IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Data science Capstone Project

Prince Wang

2022/05/26

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  SpaceX API. &Web Scraping(https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

- Perform data wrangling

  Convert outcomes into Training Labels with the booster successfully/unsuccessful landed

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

  By SVM, Classification Trees and Logistic Regression

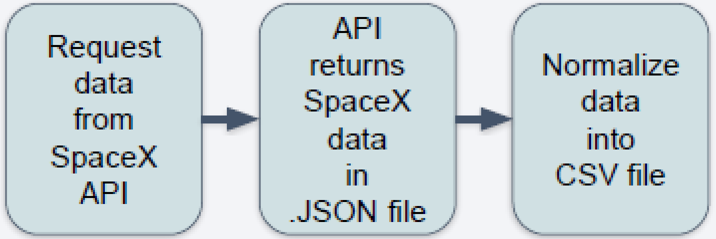
Data Collection

- Describe how data sets were collected.

1.Request to the SpaceX API :

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

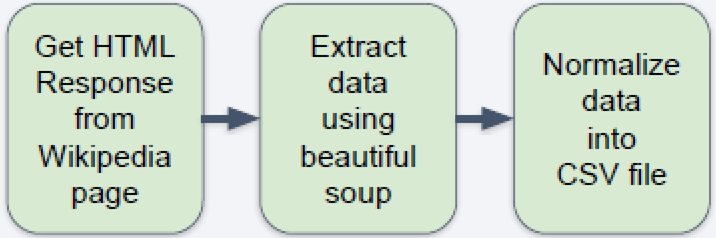
data = pd.json_normalize(response.json())   data.tocsv(‘data.csv’)



2.Web scraping

```
html_data = requests.get(static_url).text
```

```
soup = BeautifulSoup(html_data, 'html5lib')
```



Data Collection – SpaceX API

Requesting rocket launch data from SpaceX API:

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

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

Converting Response to a JSON file:

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

we combine the 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}
```

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

exporting to a CSV

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

Data Collection - Scraping

1.Getting response from HTML

```
html_data = requests.get(static_url).text
```

2.Creating a BeautifulSoup object

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3.Finding all tables and assigning the result to a list

```
html_tables = soup.find_all('table')
```

4.Extracting column name one by one

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

5.Creating an empty dictionary with keys

```
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']=[]
```

6.Filling up the `launch_dict` with launch records

