IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shiraz Wadia
10/10/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection.
  - Data wrangling.
  - Exploratory data analysis (EDA) with data visualization
  - EDA with SQL.
  - Building interactive maps with Folium
  - Building a Dashboard using Plotly Dash.
  - Building machine learning models.
- Summary of all results
  - EDA results.
  - Interactive analytics.
  - Predictive analytics

Introduction

- **Project background and context**

  Falcon 9 is a reusable, two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of people and payloads into Earth orbit and beyond. Falcon 9 is the world's first orbital class reusable rocket. Reusability allows SpaceX to re-fly the most expensive parts of the rocket, which in turn drives down the cost of space access. According to official sources the first stage can be used for at least 100 flights with refurbishment in between flights. Much depends on the successful recovery of the first stage.

- **Problems that we need to find answers for:**
  - What factors determine the success/failure of recovery of first stage?
  - Can we predict if a landing will be successful or not?
  - How accurate is our prediction about the success of the landings?

Section 1

Methodology

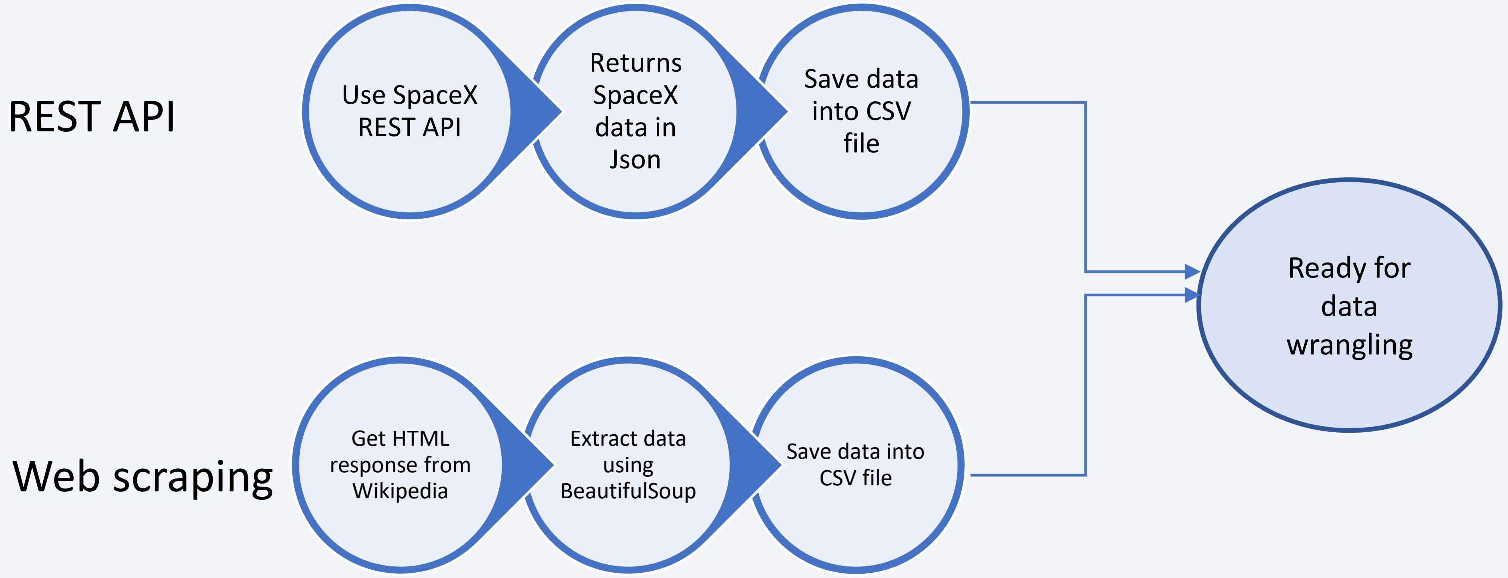
Methodology

Executive Summary

- Data collection methodology:
    - Collecting data through SpaceX REST API.
    - Collecting data through Web Scrapping from Wikipedia.
- Perform data wrangling
    - Cleaning of data by handling of Null values.
    - Removing irrelevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Using Linear Regression, K Nearest Neighbours, Support Vector Machines and Decision Tree models to find the best predictive and classification model

Data Collection

- Collection of datasets:
  - SpaceX launch data was extracted from SpaceX REST API.
  - This API has all the data about launches, type of SpaceX rockets used, payloads delivered, launch and landing specifications and landing outcomes.
  - The SpaceX REST API URL starts with api.spacexdata.com/v4/
  - Wikipedia is another data source for Falcon 9 launch data. BeautifulSoup was used for web scrapping from Wikipedia,
- Data collection process flowcharts



Data Collection – SpaceX API

- Data collection with SpaceX REST calls

1. Request and parse the SpaceX launch data using the GET request

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_
response = requests.get(static_json_url)
```

2. Convert response to .json file

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

3. Apply custom functions to get relevant data

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

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

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

4. Creating a dictionary to combine columns for dataframe

```
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. Creating a dataframe from dictionary

```
# Create a data from launch_dict
ld_df = pd.DataFrame.from_dict(launch_dict, orient = 'index')
ld_df = ld_df.transpose()
```

6. Dealing with missing data

```
# Calculate the mean value of PayloadMass column
PLMmean = data_falcon9['PayloadMass'].mean()
print(PLMmean)
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value = PLMmean, inplace = True)
data_falcon9.head()
```

7. Saving data in a .csv file

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

GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project.ipynb

Data Collection - Scraping

- Web scraping from Wikipedia

1. Request the Falcon9 Launch Wiki page from its URL and create a BeautifulSoup object from HTML response

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

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

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

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

```
column_names = []

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

3. Creating a dictionary

```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
# Let's initial the launch_dict with each
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']=[]
```

4. Creating a dataframe from dictionary

