Winning Space Race with Data Science

Vidnyani
Umathe

Outline

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Interactive map with Folium
  - Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - EDA results
  - Map and Dashboard screenshots
  - Model performance & predictive analysis

Introduction

- Background
  - 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 due to SpaceX technology that allows reuse of 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 our company wants to bid against SpaceX for a rocket launch.
- Questions
  - What is the relationship between rocket variables and the rate of success in landing the first stage?
  - How to what degree of accuracy can we predict successful first stage landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping from Wikipedia: List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
  - Determine the appropriate label for training supervised models
- Perform exploratory data analysis using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, SVM, Classification Trees, and KNN

Data Collection

- Data collected via direct request from SpaceX API
  - Key Data Collected: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSiteNumber, Outcome, LandingPad, Longitude, Latitude
- Additional data collected from the Wikipedia page: List of Falcon 9 and Falcon Heavy launches
  - Key Data collected: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Date, Time

Data Collection - SpaceX API

1. Request data from SpaceX API

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

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

2. Convert response to a JSON file

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

3. Use custom functions to extract data

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

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

```
# Call getBoosterVersion
getBoosterVersion(data)
```

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

4. Combine columns into a dictionary for data frame

```
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.from_dict(launch_dict)
```

5. Filter data frame and export to CSV

```
# Remove the Falcon 1 launches keeping only the Falcon 9 launches.
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

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

Request data from API

Convert response to JSON from API

Extract data and export to .CSV

Data Collection - Scraping

1. Get response from url

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

2. Create BeautifulSoup object

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

3. Find all tables and assign result to list

```
# Use the find_all function in the BeautifulSoup object, with element t
ype `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4. Loop to extract column name

```
column_names = []
```

```
for header in first_launch_table.find_all('th'):
    name = extract_column_from_header(header)
    if name != None and len(name) > 0:
        column_names.append(name)
```

5. Create an empty dictionary with column names as 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. Fill dictionary (not shown), Create df, export to CSV

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

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

Get Response from Wikipedia

Use Beautiful Soup to extract data

Extract and export data to .CSV

Data Wrangling

- Mission outcomes (failures and successes) are represented for specific landing areas such as Ocean, RTLS (ground pad), and ASDS (done ship)
- These outcomes will serve as our training labels:
  - Failed landing = 0
  - Successful landing = 1

Get total launches at each site for each orbit → Get number of mission outcomes for each orbit type → Extract landing outcome labels and calculate sucesss → Export to .csv

EDA with Data Visualization

- Scatter charts are utilized to assess the correlation between the following variables:
  - Flight Numbers & Launch Site
  - Payload & Launch Site
  - Flight Number & Orbit Type
  - Payload & Orbit Type
- A bar chart is utilized to to compare the Success Rate for the categorical variable Orbit Type
- A line chart is utilized to illustrate the overall Success Rate for each Year

EDA with SQL

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

- Assess launch sites and their distance to its proximities such as railways, highways, coastlines and cities
- Generated Folium map and created the following markers and lines to assess launch sites and their proximities:
  - Launch Sites (markers)
  - Successful/Failed Launches (markers)
  - Distances between launch sites and proximities (lines)
  - Go to notebook with Interactive map

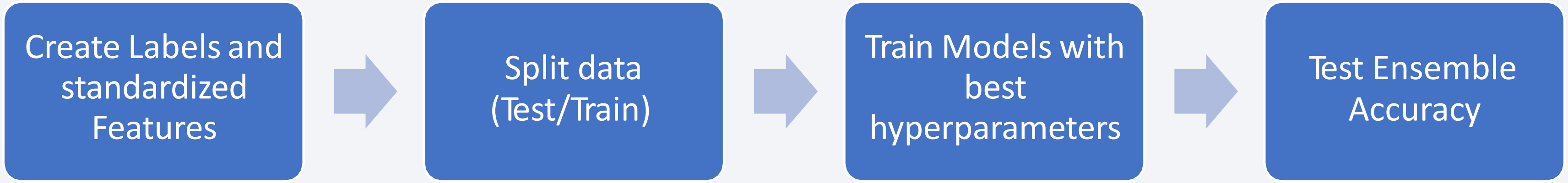
Build a Dashboard with Plotly Dash

- Plotly Dash was utilized to generate a dashboard with the following figures:
  - Pie chart showing % successful launches
    - The chart can represent the success rate of all sites or individual sites.
  - Scatter plot illustrating the correlation between Outcomes and Payload mass across different booster versions
    - Can select all sites or individual launch sites
    - Can constrain range of Payload mass values with a slider

More details on GitHub

Predictive Analysis (Classification)

- EDA to determine training labels
  - Create Class column to represent outcomes (Success/Fail = 1/0)
  - Standardize data with sklearn.preprocessing.StandardScaler()
  - Split into training and test data via train_test_split()
- Find best hyperparameters for Logistic Regression, SVM, Classification Trees, and KNN
  - Train model on training data and select hyperparameters using GridSearchCV
  - Calculate accuracy on test data using score()