```
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
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            print(date)
            launch_dict['Date'].append(date)
```

7.Creating a Dataframe and exporting it to a CSV

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

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

Data Wrangling

There are several cases in which the booster failed to successfully land on thedataset, and sometimes it attempted to land but failed because of accident

- `True Ocean`: the mission result has successfully landed in a specific area of the ocean
- `False Ocean`: the mission result has not successfully landed in a specific area of the ocean
- `True RTLS`: the mission result successfully landed on the ground pad
- `False RTLS`: the mission result has not successfully landed on the ground pad
- `True ASDS`: the mission result has successfully landed on the drone ship
- `False ASDS`: the mission result has not landed on the drone ship

EDA with Data Visualization

- **Scatter chart:**

▪ *Flight Number vs. Launch Site*
▪ *Payload vs. Launch Site*
▪ *Flight Number vs. Orbit Type*
▪ *Payload vs. Orbit Type*

A scatter plot shows how much one variable is affected by another. The relationship between two variables is called a correlation. This plot is generally composed of large data bodies.

- **Bar chart:**

Orbit Type vs. Success Rate

A Bar chart makes it easy to compare datasets between multiple groups at a glance. One axis represents a category and the other axis represents a discrete value. The purpose of this chart is to indicate the relationship between the two axes.

- **Line chart:**

Year vs. Success Rate

