

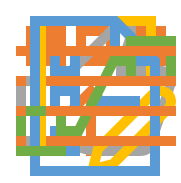


IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ziad Tarek
21-10-2021





Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

- **Summary of methodologies**
 - Data Collection Section
 - Data Wrangling
 - EDA
 - EDA with SQL
 - Data Visualization With Folium
 - Dashboarding with Dash
 - Modeling
- **Summary of all results**
 - Exploratory data analysis results
 - Interactive analytics demo in screenshots
 - Predictive analysis results



Introduction

- **Project background and context**

We predicted if the Falcon 9 first stage will land successfully. 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. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems you want to find answers**

- What affects the successful landing of the rocket ?
- Will the rocket land successfully?

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology :**

- We collected the data by web scrapping (spaceX rest API , Wikipedia)

- **Perform data wrangling :**

- we created a class label from outcome column with 1,0 's , 1 for successful landing and 0 otherwise

- **Perform exploratory data analysis (EDA) using visualization and SQL**

- **Perform interactive visual analytics using Folium and Plotly Dash**

- **Perform predictive analysis using classification models**

- We used classification models models to predict the landing outcome.

Data Collection

- We collected data from spaceX Rest API and Wikipedia

ØspaceX Rest API

- ❑ We worked with SpaceX launch data that is gathered from the SpaceX REST API.
- ❑ This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome

➤ Wikipedia

- ❑ Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.

Data Collection – SpaceX API

1. Getting Response from API

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

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

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

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

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

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

4. Assign list to dictionary then 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}
```

```
# Create a data from launch_dict  
df = pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

5. Filter dataframe and export to flat file (.csv)

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


Data Collection - Scrapping

1. Getting Response from HTML

```
page = requests.get(static_url)
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

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

4. Getting column names

```
column_names = []
count=0
temp = first_launch_table.find_all('th')
for x in range(len(temp)):
    print(count)
    count+=1
    name=extract_column_from_header(temp[x])
    if name is not None and len(name) > 0:
        column_names.append(name)
```

7. Dataframe to .CSV

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

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with empty lists
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. Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(html_tables):
    # get table row
    for rows in table.find_all('tr'):
        # check to see if first row
```

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful
- You need to present your data wrangling process using key phrases and flowcharts
- [Data Wrangling Notebook](#)

EDA with Data Visualization

1- Exploratory Data Analysis

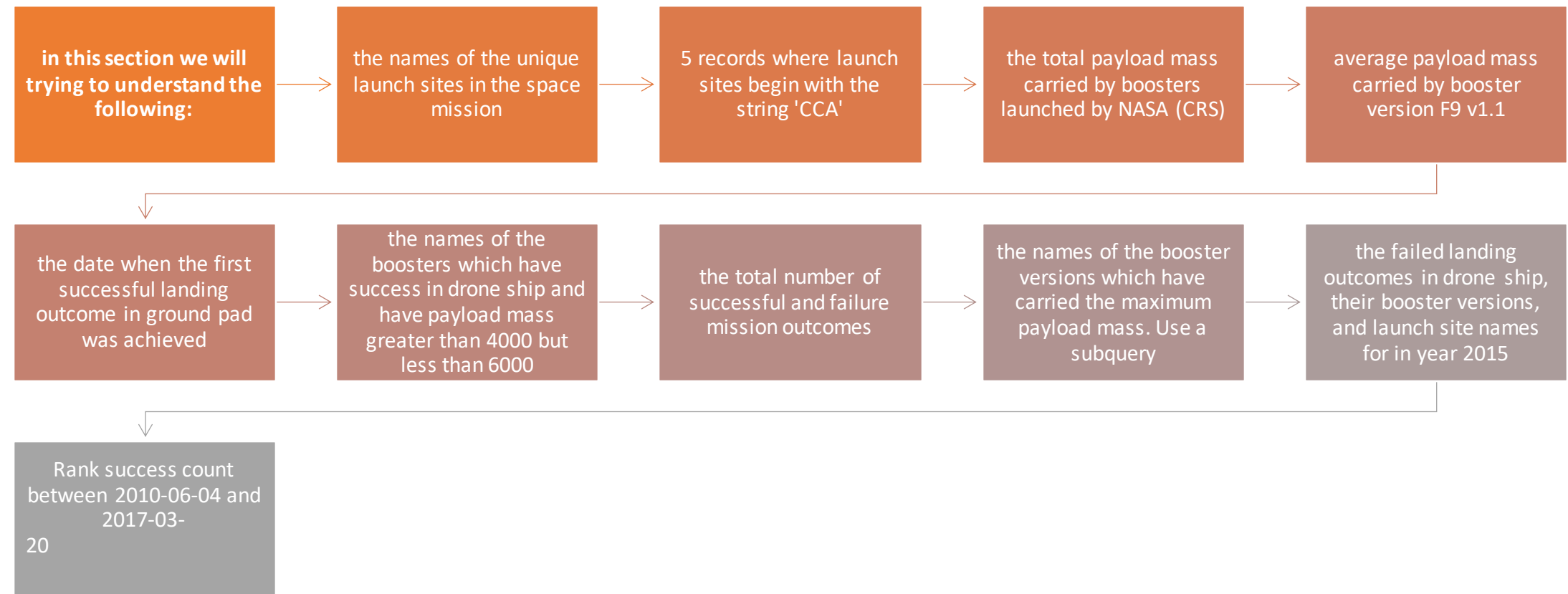
- Visualization of the relationship between Flight Number and Launch Site
- Visualization of the relationship between Payload and Launch Site
- Visualization of the relationship between success rate of each orbit type
- Visualization of the relationship between FlightNumber and Orbit type
- Visualization of the relationship between Payload and Orbit type
- Visualization of the launch success yearly trend

2- Features Engineering

- Creating a dummy variables to categorical columns
- [EDA notebook](#)

EDA with SQL

github url to notebook





Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a *Circle Marker* around each launch site with a label of the name of the launch site.
- We assigned the data frame launch outcomes (failures, successes) to *classes 0 and 1* with Green and Red markers on the map in a `MarkerCluster()`
- Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks
- **Example of some trends in which the Launch Site is situated in.**
 - Are launch sites in close proximity to railways? No
 - Are launch sites in close proximity to highways? N
 - Are launch sites in close proximity to coastline? Yes
 - Do launch sites keep certain distance away from cities? Yes

[EDA with Folium notebook](#)



Predictive Analysis (Classification)

• BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

• EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

• IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning
- Modeling Notebook

Results



EXPLORATORY DATA
ANALYSIS RESULTS



INTERACTIVE ANALYTICS
DEMO IN SCREENSHOTS



PREDICTIVE ANALYSIS
RESULTS

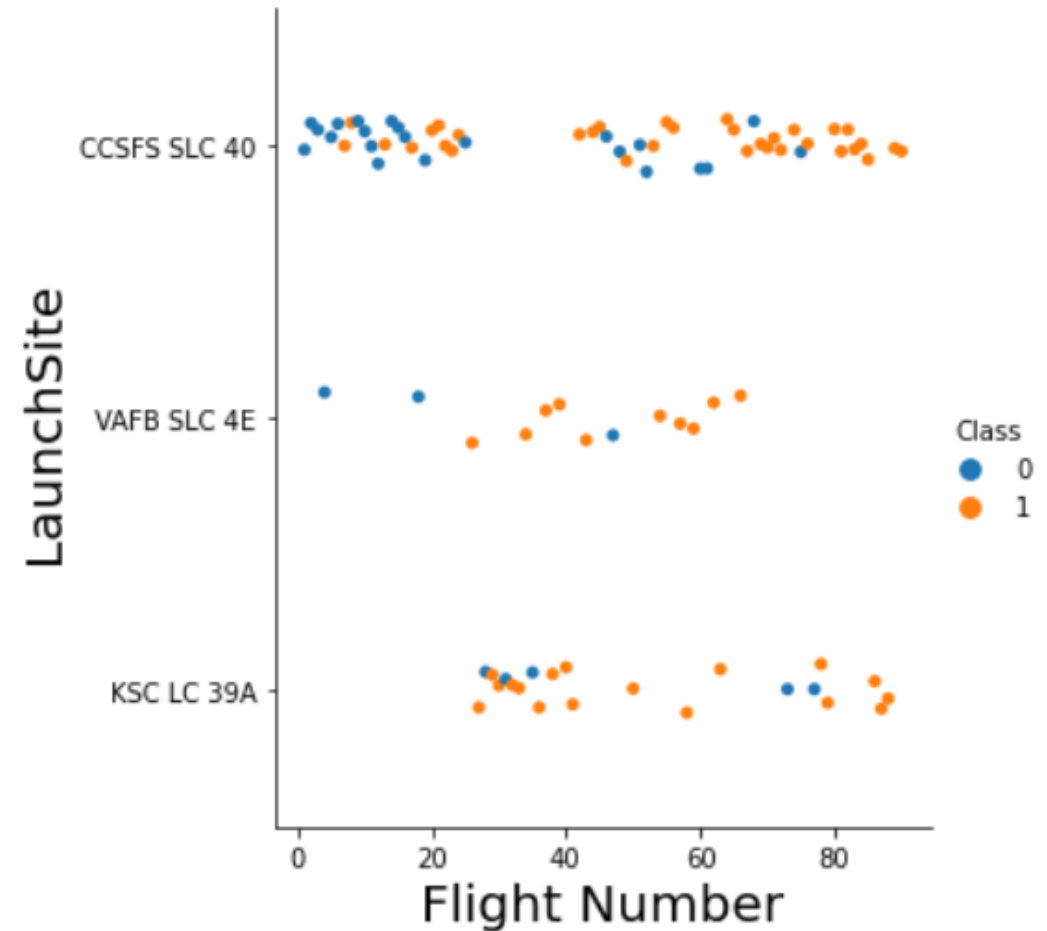
The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue, red, and teal on the right. These streaks are layered over a faint, grid-like pattern, creating a sense of depth and movement.

Section 2

Insights drawn from EDA

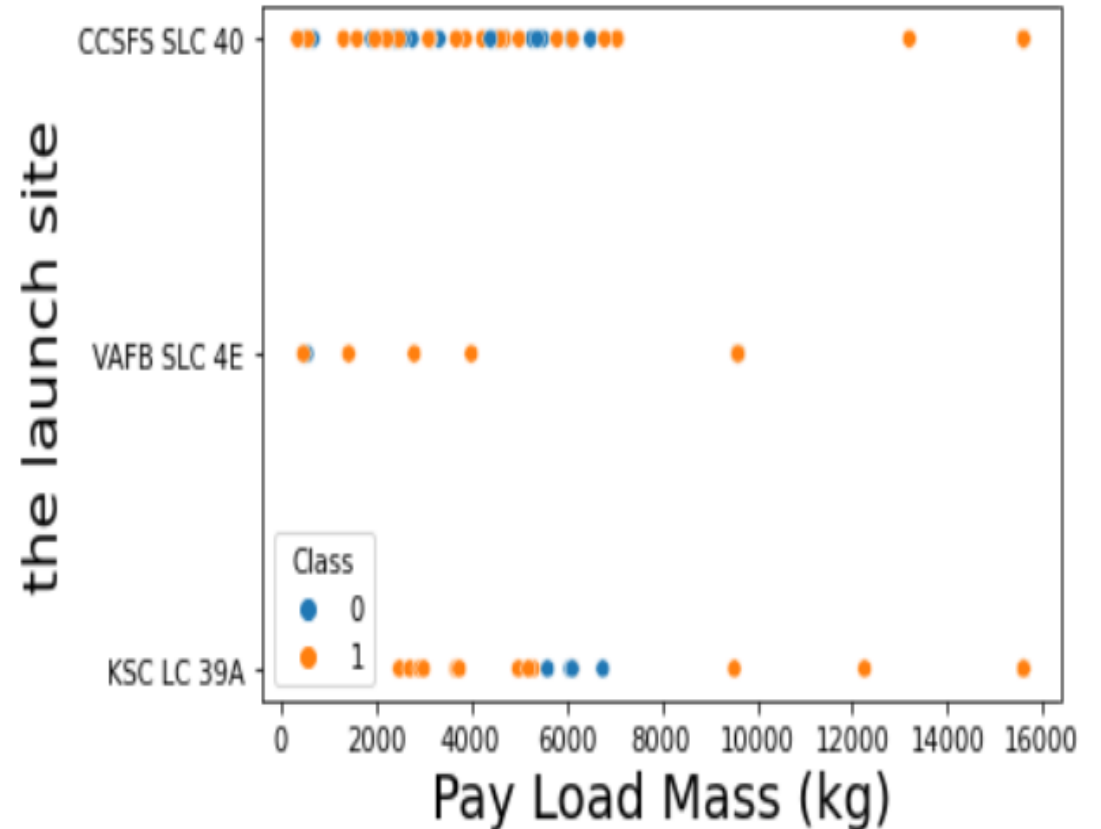
Flight Number vs. Launch Site

- The more amount of flights at a launch site the greater the success rate at a launch site.



