IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tiffany Simms Lindsey
March 25, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection: API and Web Scraping
  - Exploratory Data Analysis: SQL Queries and Data Visualization
  - Visual Analysis: Folium Lab and Interactive Dashboard
  - Predictive Analysis:
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree Classifier
    - K-Nearest Neighbor
- Summary of all results
  - Trend of launch success rate is overall positive to 2020
  - The best method for predictive analysis was determined to be the Decision Tree Classifier

Introduction

- Project background and context
  - Commercial space travel is a growing market with several companies using manned and unmanned spaceflights to accomplish a wide variety of goals, including satellite placement, suborbital spaceflights for civilians, and sending spacecraft to the International Space Station. Rocket launches are typically performed in two stages. Stage 1 is where the majority of thrust comes from, and Stage 2 utilizes more controlled thrust to get the payload to its destination (orbit, ISS, etc). SpaceX is perhaps the most successful of the companies performing these tasks, with rocket launches costing approximately 62% less than other providers. One reason SpaceX is successful is that their rocket launches are relatively inexpensive because they can reuse the first stage. If we can determine if their first stage will land successfully and be able to be reused, we can determine the approximate cost of a launch.
- Problems you want to find answers
  - What factors influence the success of a launch and ability to reuse Stage 1?
  - What method will best predict the success of a launch that has not yet occurred?

Section 1

Methodology

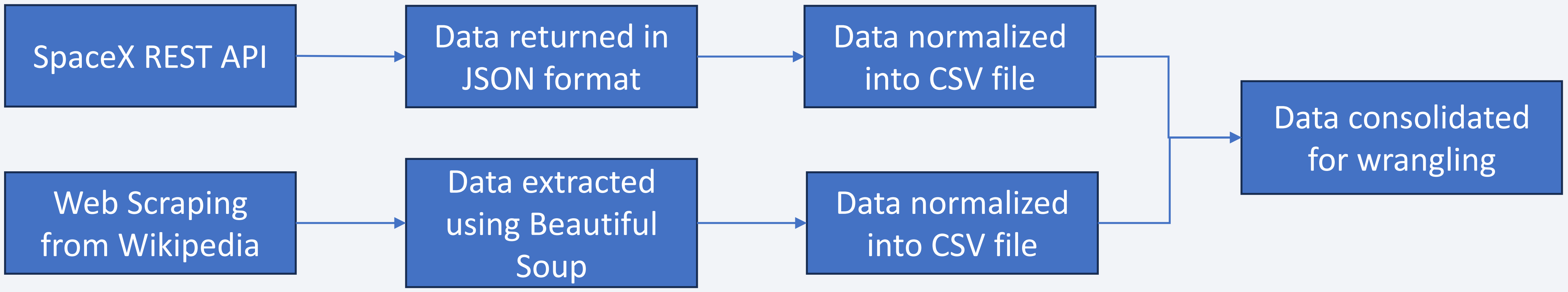
Methodology

Executive Summary

- Data collection methodology:
  - Performed request to SpaceX AP and web scraping of Wikipedia tables
- Perform data wrangling
  - Data was grouped by launch site, orbit, and landing outcomes (0=fail, 1 = success)
  - An average of landing outcome classes give us a success rate
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models are built by creating a testing and training data set from the data available. The training set is used to train the model and the testing set is used to test the model. Each model is evaluated by fitting the training data to the model and testing the model's accuracy. A confusion matrix can help see different features of the data for each model.

Data Collection

- Made request to SpaceX REST API to pull SpaceX launch data
    - Provides information about launch date, booster version, payload mass, orbit, launch site and coordinates, launch outcome, details about rocket used, and whether the rocket was able to be reused
- Web scraping with Beautiful Soup was used to retrieve data from Wikipedia tables
    - Provides information on launch date, flight number, launch site, booster version, payload and payload mass, orbit, customer, launch outcome, and booster landing



Data Collection – SpaceX API

1. API to SpaceX REST API
2. Convert API response to JSON
3. Clean data and create data dictionary
   - Use mean to replace missing data for payload mass
4. Export data to CSV file

- [API Data Collection Jupyter Notebook]()

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

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

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

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

Data Collection – Web Scraping

1. Pull text data from web page
2. Create Beautiful Soup object
3. Find all tables available
4. Pull data from launch table and find all column names
5. Create data frame by parsing HTML launch tables
6. Export data to CSV file

- [Web Scraping Jupyter Notebook]()

```python
req = requests.get(static_url)
data = req.text
```

```python
soup = BeautifulSoup(data, "html.parser")
```

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

```python
first_launch_table = html_tables[2]
print(first_launch_table)
```