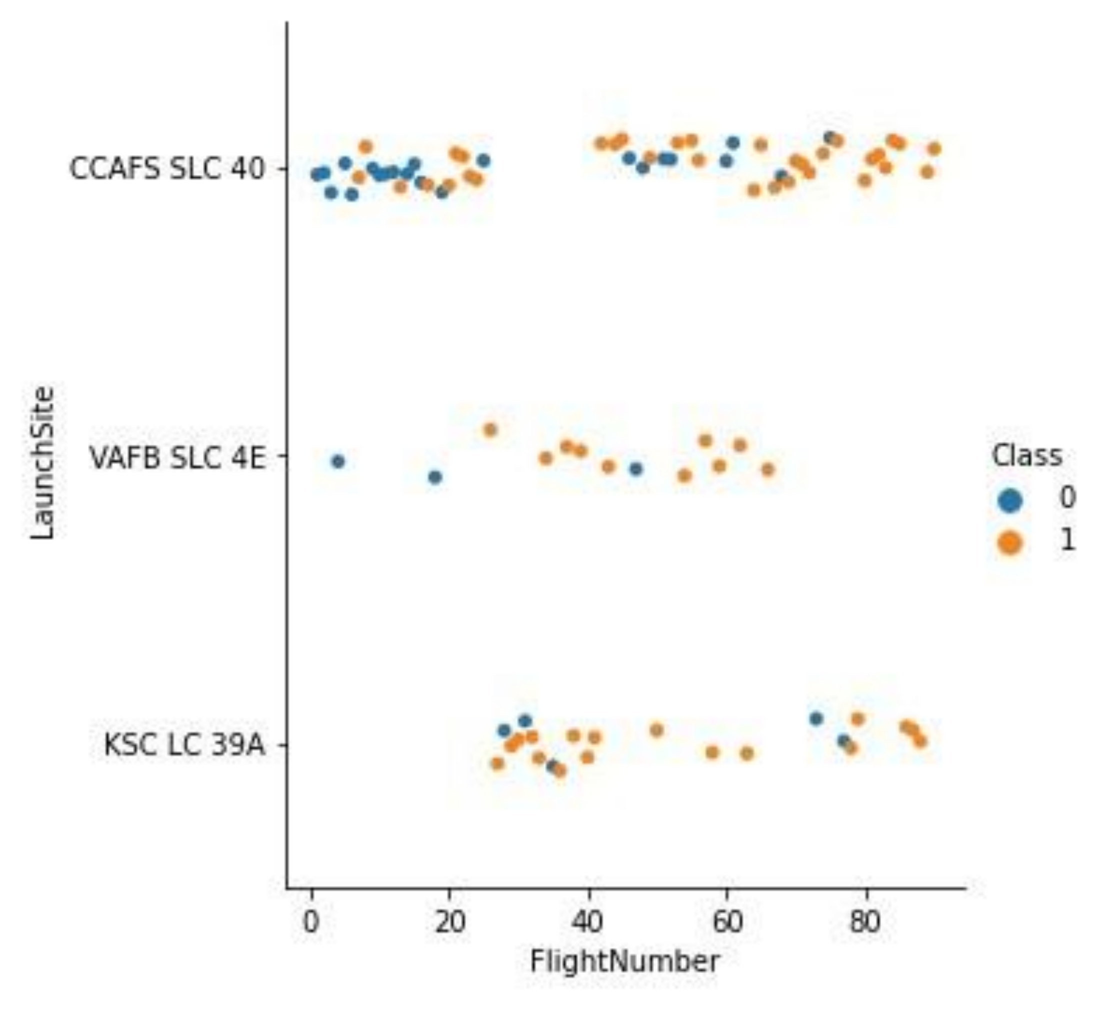
More details on GitHub

Section 2

Insights drawn from EDA

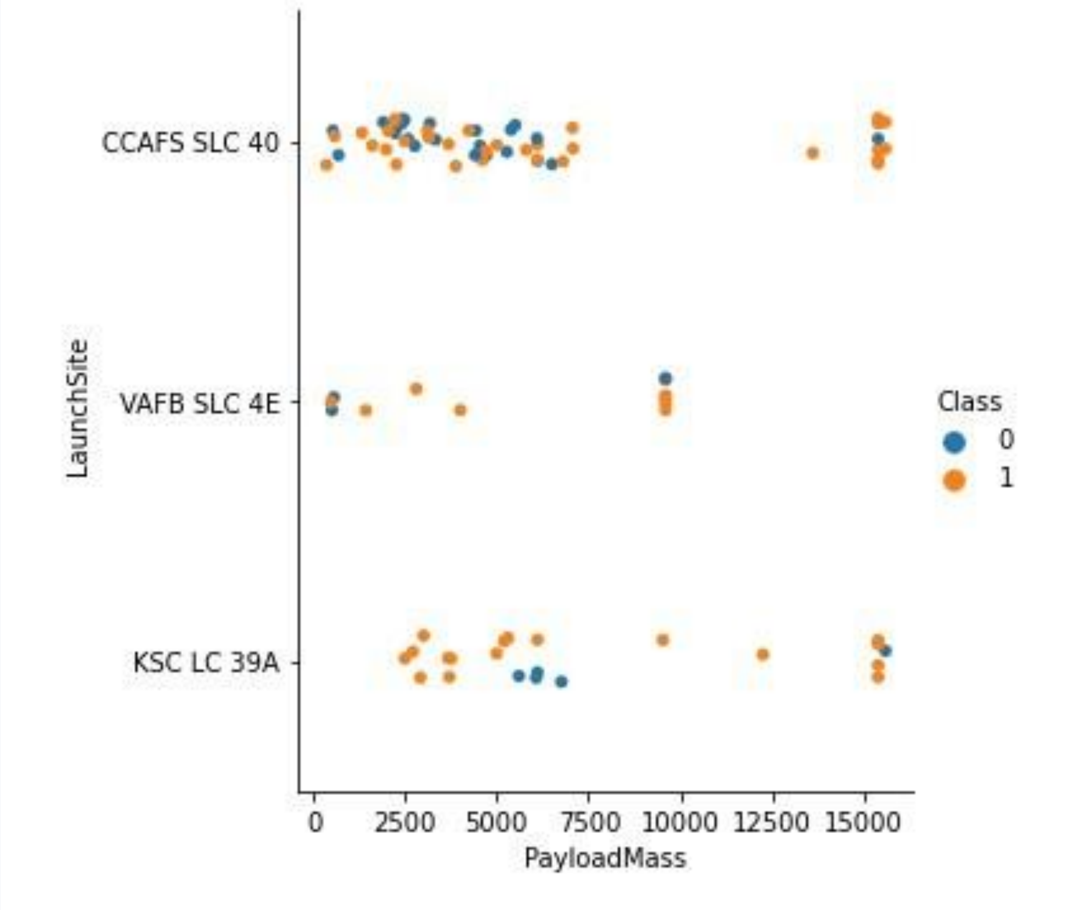
Flight Number vs. Launch Site

- Class represents launch outcomes:
    - 0 (blue) = Failure
    - 1 (orange) = Success
- With each subsequent flight, we see more successful launches
- Note increase in success after the 20th flight, suggesting a progression in methodology/technology.