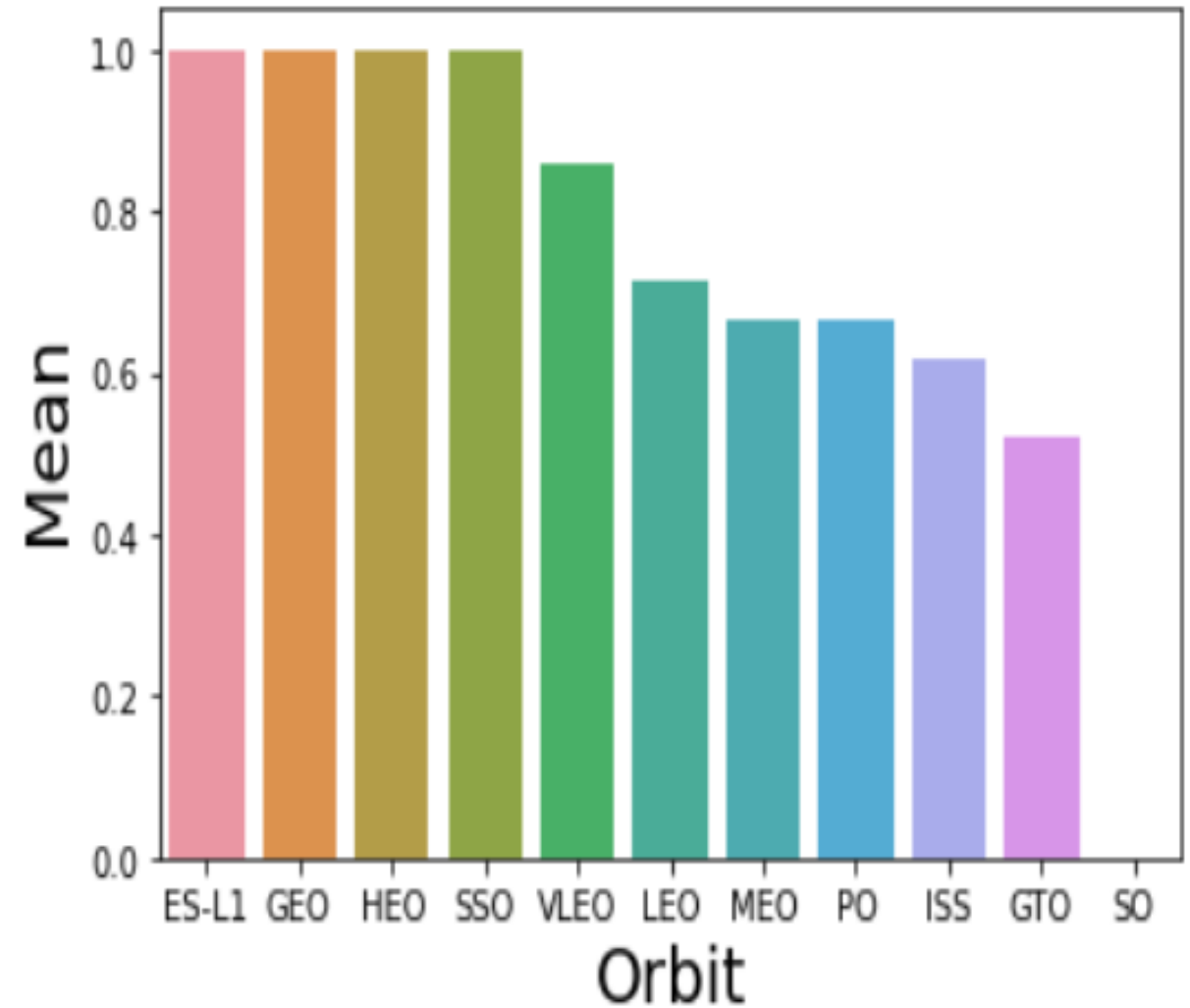
Payload vs. Launch Site

- The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.



