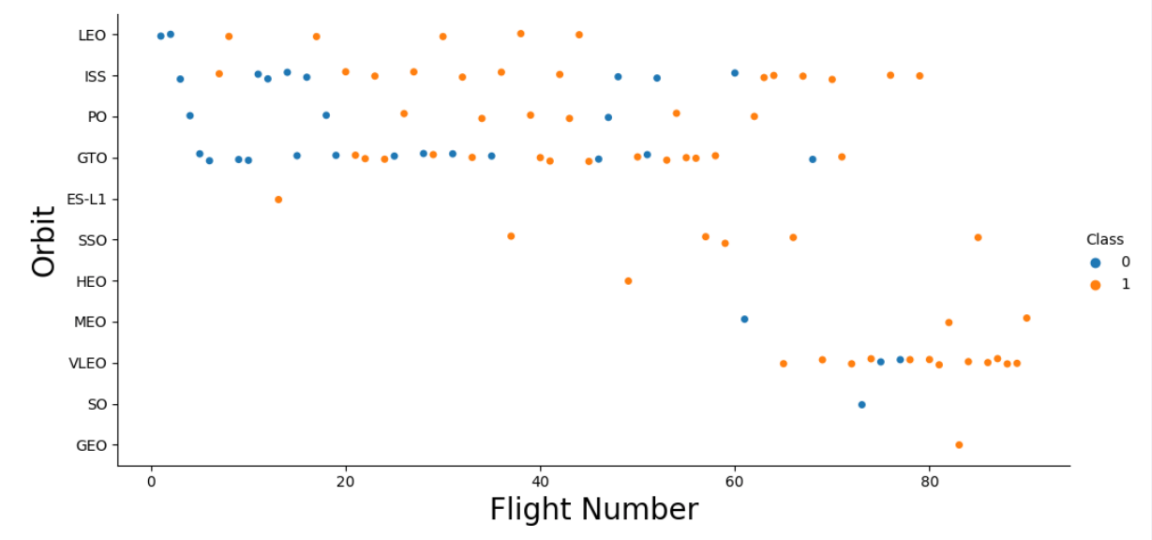
Success Rate vs. Orbit Type

→ The graphical representation reveals that launch outcomes at ES-L1, GEO, HEO, SSO, and VLEO orbit classifications exhibit the highest rates of success. This observation underscores the proficiency and effectiveness in conducting missions to these specific orbital destinations. It suggests that the technological and operational strategies implemented for launches targeting these orbits are highly optimized, leading to a consistently higher probability of successful mission completion. The success in these orbits may reflect advanced engineering solutions, precise navigational accuracy, and robust mission planning that collectively enhance the success rates for missions aimed at these orbital regions.



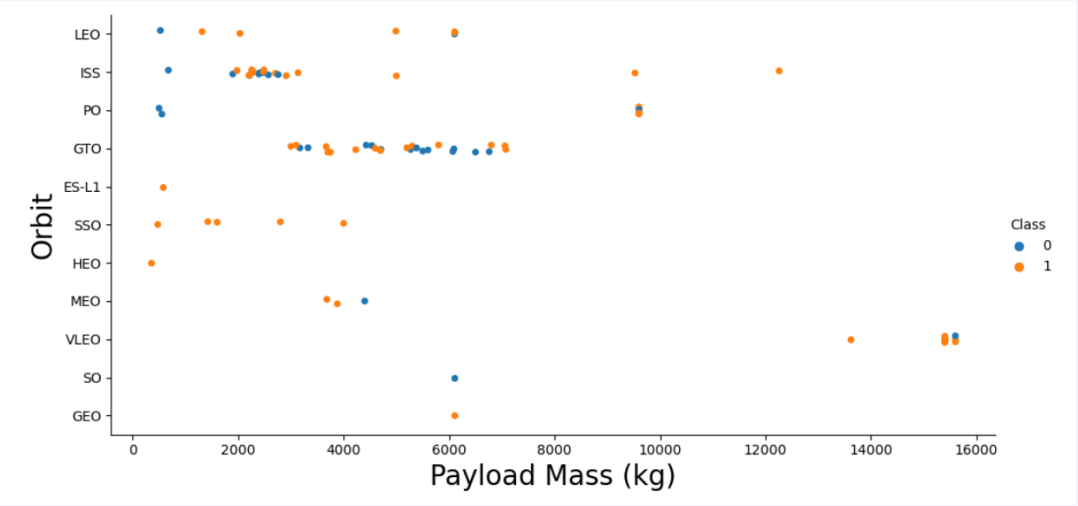
Flight Number vs. Orbit Type

→ The graph presented illustrates a comparison between flight numbers and their respective orbit types, revealing distinct patterns of success across different orbits. In the case of the Low Earth Orbit (LEO), there's a discernible correlation indicating that the success rate tends to increase with the number of flights conducted. This suggests that accumulated experience and repetitive launches in LEO contribute to refining operational procedures, thereby enhancing mission success. Conversely, for the Geostationary Transfer Orbit (GTO), the data does not display any significant link between the quantity of flights and success outcomes. This lack of correlation in the GTO suggests that success in these missions is influenced by factors beyond just the frequency of launches, possibly including the complexity of reaching such orbits, specific mission requirements, or technological challenges. The analysis highlights how success in space missions is variably influenced by orbit type, with some benefiting from operational experience while others depend more heavily on other variables.