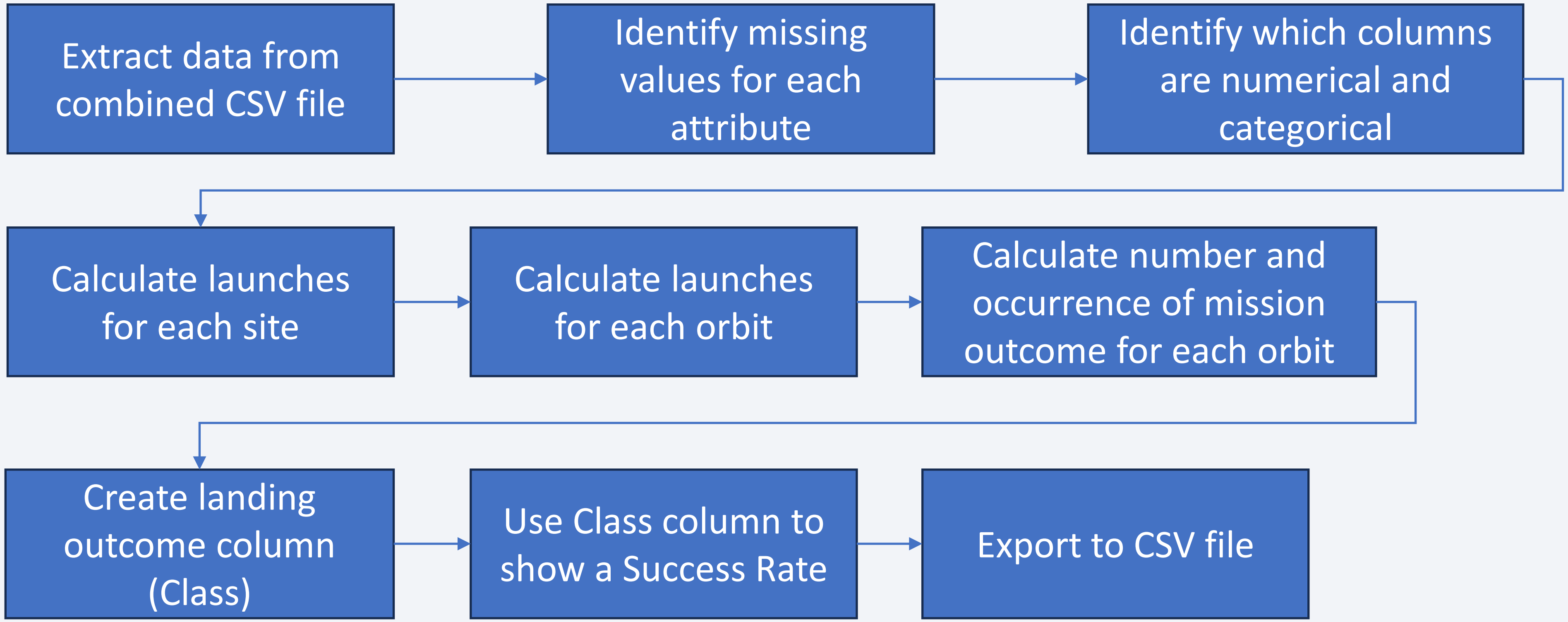
```python
...names = first_launch_table.find_all('th')

for i, col_name in enumerate(col_names):
    ...t_column_from_header(col_name)
    ...t None and len(name) >0:
    ...mes.append(name)
```

```python
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']=[]
```

```python
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

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

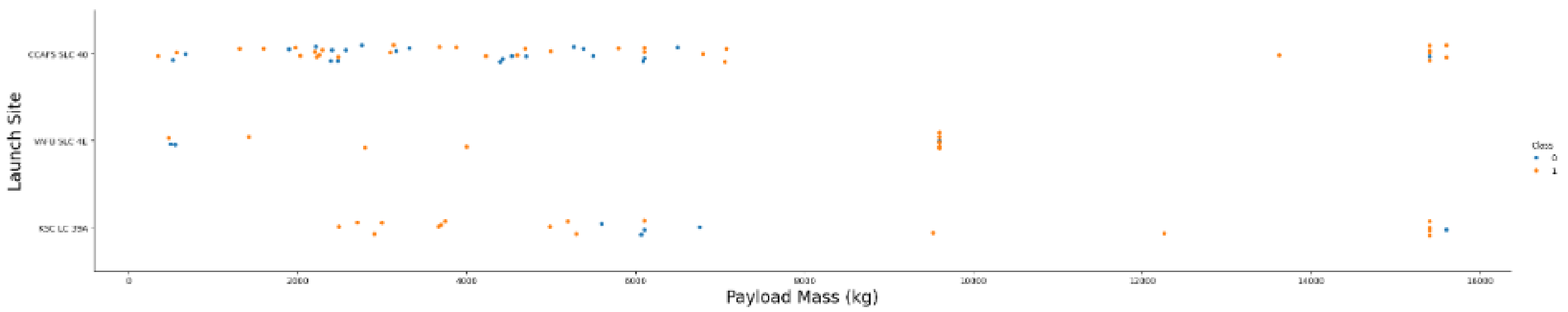
Data Wrangling



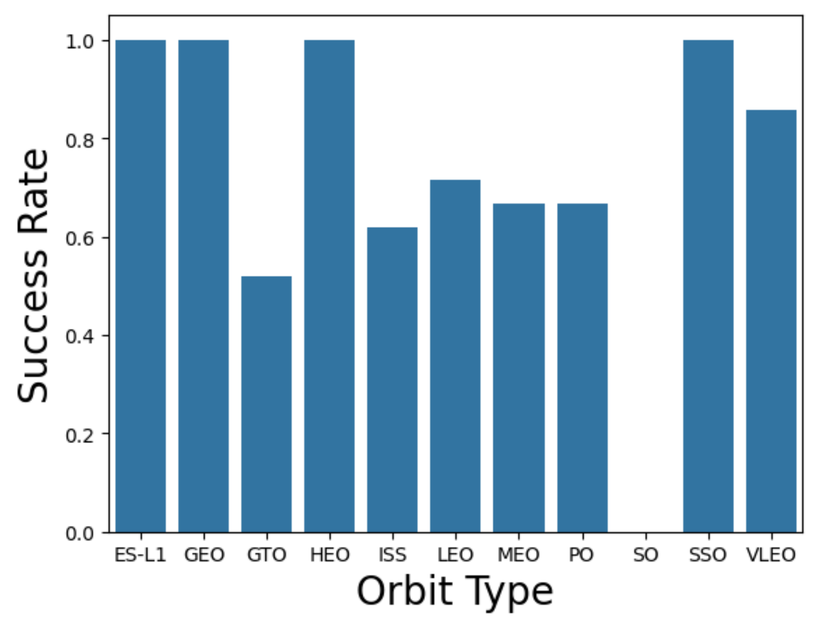
- [Data Wrangling Jupyter Notebook]

EDA with Data Visualization

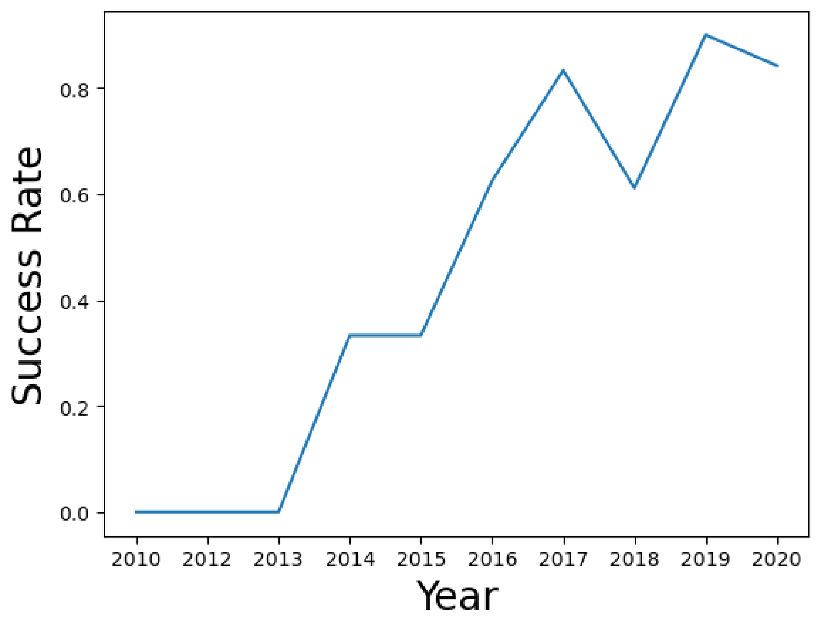
- Plotted scatter charts to explore relationships between variables



- Plotted a bar chart of Orbit Type and Success Rate to see which orbit types are more successful



- Plotted Success Rate over time