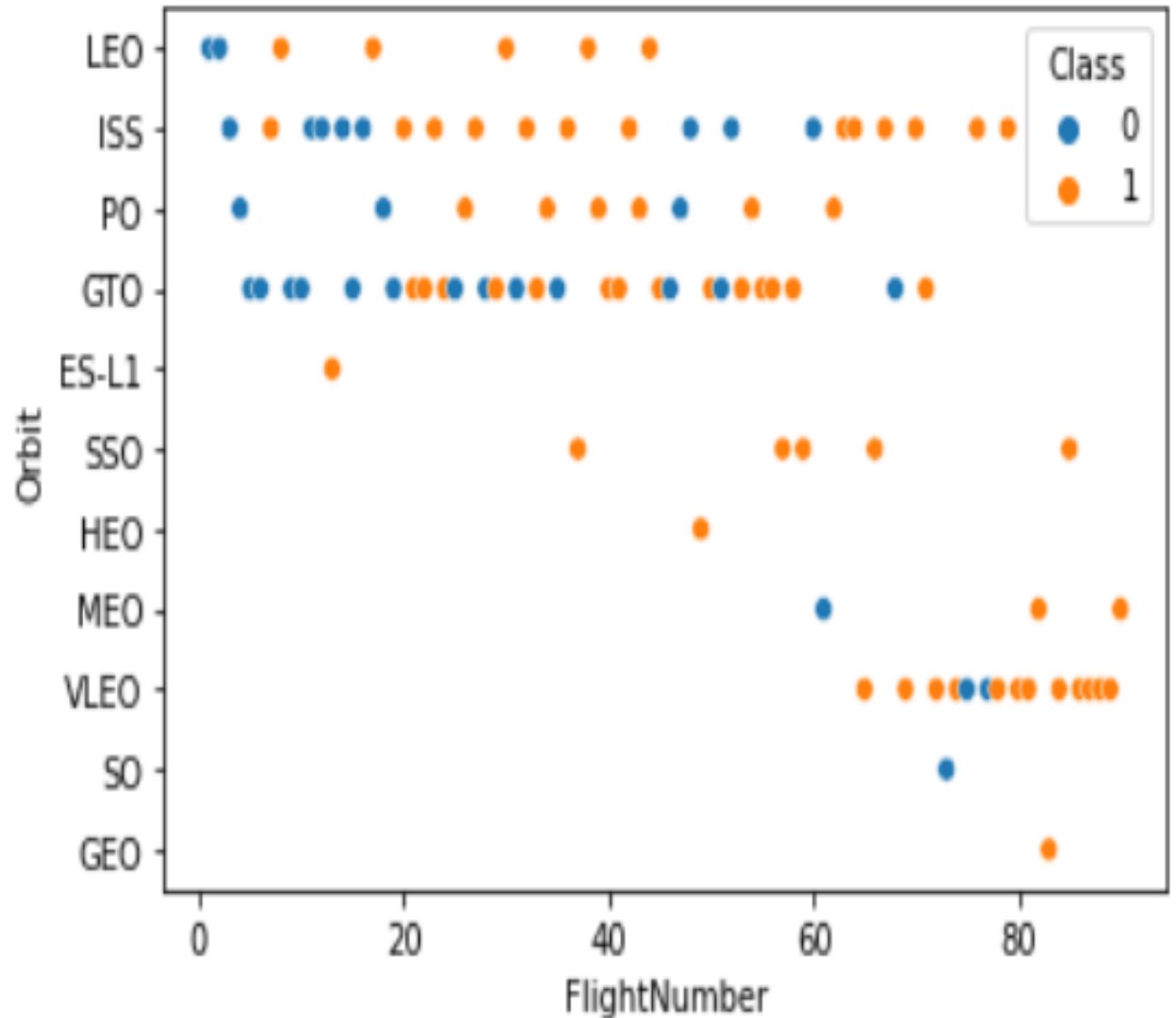
Success Rate vs. Orbit Type

- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate



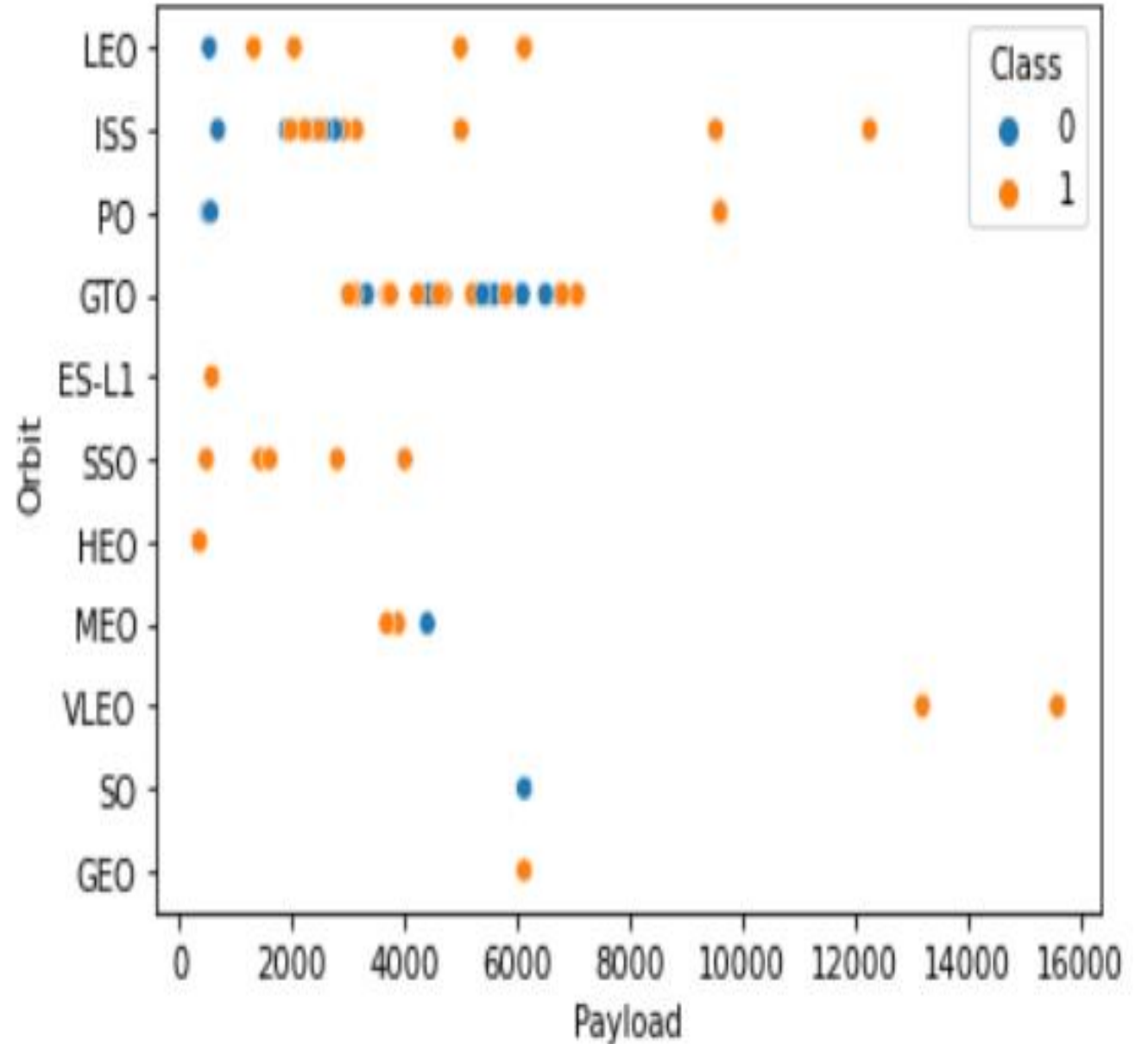
Flight Number vs. Orbit Type

- You should see 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



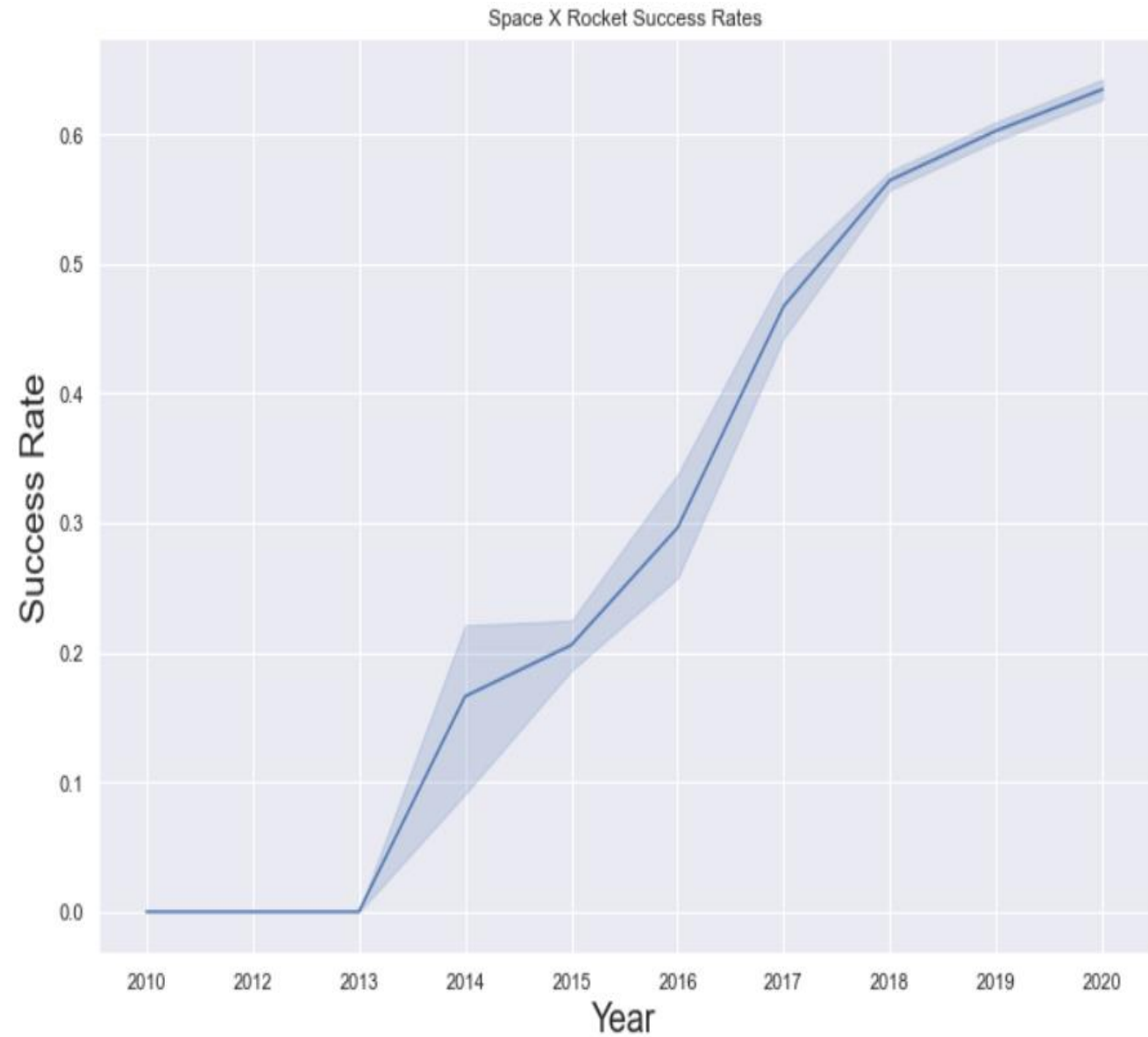
Payload vs. Orbit Type

- You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend

- you can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names



```
1 SELECT DISTINCT Launch_Site
2 From SpaceX
```


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

Launch Site Names Begin with 'CCA'

```
SELECT *  
From SpaceX  
WHERE Launch_Site LIKE 'CCA%'  
LIMIT 5
```

	Date1	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Land
1	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
3	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
4	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
5	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass



```
1 SELECT SUM(PAYLOAD_MASS_KG) as total_payload_mass
2 From SpaceX
3 WHERE Customer = 'NASA (CRS)'
```

```
4
```

total_payload_mass	
1	45596

Average Payload Mass by F9 v1.1

```
1 SELECT AVG(PAYLOAD_MASS_KG_) as average_payload_mass
2 From SpaceX
3 WHERE Booster_Version = 'F9 v1.1'
4
```

	average_payload_mass
1	2928.4

First Successful Ground Landing Date

```
1 SELECT min(Date1) as First_date
2 From SpaceX
3 WHERE Land = 'Success (ground pad)'
4
```


	First_date
