IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thái An

15 jan 2023

Outline

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

Executive Summary

- Summary of methodologies

Data collection

Data wrangling

Exploratory analysis with data visualization

Exploratory analysis with SQL

Building an interactive map with Folium

Building a Dashboard with PlotlyDash

Predictive analysis

- Summary of all results

Exploratory analysis results

Interactive analytics

Predictive analysis

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

- Problems I want to find answers

The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - SpaceX Rest API
    - Web Scrapping from Wikipedia
- Perform data wrangling
    - Data cleaning of null values and One Hot Encoding data fields
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Logistic Regression, K-NN, Support Vector Machines, Decision Trees models.

Data Collection

- The data collected includes information about SpaceX launches obtained from their

REST API, which provides details such as the rocket used, payload delivered, launch and landing specifications, and outcome of the landing. The API can be accessed through the endpoint "api.spacexdata.com/v4/“.

- An alternative data source for obtaining information on Falcon 9 launches is by using web scraping on the Wikipedia website with the library BeautifulSoup.

Data Collection – SpaceX API

1. Response from the SpaceX API

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

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

2. Obtain the json info

```python
# Use json_normalize meethod to convert the json result into a dataframe
from pandas import json_normalize
data=json_normalize(response.json())
```

3. Clean Data

```
BoosterVersion
```

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

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

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

4. Store data in a dictionary

```python
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Convert dictionary to a dataframe

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

Data Collection - Scraping

1.Response from wikipedia

```
# assign the response to a object
response=requests.get(static_url).text
```

2.BeautifulSoup Object creation

```
# Use BeautifulSoup() to create a Beautij
soup=BeautifulSoup(response,'html')
```

3.Find the tables

```
# Assign the result to a list called
html_tables=soup.findAll('table')
```

4.Find the column names

```
column_names = []
# Apply find_all() function with `th` element on firs
# Iterate each th element and apply the provided extr
# Append the Non-empty column name (`if name is not N
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.Create a dictionary with the tables

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each va
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.Adjunt data to keys

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

7.Dataset Creation

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

Data Wrangling

1.Data exploration and creation of a landing_outcomes list

```
# Apply value_counts() on co
df.LaunchSite.value_counts()
```

```
# Apply value_counts on Ort
df.Orbit.value_counts()
```

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

2.Isolate the bad outcomes types

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

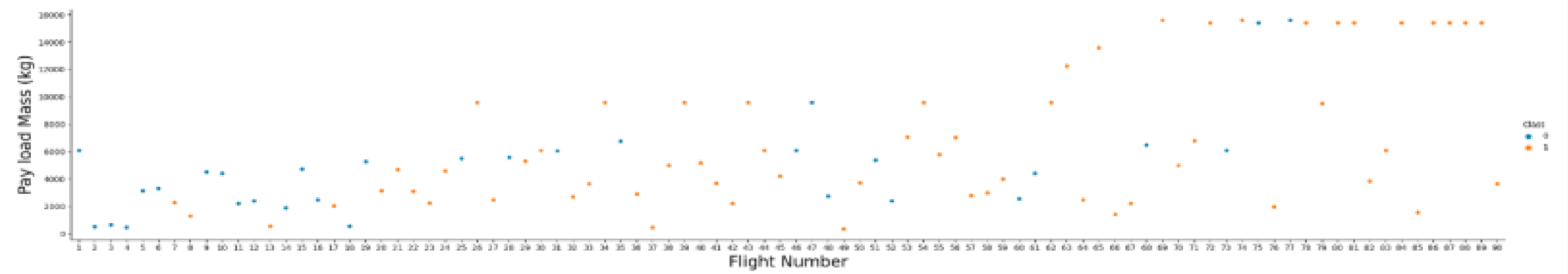
3.Creation of a list with the good and bad outcomes