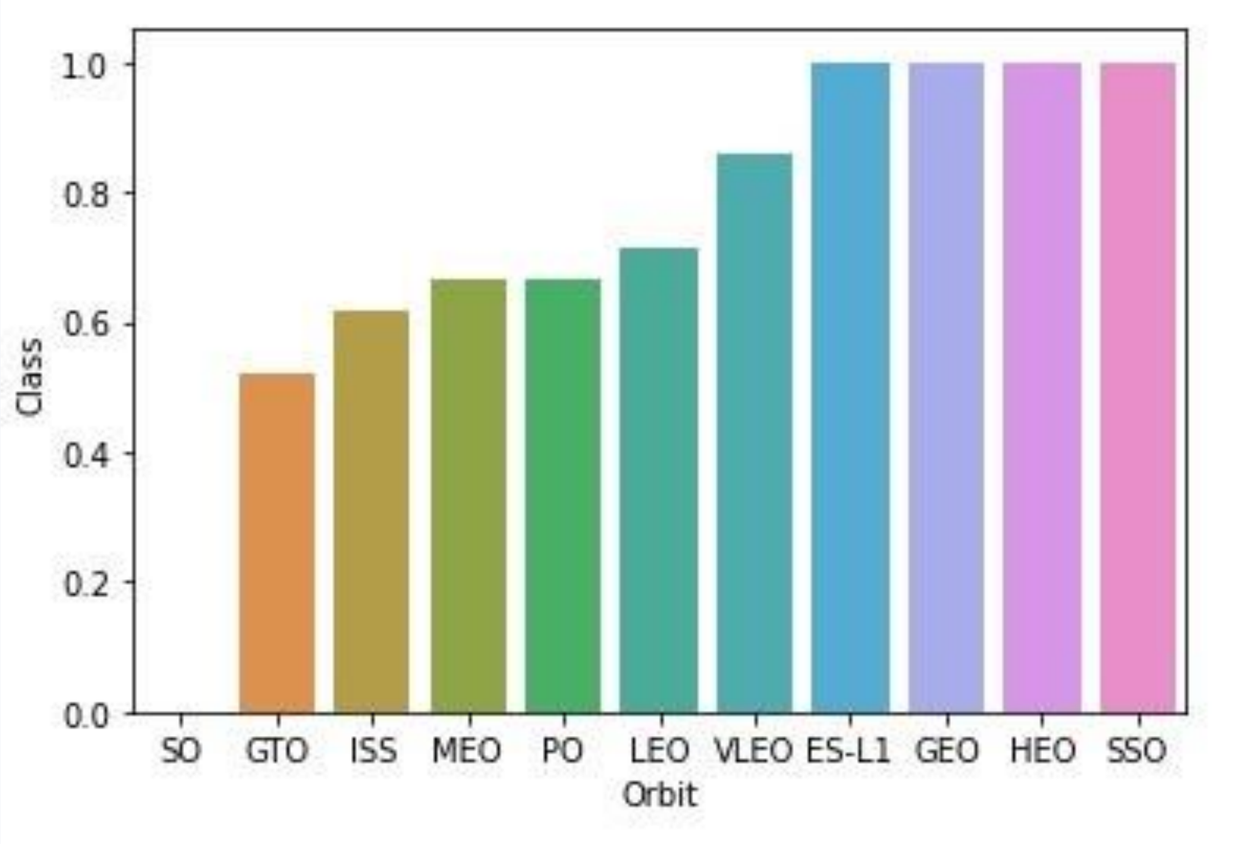
Payload vs. Launch Site

- We observe no clear correlation when assessing the relationship between launch sites and their payload mass
- Note that the VAFB-SLC launch site is limited to a payload mass under 10000 kg