1	01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
1  SELECT Booster_Version
2  From SpaceX
3  WHERE Land = 'Success (drone ship)'
4  and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
5
```

	Booster_Version
1	F9 FT B1022
2	F9 FT B1026
3	F9 FT B1021.2
4	F9 FT B1038.1
5	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes



```
1 SELECT Mission_Outcome, count(Mission_Outcome)
2 From SpaceX
3 GROUP by Mission_Outcome
4
```

	Mission_Outcome	count(Mission_Outcome)
1	Failure (in flight)	1
2	Success	99
3	Success (payload status unclear)	1

Boosters Carried Maximum Payload



```
1 SELECT Booster_Version, max(PAYLOAD_MASS__KG_)
2 From SpaceX
3 GROUP by Booster_Version
4
```

	Booster_Version	max(PAYLOAD_MASS__KG_)
1	F9 B4 B1039.2	2647
2	F9 B4 B1040.2	5384
3	F9 B4 B1041.2	9600
4	F9 B4 B1043.2	6460
5	F9 B4 B1039.1	3310
6	F9 B4 B1040.1	4990
7	F9 B4 B1041.1	9600
8	F9 B4 B1042.1	3500
9	F9 B4 B1043.1	5000
10	F9 B4 B1044	6092
11	F9 B4 B1045.1	362
12	F9 B4 B1045.2	2697
13	F9 B5 B1046.1	3600
14	F9 B5 B1046.2	5800
15	F9 B5 B1046.3	4000
16	F9 B5 B1046.4	12050

2015 Launch Records

```
1 SELECT Booster_Version, Launch_Site
2 From SpaceX
3 Where Land ='Failure (drone ship) '
4 And Date1 BETWEEN '01-01-2015' And '31-12-2015'
```

	Booster_Version	Launch_Site	
1	F9 v1.1 B1012	CCAFS LC-40	
2	F9 v1.1 B1015	CCAFS LC-40	
3	F9 v1.1 B1017	VAFB SLC-4E	
4	F9 FT B1020	CCAFS LC-40	
5	F9 FT B1024	CCAFS LC-40	

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