A Line chart shows data variables and trends very clearly and helps predict the results of data that has not yet been recorded.

EDA with SQL

Loading the dataset into the corresponding table in a Db2 database, and executing SQL queries to answer following questions:

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

Build an Interactive Map with Folium

Objects created and added to a folium map:

- Markers that show all launch sites on a map
- Markers that show the success/failed launches for each site on the map
- Lines that show the distances between a launch site to its proximities

By adding these objects, following geographical patterns about launch sites are found:

- Are launch sites in close proximity to railways? *Yes*
- Are launch sites in close proximity to highways? *Yes*
- Are launch sites in close proximity to coastline? *Yes*
- Do launch sites keep certain distance away from cities? *Yes*

Build a Dashboard with Plotly Dash

The dashboard application contains a pie chart and a scatter point chart.

- *Pie chart*

For showing total success launches by sites

This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.

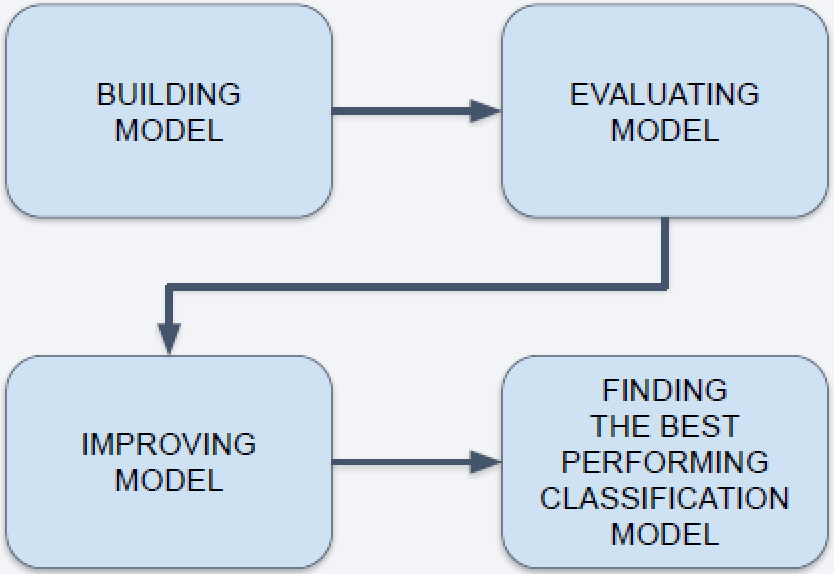
- *Scatter chart*

For showing the relationship between Outcomes and Payload mass (Kg) by different boosters

Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg

This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

Predictive Analysis (Classification)



Perform exploratory Data Analysis and determine Training Labels

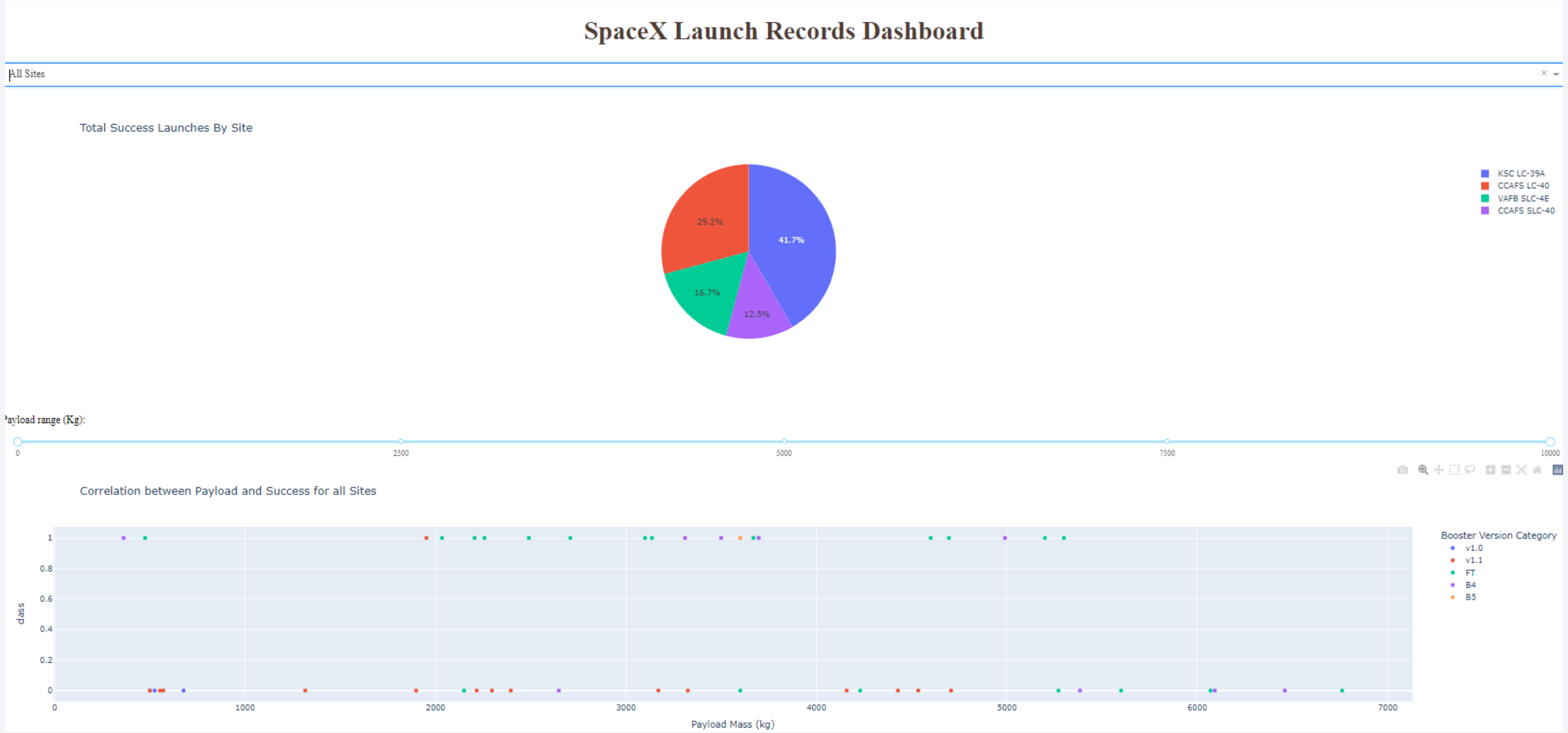
- Create a column for the class
- Standardize the data
- Split into training data and test

data

• Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