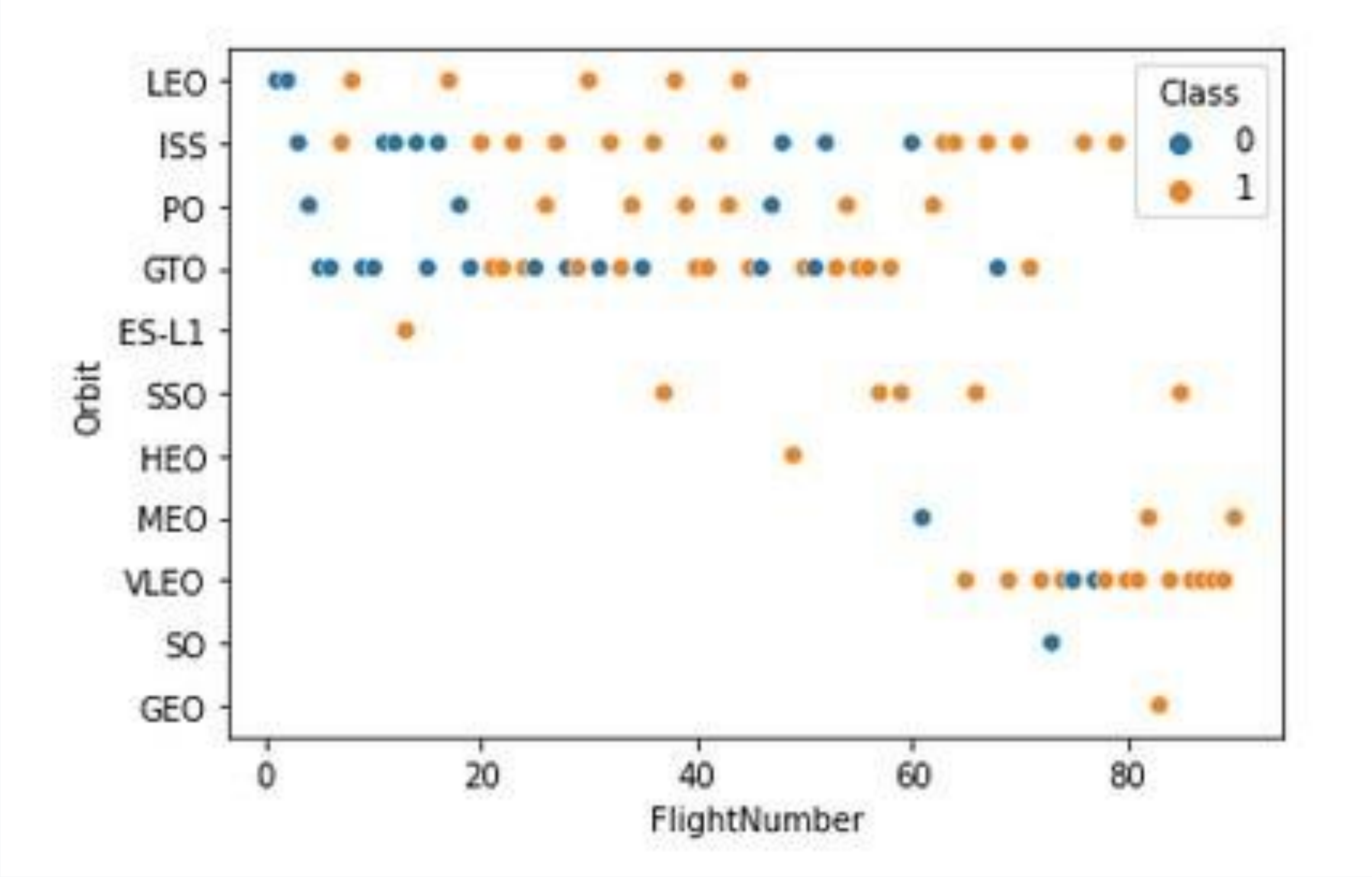
Success Rate vs. Orbit Type

- Select orbit types have 100% success rates (ES-L1, GEO, HEO, and SS0)
- Other obit types have greater than 60% success rate with the exception being GTO (at 50%) and SO which only had one launch and one failure.



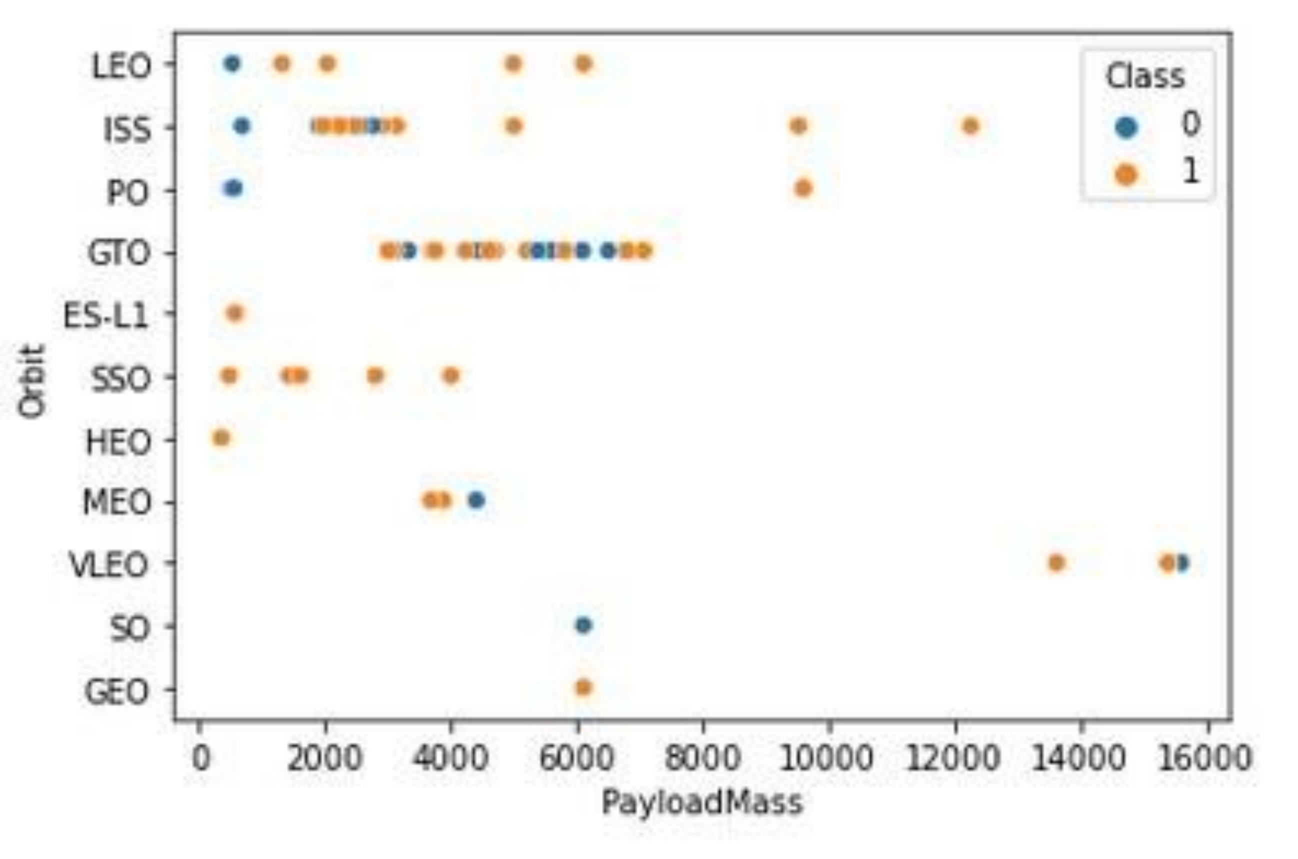
Flight Number vs. Orbit Type

- LEO orbit: 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.