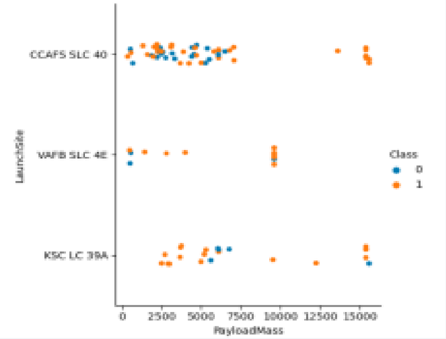
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for row in df['Outcome']:
    if row in bad_outcomes:
        landing_class.append(0)
    else: landing_class.append(1)
```

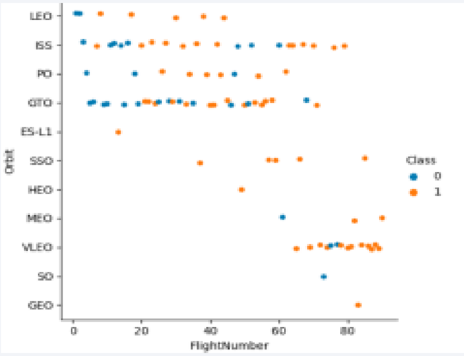
4.Append the list to the new class column

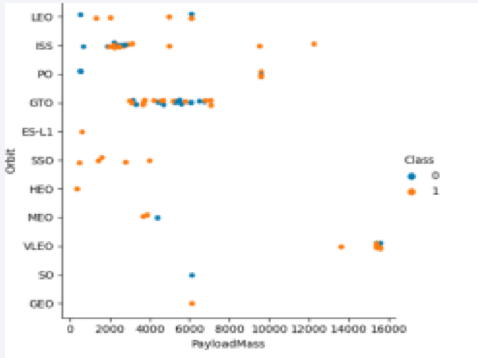
```
df['Class']=landing_class
df[['Class']].head(8)
```

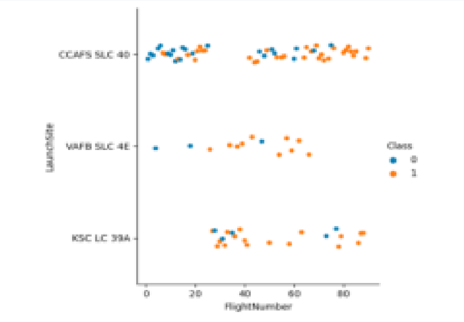
EDA with Data Visualization

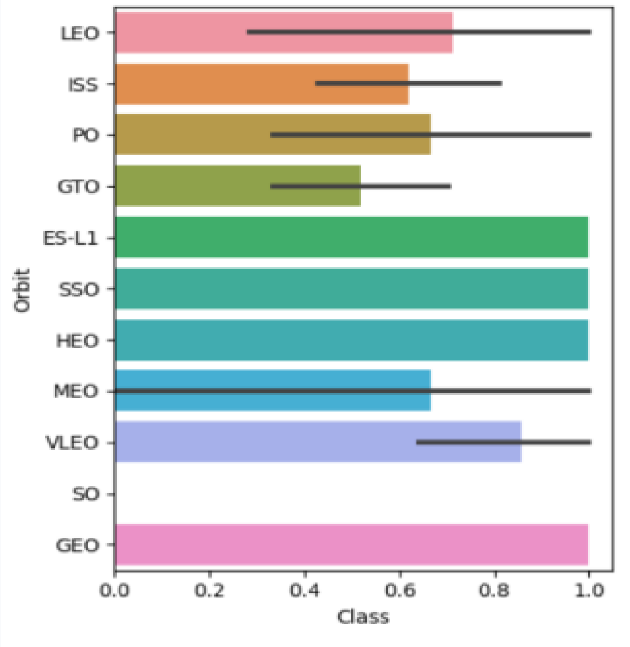


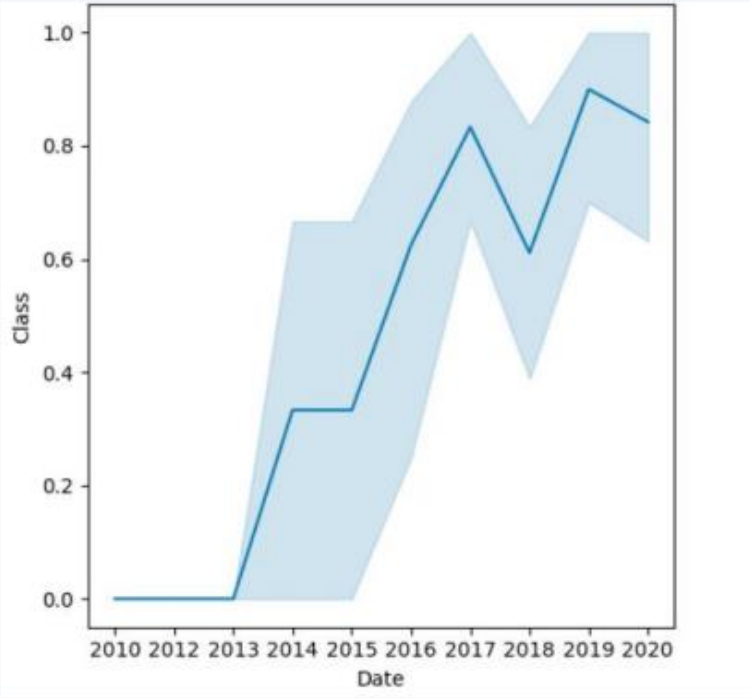












EDA with SQL

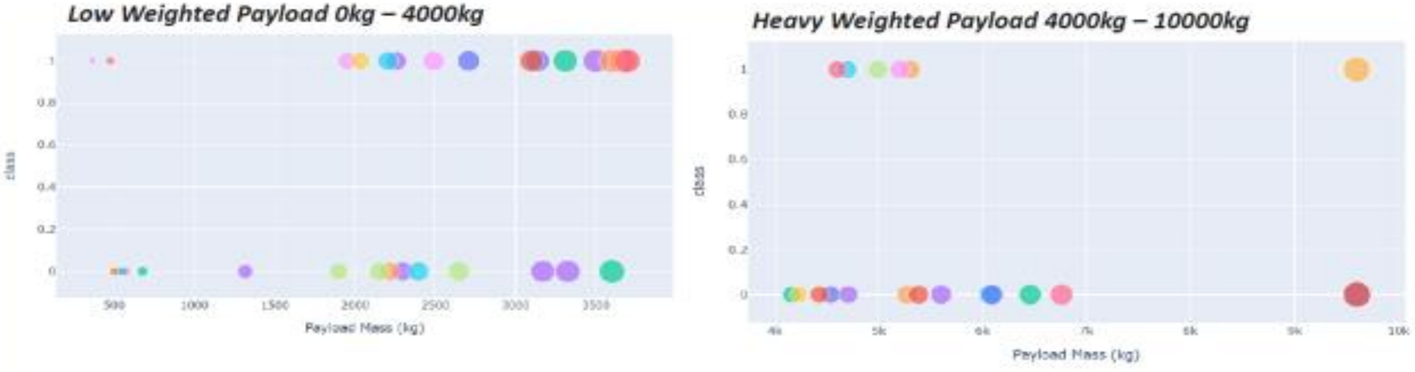
SQL queries performed:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- 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 where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad 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 records which will display the month names, successful landing_outcomes in ground pad booster
- Versions, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 indescending order

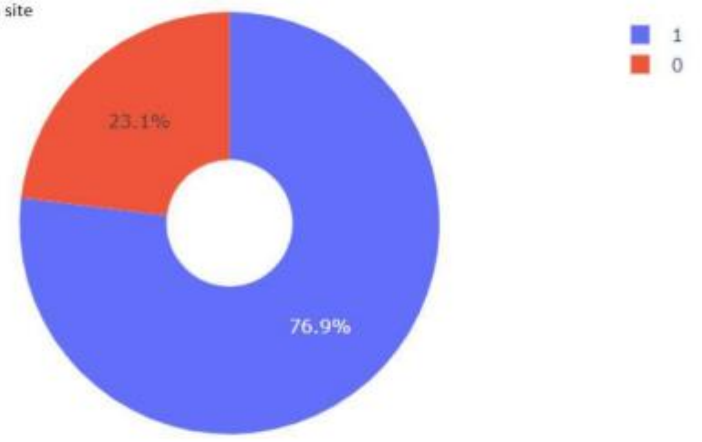
Build an Interactive Map with Folium

Launch sites and distance to the close city in the map