- Find the method performs best
- using test data

Results

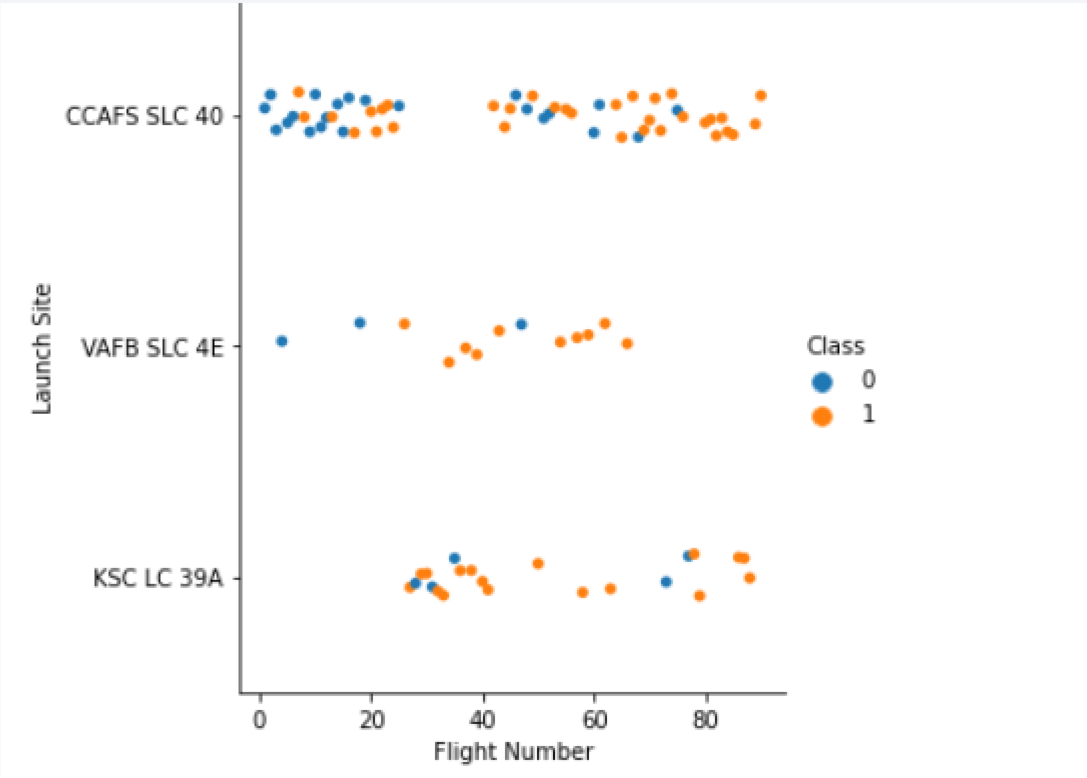


- The up screenshot is a preview of the Dashboard with Plotly Dash.
- The results of EDA with visualization, EDA with SQL, Interactive Map with Folium,and Interactive Dashboard will be shown in the next slides.
- Comparing the accuracy of the four methods, all return the same accuracy of about*83%* for test data.

Section 2

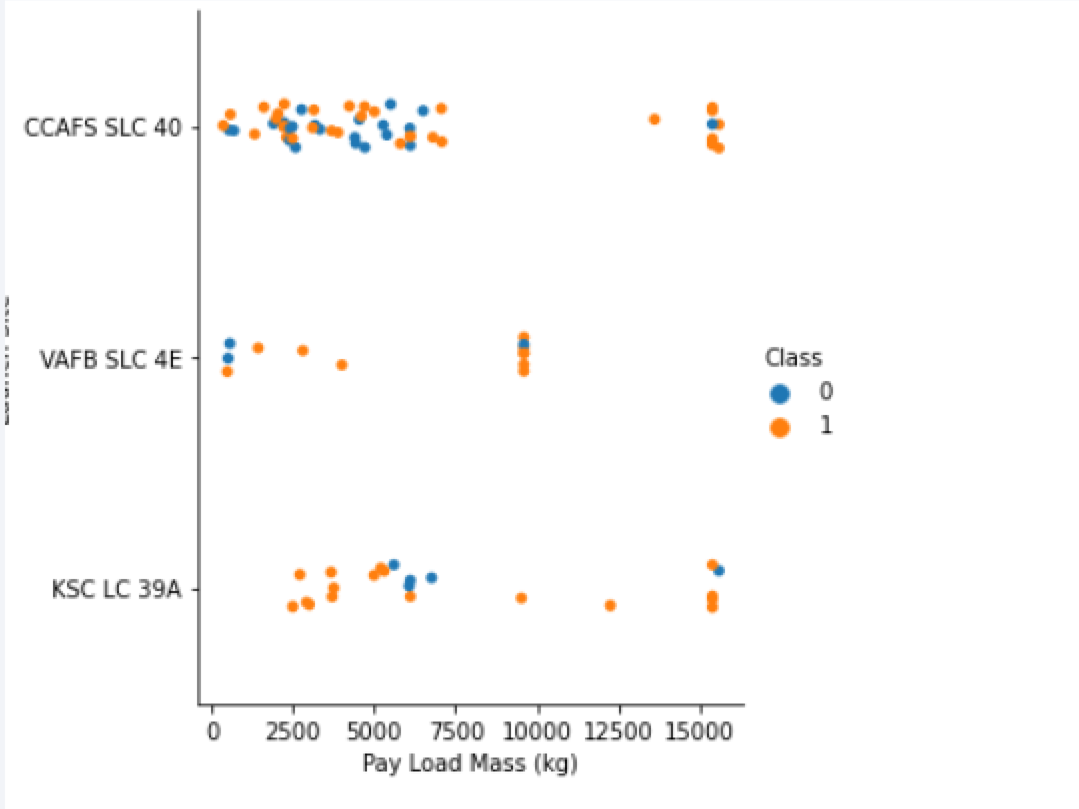
Insights drawn from EDA

Flight Number vs. Launch Site



- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- This figure shows that **the success rate increased as the number of flights increased**.
- As the success rate has increased considerably since the *20th* flight, this point seems to be a big breakthrough.

Payload vs. Launch Site

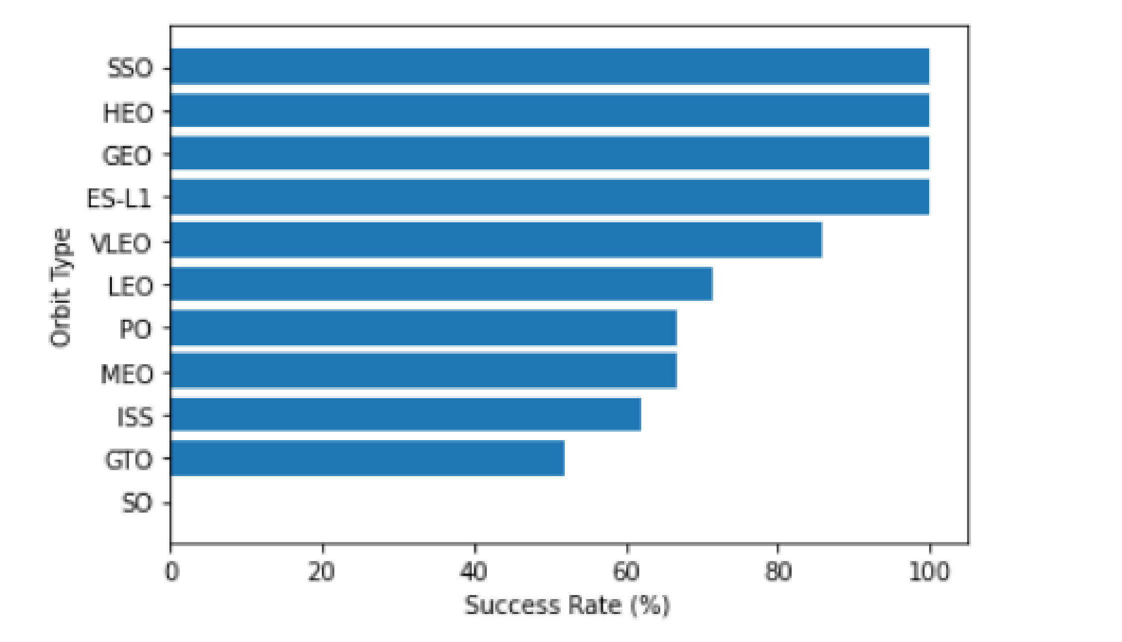


Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.

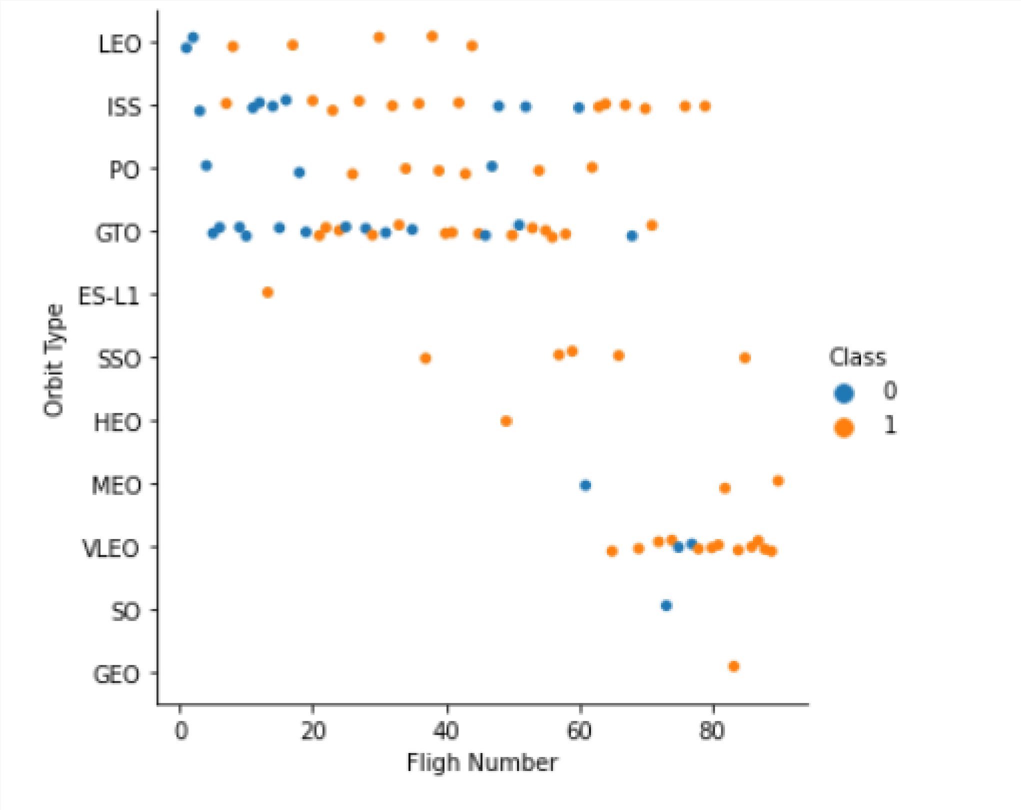
• At first glance, the larger pay load mass, the higher the rocket's success rate, but it seems difficult to make decisions based on this figure because **no clear pattern can be found between successful launch and Pay Load Mass**.

Success Rate vs. Orbit Type

- Orbit types **SSO**, **HEO**, **GEO**, and **ES-L1** have **the highest success rates**(*100%*).
- On the other hand, the success rate of orbit type **GTO** is only *50%*, and it is the **lowest** except for type SO,which recorded failure in a single attempt.



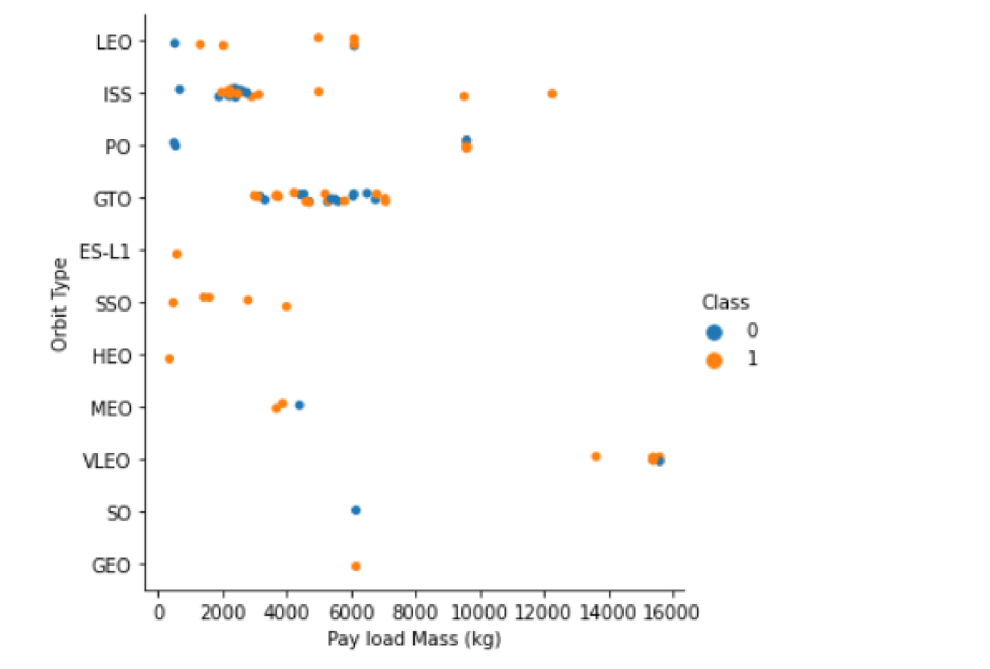
Flight Number vs. Orbit Type



- Class 0 (blue) represents unsuccessful launch, and Class 1(orange) represents successful launch.
- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in GTO orbit, there seems to be no relationship between flight numbers and success rate.
- SpaceX starts with LEO with a moderate success rate, and it seems that VLEO, which has a high success rate, is used the most in recent launches.

Payload vs. Orbit Type

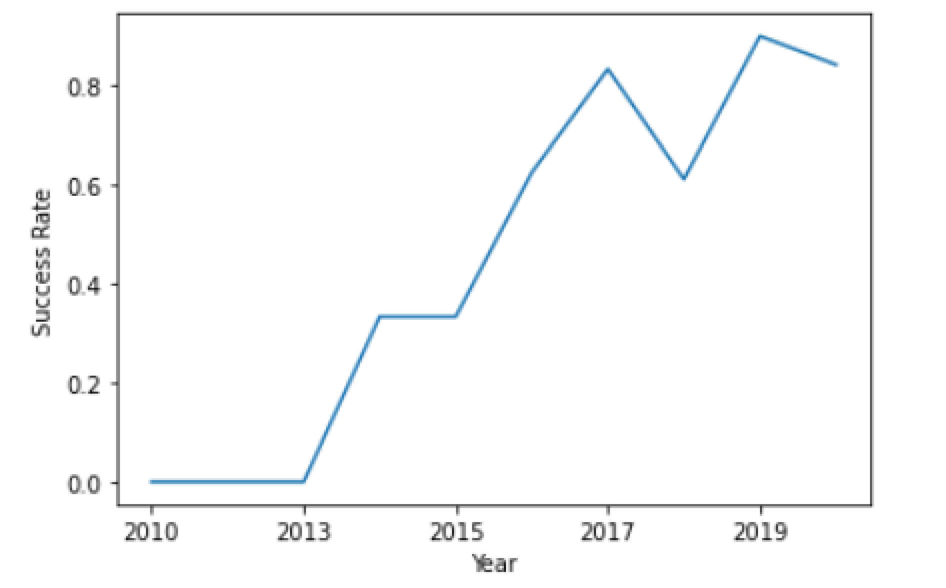
- Class 0 (blue) represents unsuccessful launch, and Class 1
- (orange) represents successful launch.
- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- However, in the case of GTO, it is hard to distinguish between the positive landing rate and the negative landing because they are all gathered together.



Launch Success Yearly Trend

Since 2013, the success rate has continued to **increase** until 2017.

- The rate decreased slightly in 2018.
- Recently, it has shown a success rate of about *80%*.



All Launch Site Names

Query SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL

	Launch Site	Lat	Long	class	marker_color
0	CCAFS LC-40	28.562302	-80.577356	0	red
1	CCAFS SLC-40	28.563197	-80.576820	1	green
2	KSC LC-39A	28.573255	-80.646895	1	green
3	VAFB SLC-4E	34.632834	-120.610746	0	red

When the SQL DISTINCT clause is used in the query, only unique values are displayed in the Launch_Site column from the SpaceX table.
There are four unique launch sites:CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

Query SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE ‘CCA%’LIMIT 5

Result

Out[18]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Only five records of the SpaceX table were displayed using LIMIT 5 clause in the query.
- Using the LIKE operator and the percent sign (%)together, the Launch_Site name starting with CAA could be called.

Total Payload Mass

Query