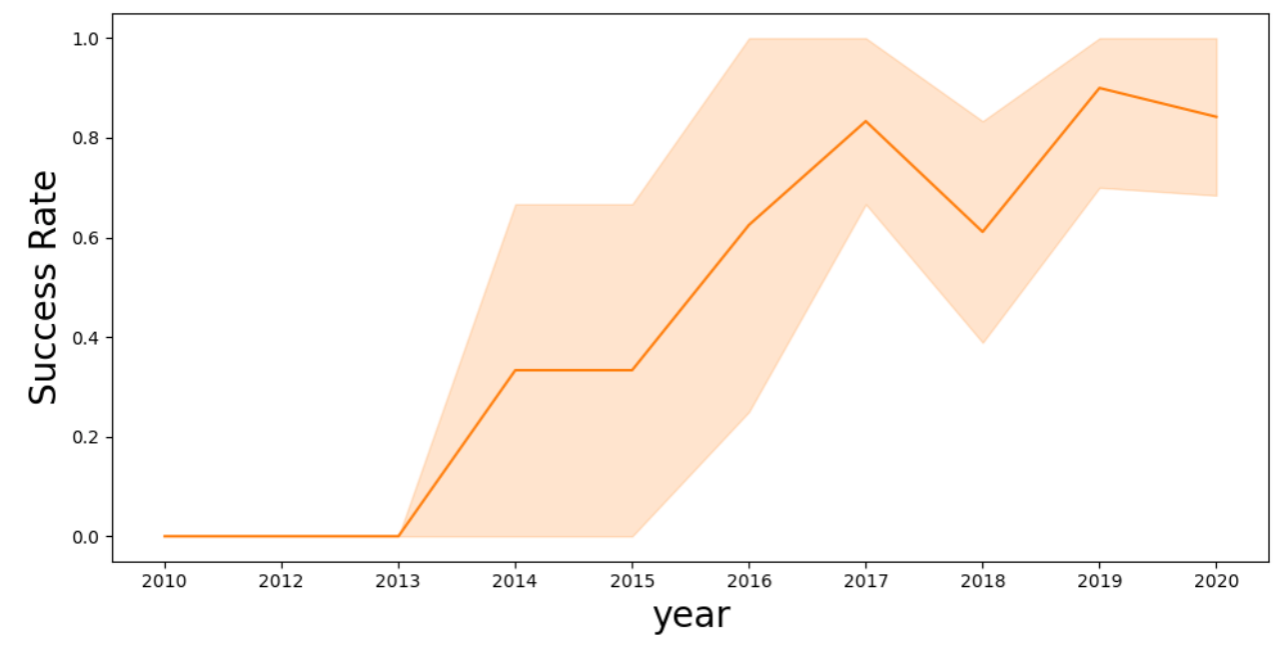
Payload vs. Orbit Type

→ The data analysis indicates that for heavier payloads, the probability of achieving a successful landing significantly increases for missions destined for Polar Orbit (PO), Low Earth Orbit (LEO), and the International Space Station (ISS) orbits. This pattern suggests that the engineering and operational strategies employed for handling substantial payloads in these specific orbits are particularly effective, leading to a higher success rate in landing. The success with heavy payloads in these orbits may reflect the robustness of the launch and landing systems designed to accommodate and manage the challenges associated with larger masses. It highlights the expertise and technological advancements that have been developed to ensure the safe return of rocket stages or landing of spacecraft in missions involving heavy payloads to PO, LEO, and ISS destinations.



Launch Success Yearly Trend

→ The graphical analysis reveals a consistent upward trend in the success rate of launches from 2013 through 2020. This progression suggests a steady improvement in mission outcomes over these years, reflecting advancements in technology, operational efficiencies, and possibly enhanced quality control measures within the space industry during this period. The increase in success rates over these years could indicate the culmination of experience, research, and development efforts that have contributed to more reliable and successful space missions.



All Launch Site Names

- The keyword DISTINCT was employed to filter and display exclusively the unique launch sites within the SpaceX dataset.
- We use the following SQL query to obtain this result:

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Note. Incorporating "DISTINCT" within the query ensures that only unique entries in the Launch_Site column from the tblSpaceX table are displayed. This keyword filters out any duplicate values, presenting a list of distinct launch sites contained within the database.