Build a Dashboard with Plotly Dash

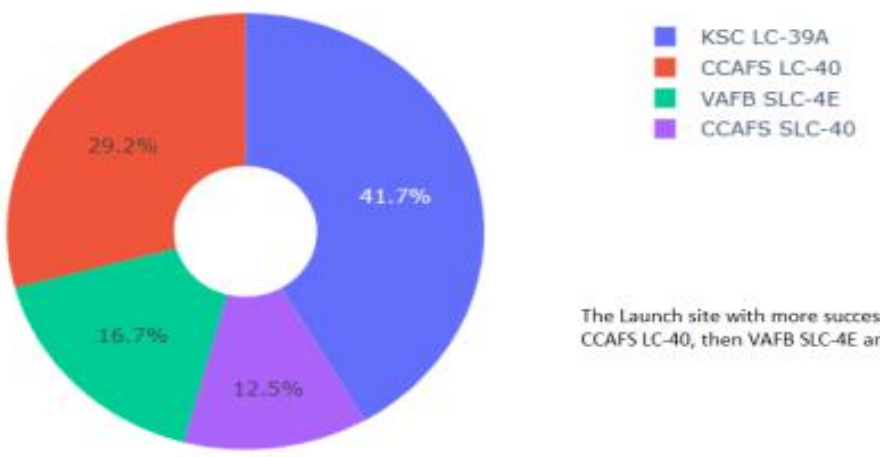


Low weighted payloads have a better success rate



KSC LC39A is the best launch site with a success rate of 76.9%

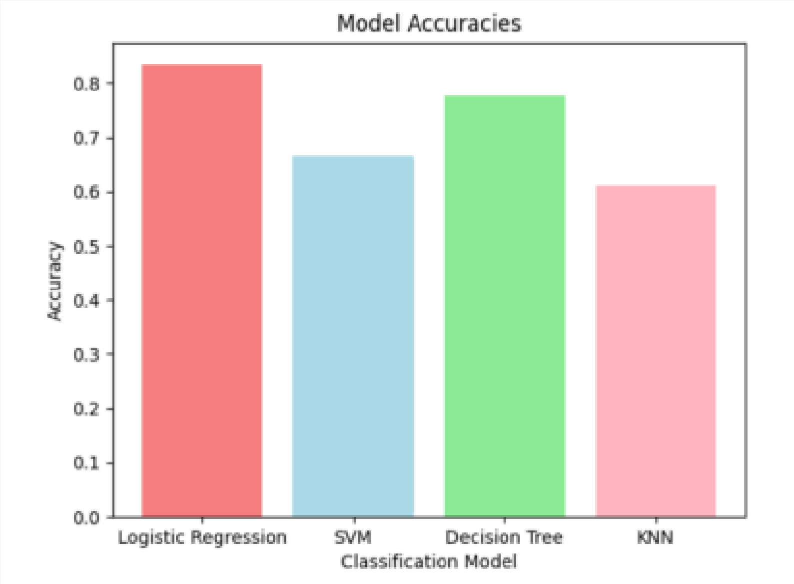
Total Success Launches By all sites



The Launch site with more success are KSC LC-39A following by CCAFS LC-40, then VAFB SLC-4E and finally CCAFS SLC-40

Predictive Analysis (Classification)

- Preprocess data
- Split data into train and test sets
- Try different models (Logistic Regression, K-NN, SVM, Decision Trees)
- Tune Hyperparameters for each model
- Evaluate each model with metrics like accuracy and confusion matrix
- Compare results and select the best performing model based on the highest accuracy and lowest confusion



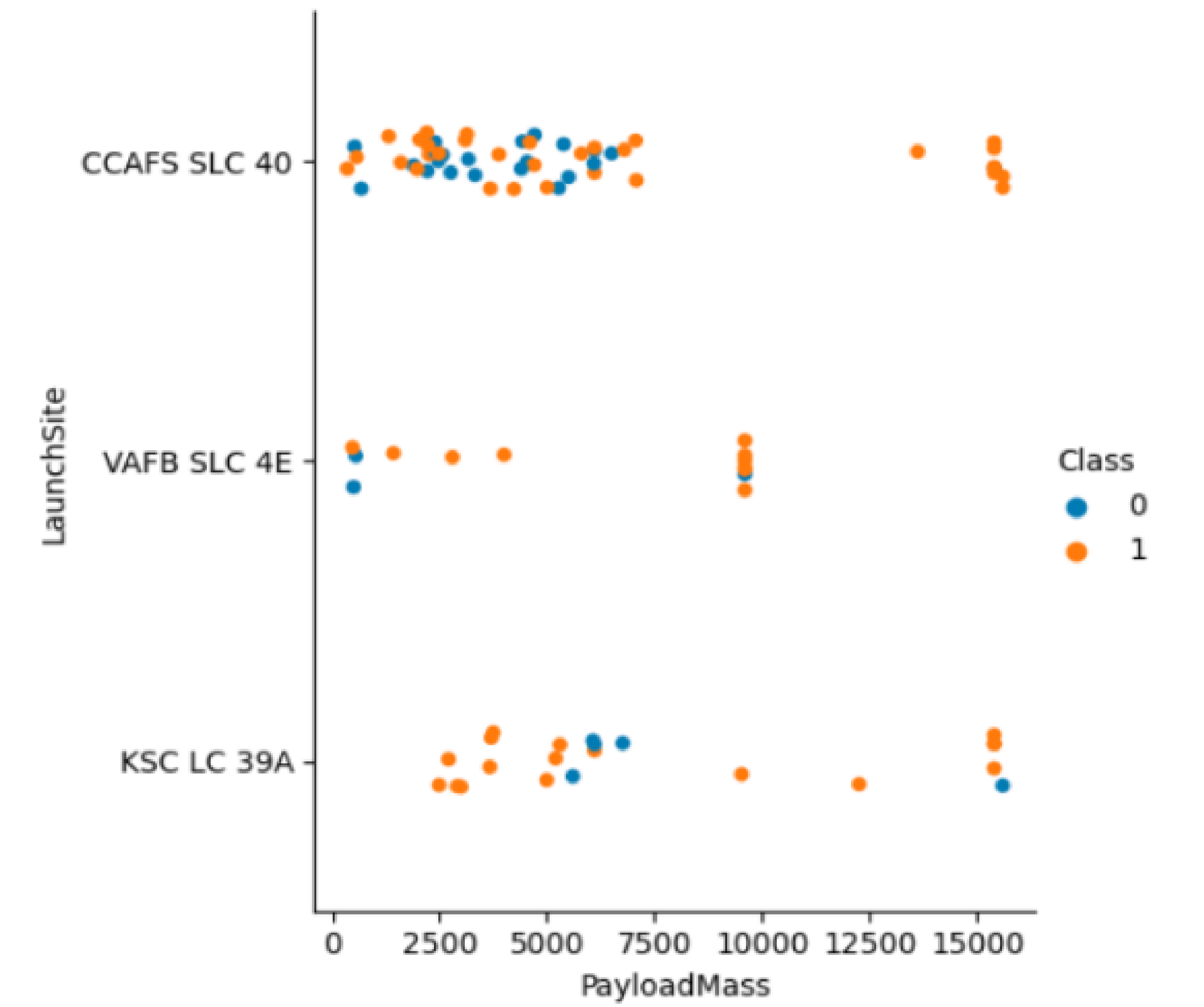
Results

- Lightweight payloads tend to perform better than heavier payloads.
- SpaceX's launch success rate has been directly proportional to the time in years.
- Kennedy Space Center's Launch Complex 39A has had the most successful launches out of all launch sites.
- Launches targeting specific orbits such as Geostationary, High Earth, SunSynchronous, and Earth-Sun L1 have had the highest success rates.