- [Data Visualization Jupyter Notebook]

EDA with SQL

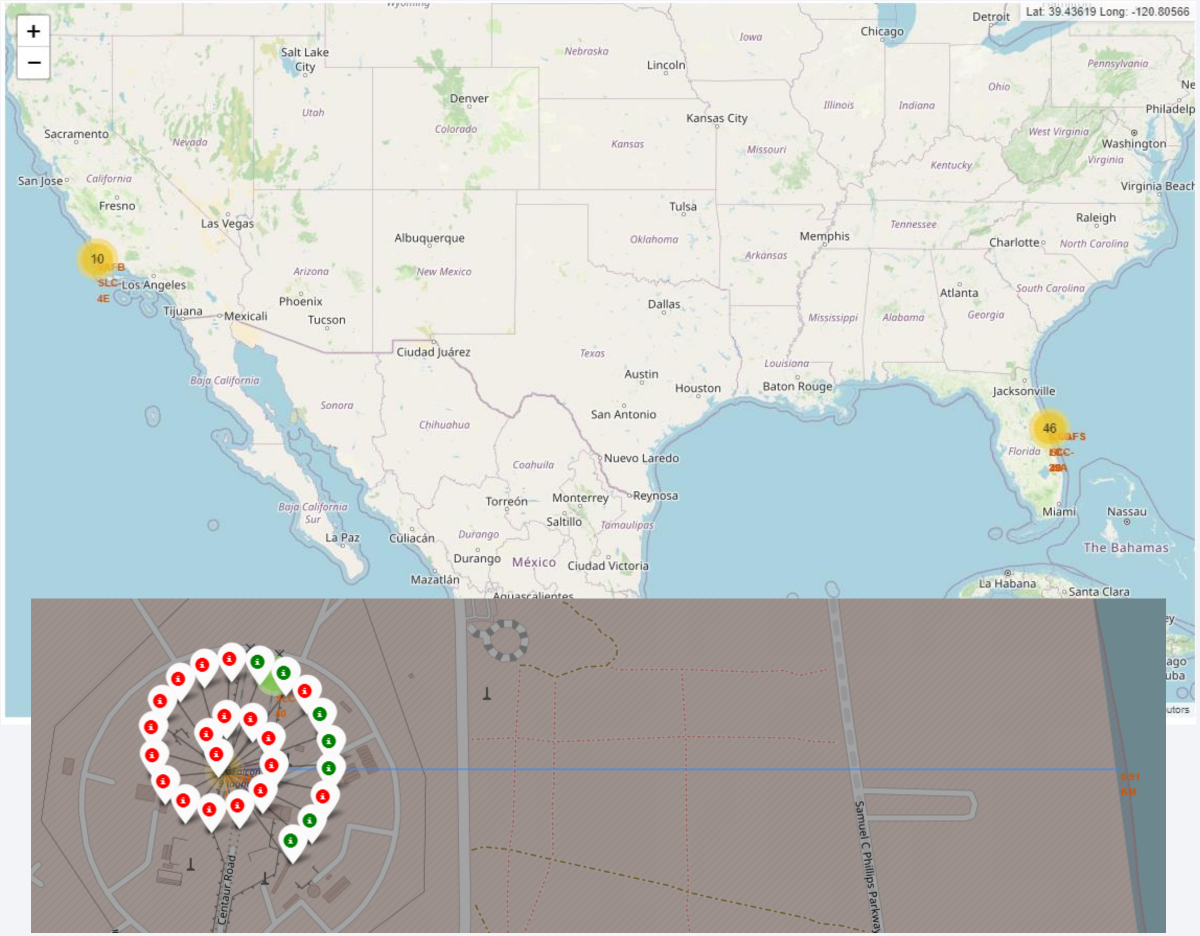
- Names of unique launch sites
- Launch sites that begin with CCA
- Total payload mass carried by NASA
- Average payload mass carried by booster version F9 v1.1
- Date of first successful landing outcome (ground pad)
- Booster versions which have success in drone ship
- Total number of successful and unsuccessful mission outcomes
- Names of booster versions which have carried the maximum payload mass
- Records which had failed landing outcomes in drone ship
- Rank of landing outcomes between 6/4/2010 and 3/20/2017

- [EDA with SQL Jupyter Notebook]

Build an Interactive Map with Folium

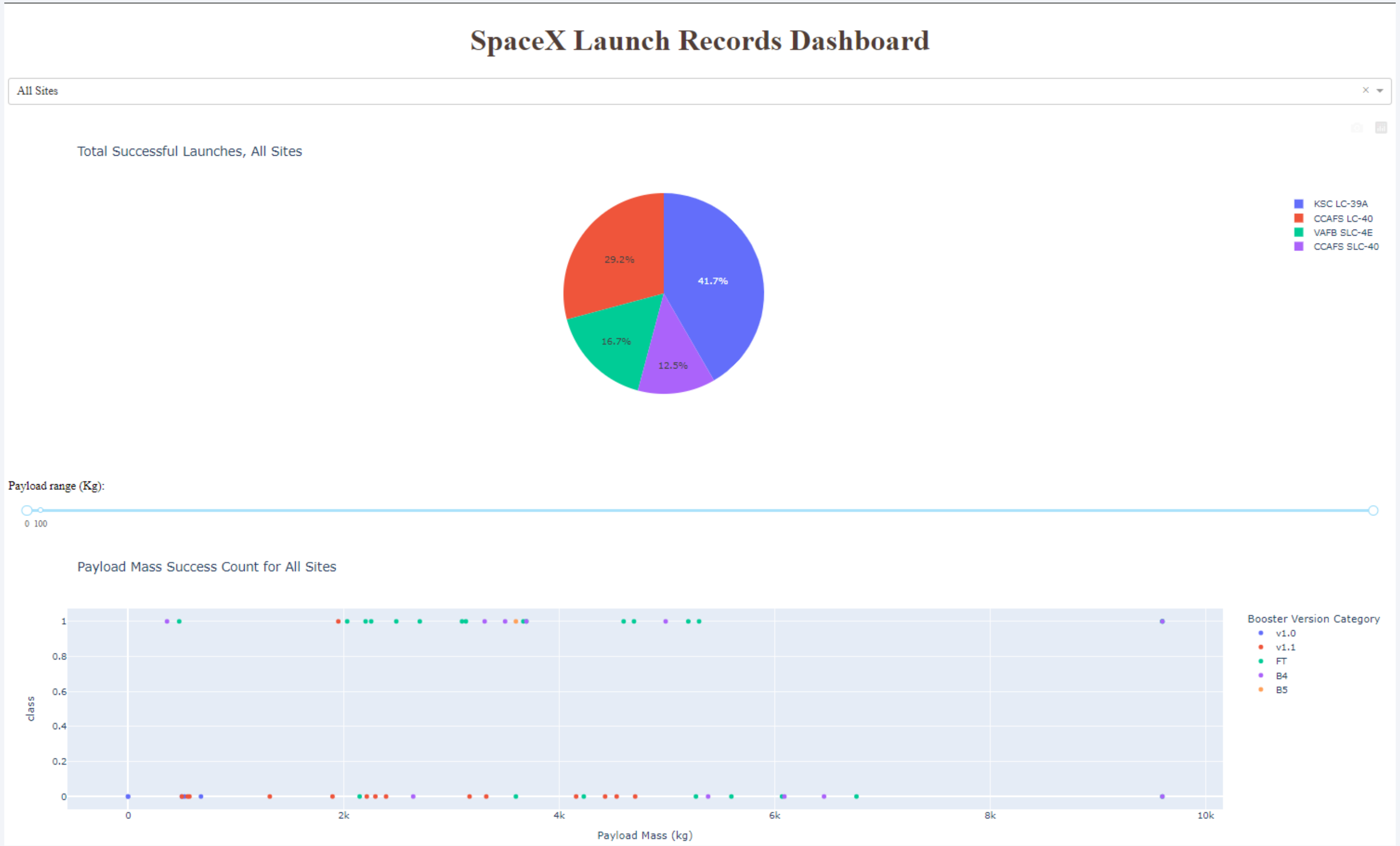
- Launch sites were added to the map and marked