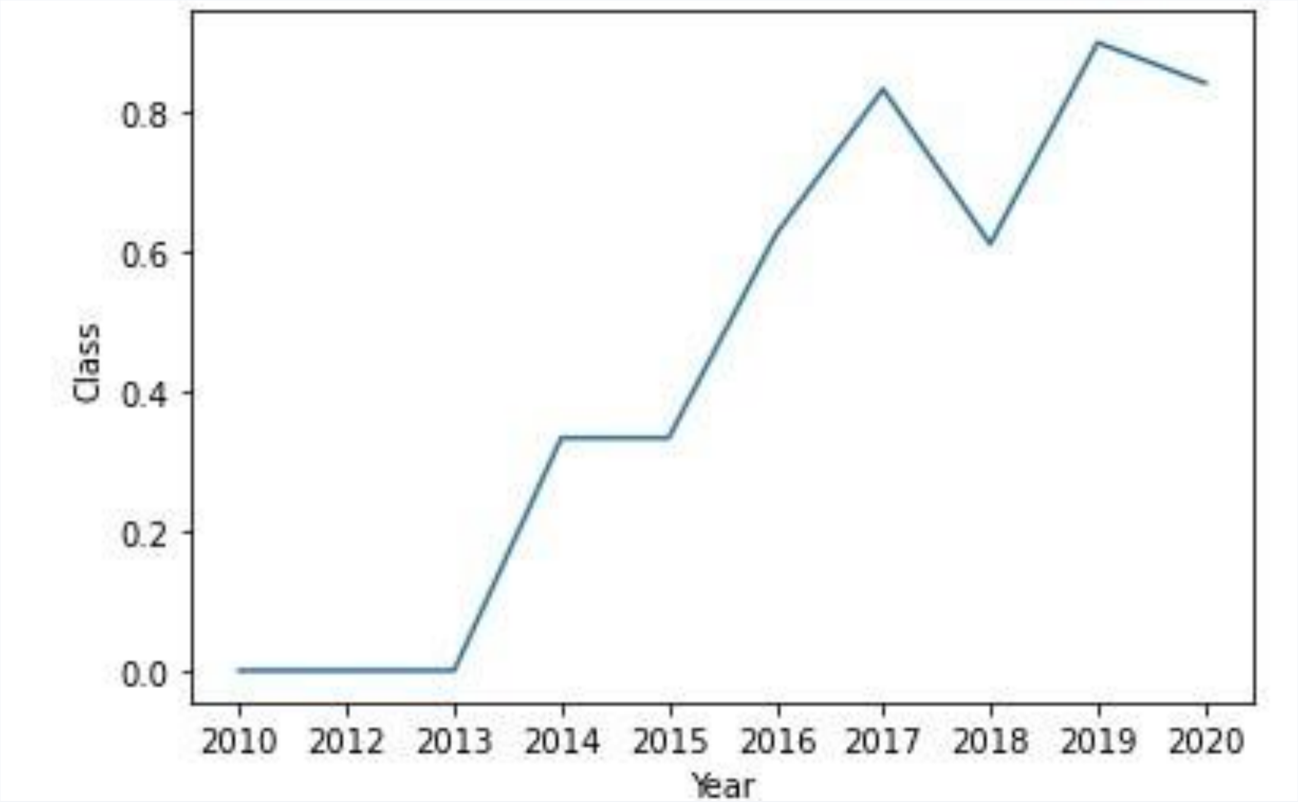
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO a clear pattern is difficult to discern.



Launch Success Yearly Trend

- Observe an increase in success rate from almost 40% in 2014 to 80% in 2020, with a sizable reduction in 2018.



All Launch Site Names

- Query data table with SQL to select unique launch site names:
  - Result displays 4 unique launch sites

```
%sql SELECT distinct(launch_site) FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- Query data table with SQL to select 5 records where launch sites begin with the string 'CCA':

```
%%sql SELECT * FROM SPACEXTBL
WHERE(launch_site like 'CCA%') LIMIT 5
```

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

- Calculate the total payload carried by boosters from NASA:
  - Use SUM() to calculate sum of payload mass
  - WHERE is used to restrict dataset to only 'NASA (CRS)' as customer

```
%%sql SELECT SUM(payload_mass__kg_)
as total_payload_mass_kg
FROM SPACEXTBL
WHERE customer = 'NASA (CRS)'
```

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1:
  - AVG() function used to calculate the average payload mass
  - WHERE is used to restrict the dataset to Booster_version F9 v1.1.

```
%%sql SELECT AVG(payload_mass__kg_)
as avg_payload_mass_kg
FROM SPACEXTBL
WHERE booster_version ='F9 v1.1'
```

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad:
  - MIN() function used to find the earliest DATE.
  - WHERE is used to restrict the dataset to Landing__outcomes that are Success (ground pad).

```
%%sql SELECT min(DATE)
as first_success
FROM SPACEXTBL
WHERE mission_outcome = 'Success'
```

first_success
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 kg but less than 6000 kg:
  - Filter the dataset to WHERE landing__outcome is Success (drone ship).
  - Use the AND operator to apply an additional WHERE condition restricting results to payload_mass__kg_ between 4000 kg and 6000 kg.

```
%%sql SELECT booster_version FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)'
AND (payload_mass__kg_ BETWEEN 4000 AND 6000)
```

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 failure mission outcomes:
  - COUNT() function to calculate the total number of mission outcomes.
  - GROUP BY rows that have the same values into to find the total number in each distcint Mission_outcome.

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass:
  - Use a subquery to find the maximum value of the payload by using MAX() function
  - Filter the dataset to perform a search if PAYLOAD_MASS__KG_ is the maximum value of the payload.

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

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:
  - Restrict the dataset to WHERE landing__outcome is Failure (drone ship).
  - Use AND operator to display an additional record WHERE YEAR is 2015.