Launch Site Names Begin with 'KSC'

- The above query was executed to showcase five entries where the launch site names start with KSC.
- We use the following SQL query to obtain this result:

```
%sql SELECT * FROM SPACEXDATASET WHERE launch_site LIKE 'KSC%' LIMIT 5;
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success

Note. Incorporating "TOP 5" within the query specifies that only the first five records from the table will be displayed. The use of the "LIKE" keyword, accompanied by the wildcard 'KSC%', indicates that the names of the Launch_Site must begin with "KSC". The '%' symbol at the end signifies that any characters can follow "KSC", allowing for a range of names that start with these letters to be included in the results.

Total Payload Mass

- Using the following SQL query, we determined that boosters from NASA have transported a total payload of 48,213 kilograms.

```
%sql SELECT sum(payload_mass__kg_) AS "Total Payload Mass (kg)" FROM SPACEXDATASET
WHERE customer LIKE '%NASA (CRS)%';
```

Total Payload Mass (kg)
48213

Note. The SUM function is utilized to calculate the cumulative total in the column PAYLOAD_MASS_KG_. Through the application of the WHERE clause, the dataset is specifically filtered to perform operations only on records associated with the customer NASA (CRS), thereby allowing for a more targeted and efficient evaluation of relevant data. This approach ensures that calculations focus on the specific payload mass carried by NASA's Commercial Resupply Services (CRS), providing a clear and accurate view of the cumulative payload mass that NASA has transported through these missions. This method of filtering and summing is crucial for analyzing and gaining a better understanding of logistical operations and the impact of resupply missions on payload transportation to space.

Average Payload Mass by F9 v1.1

- Using the following SQL query, we determined that the average mass of the payload transported by the booster version F9 v1.1 was computed to be 2928.4 kilograms.

```
%sql SELECT sum(payload_mass__kg_) / count(payload_mass__kg_) AS "Average Payload Mass (kg)" FROMSPACEXDATASET WHERE booster_version LIKE 'F9 v1.1';
```

Average Payload Mass (kg)
2928

Note. The AVG function is employed to calculate the average value in the column PAYLOAD_MASS_KG_. By incorporating the WHERE clause, the dataset undergoes a precise filtration process to ensure that calculations are exclusively executed on records pertaining to the Booster_version F9 v1.1. This specific targeting facilitates an accurate and insightful analysis of the average payload mass associated with this particular version of the Falcon 9 rocket. Such an analysis is instrumental in understanding the performance and capacity trends of the F9 v1.1 booster over various missions. It enables stakeholders to assess the efficiency, reliability, and evolution of payload capabilities within the context of SpaceX's rocket development program. By focusing on this distinct booster version, the calculation sheds light on its operational effectiveness and its contribution to space logistics and payload deployment strategies.

First Successful Ground Landing Date

- It was noted that the initial successful landing on a ground pad occurred on the 22nd of December, 2015.
- We use the following SQL query:

```
%sql SELECT min(DATE) AS "First Successful Landing Outcome Date"
FROM SPACEXDATASET WHERE landing__outcome LIKE'Success (ground
pad)';
```

First Successful Landing Outcome Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We use the following SQL query:
- `%sql SELECT DISTINCT booster_version FROM SPACEXDATASET WHERE landing__outcome = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;`
- The WHERE clause was employed to isolate boosters that achieved a successful landing on a drone ship, and the AND condition was utilized to identify instances of successful landings carrying a payload mass exceeding 4000 kilograms yet not surpassing 6000 kilograms.

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Note. Applying a selection criterion to Booster_Version, the WHERE clause refines the dataset to consider entries where Landing_Outcome equals 'Success (drone ship)'. Concurrently, the AND clause introduces further filtering parameters, demanding that Payload_MASS_KG_ must be greater than 4000 AND less than 6000. This structured query setup ensures the extraction of specific data points, focusing on successful drone ship landings by boosters that are tasked with transporting payloads within the defined mass range, thereby facilitating a focused analysis on operational efficiencies and payload delivery capabilities under these precise parameters.

Total Number of Successful and Failure Mission Outcomes

- We use the following SQL query:

```
%sql SELECT (SELECT count(*) FROM SPACEXDATASET WHERE lcase(landing__outcome) LIKE '%success%') AS "Success", count(*) AS "Failure" FROM SPACEXDATASET WHERE lcase(landing__outcome) NOT LIKE '%success%';
```

- Wildcard characters such as '%' were utilized in the filtering process to select records WHERE the MissionOutcome indicated either success or failure.

Success	Failure
61	40

Boosters Carried Maximum Payload

- We use the following SQL query:

```
%sql SELECT booster_version, payload_mass__kg_ FROM SPACEXDATASET WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEXDATASET);
```

- Through the use of a subquery within the WHERE clause and employing the MAX() function, we identified the booster responsible for transporting the heaviest payload.

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- We use the following SQL query:

```
%sql SELECT DISTINCT booster_version FROM SPACEXDATASET WHERE landing__outcome
= 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;
```

- A blend of the WHERE clause, LIKE operator, AND conjunction, and BETWEEN criteria was applied to sift through the data for instances of unsuccessful landings on drone ships, including details of their booster models and names of the launch sites, specifically for the year 2015.

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We use the following SQL query:

```
%sql SELECT landing__outcome, count(landing__outcome) AS
"Count" FROM SPACEXDATASET WHERE DATE BETWEEN '2010-06-
04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY
count(landing__outcome) DESC;
```

- We extracted information on landing outcomes along with their frequency from the dataset, employing the WHERE clause to narrow down events occurring from March 20, 2010, to June 4, 2010.
- The data was then organized using the GROUP BY clause to consolidate identical landing outcomes, followed by the implementation of the ORDER BY clause to arrange these consolidated outcomes in a descending sequence.

Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

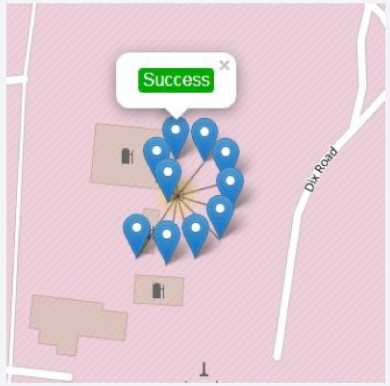
Note. The COUNT function tallies the number of records within a column. The WHERE clause is used to narrow down the dataset based on specific criteria. The LIKE operator allows for pattern matching using wildcards, enabling flexible search conditions. The AND keyword is employed to combine multiple conditions, ensuring that all specified criteria are met simultaneously within the query's constraints.

Section 4

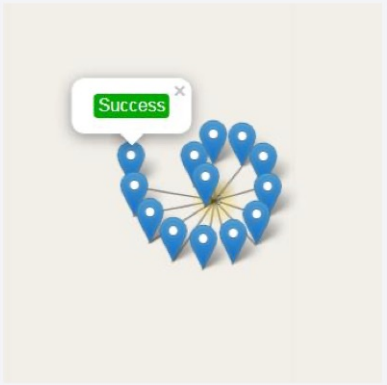
Launch Sites Proximities Analysis

Map Markets of Success/Failed Landings

- The indicators on the map represent the results of Falcon 9 first stage landings, denoted as either Success or Failure. These markers are organized according to the geographical locations of the launch sites.