- Successful and unsuccessful launch coordinates were added to the map and marked red (failures) and green (successes)
- Distance to closest coastline, railroad, city, and major highway were measured using lines
    - Launch sites are typically close to coastlines, railroads, and highways
    - Launch sites studied are more than 14 miles away from the nearest city center

- [Interactive Map with Folium Jupyter Notebook]()



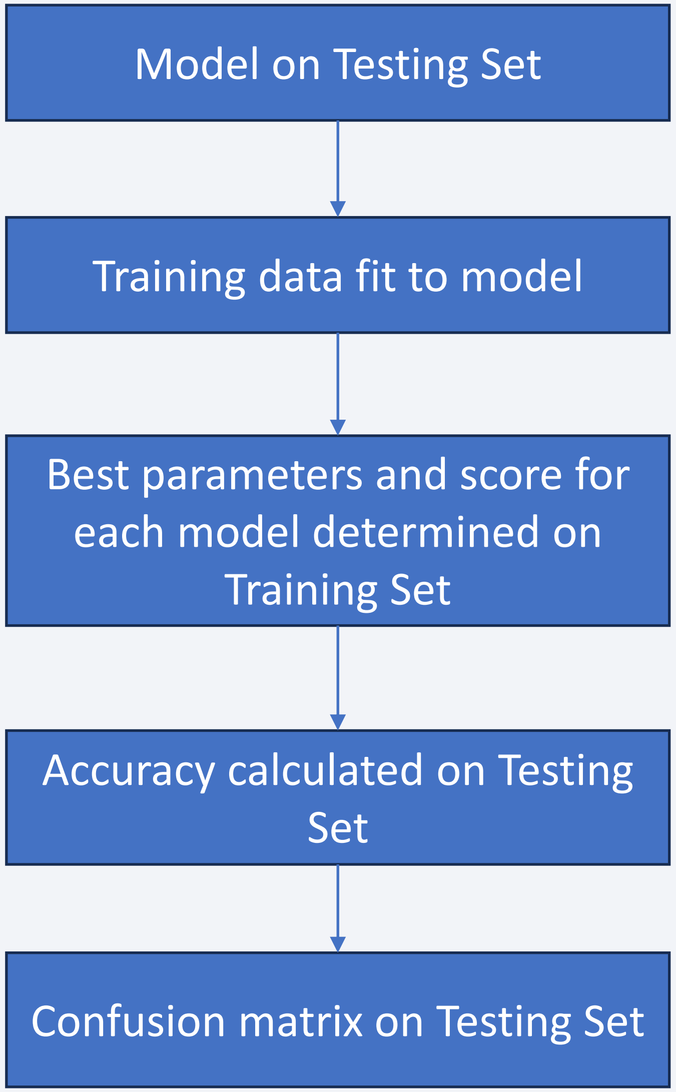
Build a Dashboard with Plotly Dash

- Pie chart with percentage of successful launches filterable by site
- Scatter chart showing failed and successful launches filterable by payload range, also shows booster version
- [Plotly Dash Python File]()



Predictive Analysis (Classification)

- Several models were used to analyze the SpaceX data: Decision Tree, K-Nearest Neighbor (KNN), Logistic Regression (LR), and Support Vector Machines (SVM). Each model was evaluated individually, then results were compared to each other.