```
SELECT SUM(PAYLOAD_MASS__KG_)AS total_payload_mass_kg FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

total_payload_mass_kg
45596

Using the SUM() function to calculate the sum of column PAYLOAD_MASS__KG_.
In the WHERE clause, filter the dataset to perform calculations only if Customer is NASA (CRS).

Average Payload Mass by F9 v1.1

Query

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

Result

avg_payload_mass_kg
2928

- Using the AVG() function to calculate the average value of column PAYLOAD_MASS__KG_.
- In the WHERE clause, filter the dataset to perform calculations only if Booster_version is F9 v1.1.

First Successful Ground Landing Date

Query

```
%%sql
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

Result

first_successful_landing_date
2015-12-22

- Using the MIN() function to find out the earliest date in the column DATE.
- In the WHERE clause, filter the dataset to perform a search only if Landing__outcome is Success (ground pad).

Successful Drone Ship Landing with Payload between 4000 and 6000

Query

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

Result

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

Total Number of Successful and Failure Mission Outcomes

Query

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

Result

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

Boosters Carried Maximum Payload

Query

```
%%sql
SELECT DISTINCT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

- Using a subquery, first, find the maximum value of the payload by using MAX() function, and second, filter the dataset to perform a search if PAYLOAD_MASS__KG_ is the maximum value of the payload.
- According to the result, version F9 B5 B10xx.x boosters could carried the maximum payload.

Result

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

Query

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

Result

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

In the WHERE clause, filter the dataset to perform a search if Landing__outcome is Failure (drone ship). Using the AND operator to display a record if additional condition YEAR is 2015.

In 2015, there were two landing failures on drone ships.

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

Query

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

Result

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

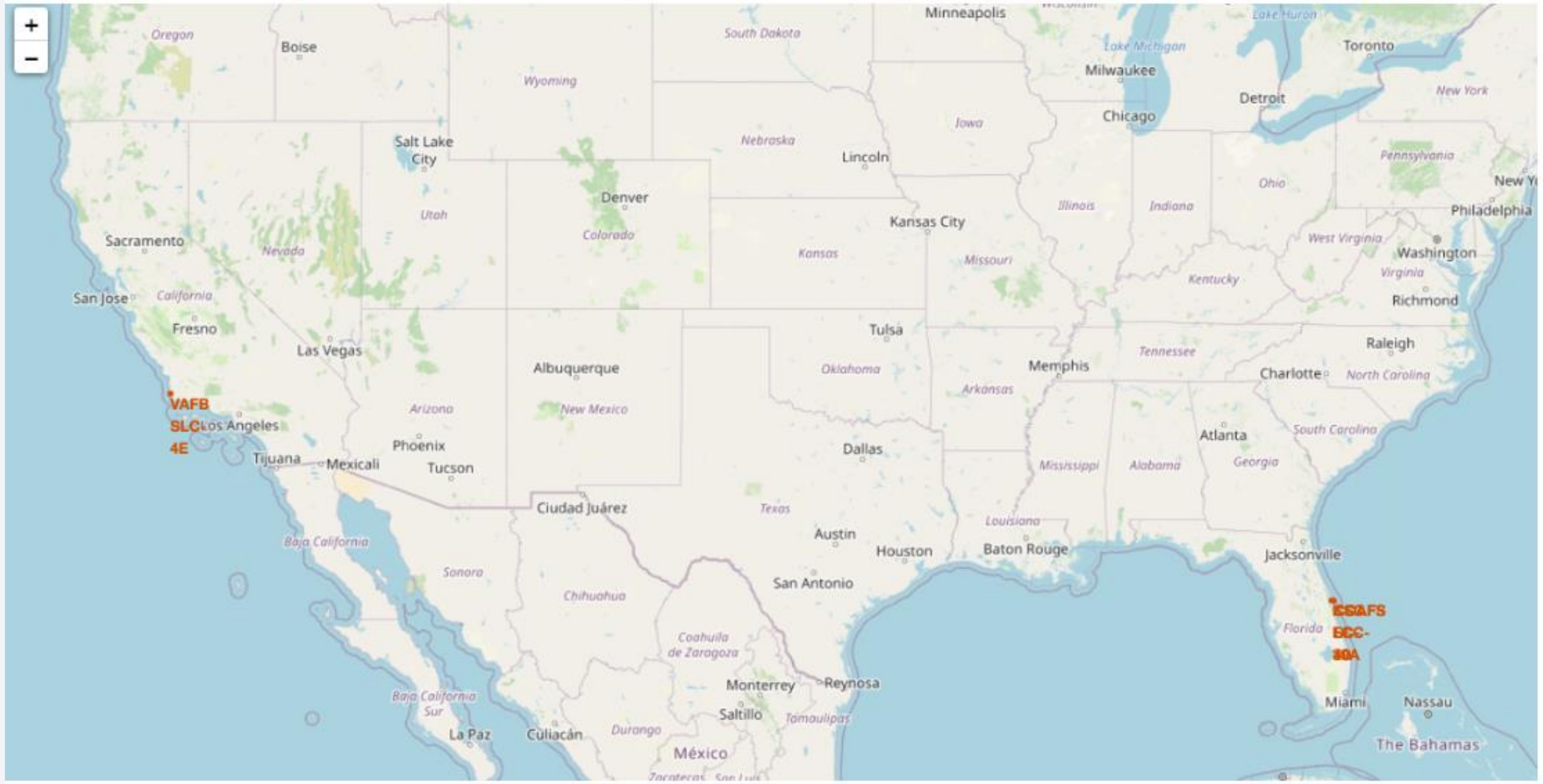
In the WHERE clause, filter the dataset to perform a search if the date is between 2010-06-04 and 2017-03-20.
• Using the ORDER BY keyword to sort the records by total number of landing, and using DESC keyword to sort the records in descending order.
• According to the results, the number of successes and failures between 2010-06-04 and 2017-03-20 was similar.