```
df=pd.DataFrame.from_dict(launch_dict, orient = 'index')
df = df.transpose()
```

5. Converting dataframe to .csv file

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

GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_Webscrapping.ipynb

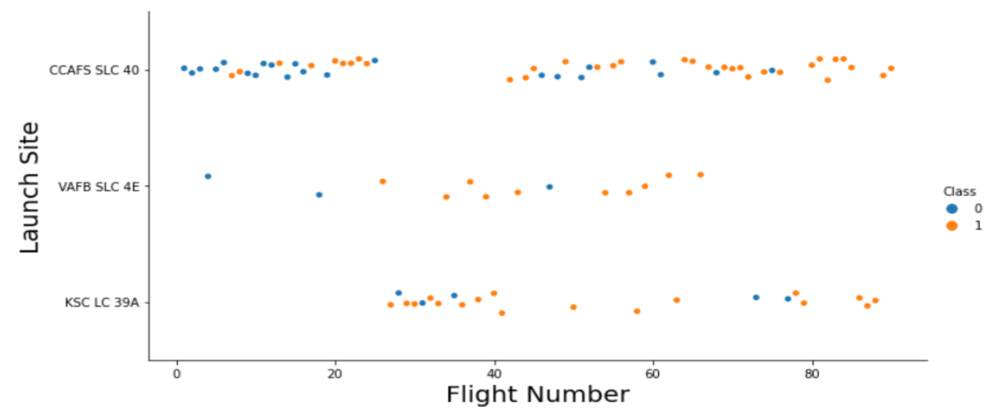
Data Wrangling

Exploratory Data Analysis Process Flow

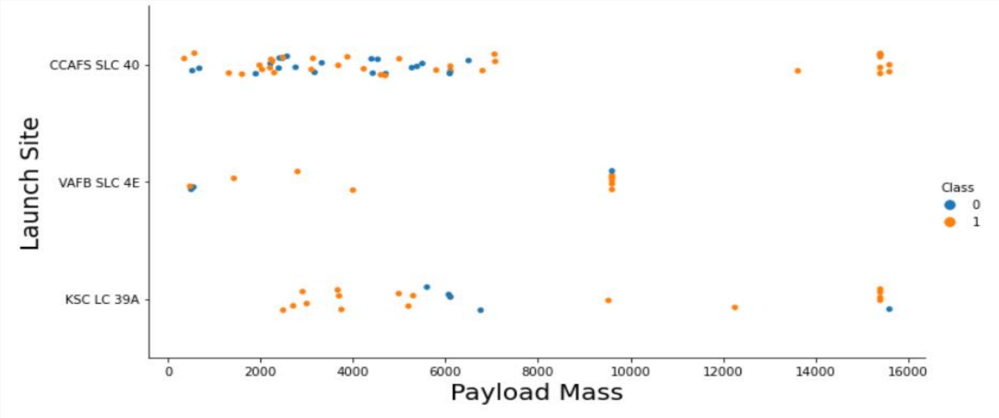
GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_DataWrangling.ipynb

EDA with Data Visualization

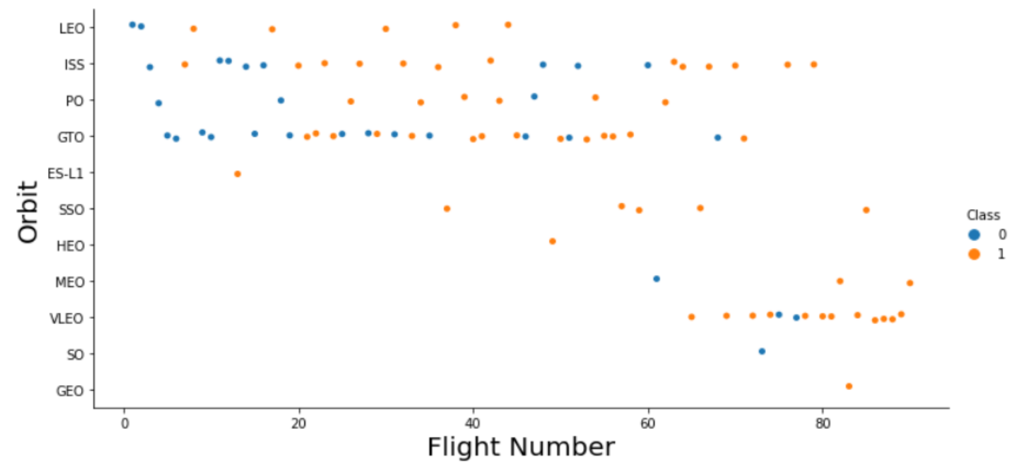
Relationship between Flight Number and Launch Site



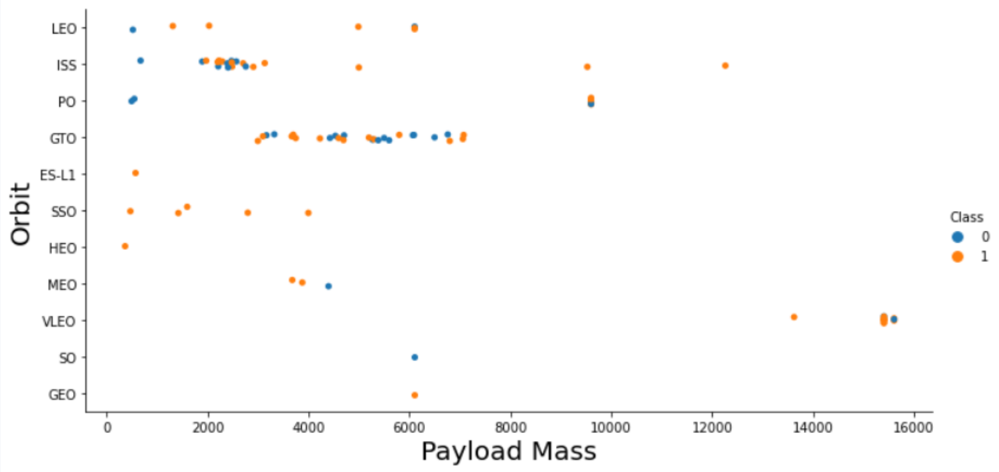
Relationship between Payload and Launch Site



Relationship between Flight Number and Orbit type

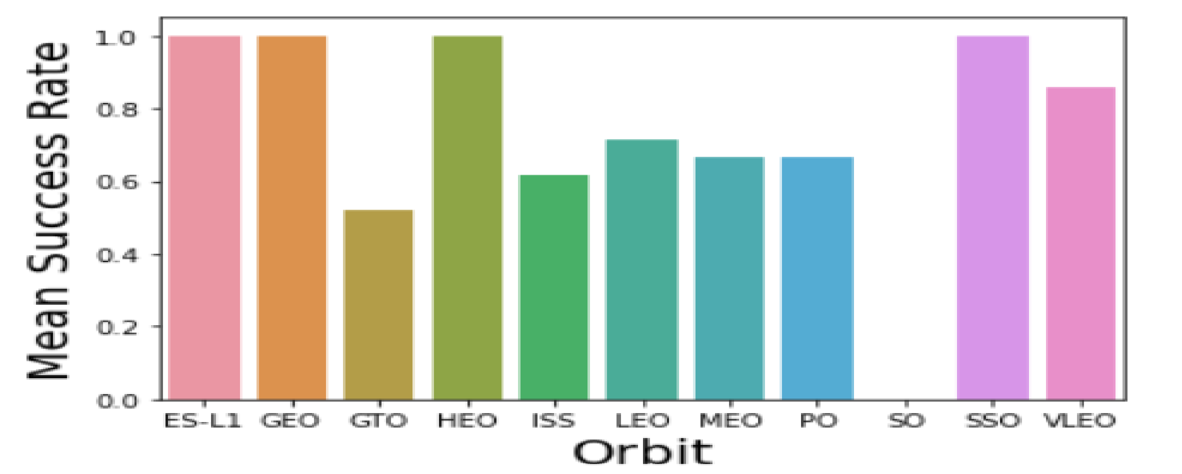


Relationship between Payload and Orbit type

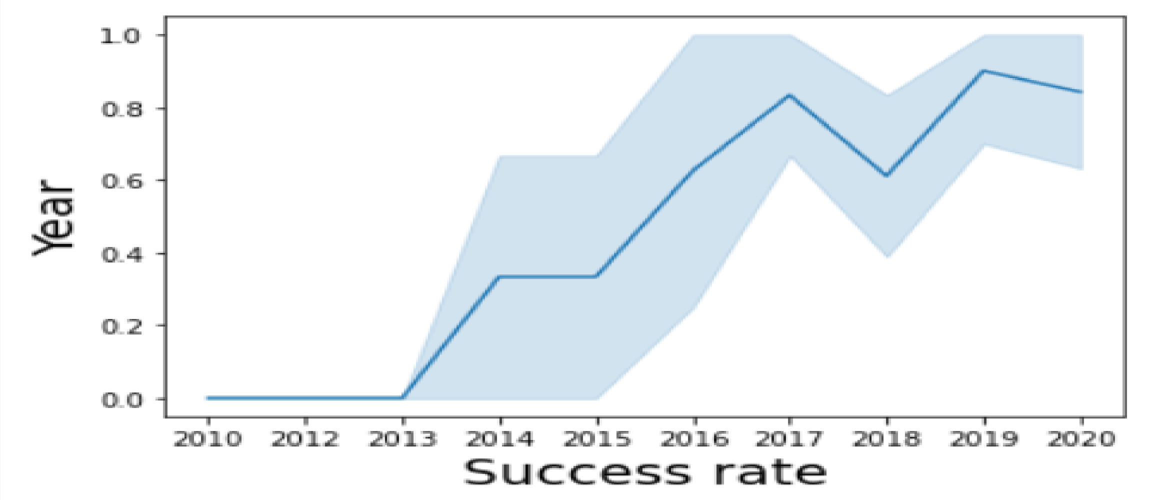


EDA with Data Visualization

Relationship between success rate of each orbit type



Launch success yearly trend



	Orbit	SuccessRate
0	ES-L1	1.000000
1	GEO	1.000000
2	GTO	0.518519
3	HEO	1.000000
4	ISS	0.619048
5	LEO	0.714286
6	MEO	0.666667
7	PO	0.666667
8	SO	0.000000
9	SSO	1.000000
10	VLEO	0.857143

EDA with Data Visualization

- Summary of plotted charts:
  - The scatter plot for relationship between flight Number and Launch Site shows that the maximum number of payloads were launched from the site CCAFS SLC 40.
  - Observing Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
  - The bar chart shows that launches to orbits ES-L1, GEO, HEO, SSO have a 100% success rate. The launch to GTO have been least successful.
  - Scatter plot for the relationship between Flight Number and Orbit type shows that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
  - The line plot for yearly success trend shows that the success rate since 2013 kept increasing till 2020

GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_EDA%20with%20Visualization%20lab.ipynb

EDA with SQL

SQL queries performed in the project:

- 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 first 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. Use a subquery
- 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 in descending order.

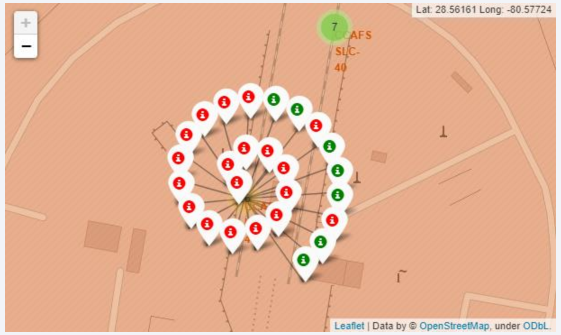
GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_ExploDataAnalysis.ipynb

Build an Interactive Map with Folium

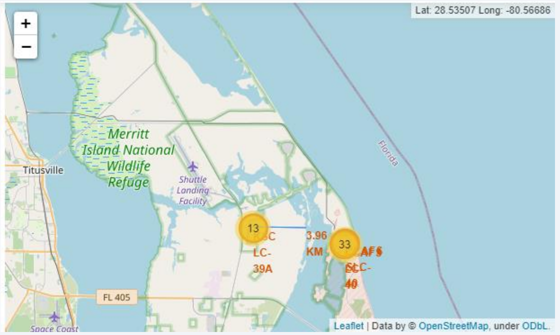
The following maps were plotted using Folium:

- Marked all launch sites on a map based on their co-ordinates.
- Creating and adding folium.Circle and folium.Marker for each launch site on the site map so that the sites are clearly distinguishable. The map can be explored using zoom in/out feature. This shows that the launch sites are located to within 29 degrees of Equator and in close proximity to the coastline.
- Marking the successful and failed launches for each site,
- Calculating the distances between a launch site to its proximities such as roads, coastline, railway etc. This also shows that the launch sites are at a safe distance from human settlements to avoid any loss or damages due to unforeseen accidents.

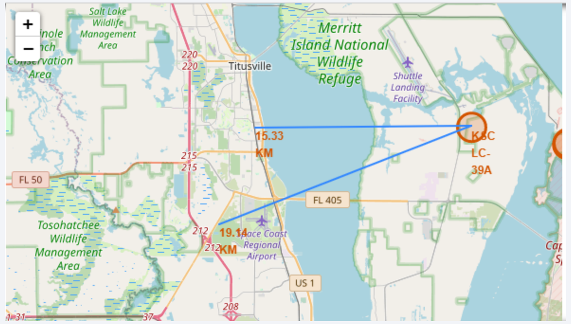
Map of launch sites



Map showing success/failures



Map of distance between launch site and coast



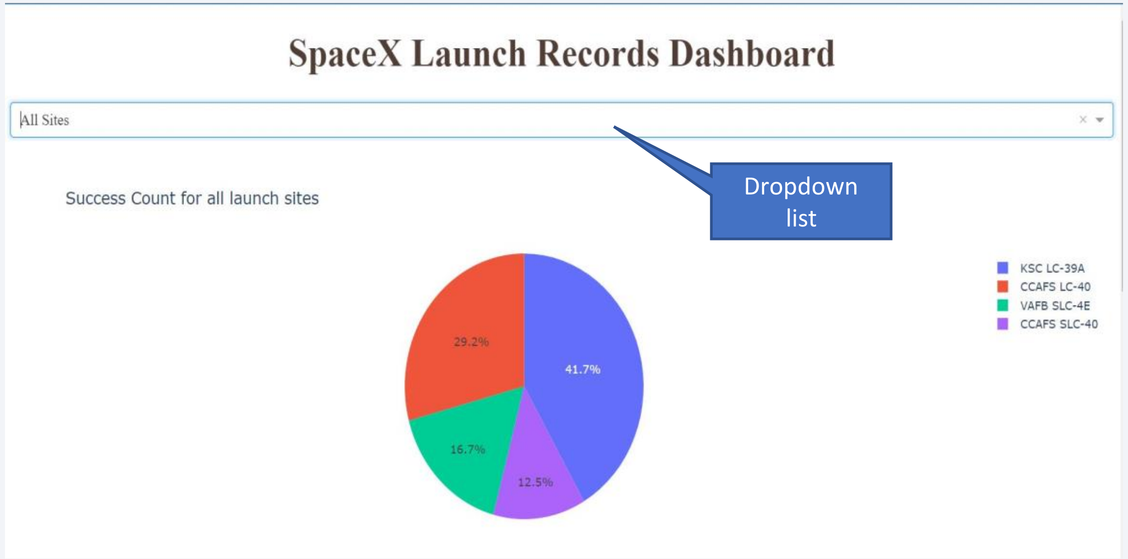
Map of distance: launch, road, rail and coast

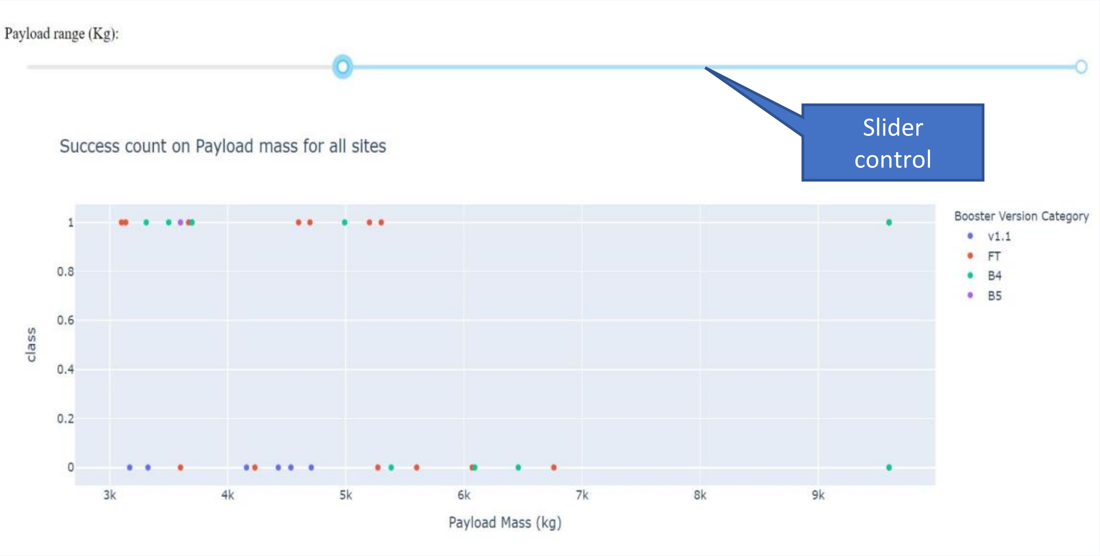
GITHUB URL:
https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Assignment_Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Following plots and graphs were added to the dashboard:

- Dropdown list containing the launch sites, to select the individual or all launch sites for displaying the success rate of the launch/recovery.
- Pie charts to display the success of launches for the different or all sites.
- A slider control to select the payload mass.
- Scatter plots of the payloads carried by the launches.





GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/FinalProject_Build%20an%20Interactive%20Dashboard_%20Ploty.ipynb

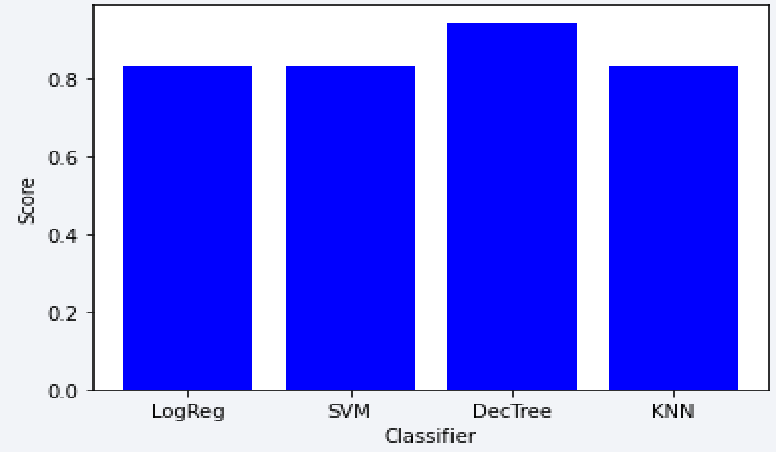
Predictive Analysis (Classification)

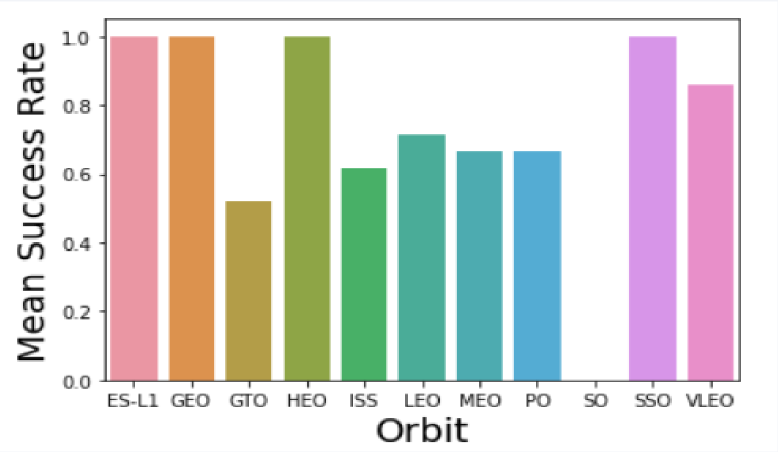
Performing Predictive Analysis – Process Flow:

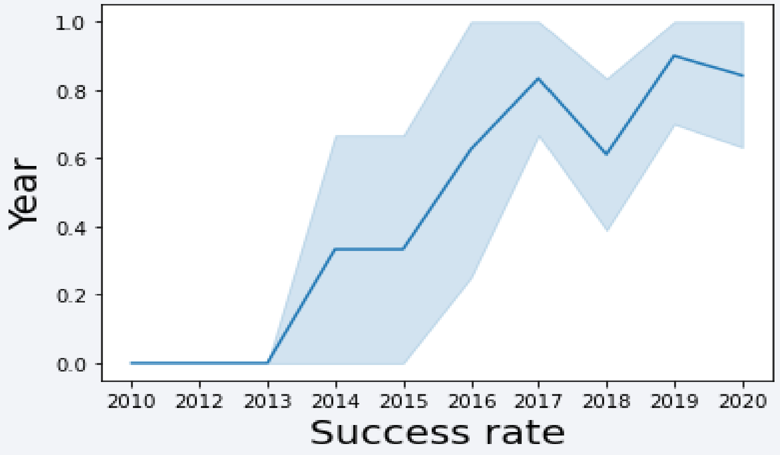
GITHUB URL: https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Assignment_Machine%20Learning%20Prediction.ipynb

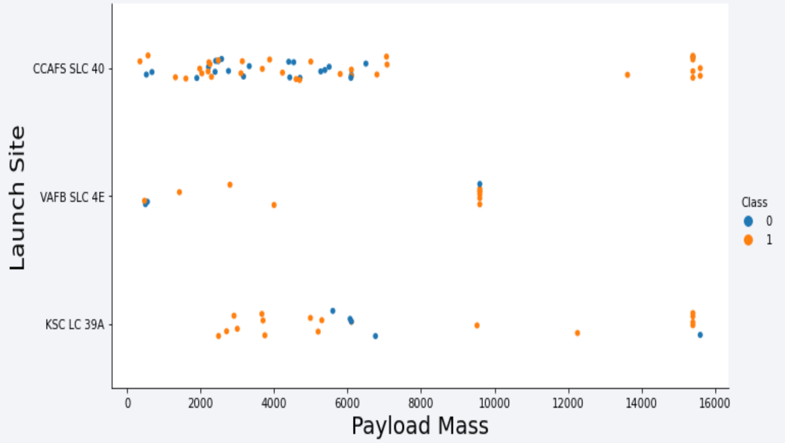
Results

- The SVM, KNN and Logistic Regression models deliver almost same results in terms of prediction accuracy for this dataset.