- Data was split into a Training Set and a Testing Set
- Data was fit to the Training Set, then accuracy was calculated for each by finding the best score.
- The accuracy score was then calculated on the Test Set.
- Confusion matrices were developed for each model to see which features can be easily distinguished from each other and whether there are false positives or negatives.
- Predictive Analysis Jupyter Notebook



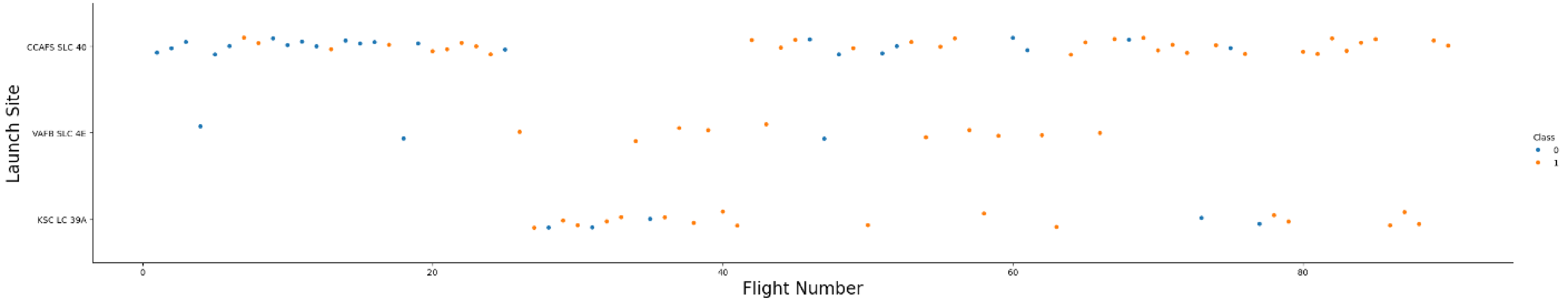
Results

- The first successful landing for SpaceX was in December of 2015
- The F9 Booster has success in drone ship and where the payload mass is between 4000 and 6000 kg
- The F9 Booster has also been able to carry the maximum payload mass