- By observing the proportion of green markers (indicating success) to red markers (indicating failure), one can deduce the success rate of Falcon 9 first stage landings at each launch site.
- This visual arrangement allows for an intuitive understanding of how each launch site performs in terms of successfully landing the Falcon 9's first stage.

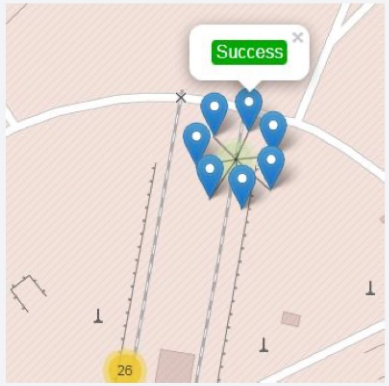


VAFB SLC-4E



KSC LC-39A

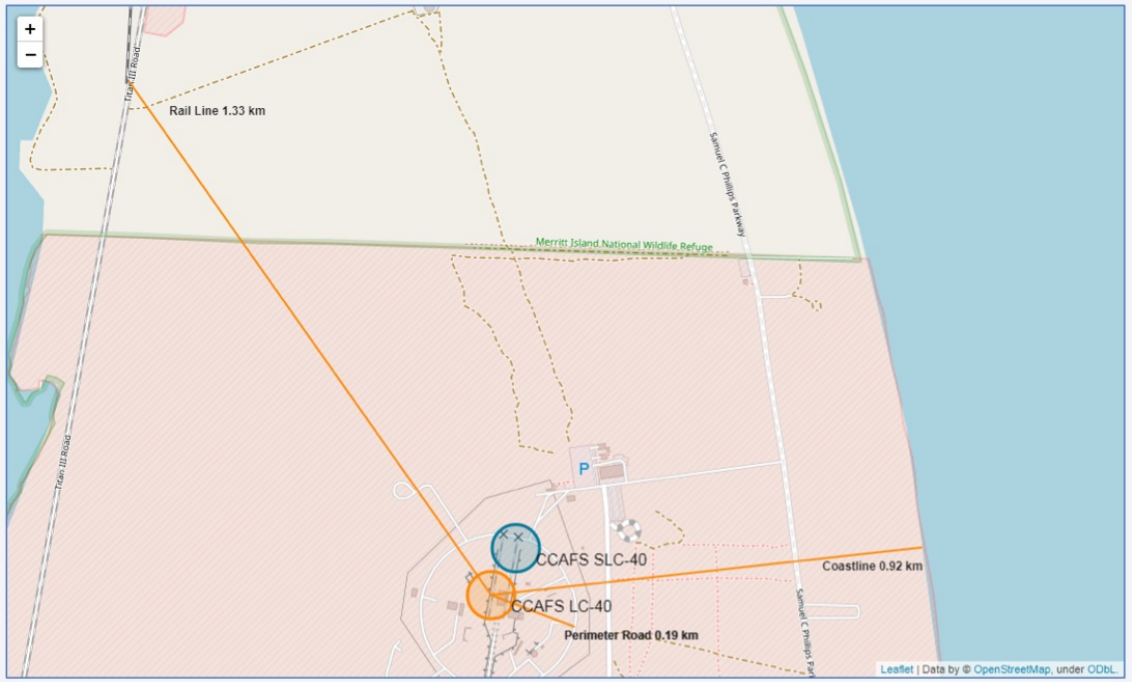
CCAFS LC-40



CCAFS SLC-40

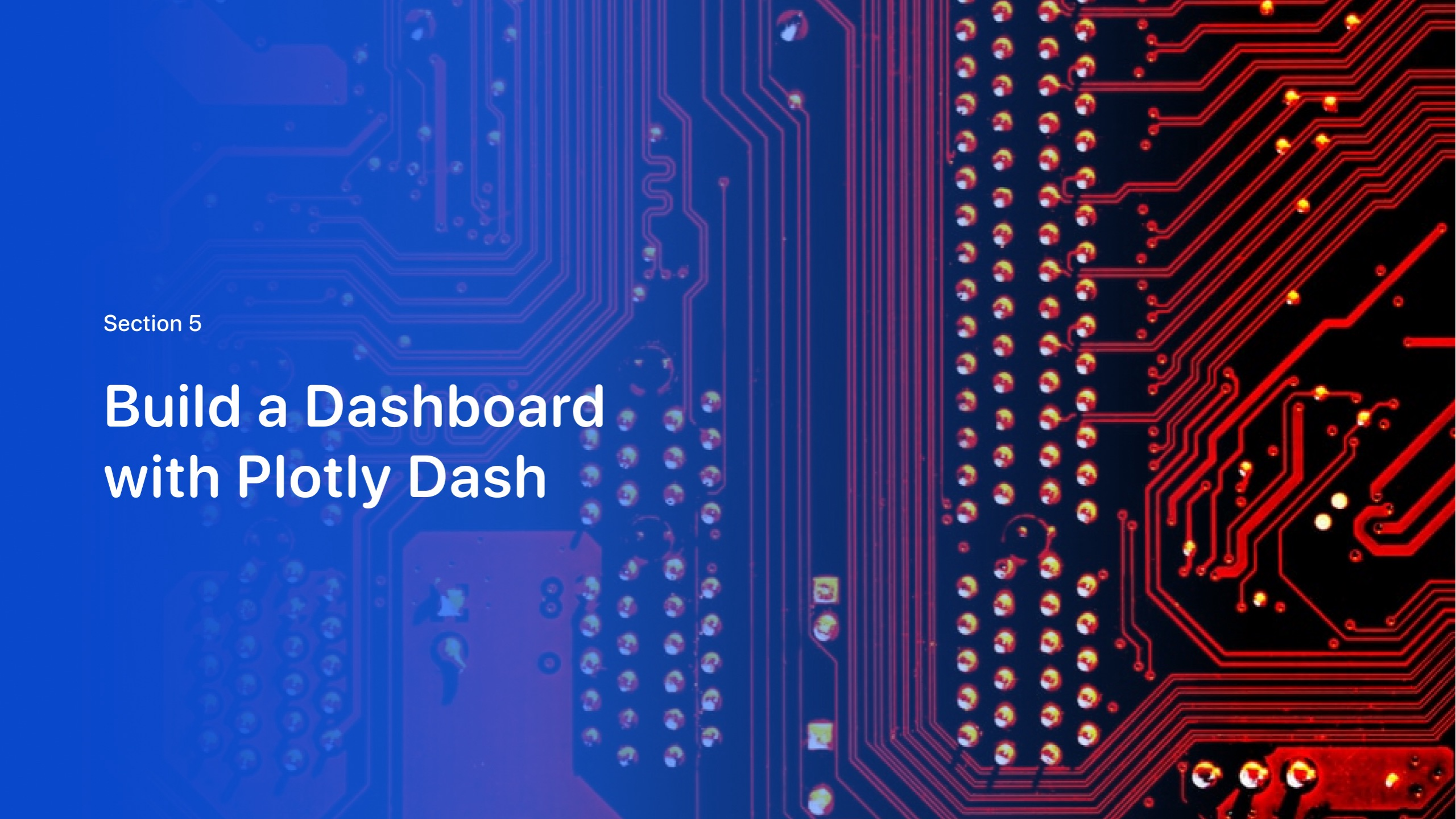
Distance from Launch Site to Proximities

- The launch sites CCAFS LC-40 and CCAFS SLC-40 are situated in proximity to each other, their coordinates nearly overlapping but not precisely identical.
- The boundary road encircling CCAFS LC-40 lies at a distance of 0.19 kilometers from the site's exact coordinates.
- The shore is located 0.92 kilometers distant from CCAFS LC-40.
- The railway track is positioned 1.33 kilometers away from CCAFS LC-40.



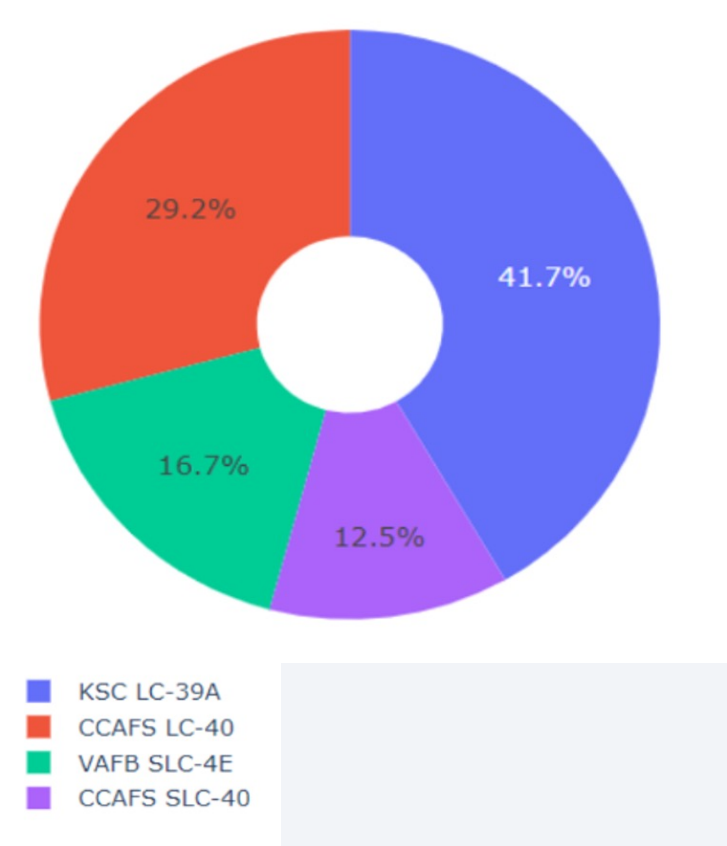
Falcon 9 Launch Site Locations

- The launch sites CCAFS LC-40 and CCAFS SLC-40 are situated in proximity to each other, their coordinates nearly overlapping but not precisely identical.
- The boundary road encircling CCAFS LC-40 lies at a distance of 0.19 kilometers from the site's exact coordinates.
- The shore is located 0.92 kilometers distant from CCAFS LC-40.