- However, the Tree model has the highest score among all the models.
- Success rate of launches has increased significantly from 2013 onwards.
- Launches to GEO, HEO, SSO and ES-L1 has the highest success rates.
- Launch of low payload mass has better chance of success.
- KSC LC 39A had the most successful launches from all the sites.







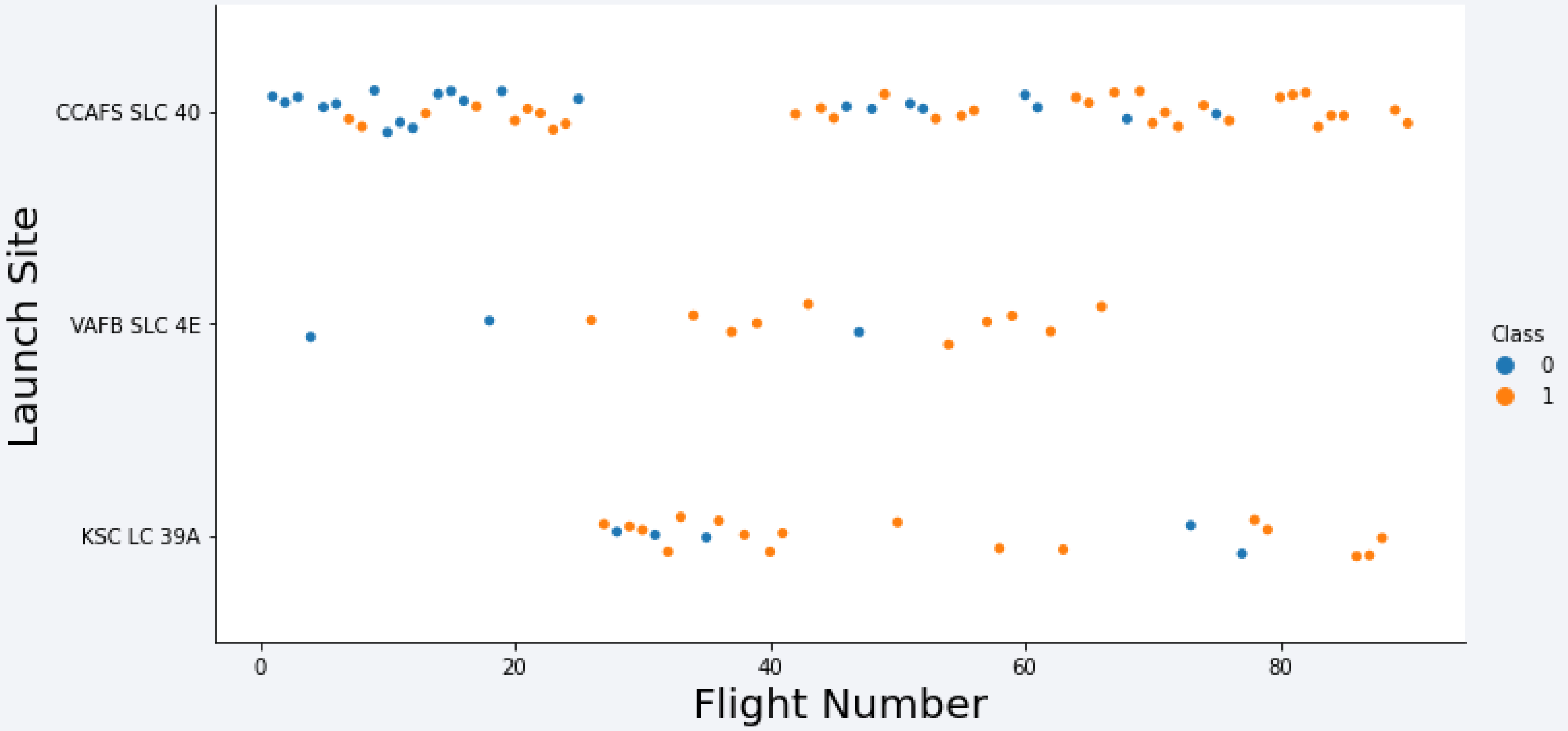


Section 2

Insights drawn from EDA

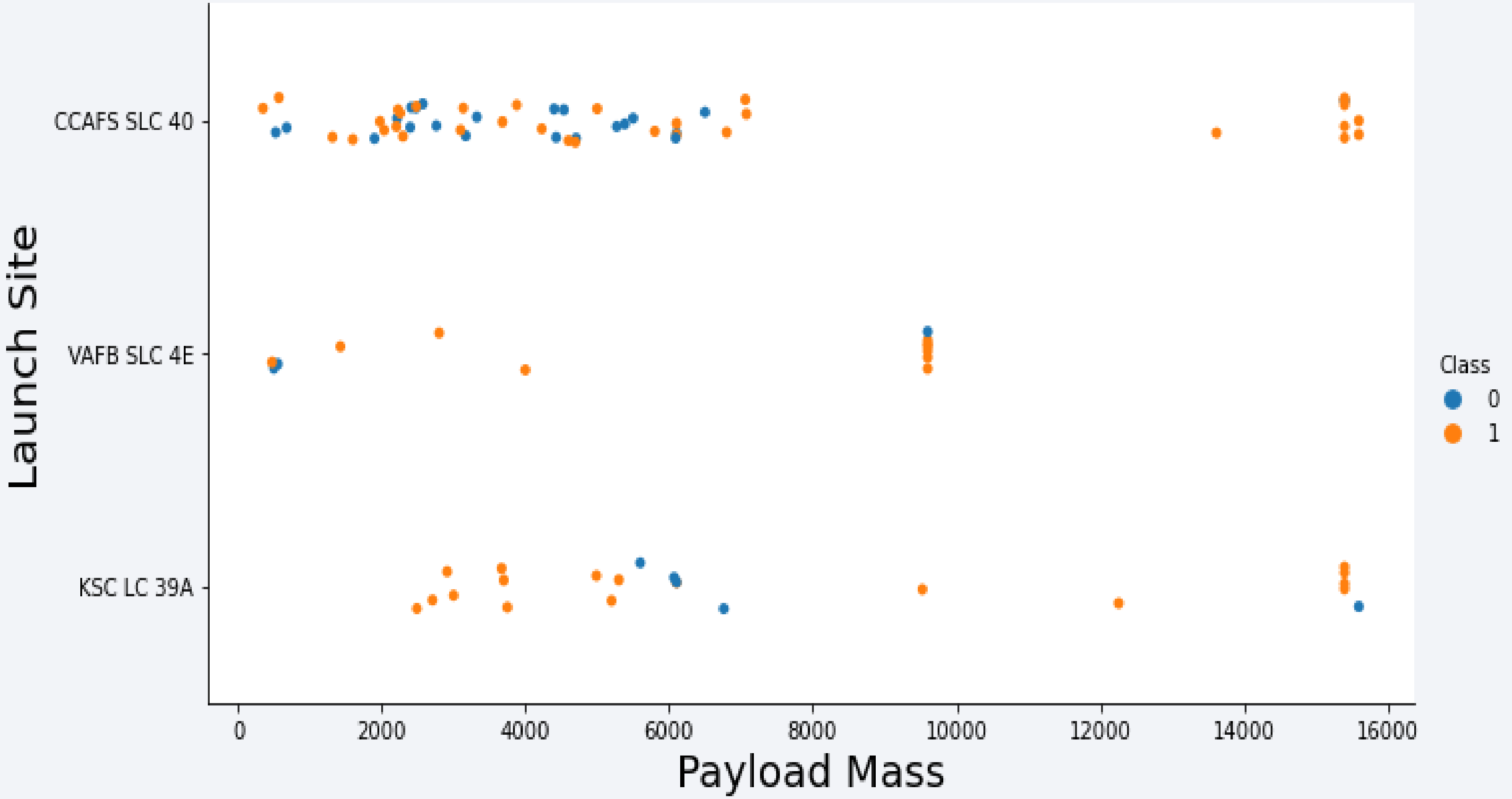
Flight Number vs. Launch Site

- Launch site CCAFS SLC 40 has the highest number of launches compared to other sites.
- Site VFAB SLC 4E has the least number of launches.



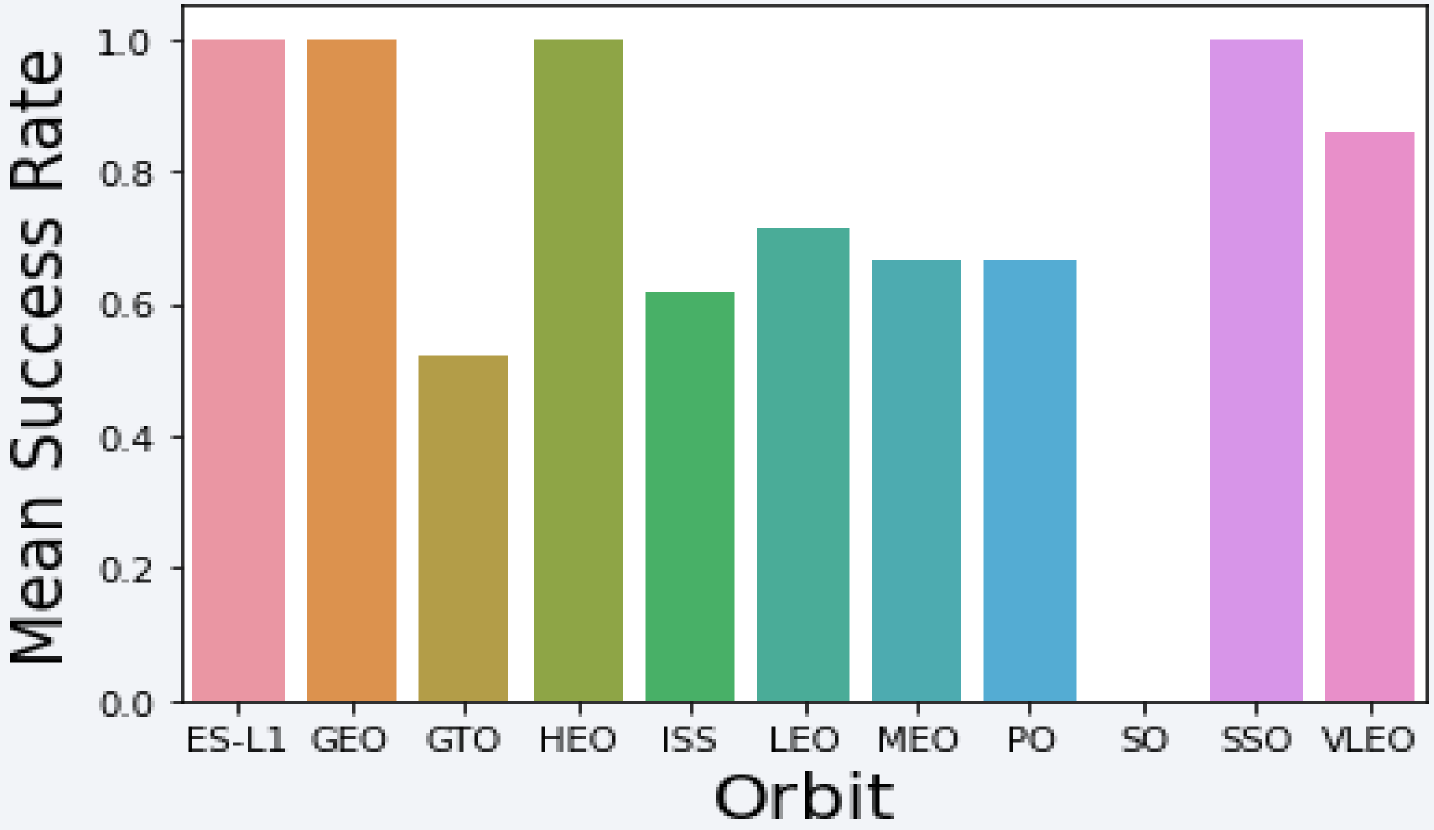
Payload vs. Launch Site

- Site CCAFS SLC 40 has launched most payloads of 9000 or less.
- CCAFS SLC 40 also has the greatest number of unsuccessful launches.
- VFAB SLC 4E has never been used to launch payloads greater than 10000.
- KSC LC 39A has been used for launching payloads of all sizes.
- CCAFS SLC 40 has been most successful in launching heavy payloads without any failure.



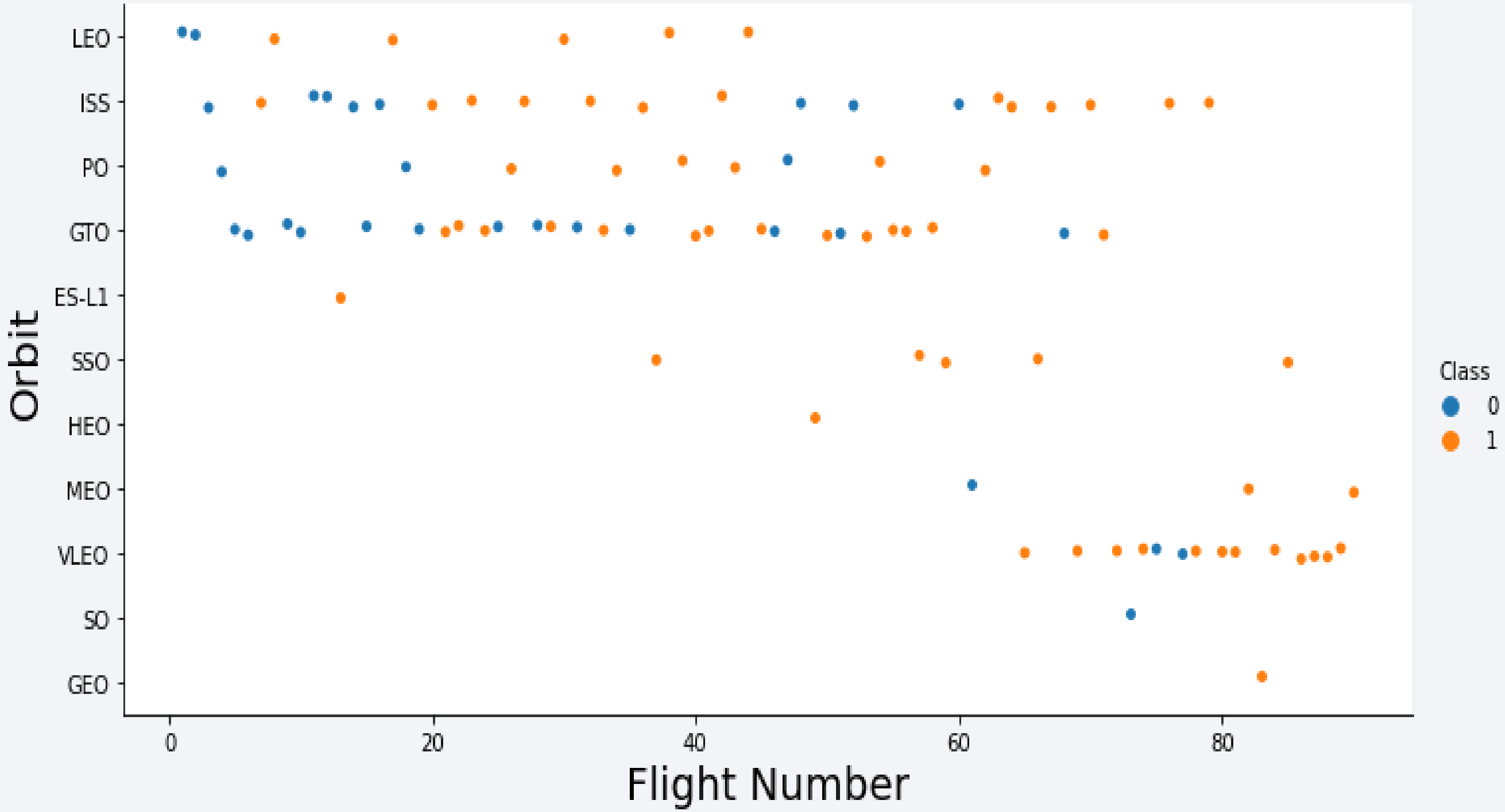
Success Rate vs. Orbit Type

- Launches to ES-L1, GEO, HEO, SSO have a 100% success rate.
- Launch to GTO has been least successful with a success rate of 52%.
- SO has a success rate of 0 due to a single failed launch.



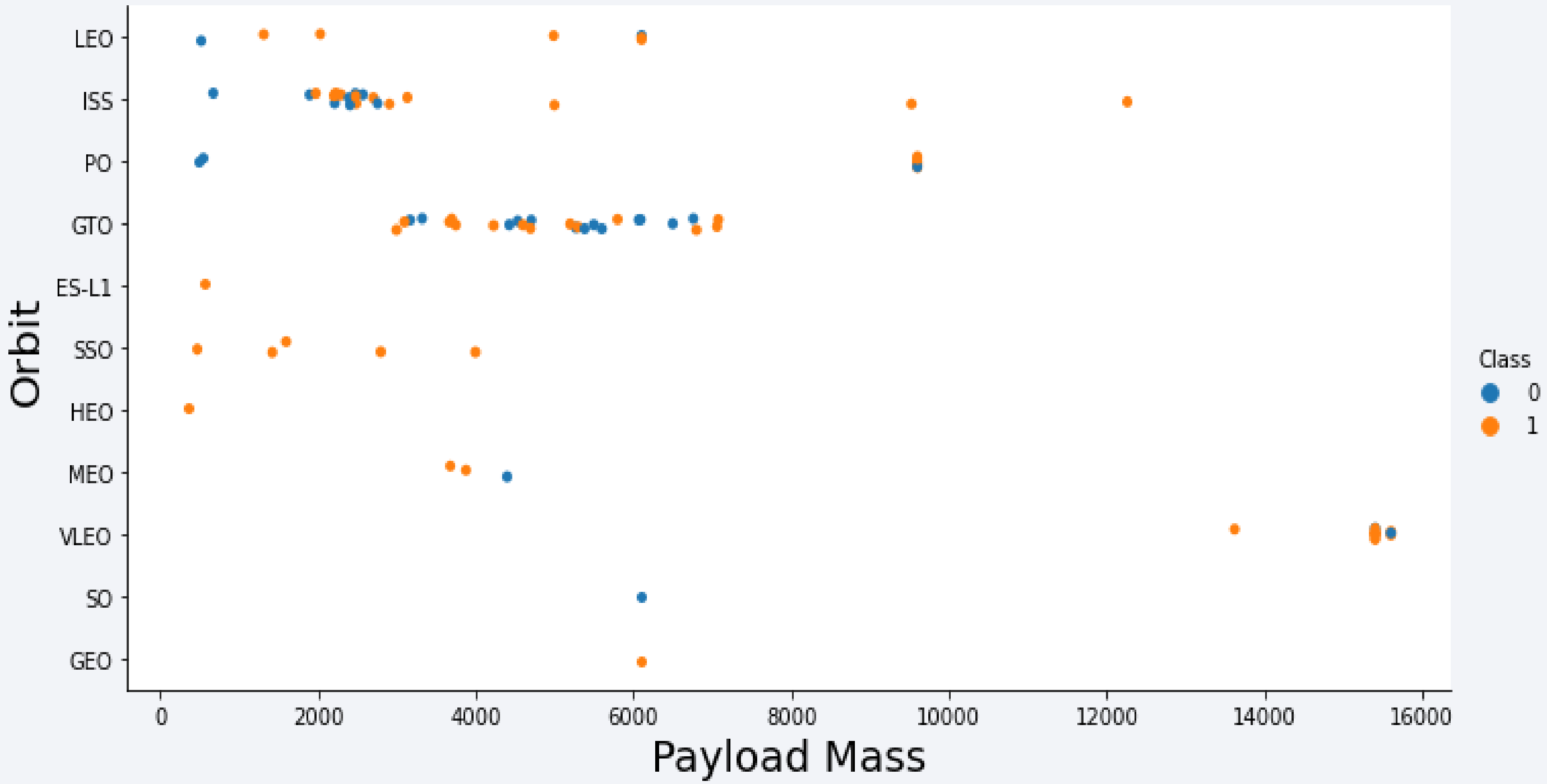
Flight Number vs. Orbit Type

- Most of the flights have been launched to GTO.
- ES-L1, HEO, SO, GEO, each had only one launch. With the launch to SO being unsuccessful.
- Almost 50% of launches to GTO have failed.



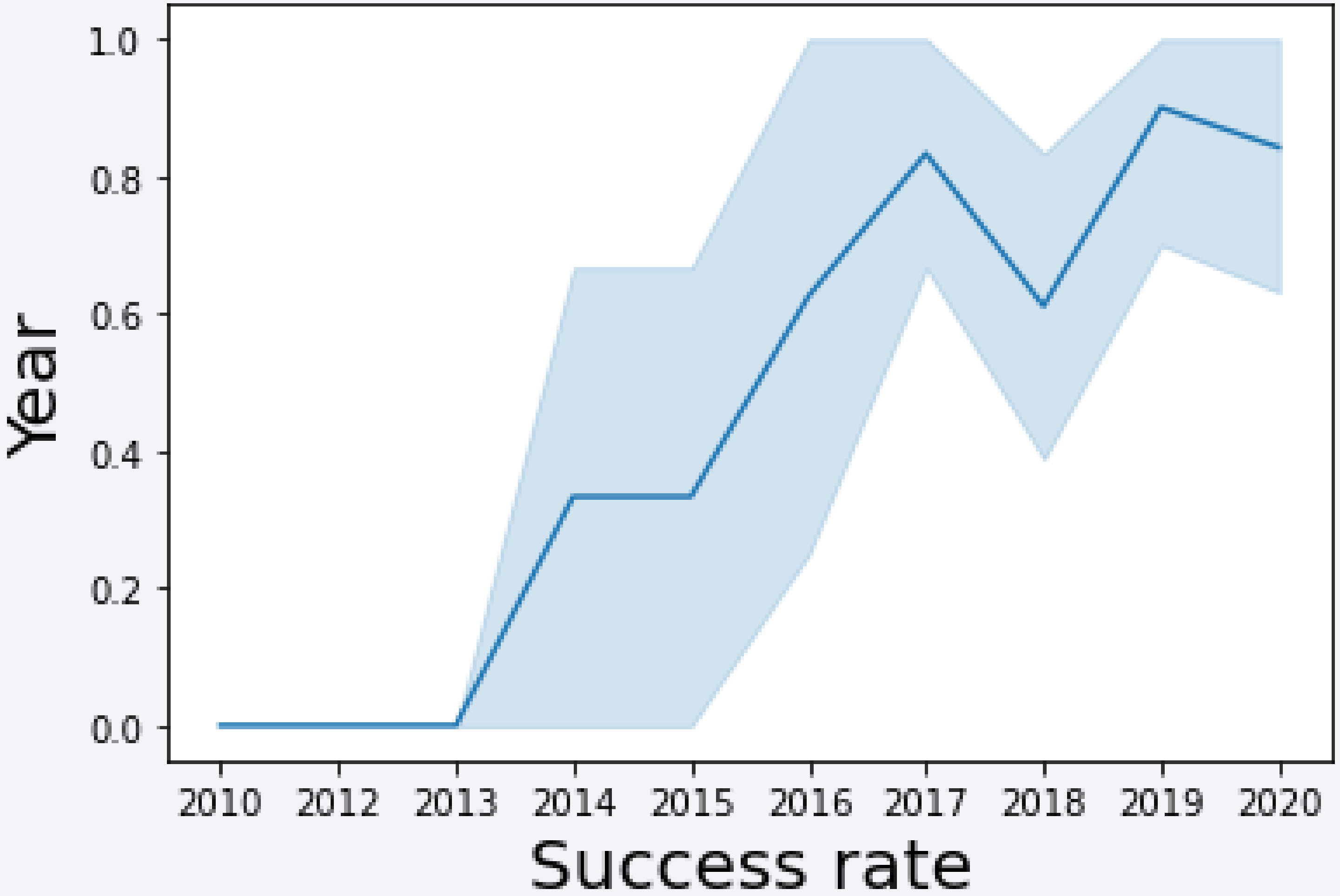
Payload vs. Orbit Type

- Heaviest payload has been launched to VLEO.
- Maximum number of launches have been in the payload range of 2000 – 10000.



Launch Success Yearly Trend

- Launch success has greatly increased from 2013 onwards.
- There has been a steady growth in success rate up to 2017, where it declined till 2018.
- After 2018 the success rate has increased again.



All Launch Site Names

- The names of the unique launch sites can be found using DISTINCT or UNIQUE keywords.
- The UNIQUE constraint ensures that all values in a column are different and is preferable in this scenario.