- Overall, success is higher when payload mass is lighter
- The site KSC LC-39A has the most successful launches
- The F9 Booster type FT has the most successes at payloads below 6000 kg
- The best model was determined to be a Decision Tree, which had an accuracy score of 88.9%
- ES-L1, GEO, HEO, and SSO orbits are the most successful

Section 2

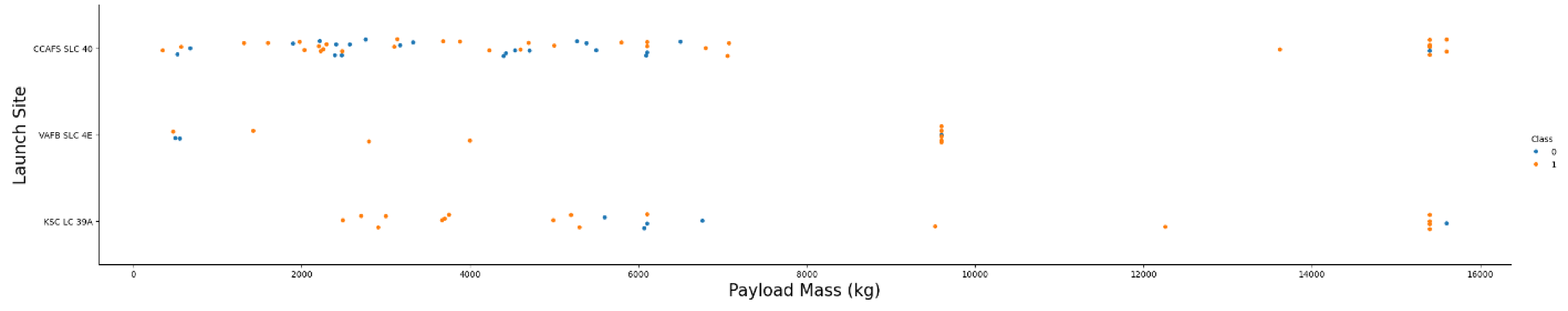
Insights drawn from EDA

Flight Number vs. Launch Site



- Flight numbers correlate with the number of launches
- At most launch sites, successful launches increase in number with more launches (higher flight numbers)
- CCAFS SLC-40 has had the most launches

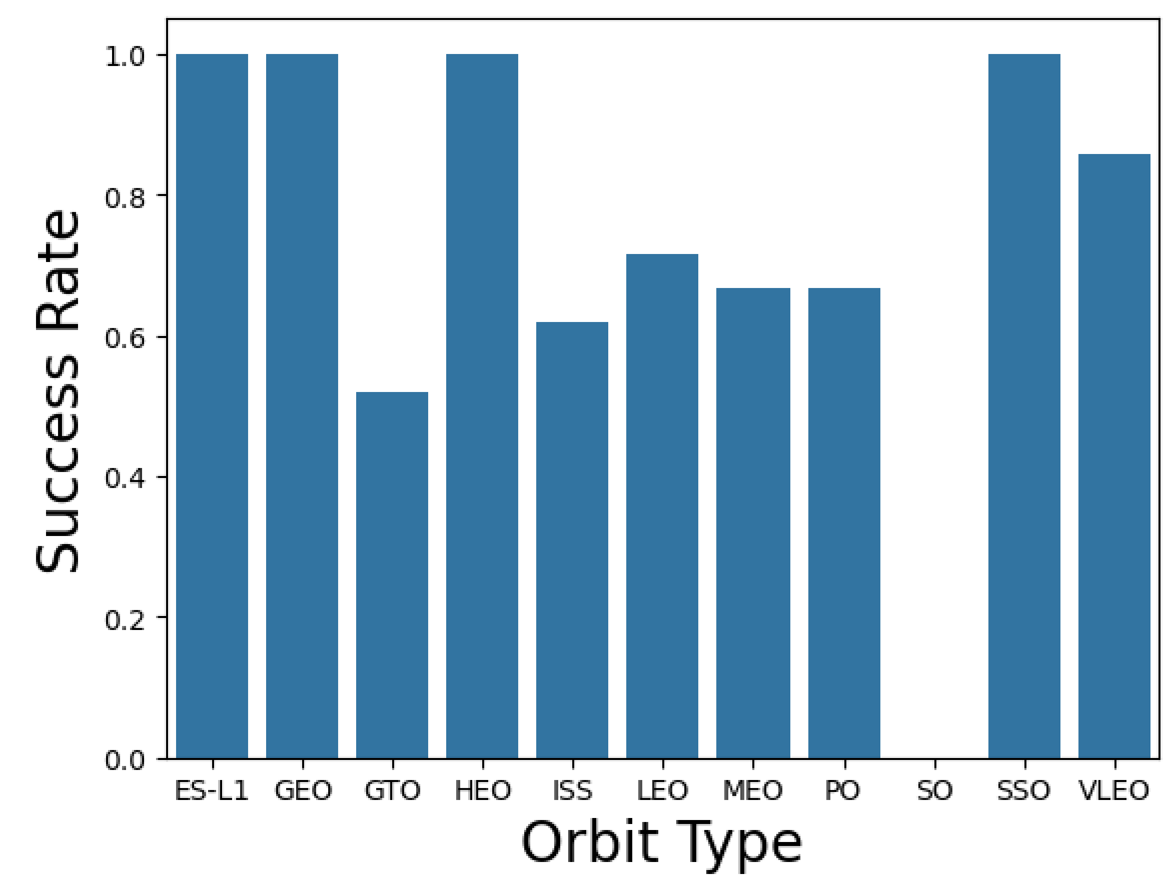
Payload vs. Launch Site



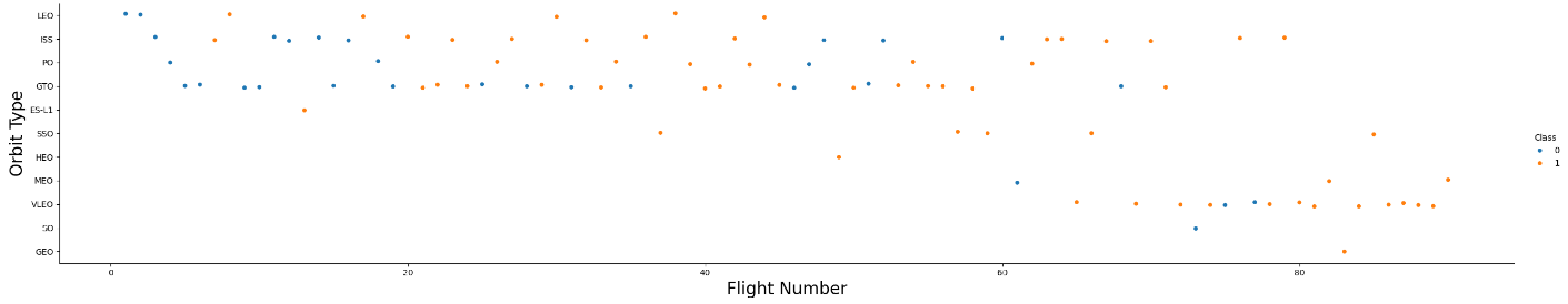
- Some launch sites have more successful launch numbers with certain payload ranges