Section 3

Launch Sites Proximities Analysis

All Launch Sites’ Locations

All Launch Sites’ Locations

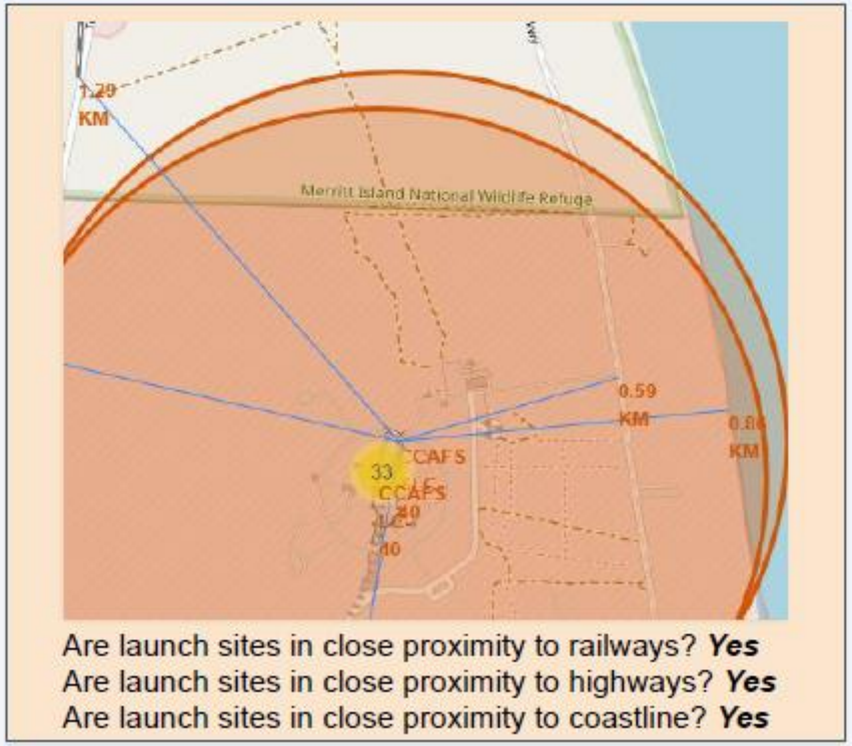


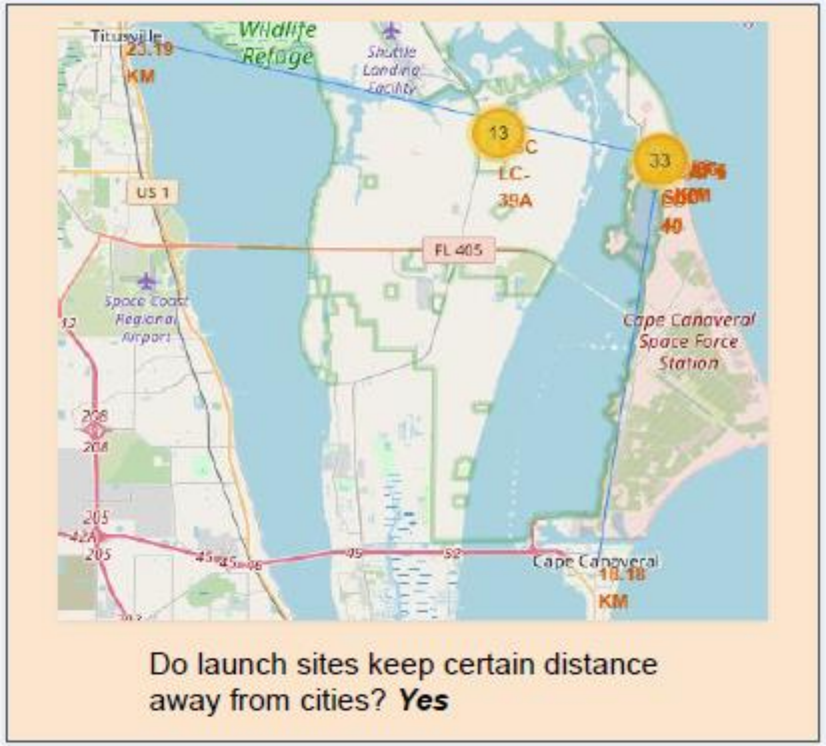
- The left map shows all *SpaceX* launch sites, and the right map also shows that all launch sites are in the United States.
- As can be seen on the map, all launch sites are near the coast.

Color-labeled Launch Outcomes

By clicking on the marker clusters,successful landing (green)or failed landing (red) are displayed.

Proximities of Launch Sites





It can be found that the launch site is **close to railways and highways** for transportation of equipment or personnel, and is also **close to coastline** and relatively **far from the cities** so that launch failure does not pose a threat.