```
%sql select unique(launch_site) from spacextbl
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-af
main.cloud:31929/bludb
Done.
```

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

```
%sql select distinct launch_site from spacextbl
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-af
main.cloud:31929/bludb
Done.
```

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

Launch Site Names Begin with 'KSC'

- The LIKE command is used in a WHERE clause to search for a specified pattern in a column. You can use two wildcards (%) with LIKE
- The keyword LIMIT restricts the number of rows to the number preceding the keyword. In this case 5

```
%sql select * from spacextbl where launch_site like 'KSC%' limit 5
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__ou
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No a
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No a

Total Payload Mass

- The function SUM has been used to calculate and display the sum of payloads carried with the condition WHERE used to add only the values where the customer is NASA(CRS).

```
%sql select sum(payload_mass__kg_) from spacextbl where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1
main.cloud:31929/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- AVG function was used to calculate the average payload mass carried by booster version F9 v1.1
- The LIKE command is used in a WHERE clause to search for a specified pattern in a column.

```
%sql select avg(payload_mass__kg_) from spacextbl where booster_version like 'F9 v1.1'
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.c
main.cloud:31929/bludb
Done.
```

1
2928

First Successful Drone Ship Landing Date

- To find the dates of the first successful landing outcome on drone ship, MIN function was used with WHERE clause.

List the date where the first successful landing outcome in drone ship was acheived.