```
%%sql SELECT landing__outcome,
    booster_version,
    launch_site FROM SPACEXTBL
WHERE YEAR(DATE) = '2015'
AND landing__outcome = 'Failure (drone ship)'
```

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

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

- Rank 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:
  - WHERE used to restrict the dataset to dates between 2010-06-04 and 2017-03-20.
  - ORDER BY used to sort the records by total number of landings
  - DESC used to sort the records in descending order.

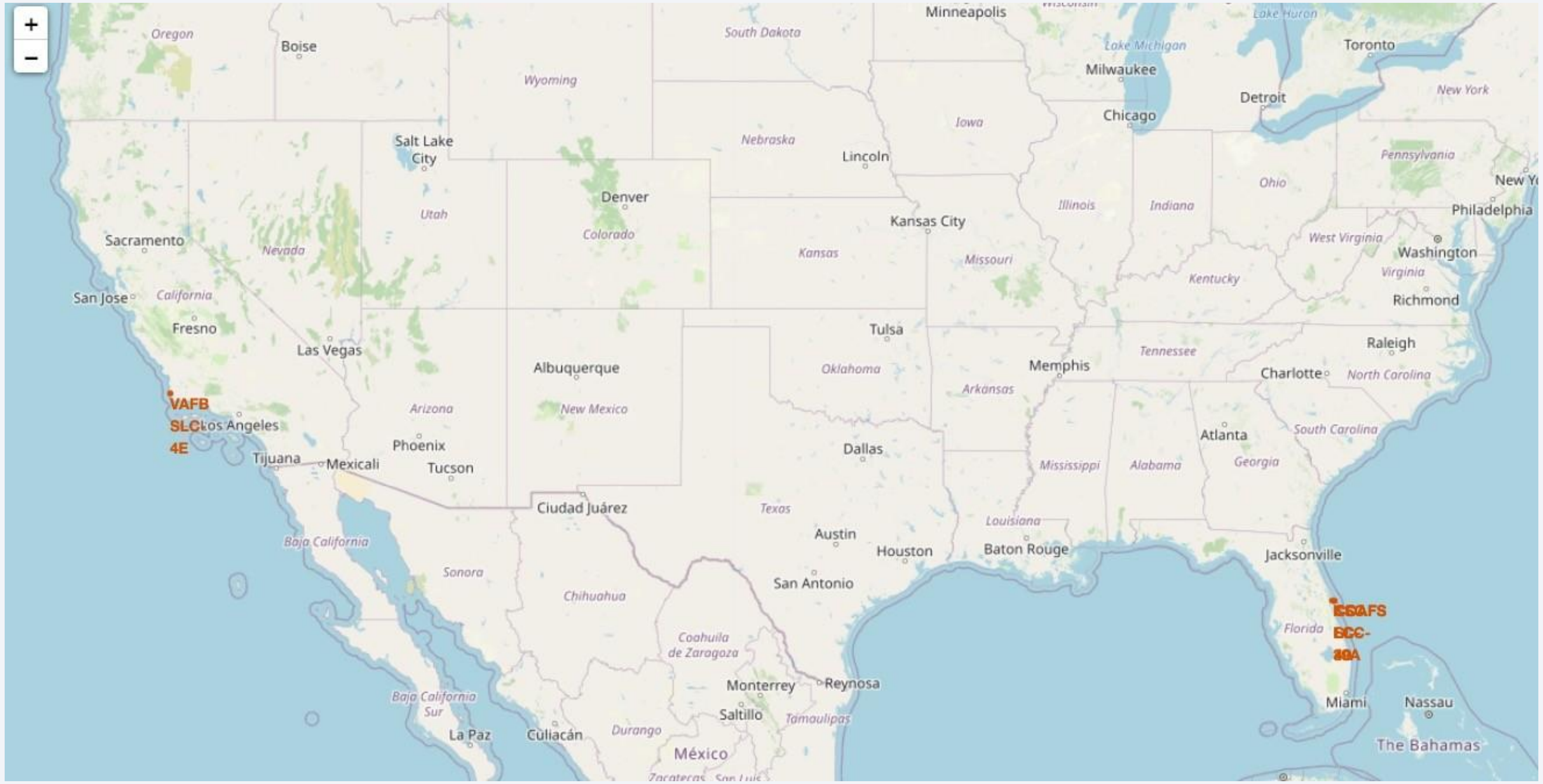
```
%%sql SELECT landing__outcome,
    COUNT(landing__outcome)
    AS total
FROM SPACEXTBL
GROUP BY landing__outcome
ORDER BY total DESC
```

landing__outcome	total
Success	38
No attempt	22
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Location of SPACE X Launch Sites

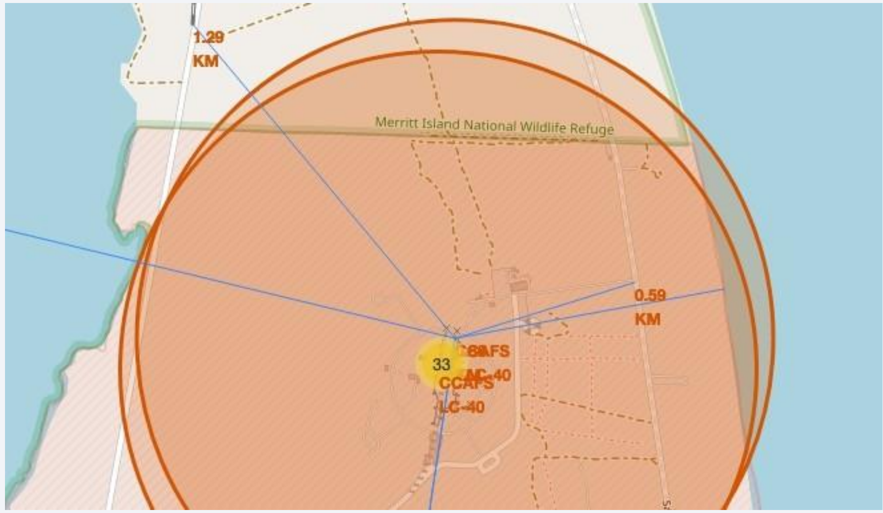


All launch sites are near the coastal United States