```
1 SELECT count(Land)
2 From SpaceX
3 Where Date1 BETWEEN '04-06-2010' And '20-03-2017'
```

	count(Land)
1	57

Section 4

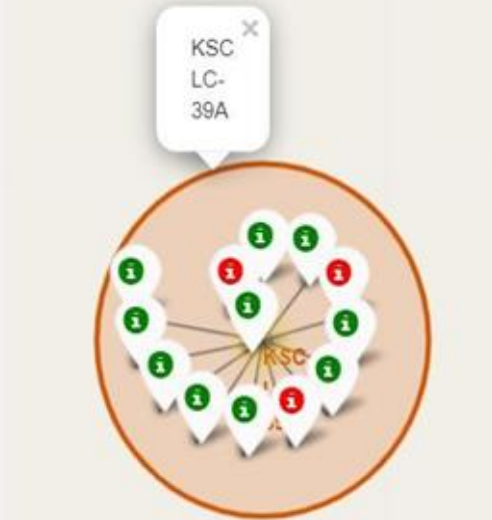
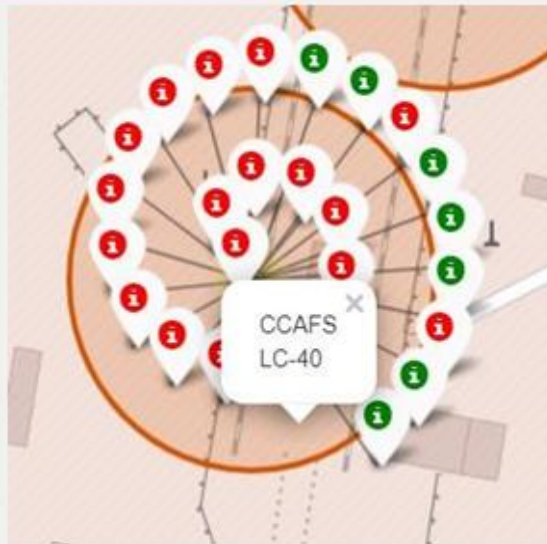
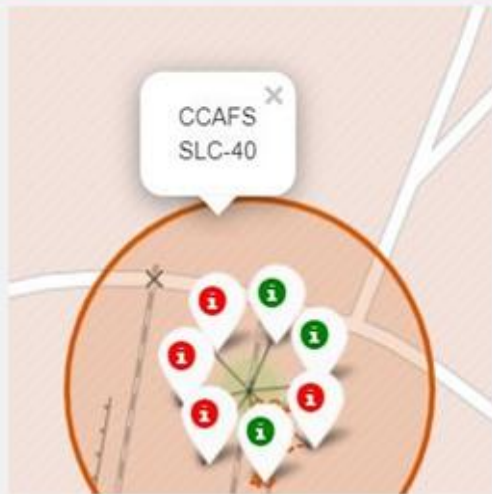
Launch Sites Proximities Analysis



All launch sites global map markers



Color Labelled Markers



Green Marker shows successful Launches and Red Marker shows Failures

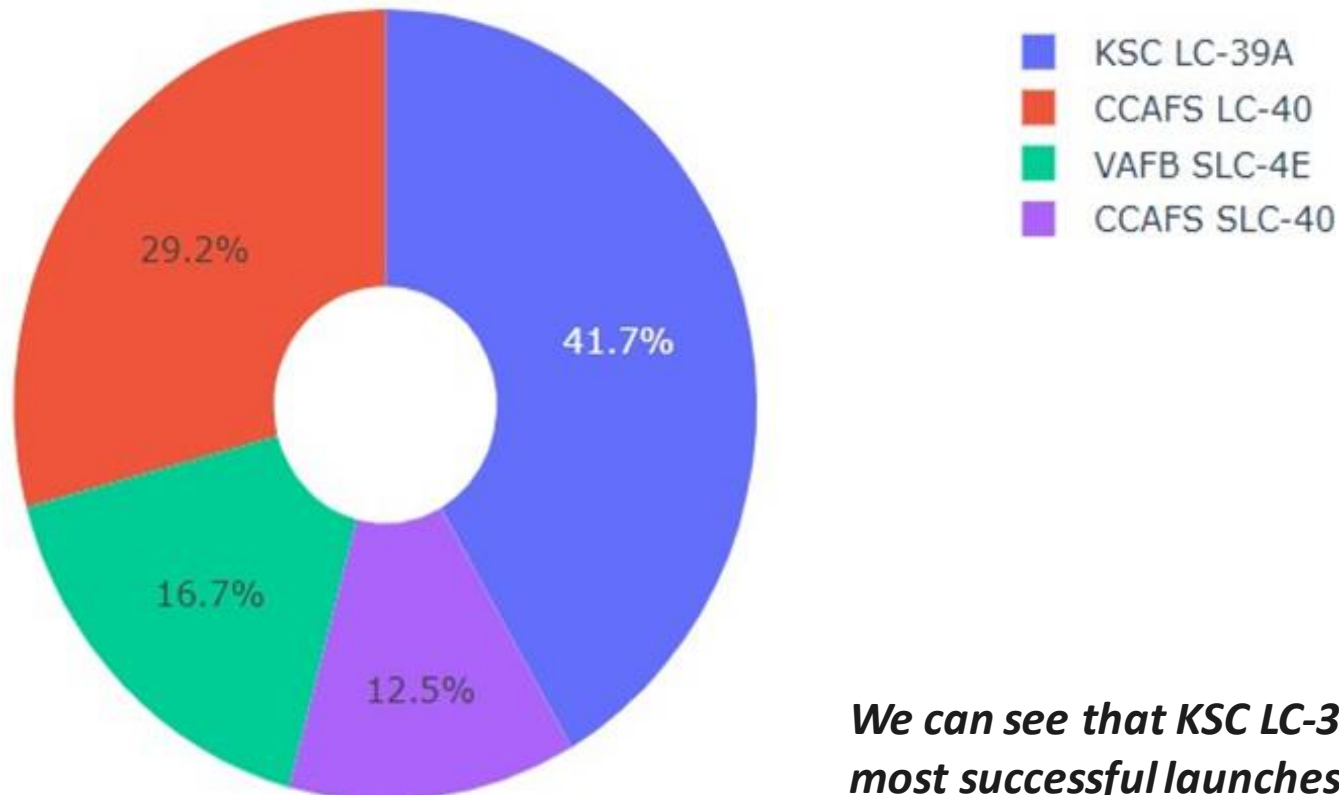


Section 5

Build a Dashboard with Plotly Dash

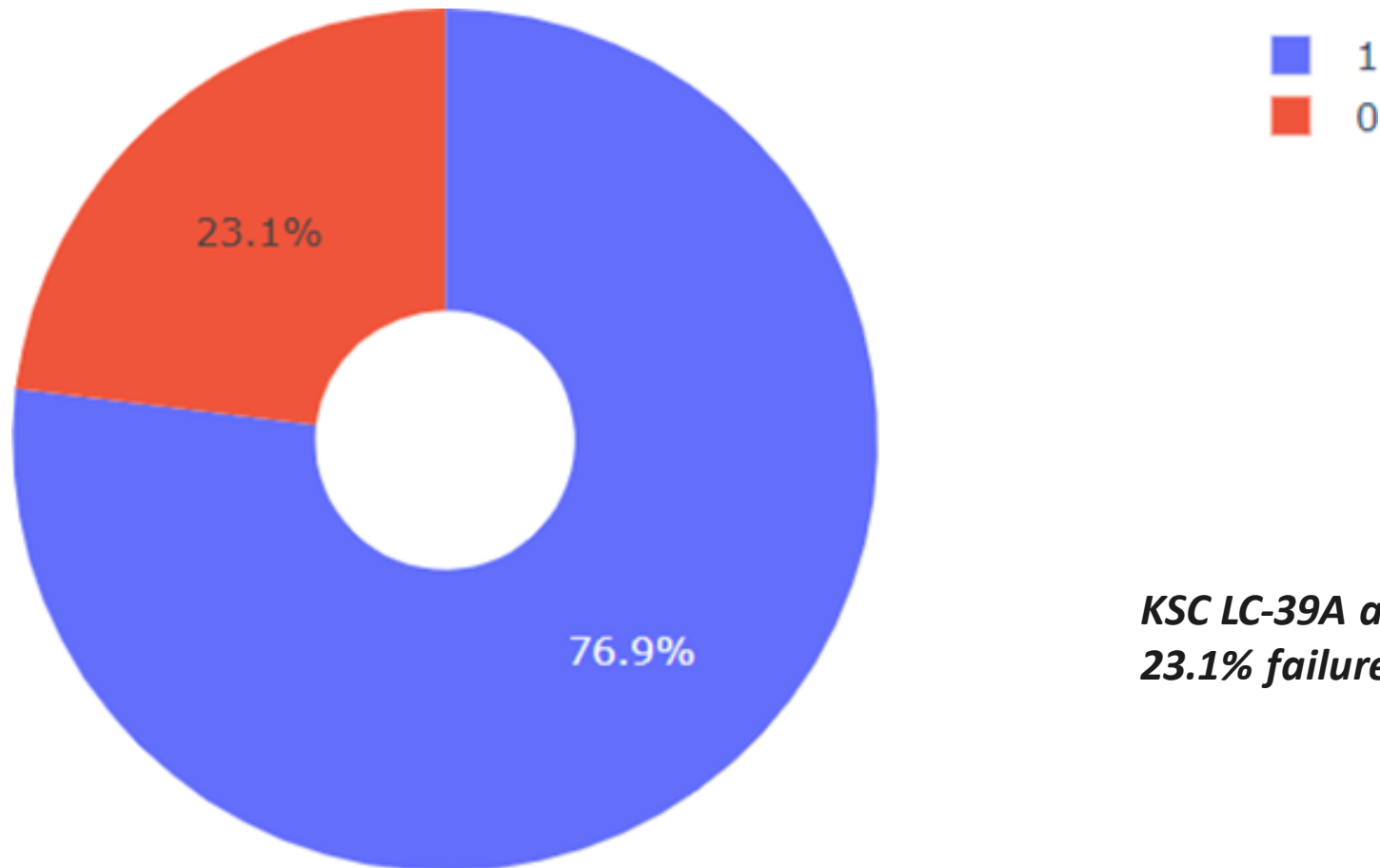