```
%sql select min(DATE) from spacextbl where landing__outcome = 'Success (drone ship)'
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.
main.cloud:31929/bludb
Done.
```

1
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

- AND and BETWEEN conditions were used with WHERE clause in the query to find the result.

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

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-
main.cloud:31929/bludb
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

- Calculate the total number of successful and failed mission outcomes

```
%sql select landing__outcome, count(*) as COUNT_ from spacextbl \
where \
landing__outcome like 'Success%' \
or \
landing__outcome like 'Fail%' \
group by landing__outcome
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.b
main.cloud:31929/bludb
Done.
```

landing__outcome	count_
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

- Subquery was used to list the names of the booster which have carried the maximum payload mass

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

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs:
main.cloud:31929/bludb
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

2017 Launch Records

- 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

```
%sql select monthname(DATE) as Month, landing__outcome, \
booster_version, launch_site from spacextbl \
where (landing__outcome = 'Success (ground pad)' \
       and extract(year from DATE) = '2017')
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e
main.cloud:31929/bludb
Done.
```

MONTH	landing__outcome	booster_version	launch_site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

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

```
%sql select extract(year from DATE)as YEAR,\
count(landing__outcome)as Success_Count from spacextbl \
where (DATE between '2010-06-04'and '2017-03-20') and \
landing__outcome like 'Success%' group by extract(year from DATE) \
order by count(landing__outcome) desc
```