- Logistic regression is typically the most accurate method for predicting launch success.

Section 2

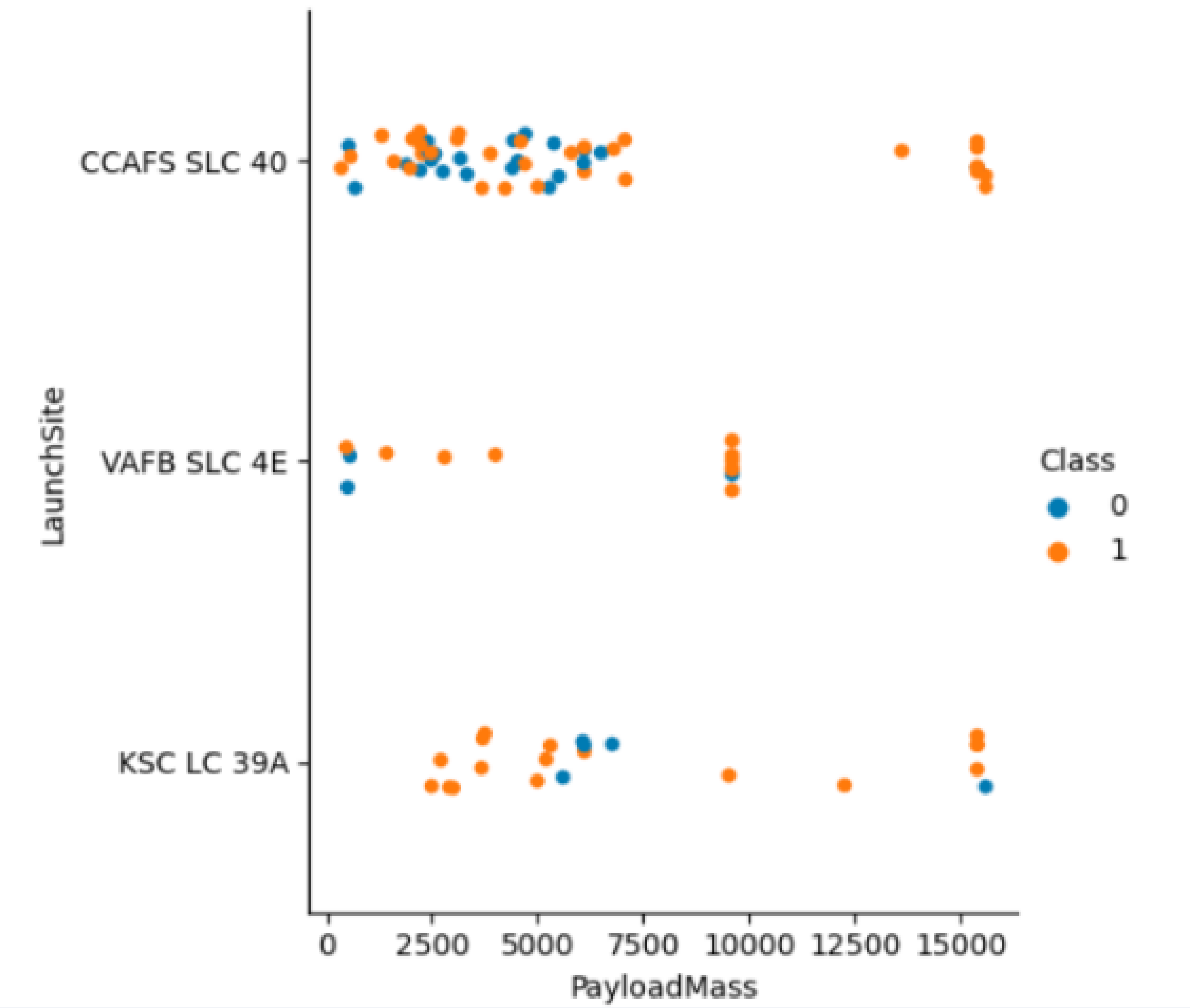
Insights drawn from EDA

Flight Number vs. Launch Site



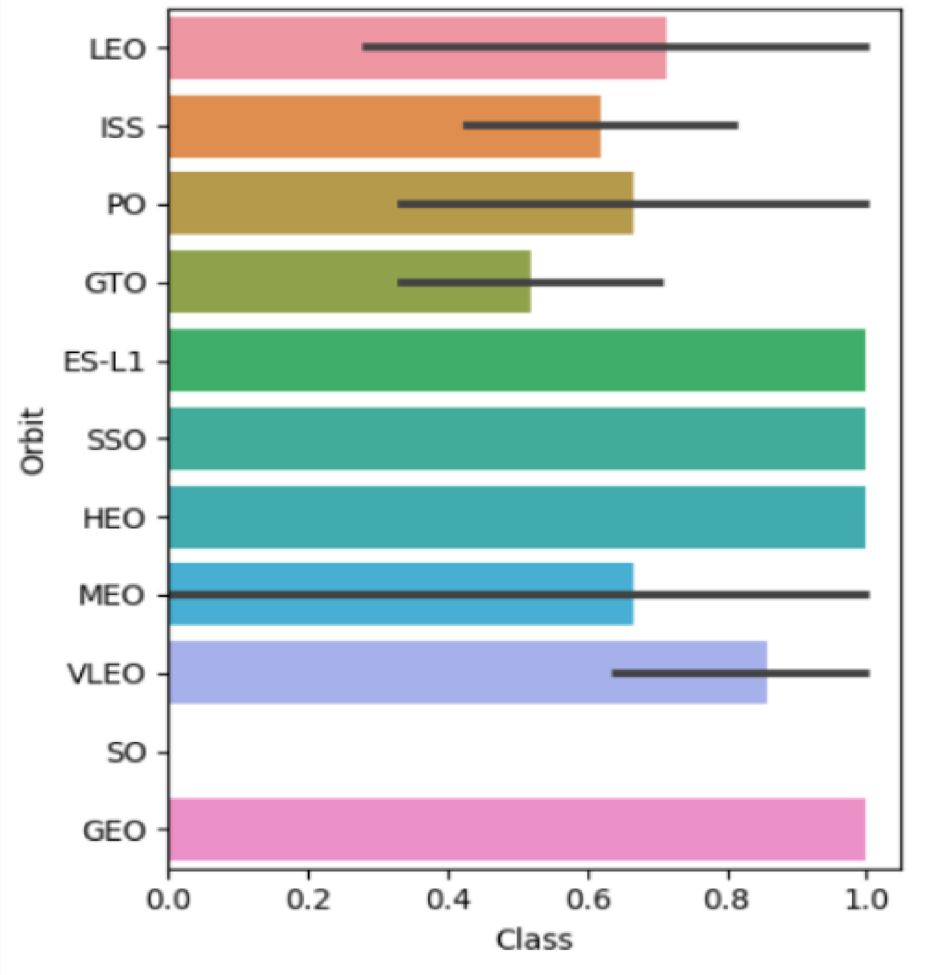
A majority of payloads with lower mass have been launched from the Cape Canaveral Air Force Station's Space Launch Complex 40 (CCAFS SLC 40)

Payload vs. Launch Site



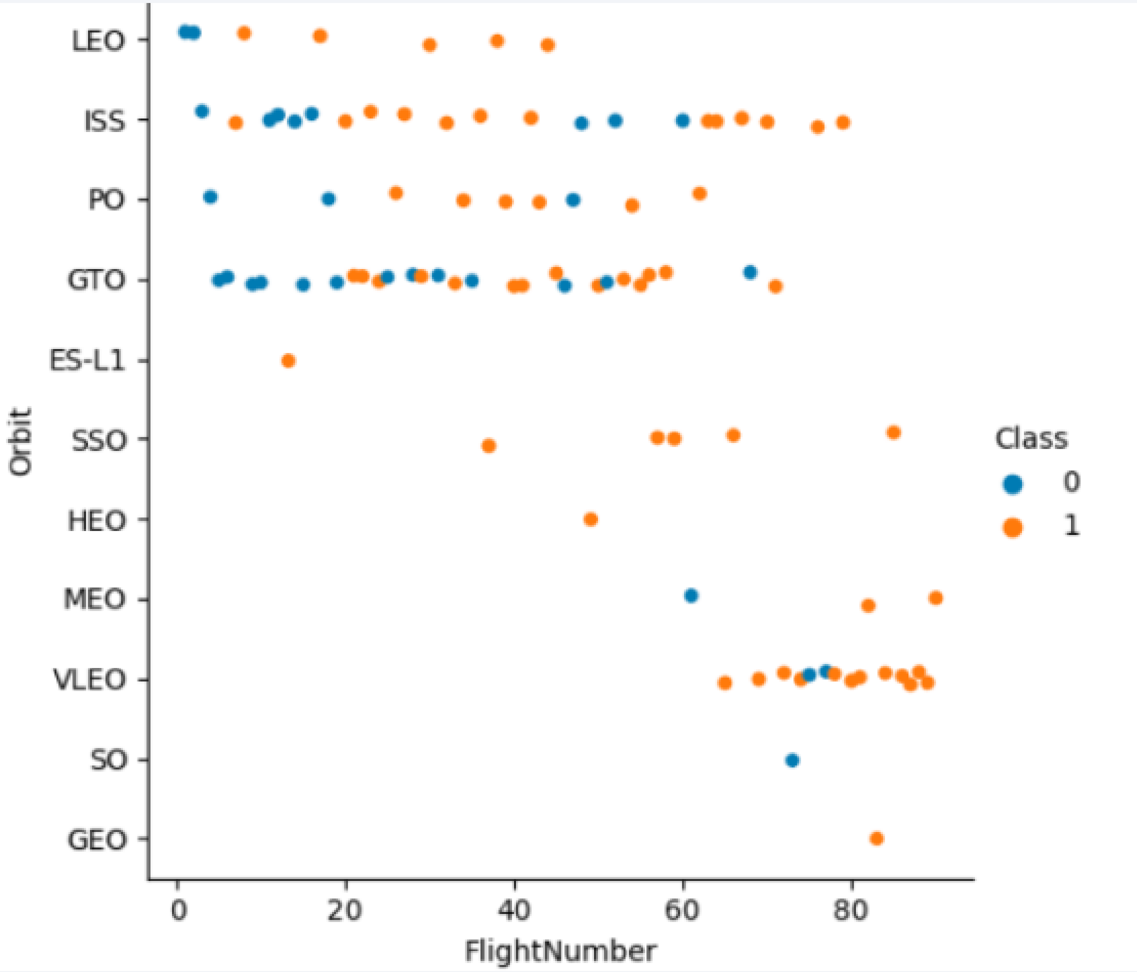
A majority of payloads with lower mass have been launched from the Cape Canaveral Air Force Station's Space Launch Complex 40 (CCAFS SLC 40)

Success Rate vs. Orbit Type



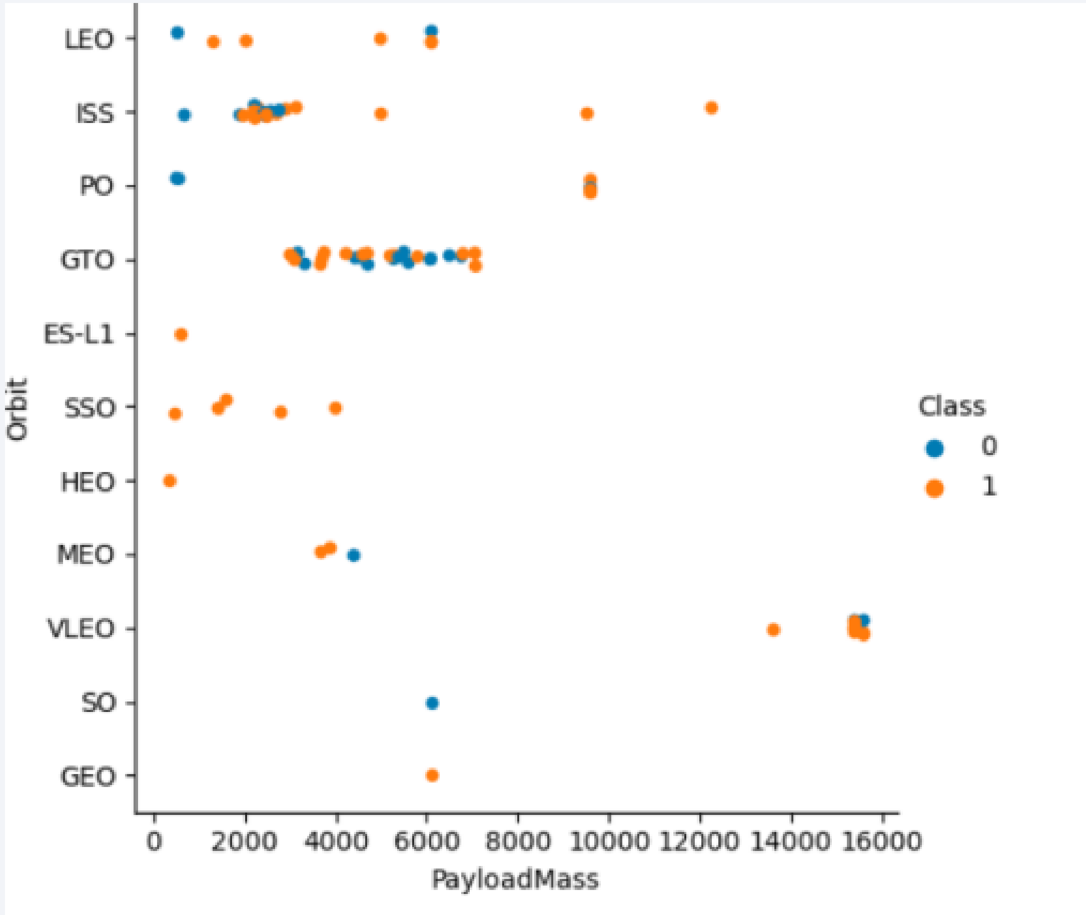
The orbits of Earth-Sun L1 (ES L1), Geostationary Earth Orbit (GEO), High Earth Orbit (HEO) and Sun-Synchronous Orbit (SSO) have demonstrated a high success rate in terms of launches.

Flight Number vs. Orbit Type



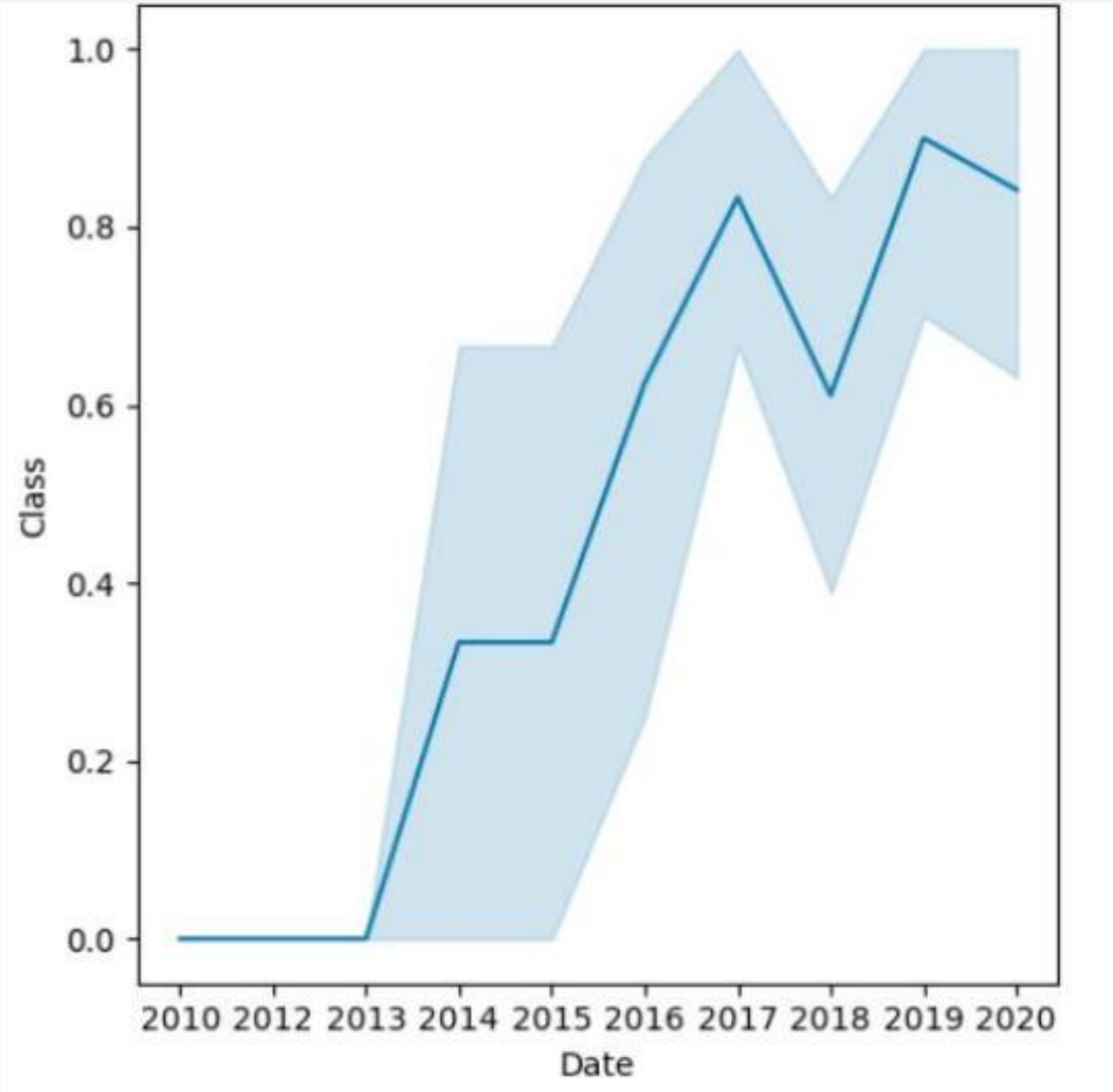
There is a trend of increasing use of VLEO (Very Low Earth Orbit) launches in recent flight numbers.

Payload vs. Orbit Type



The trend of VLEO launches has been increasing in recent flight numbers and these launches tend to carry more payload and have higher success rates.

Launch Success Yearly Trend



The launch success rate has seen a significant increase since 2013. This can be attributed to a number of factors such as advancements in technology, and the lessons learned from previous launches. The improvement in technology has allowed for more accurate and efficient launches, resulting in a higher success rate. Furthermore, the lessons learned from previous launches have been implemented to prevent similar failures from happening in the future. Since 2019, the launch success rate has stabilized, indicating that the industry has reached a level of maturity and efficiency. The increasing success rate is not only beneficial for the companies involved in the launches, but also for the payloads and the organizations that rely on them. This trend is likely to continue as the industry continues to evolve and improve.

All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTBL
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-8
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

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

Total Payload Mass