Color-labeled Markers for Launch Outcomes

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

Launch Site Proximities

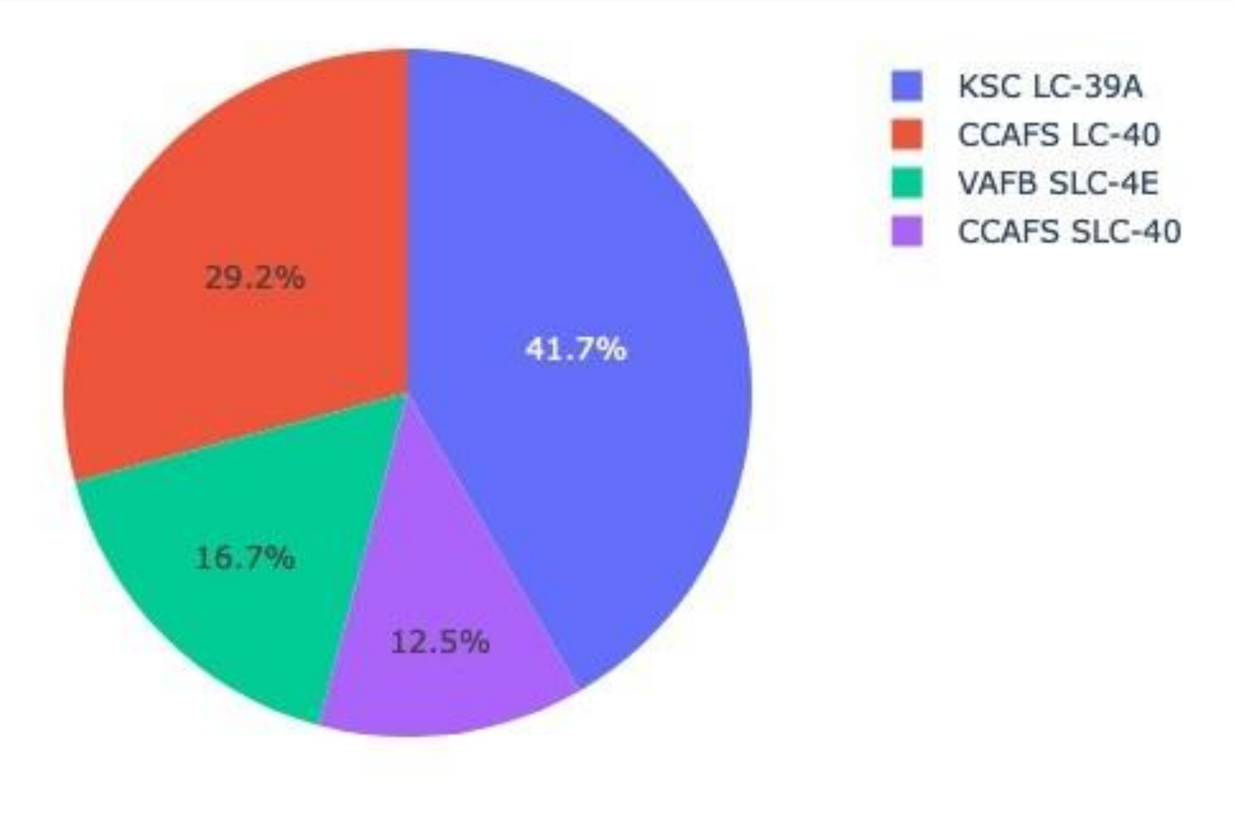


It can be found that the launch site is close to railways and highways for transportation of equipment or personnel, and is also close to coastline and relatively far from the cities so that launch failure does not pose a threat.

Section 4

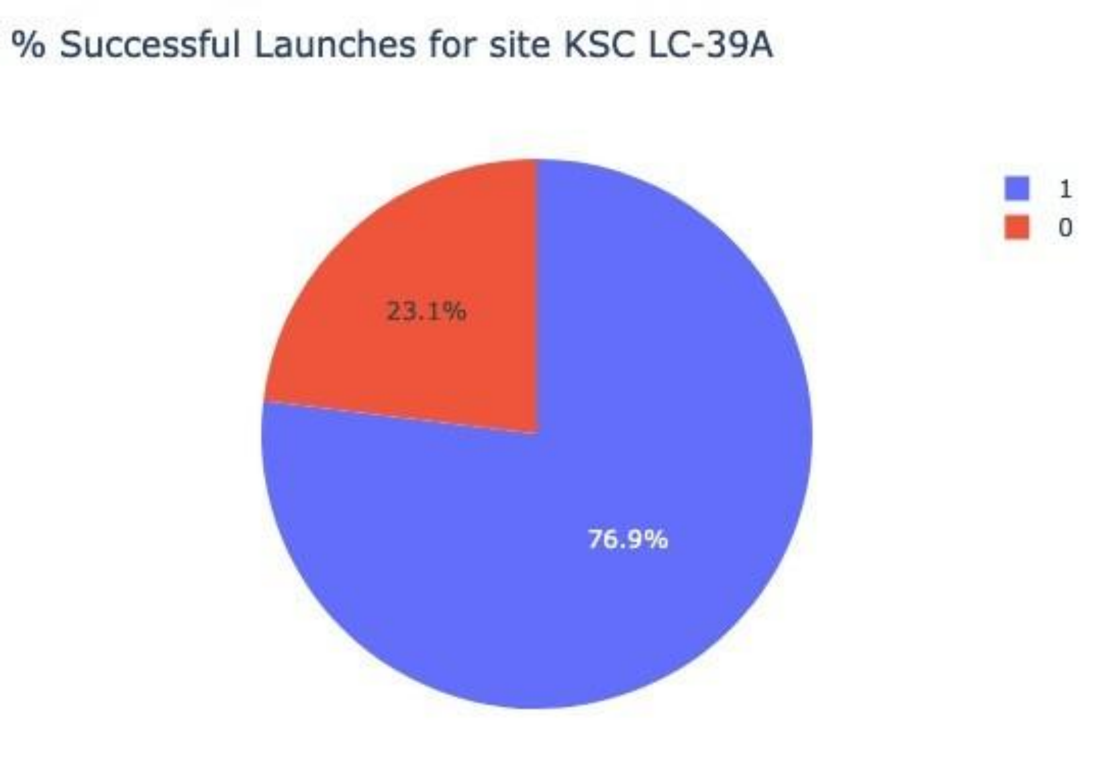
Build a Dashboard with Plotly Dash

% Success for All Launch Sites



KSLC-39A records the most launch success among all sites. On the other hand, VAFB SLC-4E has the fewest launch success, possibly likely due to small sample size

Launch Site with Highest % Success



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

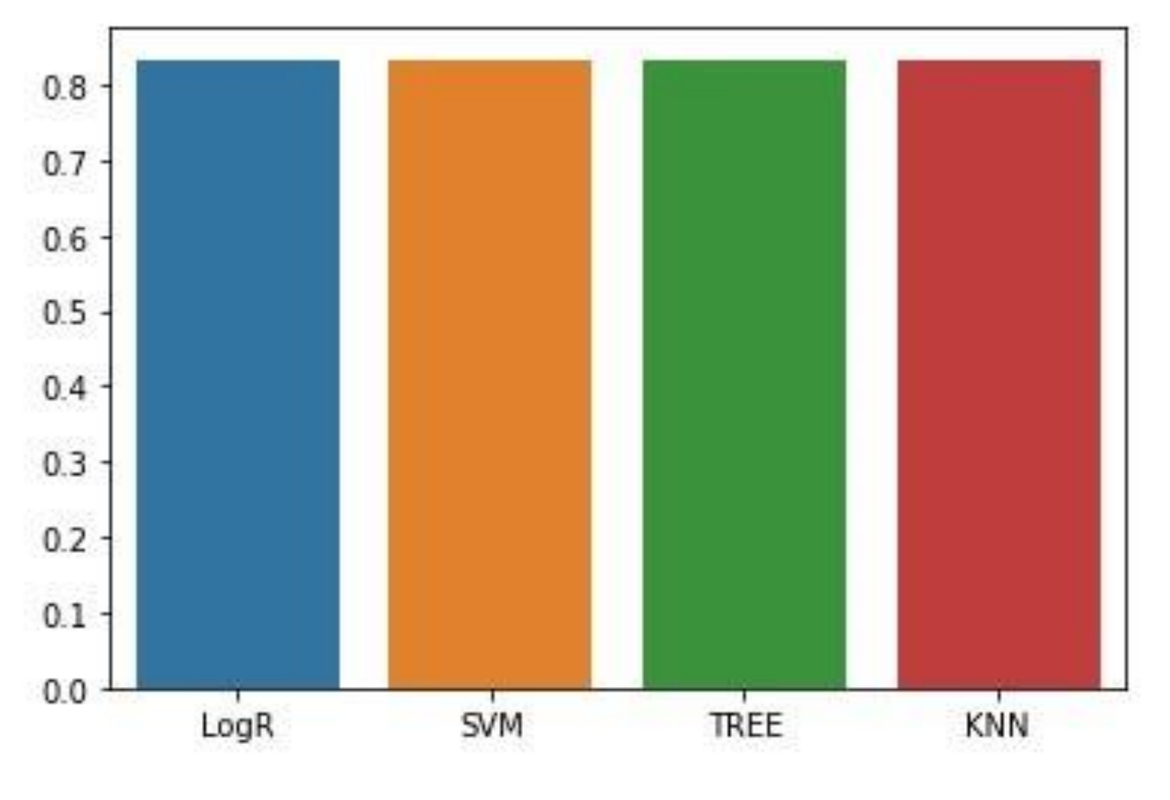
Payload v Launch Outcomes by Booster Version