```
 * ibm_db_sa://mhn78688:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2
main.cloud:31929/bludb
Done.
```

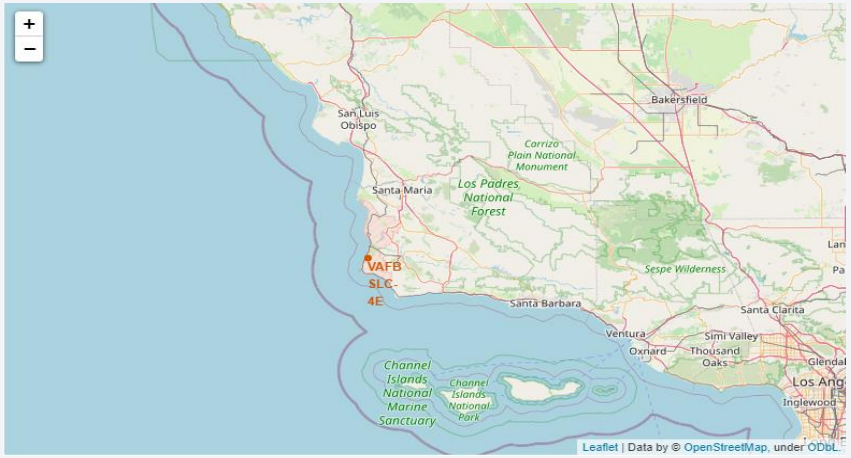
YEAR	success_count
2016	5
2017	2
2015	1

Section 3

Launch Sites Proximities Analysis

Marking all launch sites on a map

- The maps show that all the launch sites are close to the coastline.
- They are also close to equator, situated within 29-34 degrees North latitude.

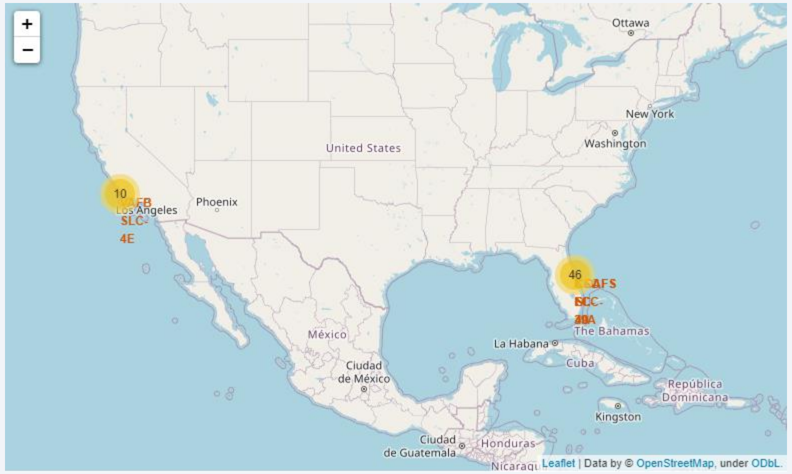


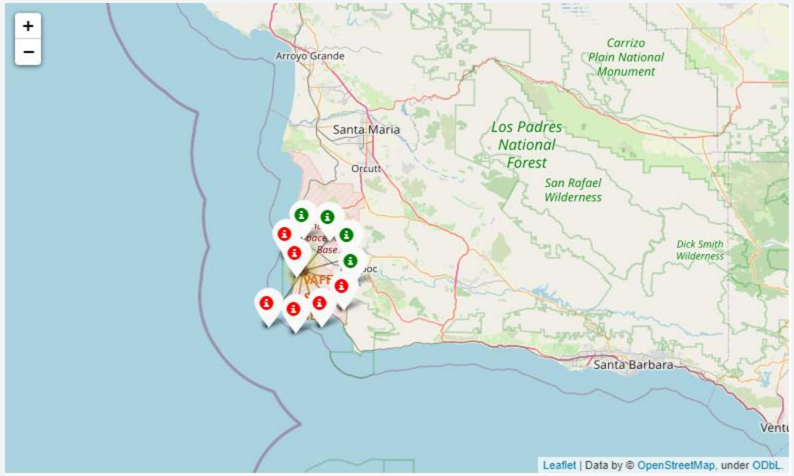
	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

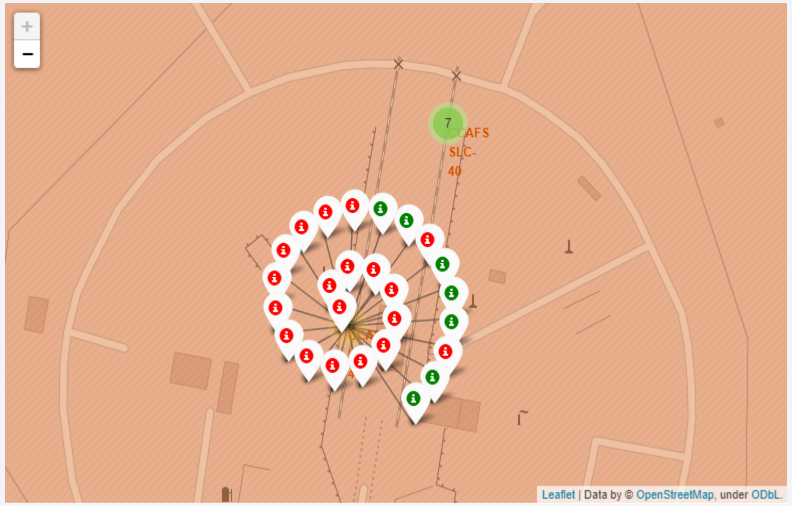
Marking the successful/failed launches for each site on the map

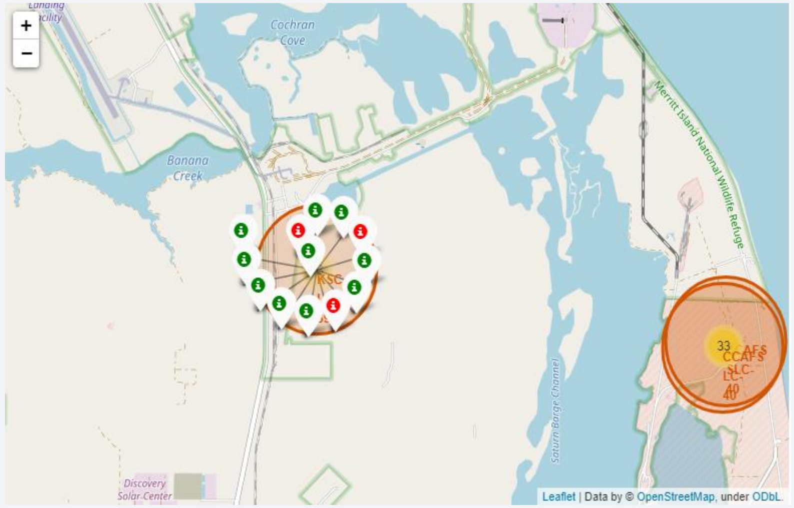
All the successful launches are marked in green, and the failed ones are marked in red

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red



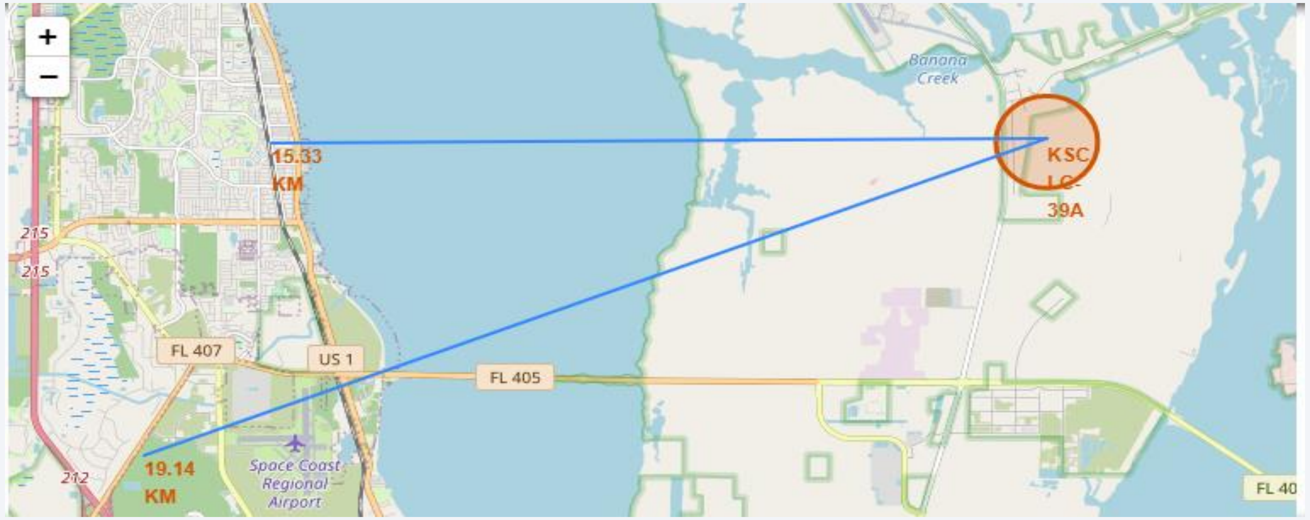






Calculating the distances between a launch site and its proximities

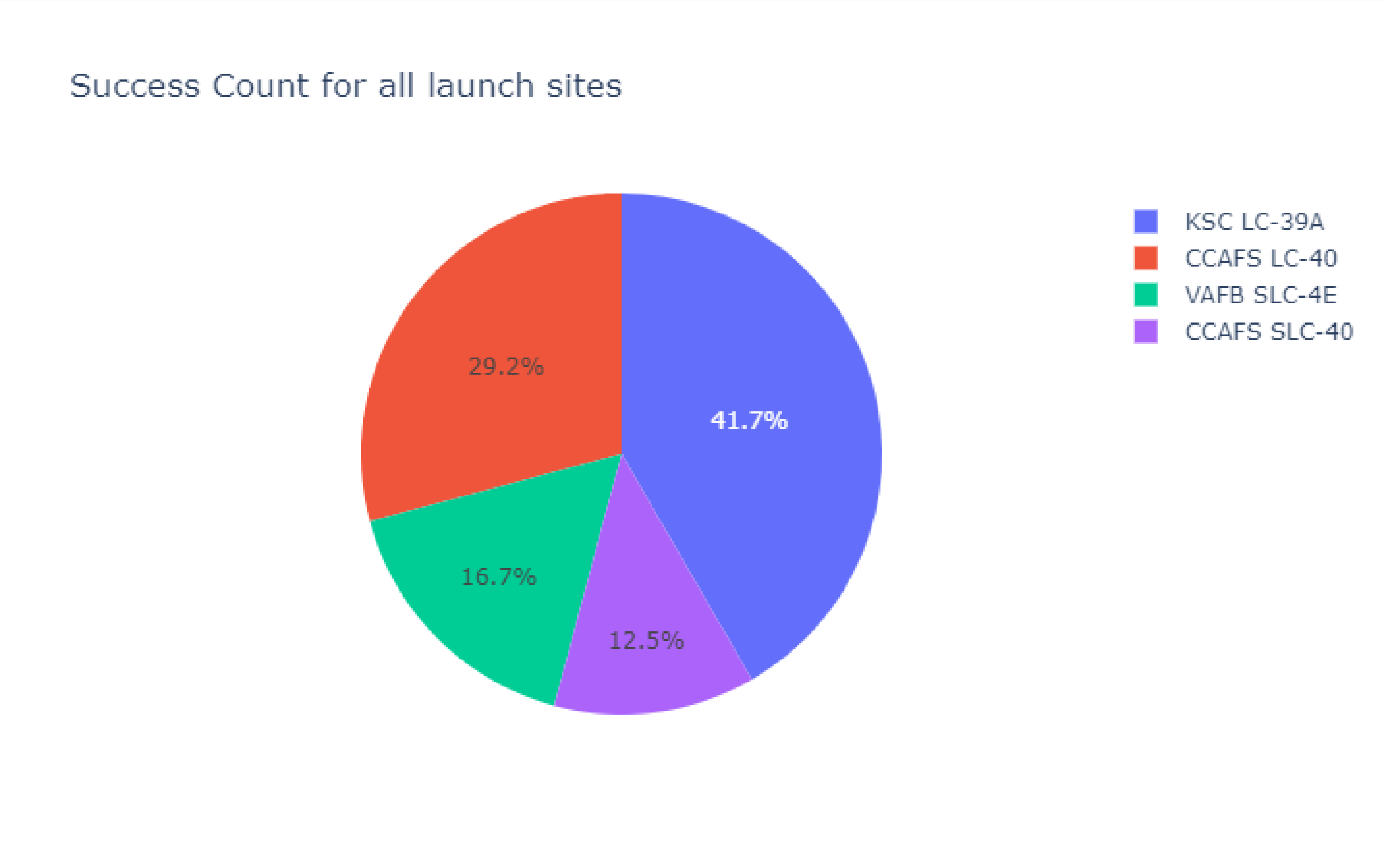
- The launch sites are situated in protected areas and are well connected by a network or railways and roads.
- They are away from human settlements to avoid any loss or damages during failed launches.
- All launch sites are located near the coastline and equator.
- If a spacecraft is launched from a site near Earth's equator, it can take optimum advantage of the Earth's substantial rotational speed. Sitting on the launch pad near the equator, it is already moving at a speed of over 1650 km per hour relative to Earth's center.
- The launching stations are located near the eastern coastline so that in case of a failure, the satellite does not fall on a built-up area.



Section 4

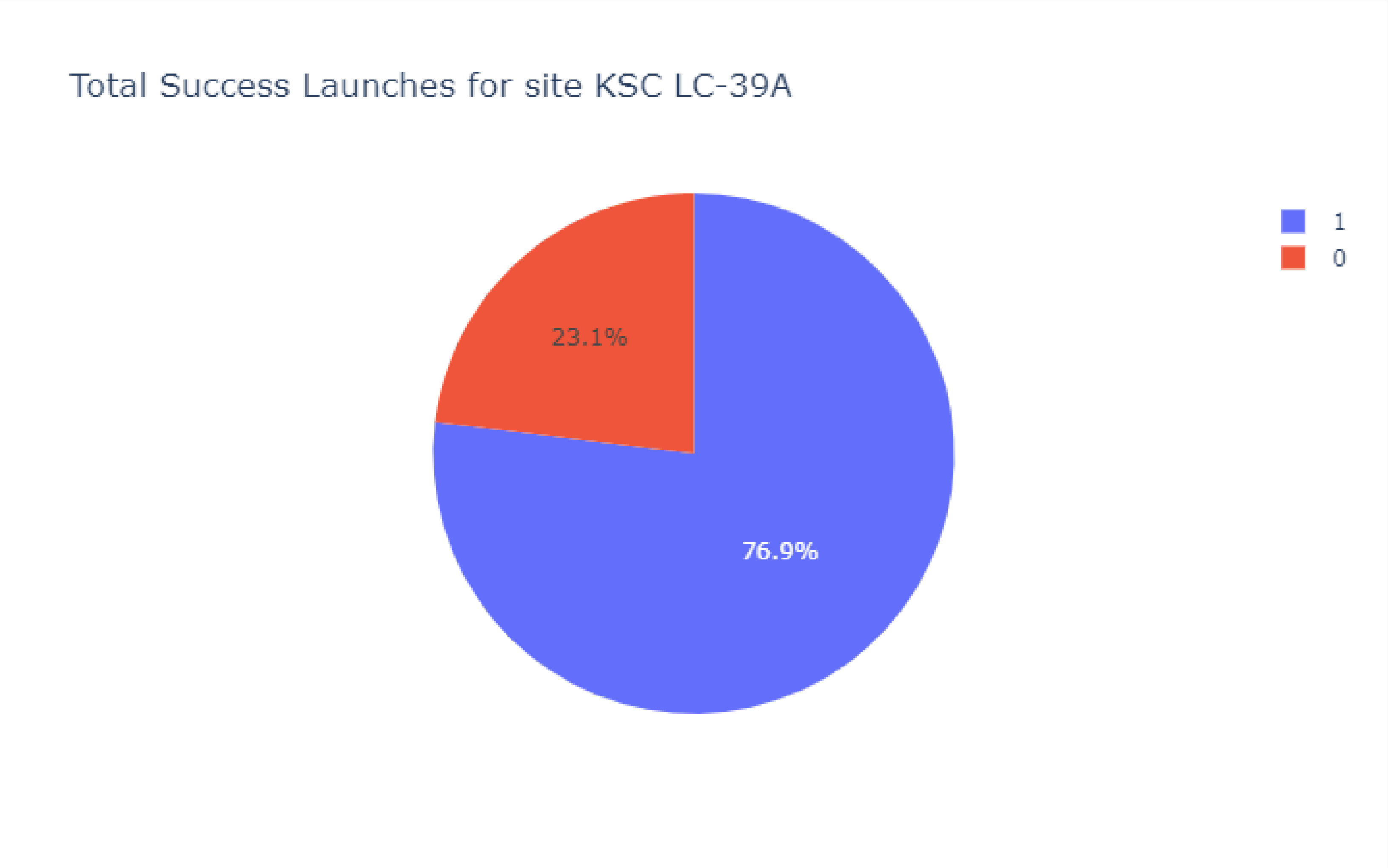
Build a Dashboard with Plotly Dash

Success count for all launch sites



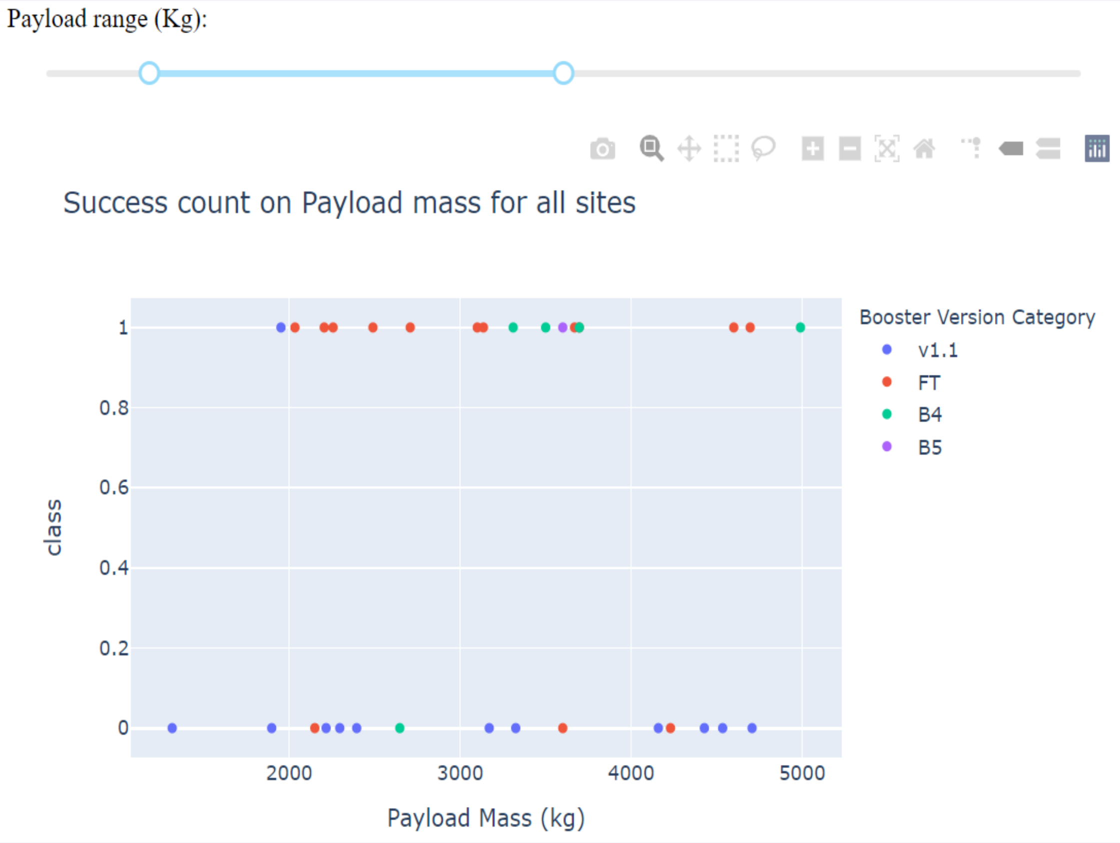
- The success rate of the launch site KSC LC 39A is the highest.
- The site CCAFS SLC-40 has the lowest rate of success.

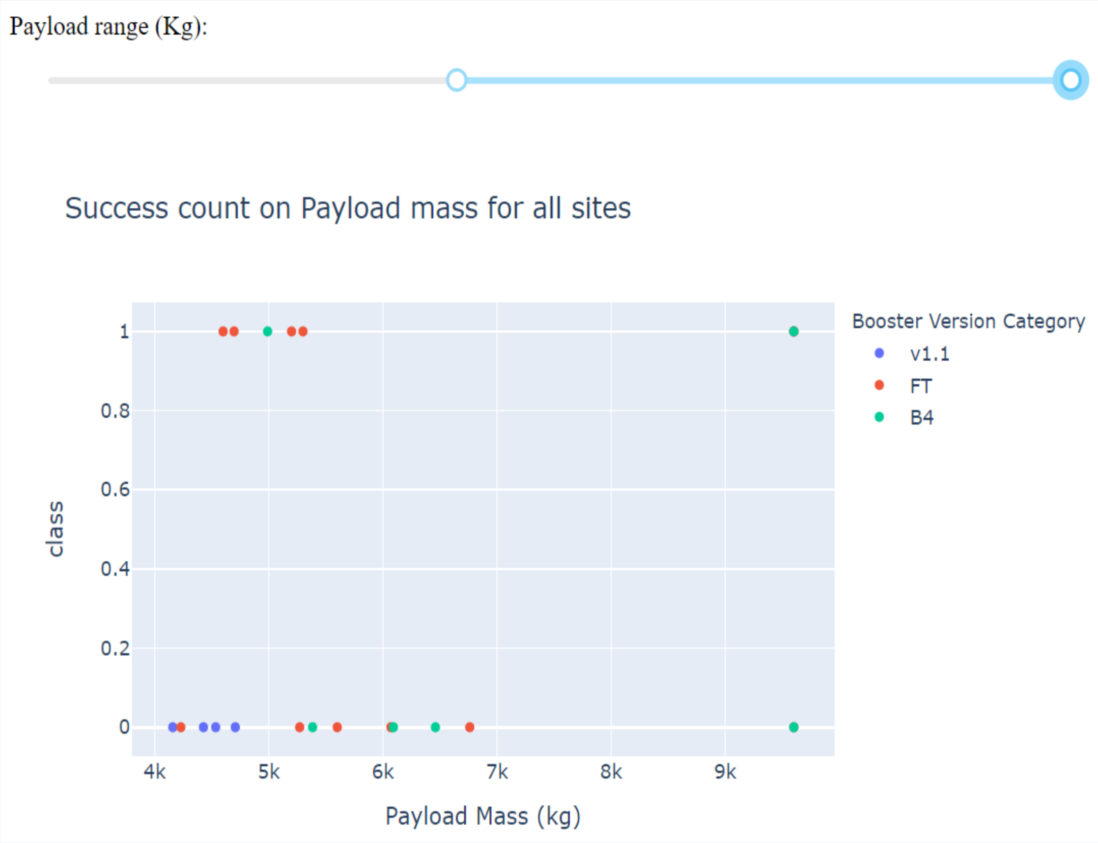
The launch site with highest launch success ratio



Launch site KSC LC-39A has the highest success rate with 76.9% of launches being successful and 23.1% failing.

Payload vs. Launch Outcome scatter plot for all sites



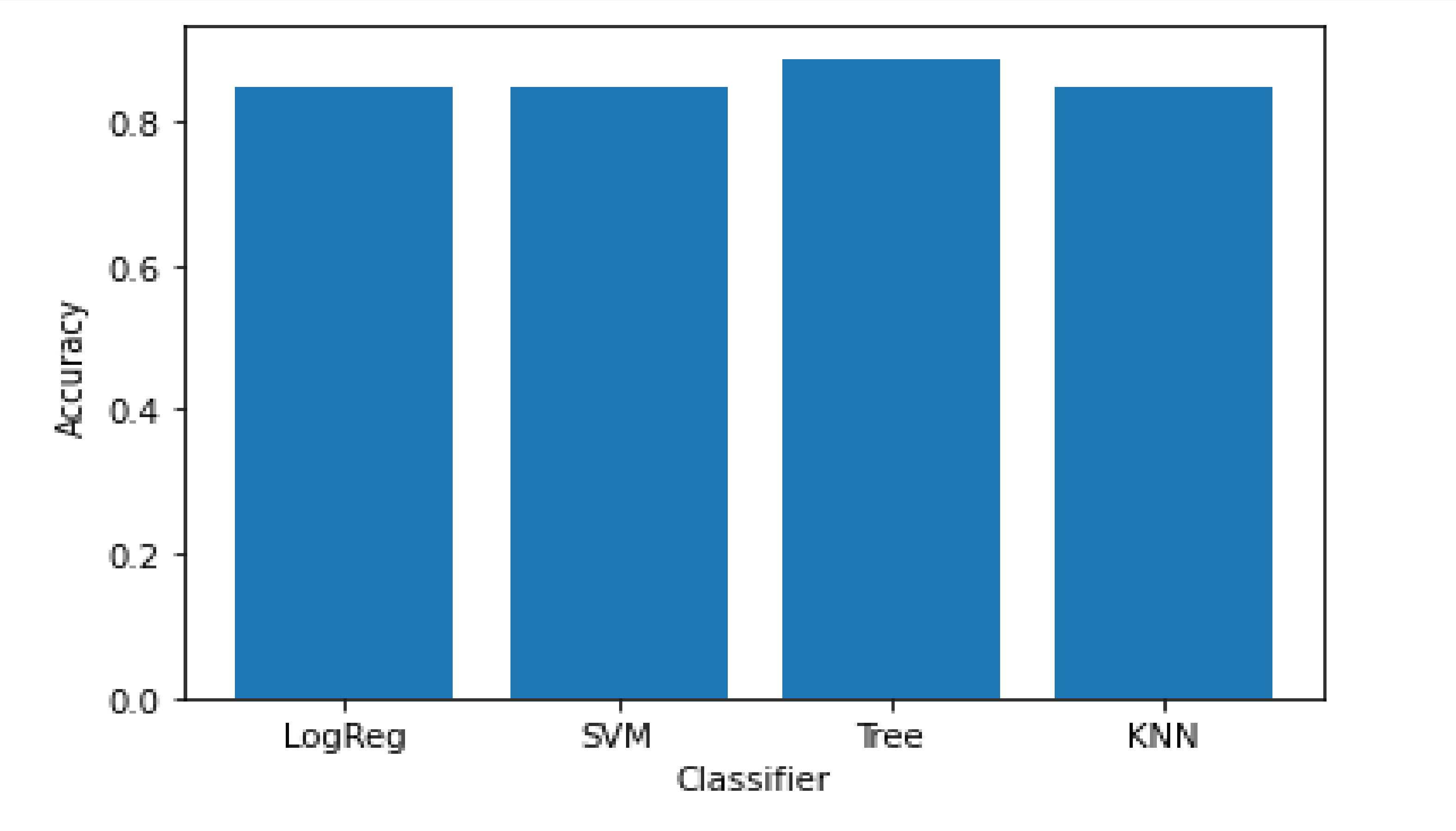


The success rate for payloads between 2000 – 5000 KG is better than the success rate of payloads in the range 5000-10000KG

Section 5

Predictive Analysis (Classification)

Classification Accuracy

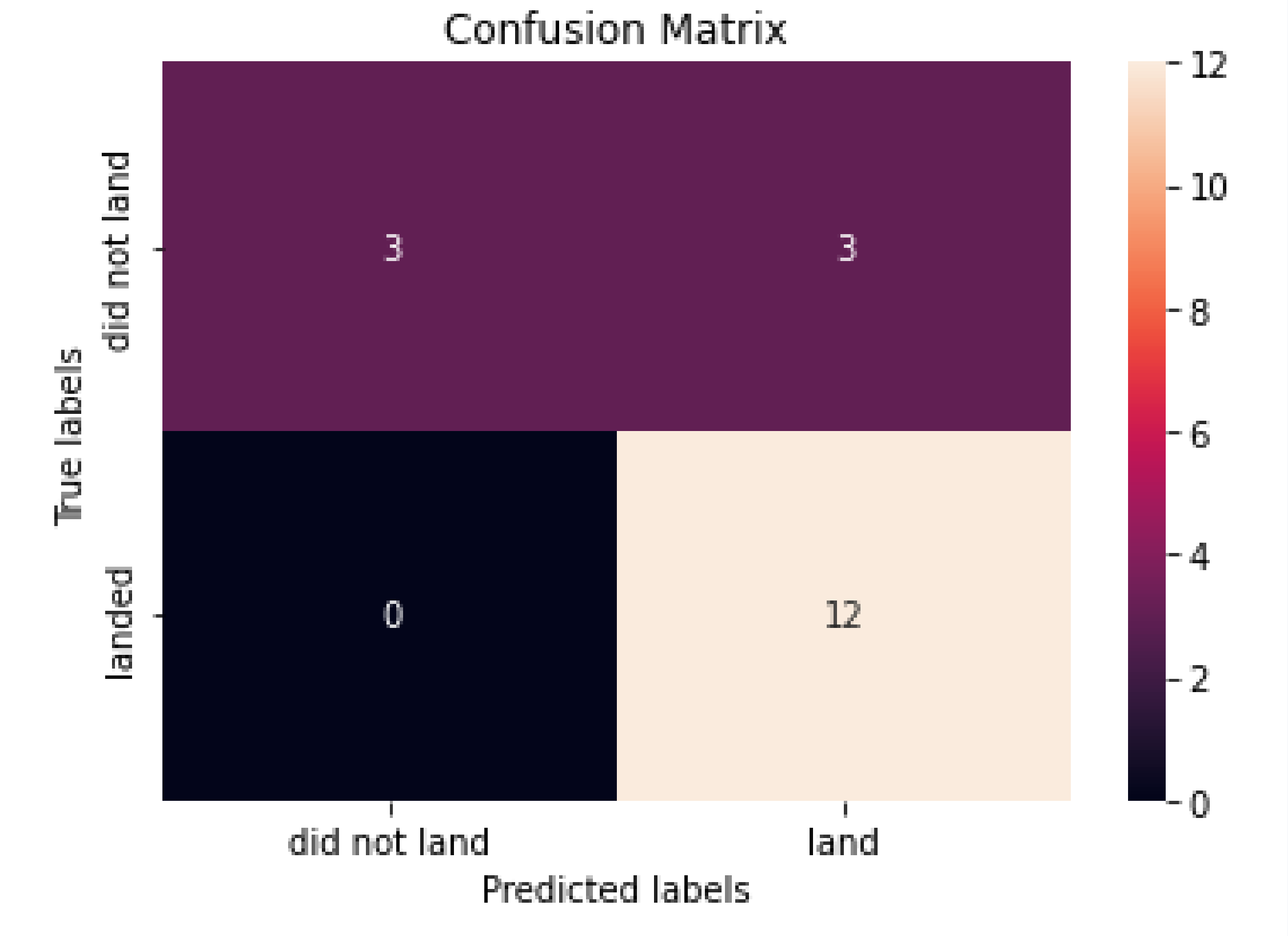