- Most low payload mass launches have been performed at CCAFS SLC-40

Success Rate vs. Orbit Type

- Success rate can also vary depending on the orbit type
- ES-L1, GEO, HEO, and SSO orbits are the most successful

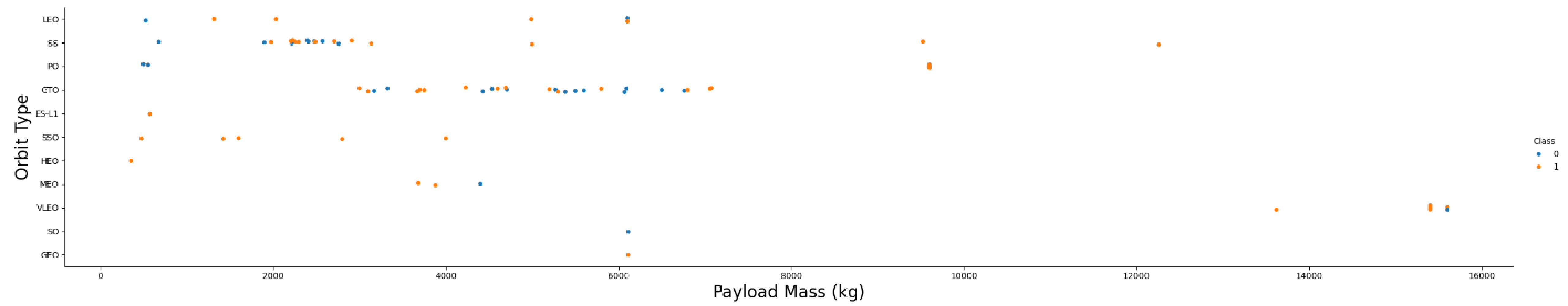


Flight Number vs. Orbit Type



- Earlier flights were generally performed for LEO, ISS, PO, and GTO orbits
- In recent years it has become more common to target the VLEO orbit type

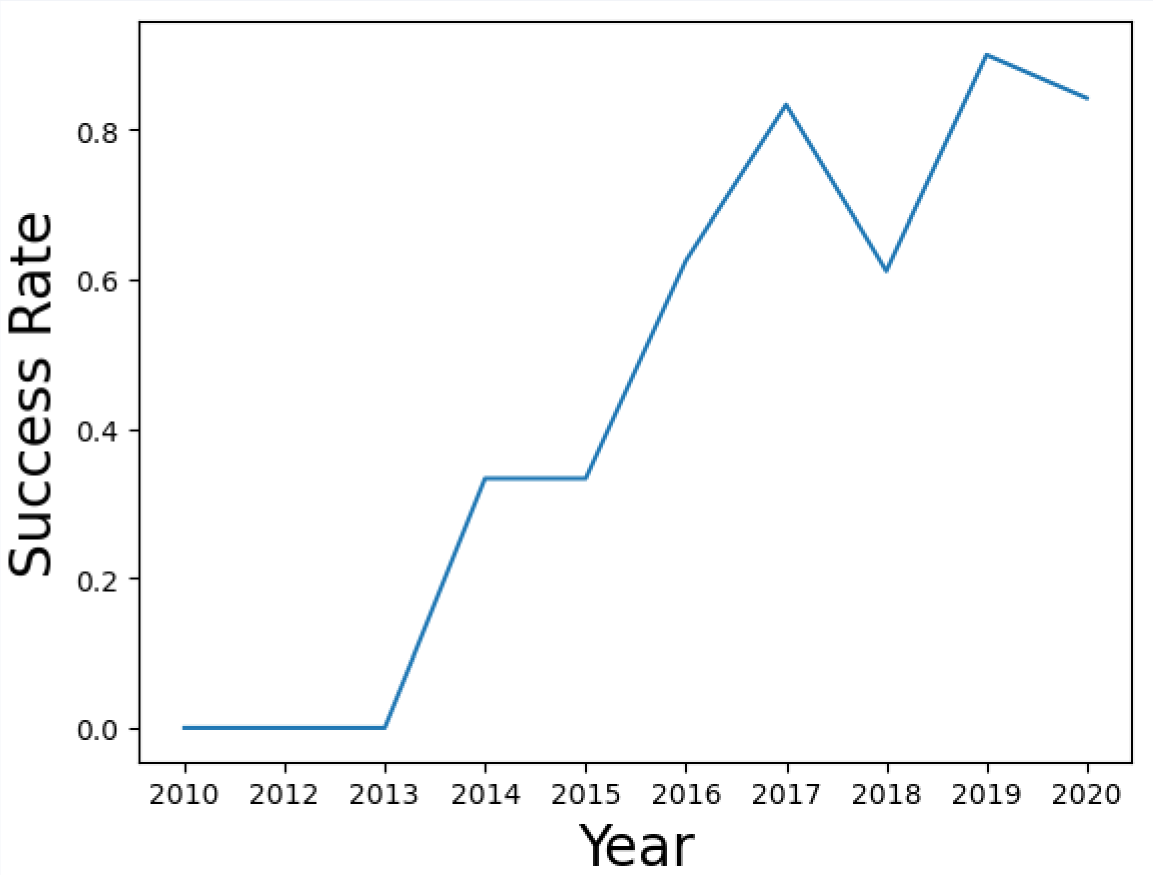
Payload vs. Orbit Type



- ISS payload mass is typically around 2000 kg
- GTO payload is typically between 3000 and 8000 kg

Launch Success Yearly Trend

- Launch success rate has increased significantly since 2013
- Since 2017, success rates have been consistently high



All Launch Site Names

- This query returns the distinct launch site names