- The railway track is positioned 1.33 kilometers away from CCAFS LC-40.


Section 5
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

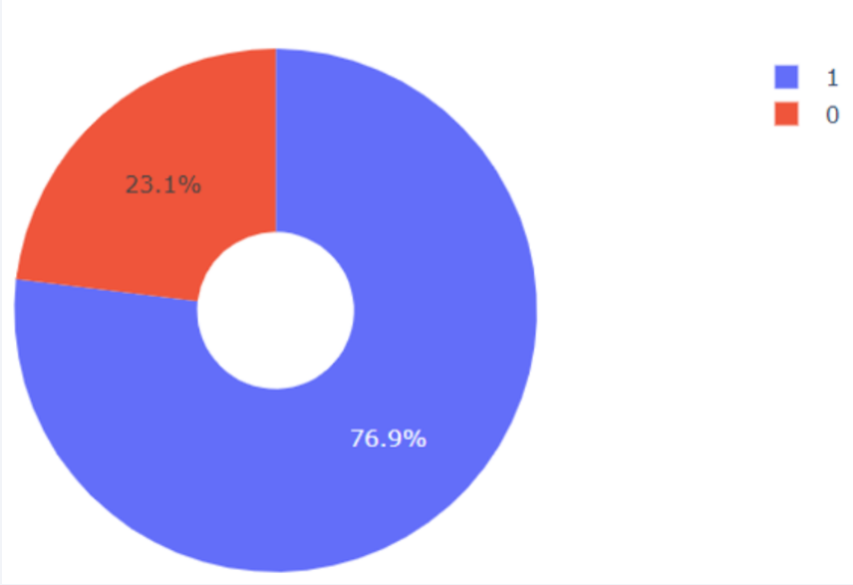
Total Success Launches By all sites:



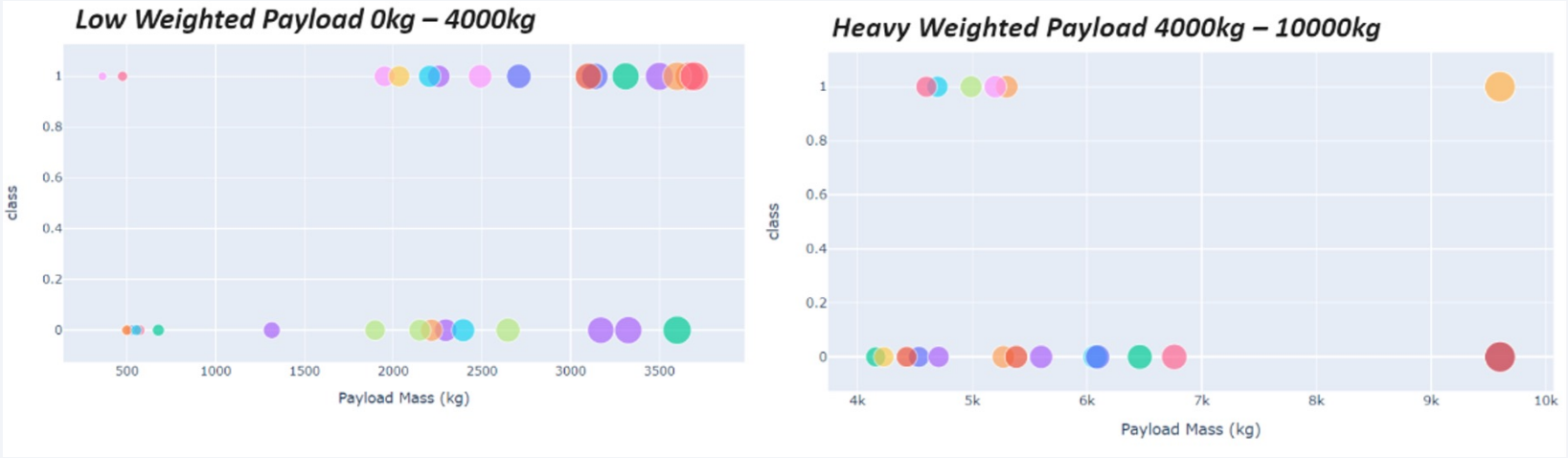
→ The data clearly indicates that **KSC LC-39A** has distinguished itself as the premier launch site by recording the highest number of successful launches among all sites. This achievement underscores the site's exceptional operational capabilities and technological advancements, positioning KSC LC-39A at the forefront of space exploration endeavors. The success of KSC LC-39A not only reflects its robust infrastructure and efficient mission planning but also its role in advancing the frontiers of space science by consistently achieving mission objectives. This standout performance sets KSC LC-39A as a benchmark for reliability and success in the highly competitive and challenging arena of space launches.

Pie chart showing the Launch site with the highest launch success ratio

→ The pie chart illustrates a comparative analysis of launch success rates across various sites, with a special focus on the site exhibiting the highest success rate. It visually represents how each launch site stacks up in terms of mission success, making it clear which one leads in operational reliability. Specifically, **KSC LC-39A stands out with a 76.9% success rate and a 