```
%sql select sum(payload_mass__kg_) from SPACEXTBL WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1o
Done.
```

1
45596

Average Payload Mass by F9 v1.1

```
%sql select avg(payload_mass__kg_) from SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lc
Done.
```

1
2928

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%';
```

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

MIN(Date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)'\
and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databa
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP BY mission_outcome
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databa
Done.
```

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

Boosters Carried Maximum Payload

```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL\
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l
Done.
```

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

2015 Launch Records

```
%sql select booster_version, launch_site from SPACEXTBL where landing__outcome = 'Failure (drone ship)' and year(DATE) = 2015
```

```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql select count(landing__outcome), landing__outcome from SPACEXTBL \
where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome\
order by count(landing__outcome) desc
```

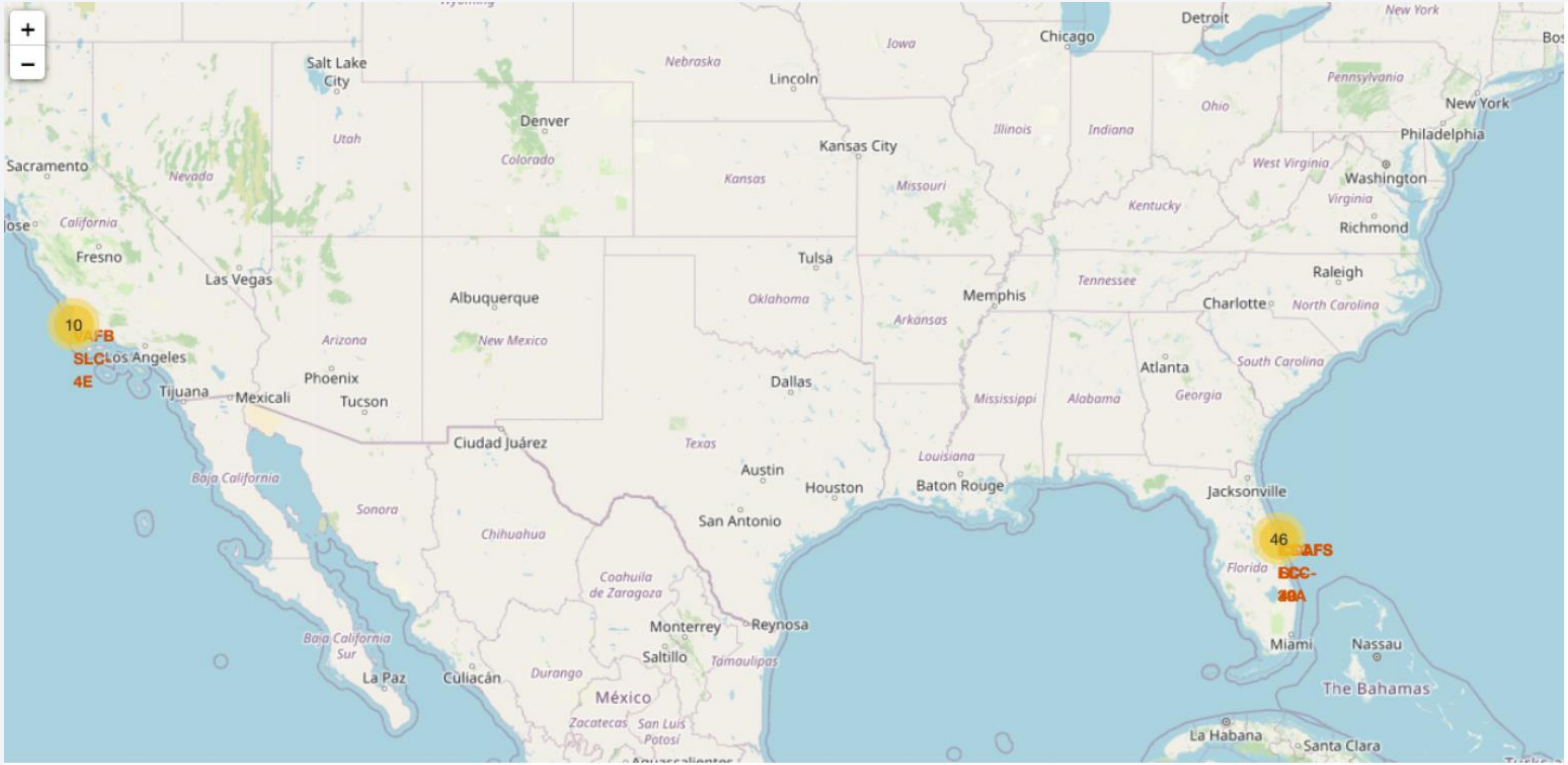
```
 * ibm_db_sa://gtc27297:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08k
Done.
```

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

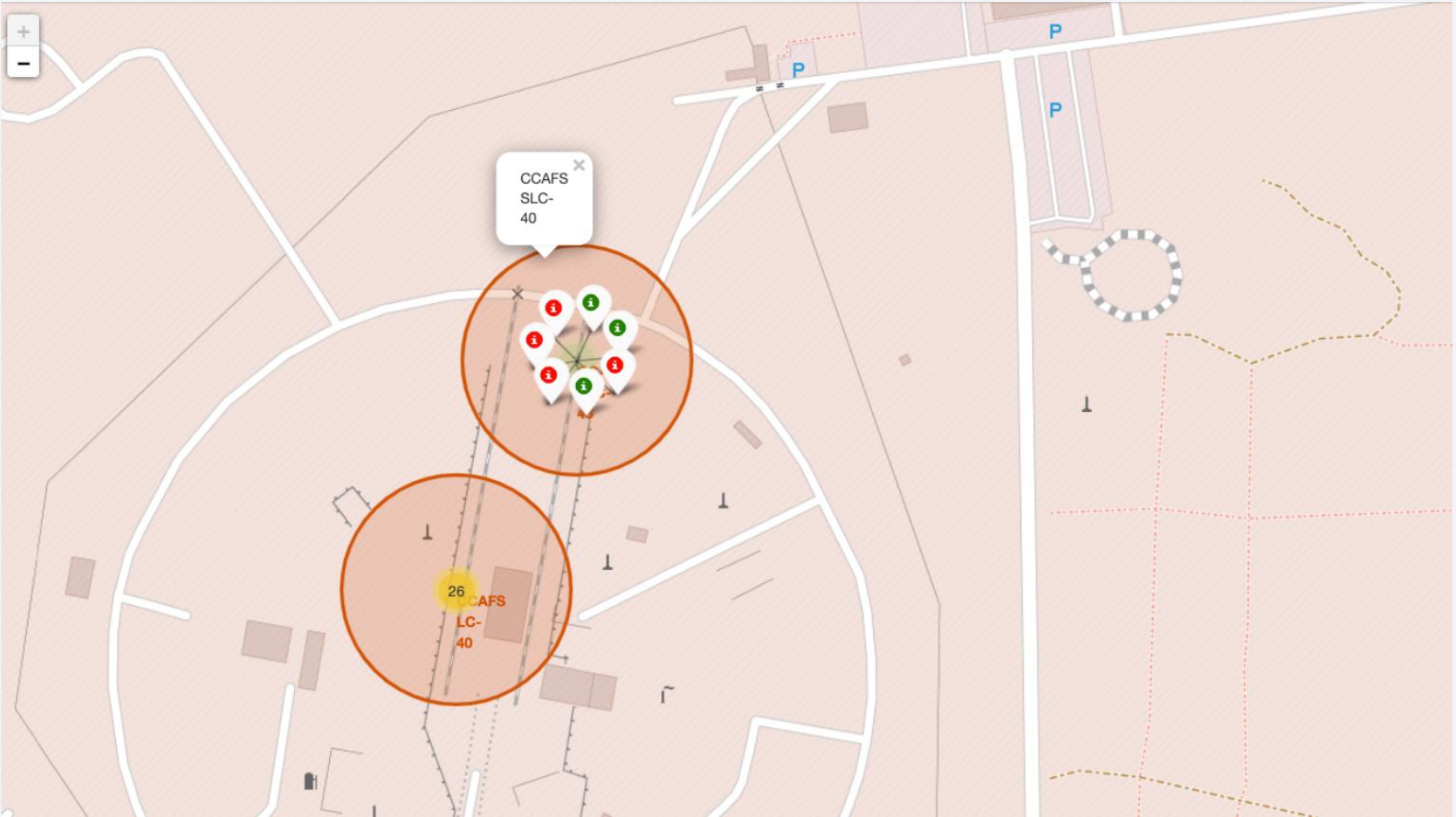
Section 3

Launch Sites Proximities Analysis

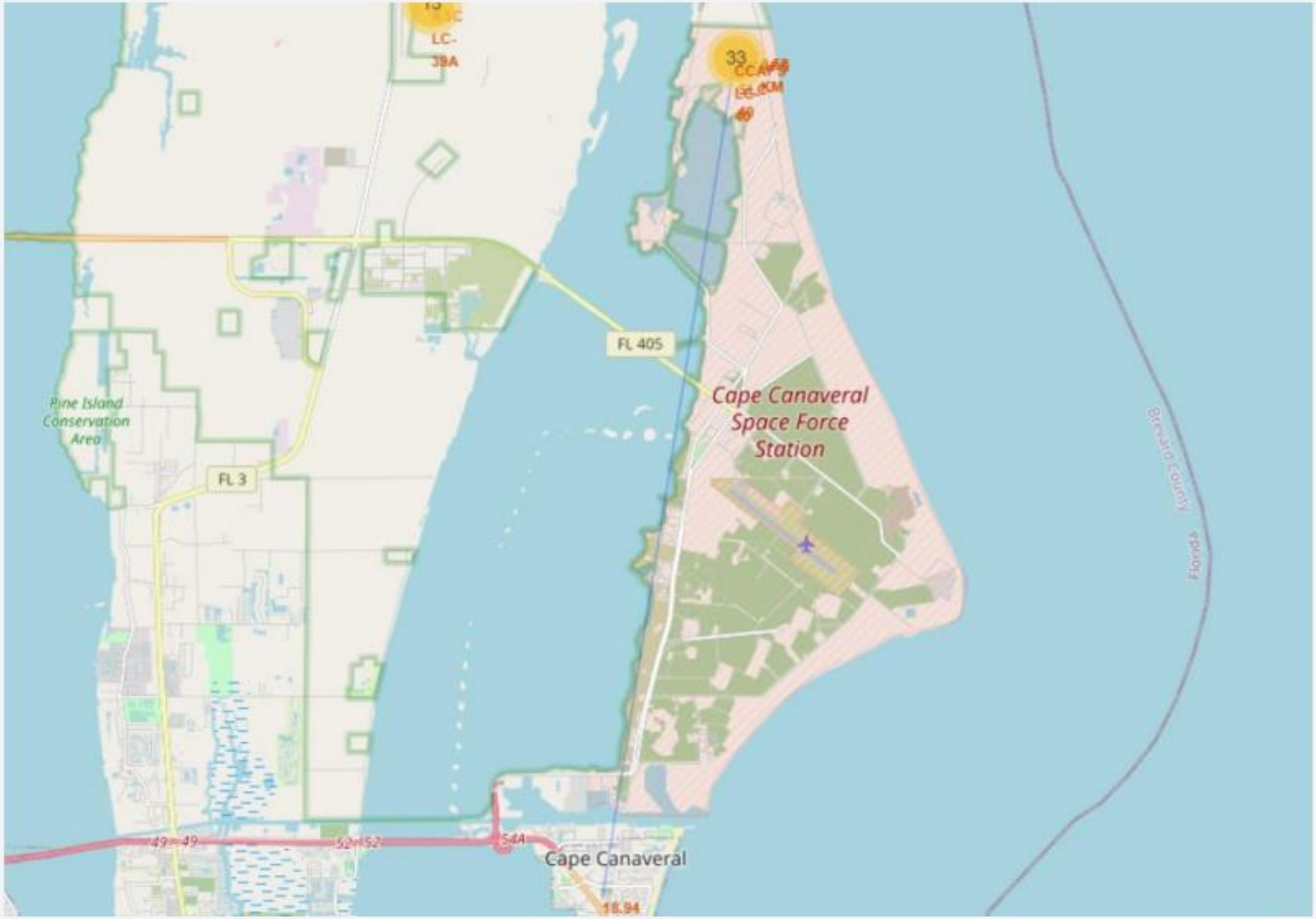
<Folium Map Screenshot 1>



<Folium Map Screenshot 2>



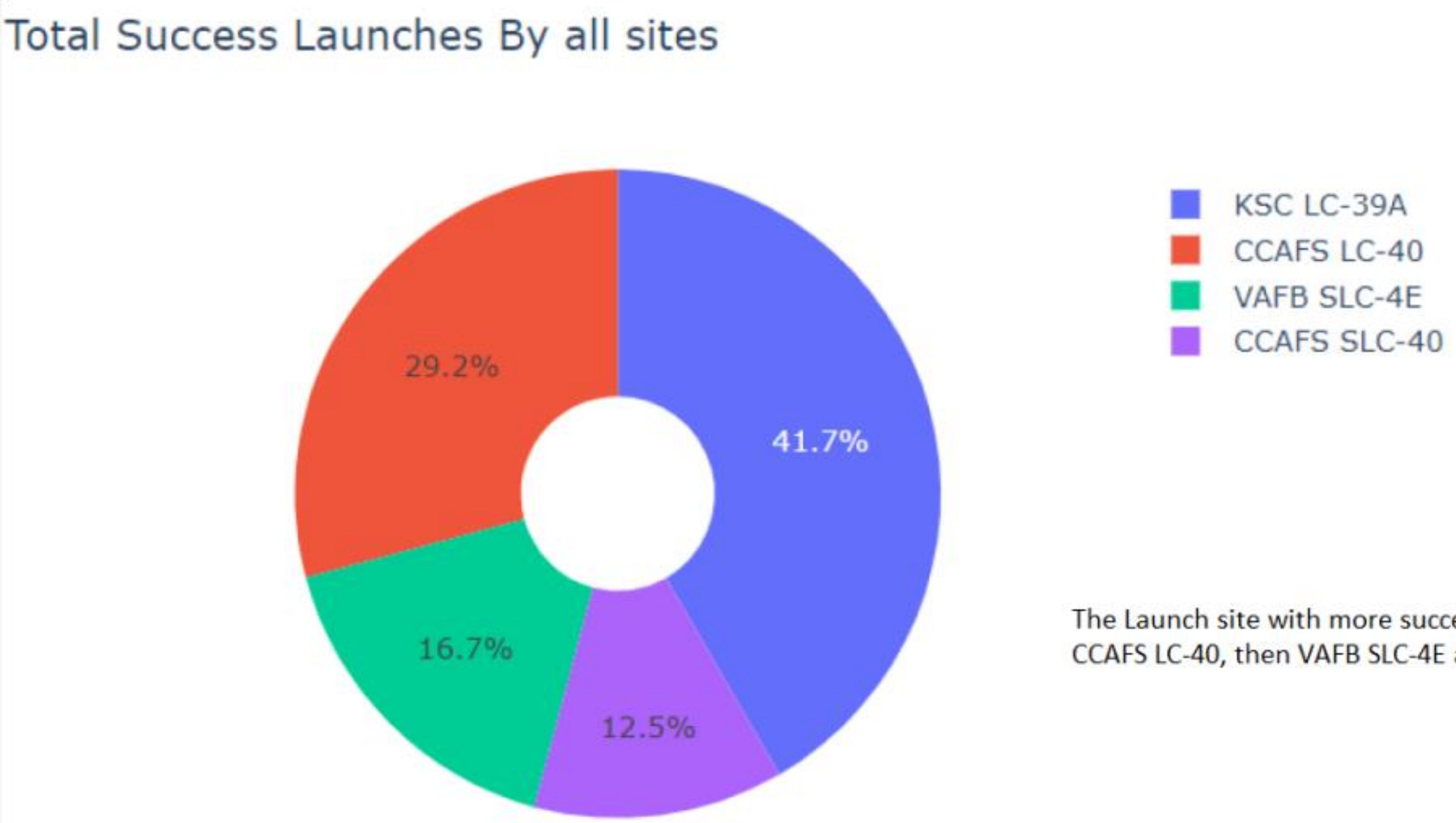
<Folium Map Screenshot 3>



Section 4

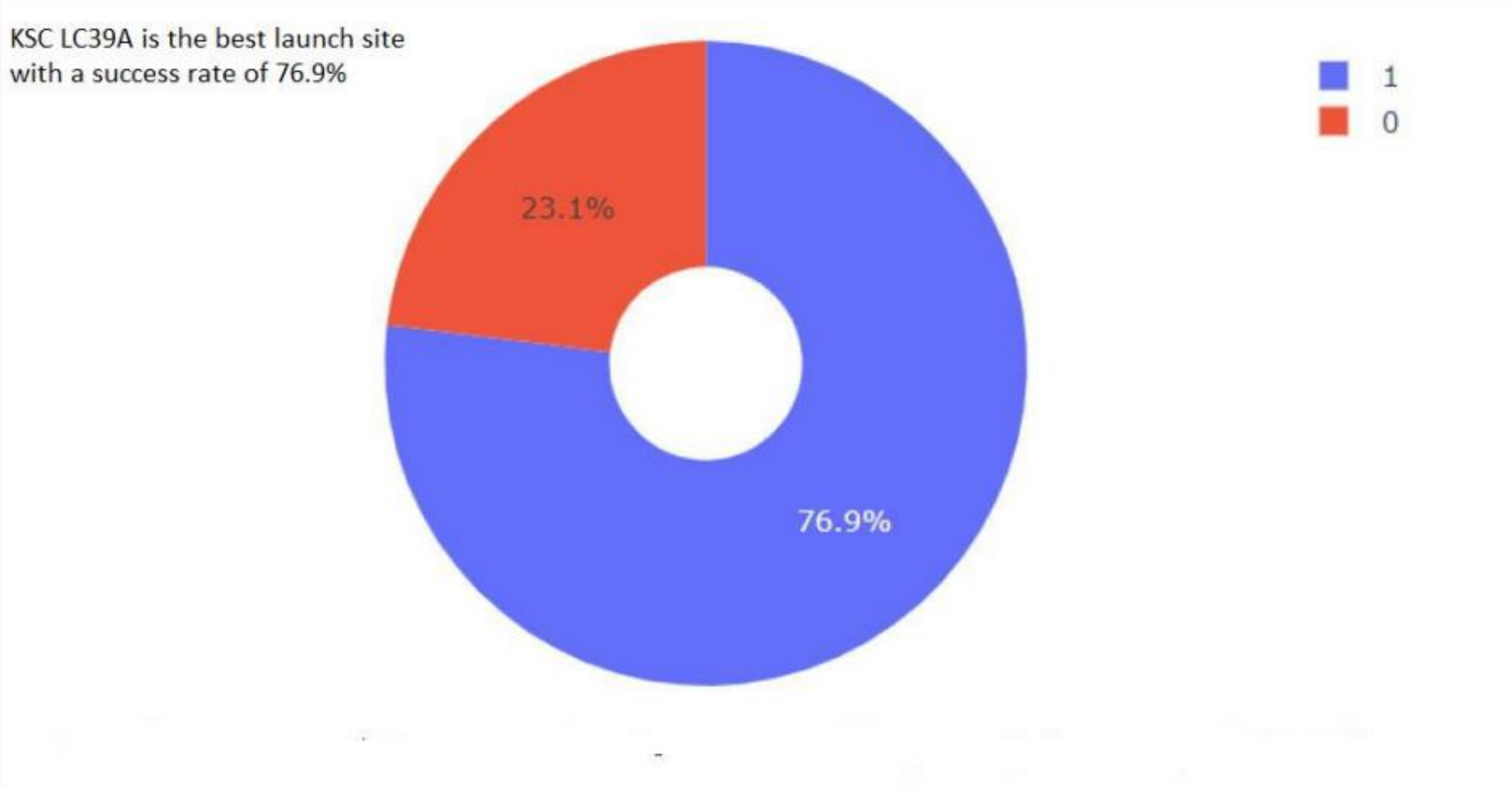
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

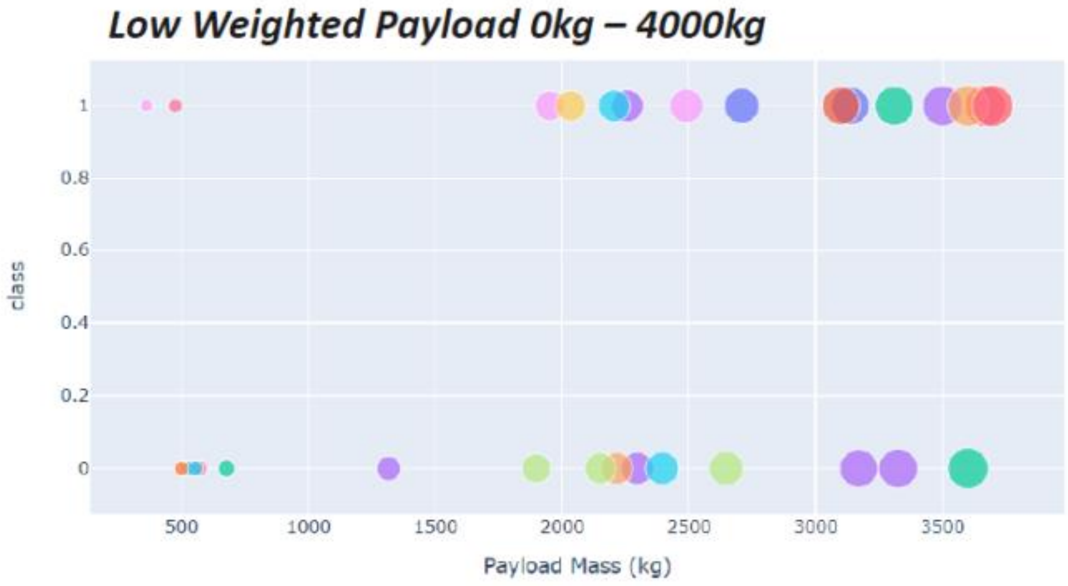


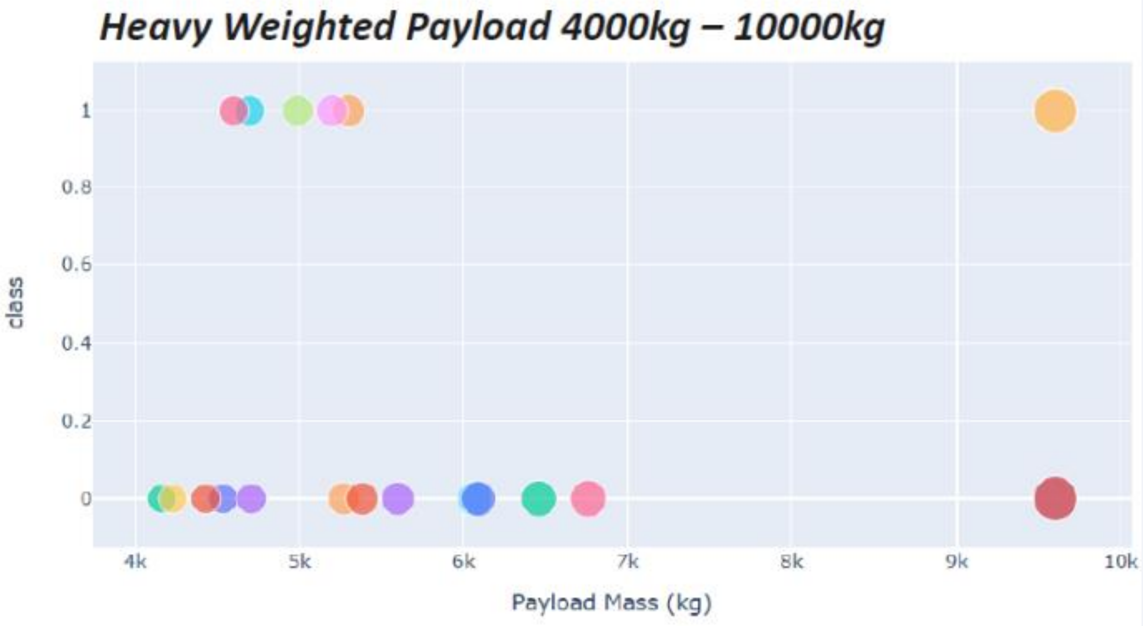
The Launch site with more success are KSC LC-39A following by CCAFS LC-40, then VAFB SLC-4E and finally CCAFS SLC-40

<Dashboard Screenshot 2>



<Dashboard Screenshot 3>



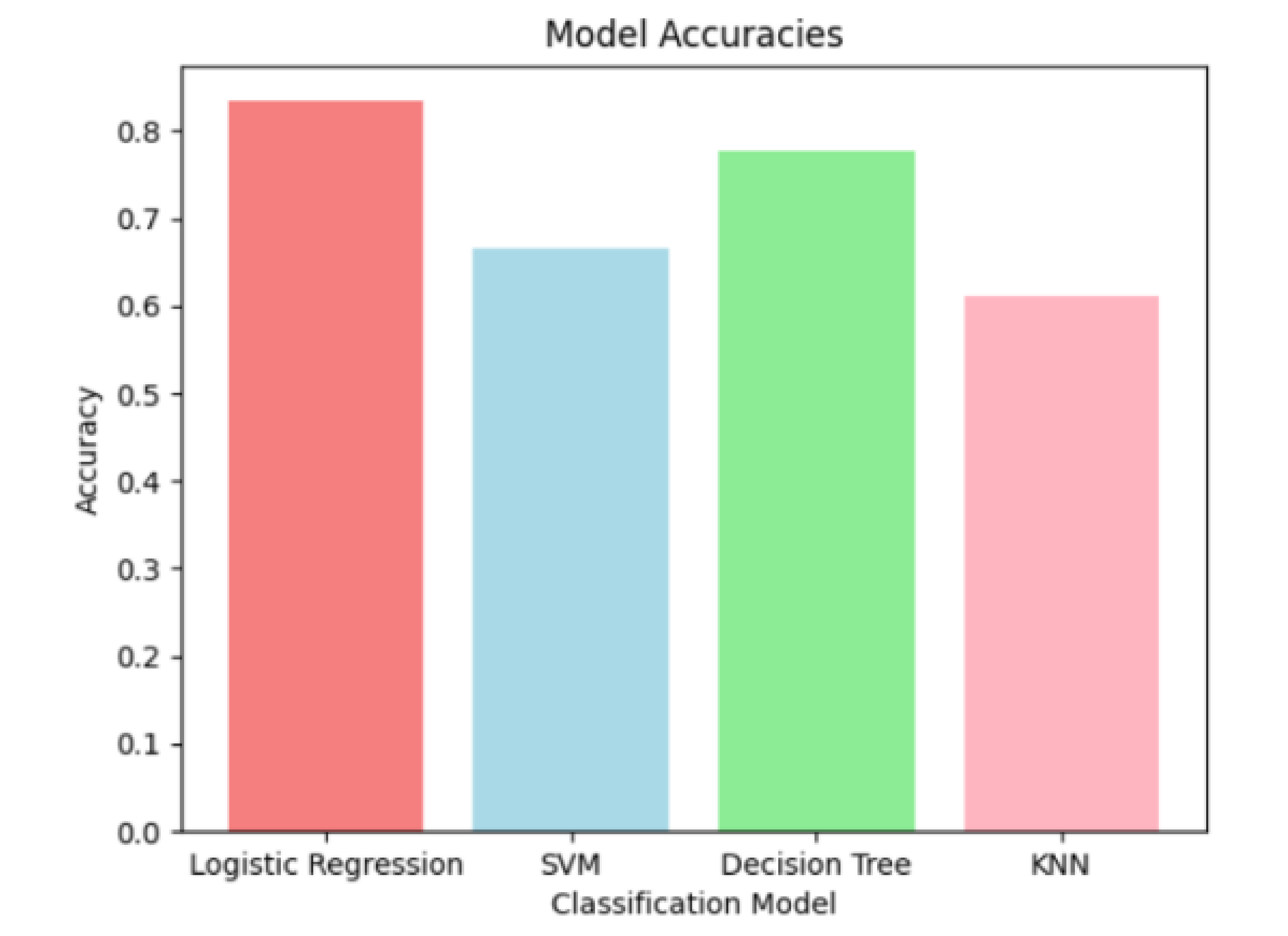