DASHBOARD – Pie chart showing the success percentage achieved by each launch site

Total Success Launches By all sites



We can see that KSC LC-39A had the most successful launches from all the sites

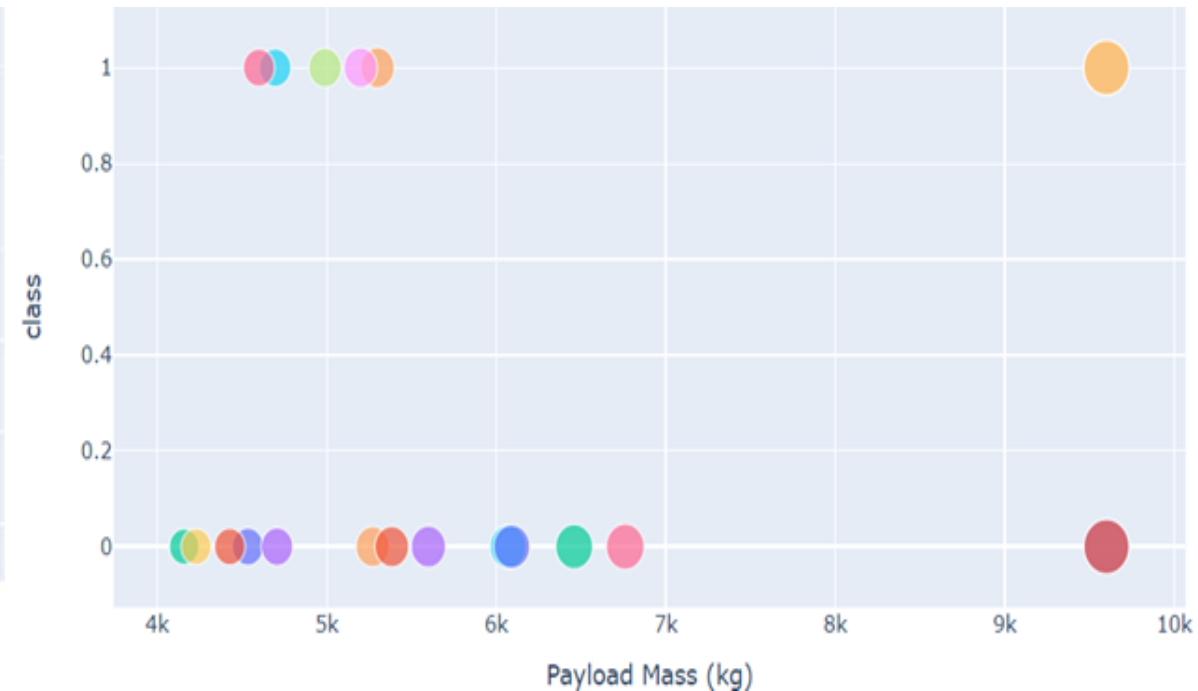
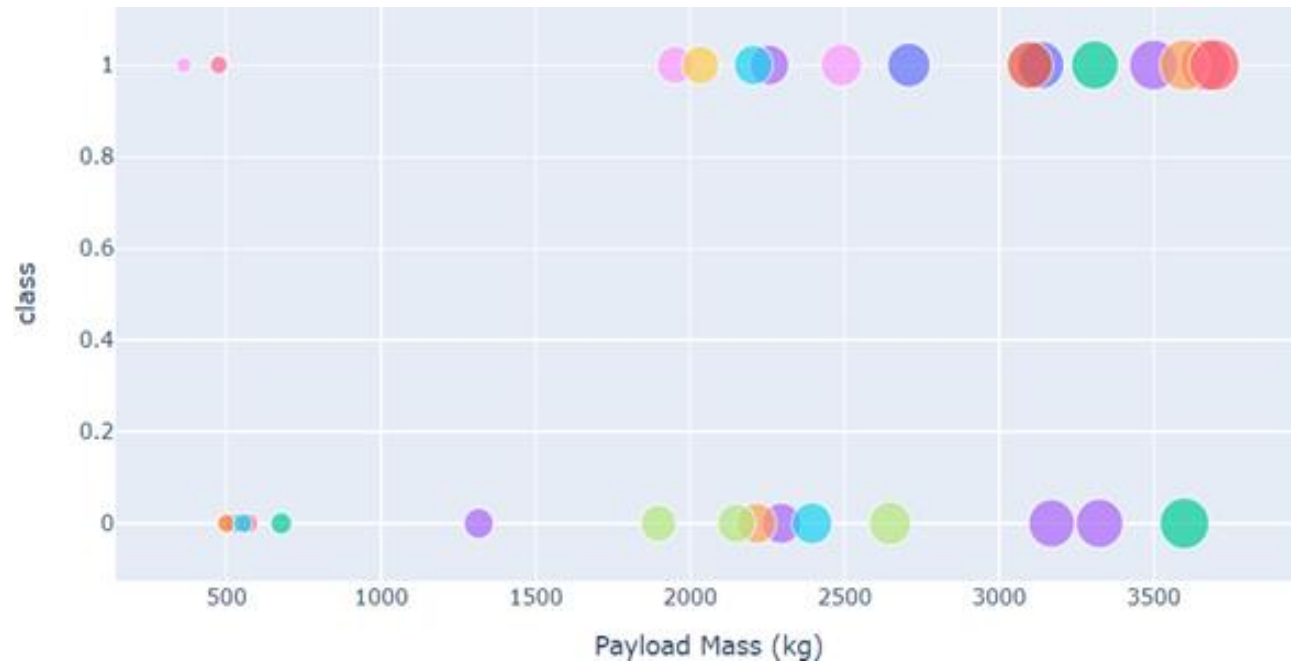
DASHBOARD – Pie chart for the launch site with highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

DASHBOARD – Payload vs. Launch Outcome

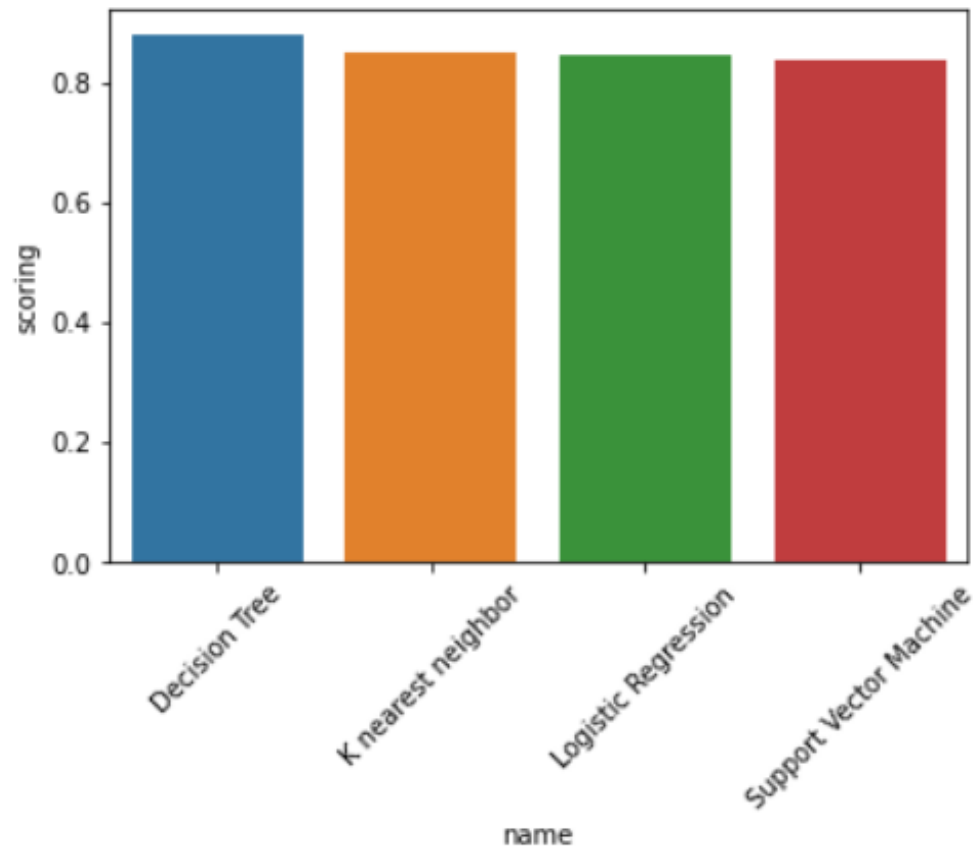
scatter plot for all sites, with different payload selected in the range slidero add text