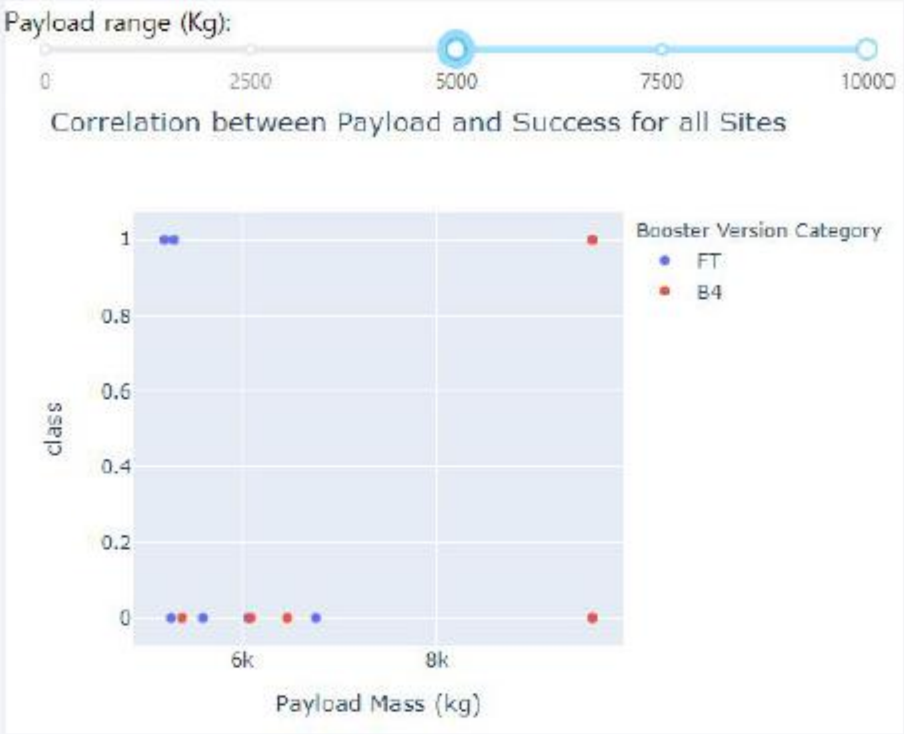
Section 4

Build a Dashboard with Plotly Dash

Launch Site with Highest Launch Success Ratio

KSLC-39A has the highest success rate with 10 landing successes *(76.9%)* and 3 landing failures *(23.1%)*.

Payload vs. Launch Outcome Scatter Plot for all sites

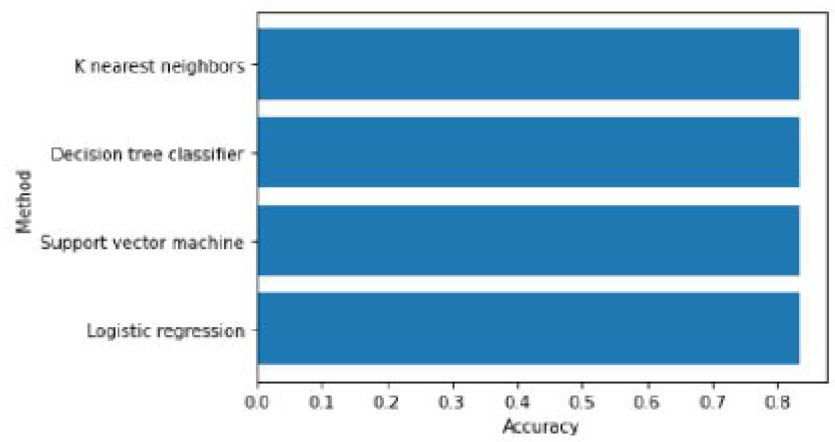


These figures show that **the launch success rate** *(class 1)* **for low weighted payloads*(0-5000 kg)* is higher** than that of heavy weighted payloads*(5000-10000 kg)*.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



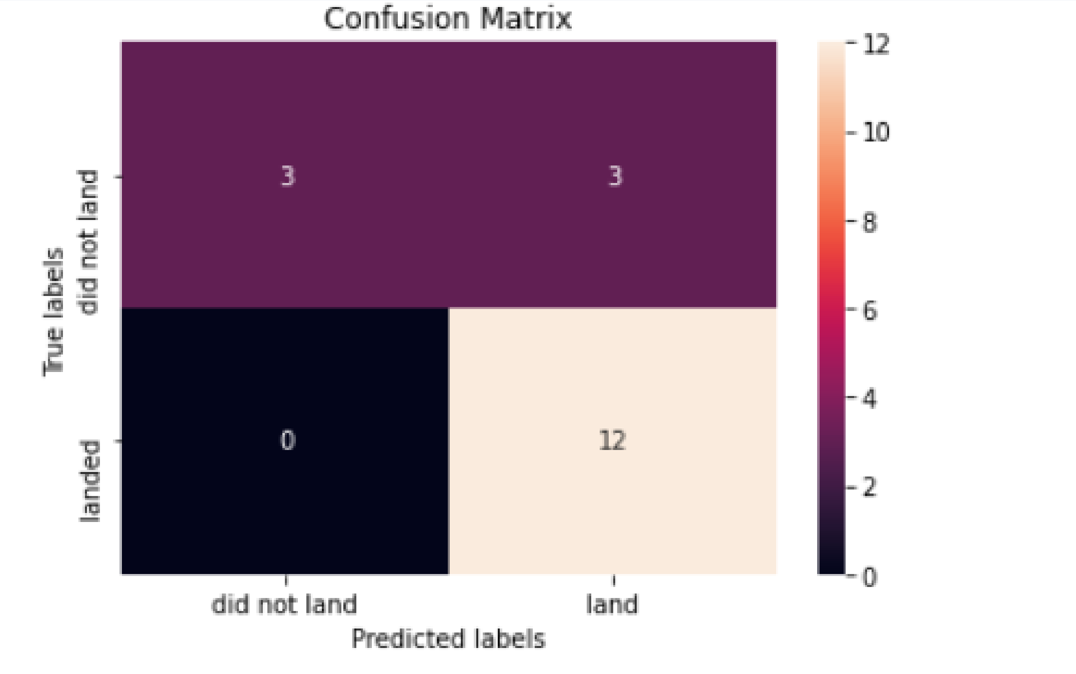
	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

In the test set, **the accuracy of all models** was virtually the **same** at **83.33%**.

• It should be noted that the test size was small at 18.

• Therefore, more data is needed to determine the optimal model.

Confusion Matrix



The confusion matrix is the same for allmodels because all models performed the same for the test set.

• The models predicted 12 successful landings when the true label was successful and 3 failed landings when the true label was failure. But there were also 3 predictions that said successful landings when the true label was failure *(false positive)*.

• Overall, **these models predict successful landings**.

Conclusions

- As the number of flights increased, the success rate increased, and recently it has exceeded 80%.
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).
- The launch site is close to railways, highways, and coastline, but far from cities.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- In this dataset, all models have the same accuracy (83.33%), but it seems that more data is needed to determine the optimal model due to the small data size.

Appendix

- GitHub URL

Thank you!