Low weighted payloads have a better success rate

Section 5

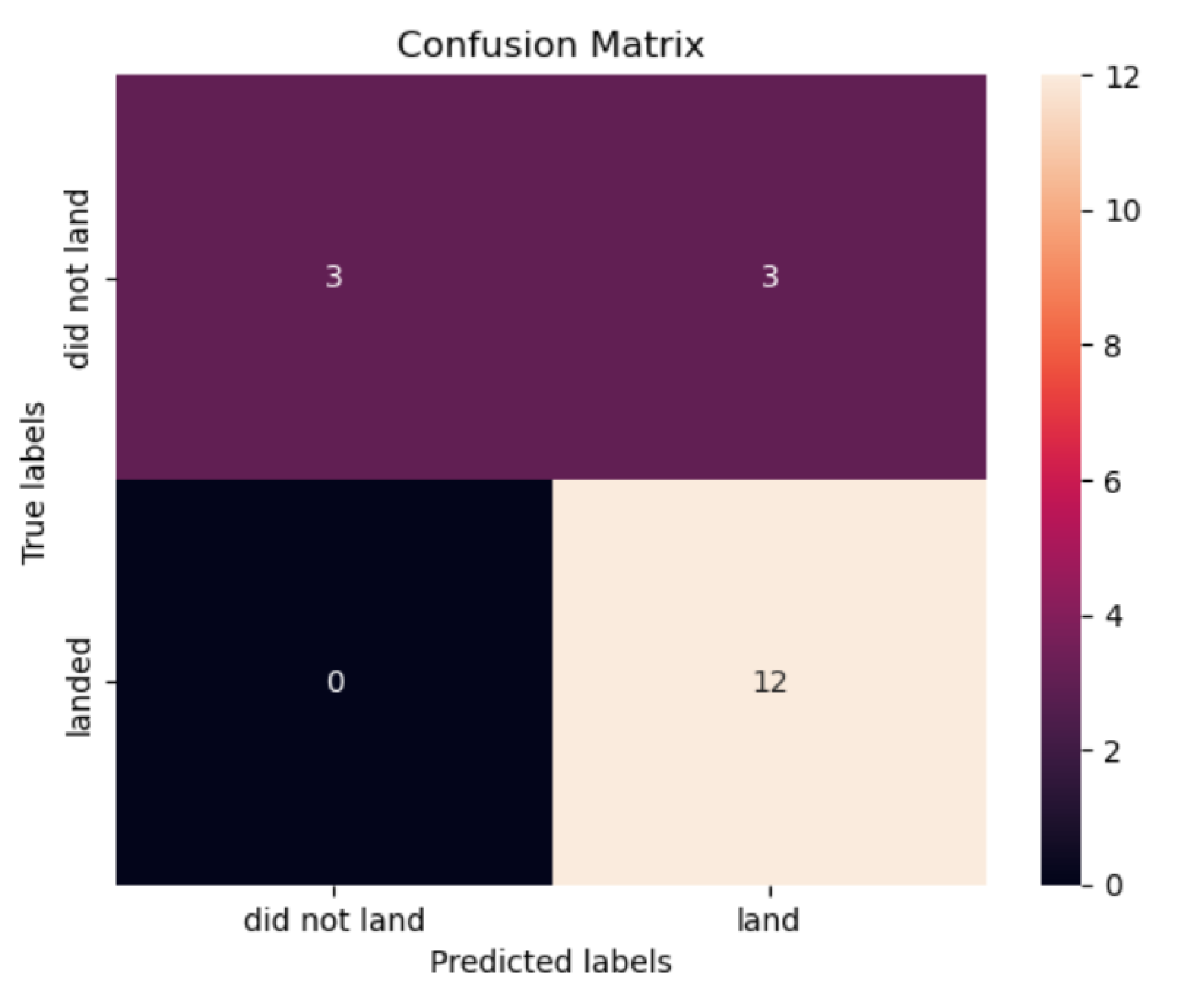
Predictive Analysis (Classification)

Classification Accuracy



The logistic regression is clearly the most accurate model in this case following the decisión tree, them the SVM and finally the K-NN algorithm

Confusion Matrix



In this case, the sensitivity is 0.8 means the model is correctly identifying 80% of the actual positive cases. Specificity is 1 means the model is correctly identifying 100% of the actual negative cases

Conclusions

- Low weighted payloads perform better than heavier payloads in terms of success rates for launches.
- The success rate for SpaceX launches is directly proportional to the time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO, HEO, SSO, and ES L1 have the best success rate.
- Logistic Regression models are the best method in terms of prediction accuracy for this dataset. It has been tested and compared with other methods such as Support Vector Machines, K-NN and Decision Trees, and it has shown to be more accurate.

Appendix

- All assets in IBM's Resource

Thank you!