23.1% failure rate,** highlighting its superior performance in space launch missions. This data not only showcases KSC LC-39A's technological and operational prowess but also provides insights into the critical factors that contribute to its success, offering a concise overview of the effectiveness of different launch sites in achieving mission objectives.



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



→ In the analysis, it becomes evident that the success rates for launching low-weighted payloads are higher than those for heavy-weighted payloads. This observation suggests that lighter payloads are associated with a greater likelihood of mission success. The distinction in success rates between the two categories indicates potential factors such as the challenges of launching heavier payloads, which may include increased structural and engineering demands, more complex logistics, and higher fuel requirements. This pattern underscores the importance of considering payload weight in the planning and execution of space missions, highlighting how the mass of the payload can significantly impact the outcome of a launch.

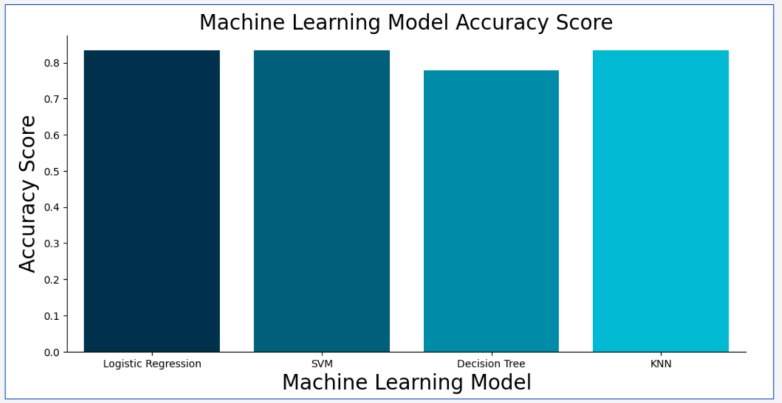
Section 6

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier stands out as the algorithm with the superior accuracy in classification tasks.

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

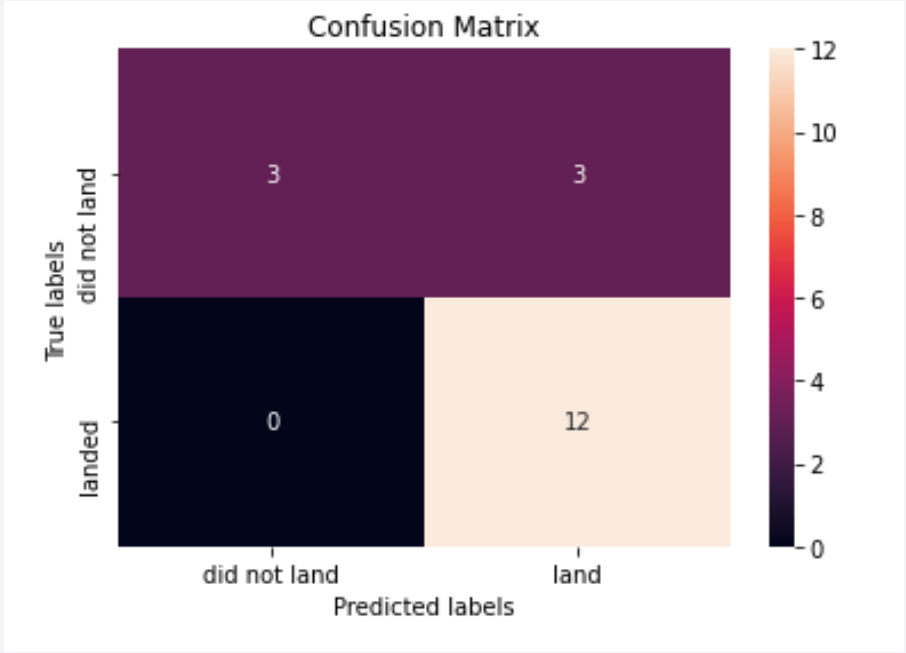