Low weighted payloads (left) appear to have higher success rates than that of heavy weighted payloads (right). Note that there are more launches with lower payloads

Section 5

Predictive Analysis (Classification)

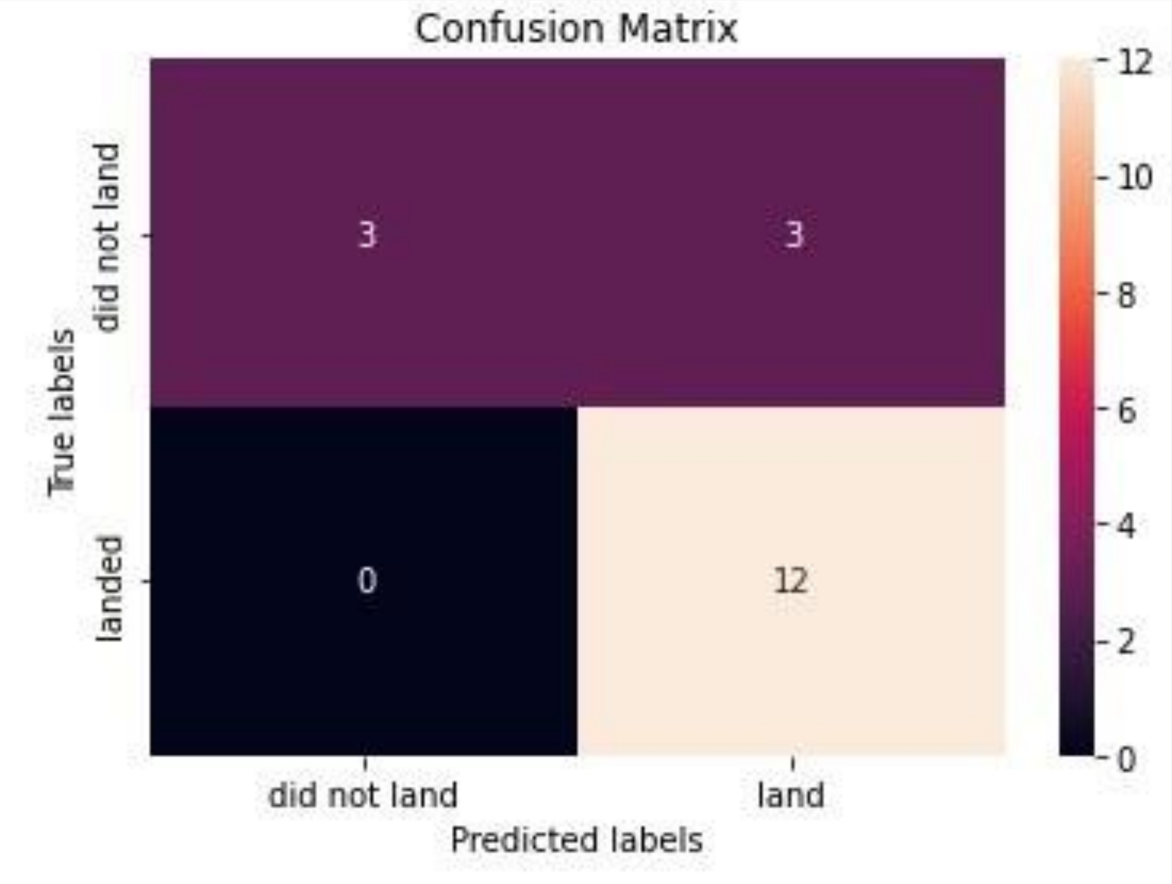
Classification Accuracy



In the test set, the accuracy of all models was 83.33%. Given the small size of the data set, bootstrapping for more data may be key to obtain a better model.

Confusion Matrix

- The confusion matrix consistent across all models due to almost identical model performance.
- The models predicted:
  - 12 successful landings when the true label was successful
  - 3 failed landings when the true label was failure.
  - 3 incorrect predictions for 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 up to 80% around 2020.
- 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.
- All models have the same accuracy (83.33%), but more data may be necessary to determine the optimal model.

Appendix

- GitHub URl

Thank you!