```
{'LogReg': 0.8464285714285713, 'SVM': 0.8482142857142856, 'Tree': 0.8875, 'KNN': 0.8482142857142858}
```

- The accuracy of all models is above 80%.
- Logistic Regression, KNN and SVM have almost the same accuracy.
- Tree model has the highest accuracy at 88.7%

Confusion Matrix

Confusion matrix of Decision Tree model



- There is no False Negative
- True Negative is 100% correct.
- True Positive and False Positive are equal

Conclusions

- The Decision Tree model gives the highest accuracy, SVM, KNN, Logistic Regression have nearly the same accuracy.
- Low weighted payloads have a better chance of success than the heavier ones
- Success rates of launches have increased rapidly from 2013 onwards.
- KSC LC-39A has the most successful launches.
- Launches to orbits GEO, HEO, SSO, ES-L1 have the best success rates.
- Recovery via drone ship is more successful compared to ground pad and parachute landings.

Appendix

GITHUB URL:

- **FOLIUM:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Assignment_Interactive%20Visual%20Analytics%20with%20Folium.ipynb
- **PREDICTIONS:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Assignment_Machine%20Learning%20Prediction.ipynb
- **DATA COLLECTION:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project.ipynb
- **DATA WRANGLING:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_DataWrangling.ipynb
- **DASHBOARD:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_EDA%20with%20Visualization%20lab.ipynb
- **SQL:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_ExploDataAnalysis.ipynb
- **WEBSCRAPPING:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/Final%20Project_Webscrapping.ipynb
- **PLOTLY:** https://github.com/shirazwadia/ibm_ds_capstone/blob/master/FinalProject_Build%20an%20Interactive%20Dashboard_%20Ploty.ipynb

Thank you!