```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Best model is DecisionTree with a score of 0.8732

Confusion Matrix

- The confusion matrix generated from the decision tree classifier provides insight into its ability to accurately identify and separate the distinct classes within the dataset. This analytical tool highlights the classifier's proficiency in categorization, underscoring its effectiveness in distinguishing between successful and unsuccessful landings.
- Despite these strengths, a notable challenge arises with the occurrence of false positives. In these instances, the classifier erroneously labels landings that failed as if they had succeeded. This misclassification can skew the accuracy of the model's predictions, leading to potential misinterpretations of the effectiveness of landing strategies.
- Addressing this issue is crucial for refining the classifier's performance, ensuring more reliable predictions, and minimizing the impact of incorrect assessments on decision-making processes related to launch and landing operations.



True Negative	False Positive
False Negative	True Positive

Prediction breakdown:

→ 12 True Positives and 3 True Negatives

→ 3 False Positives and 0 False Negatives

Conclusions

Based on the data and analysis conducted, several key findings emerge, painting a comprehensive picture of launch success factors and outcomes:

1. **Correlation Between Launch Frequency and Success:** There is a clear correlation observed between the number of flights conducted at a particular launch site and its corresponding success rate. This indicates that sites with a higher volume of launches tend to have a higher likelihood of successful missions, possibly due to accumulated operational experience and refined processes over time.
2. **Trend of Increasing Success Over Time:** The period from 2013 to 2020 marks a significant phase of improvement in launch success rates. This upward trend suggests advancements in technology, engineering practices, and mission planning that have collectively contributed to enhancing the reliability of launches over these years.

Conclusions

3. **Orbital Success Patterns:** The orbits designated as ES-L1, GEO, HEO, SSO, and VLEO stand out for having the highest rates of launch success. This pattern may reflect the specialized expertise and technological capabilities developed for missions targeting these specific orbits, indicating a strategic focus on achieving high success rates in these areas.
4. **Leading Launch Site:** Among the various launch sites analyzed, KSC LC-39A is distinguished by having the highest number of successful launches. This achievement underscores the site's pivotal role in space exploration efforts, backed by superior infrastructure, strategic location, and historical significance in spaceflight history.
5. **Optimal Machine Learning Model:** The Decision Tree Classifier has been identified as the most effective machine learning algorithm for predicting launch success in this context. Its superiority in performance suggests that it is particularly adept at handling the complexities and nuances of the dataset, making it an invaluable tool for forecasting outcomes and guiding decision-making processes.

Appendix

- Jupyter Notebooks and Dashboard Python File
  - **GitHub URL (Data Collection):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/01_Data_Collection.ipynb
  - **GitHub URL (Web Scraping):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/02_Web_Scraping.ipynb
  - **GitHub URL (Data Wrangling):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/03_Data_Wrangling.ipynb
  - **GitHub URL (EDA with SQL):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/04_SQL.ipynb
  - **GitHub URL (EDA with Data Visualization):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/05_Data_Visualization.ipynb
  - **GitHub URL (Folium Maps):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/06_Interactive_Visual_Analytics.ipynb
  - **GitHub URL (Dashboad File):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/07_Interactive_Visual_Analytics_Plotly.py
  - **GitHub URL (Machine Learning):** https://github.com/pablorcordoves/IBM_SpaceX_Data_Science_Capstone/blob/main/08_Predictive_Analytics.ipynb

Thank you!