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)



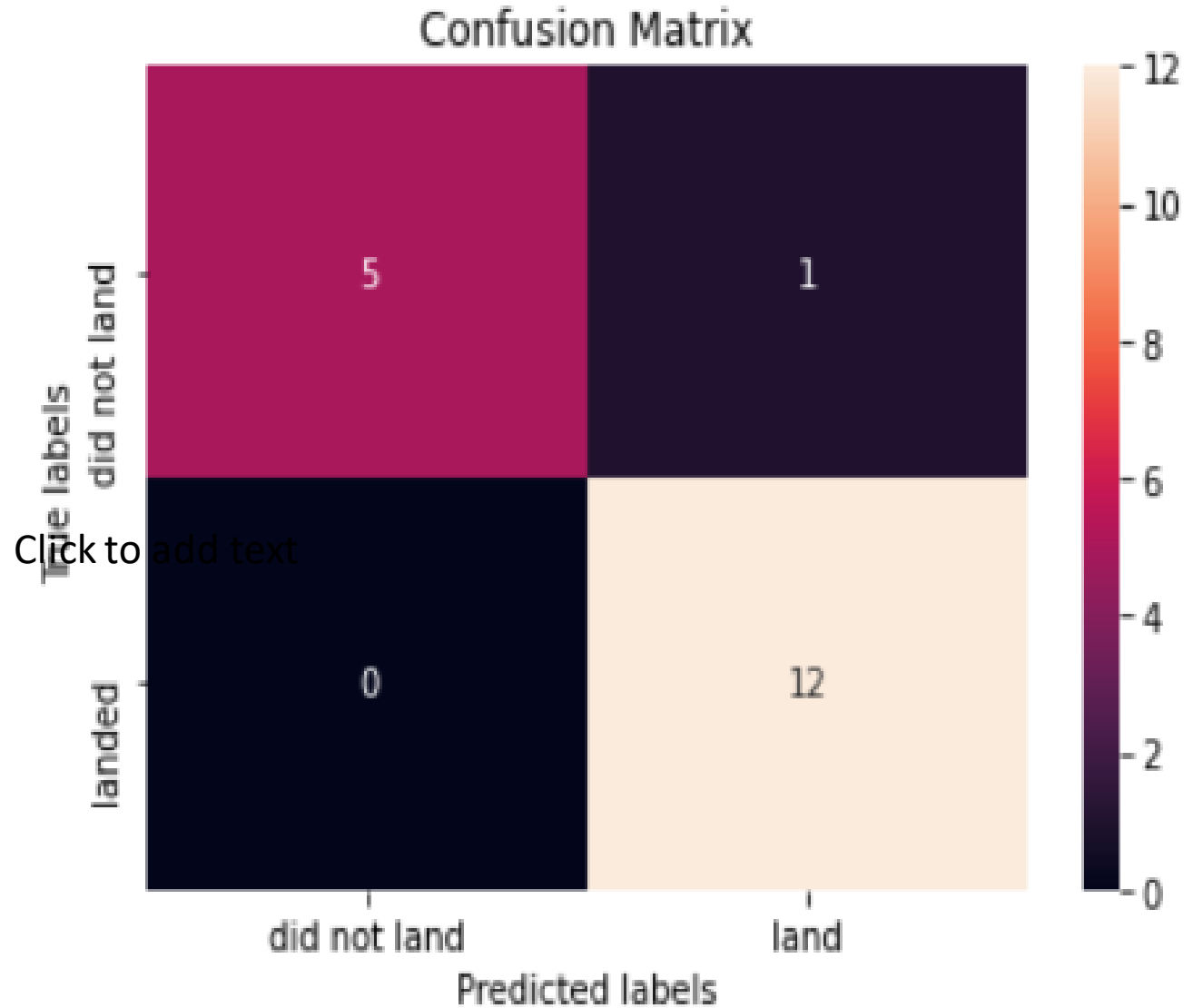
name	scoring
Decision Tree	0.876786
K nearest neighbor	0.848214
Logistic Regression	0.846429
Support Vector Machine	0.835714

Classification Accuracy

➤ So, we can see that Decision Tree gives us the highest accuracy score

Confusion Matrix

- Examining the confusion matrix, we see that Tree can distinguish between the different classes very well



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Thank you!