```
sql_launch_sites = """
    SELECT DISTINCT launch_site
    FROM spacextbl
    Where Date is not null"""
cur.execute(sql_launch_sites)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('CCAFS LC-40',)

('VAFB SLC-4E',)

('KSC LC-39A',)

('CCAFS SLC-40',)
```

Launch Site Names Begin with 'CCA'

- This query pulls the first 5 data points that have a launch site name that starts with ‘CCA’

```
q2 = """
    SELECT *
    FROM spacextbl
    Where Date is not null AND launch_site like 'CCA%' limit 5"""
cur.execute(q2)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('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', '7:44:00', 'F9 v1.0  B0005', 'CCAFS LC-40', 'Dragon demo flight C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt')
('2012-10-08', '0:35:00', '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

- This query calculates the total payload carried by boosters from NASA, 45596 kg

```
q3 = """
    SELECT
    sum(PAYLOAD_MASS__KG_)
    FROM spacextbl
    Where Date is not null AND Customer = 'NASA (CRS)'"""
cur.execute(q3)
items = cur.fetchall()
for i in items:
    print(i)
```

```
(45596,)
```

Average Payload Mass by F9 v1.1

- This query calculates the average payload mass carried by booster version F9 v1.1, which is 2928.4 kg

```
q4 = """
    SELECT
    avg(PAYLOAD_MASS__KG_)
    FROM spacextbl
    Where Date is not null AND Booster_Version = 'F9 v1.1'"""
cur.execute(q4)
items = cur.fetchall()
for i in items:
    print(i)
```

```
(2928.4,)
```

First Successful Ground Landing Date

- This query looks for the date of the first successful landing outcome on ground pad, Dec 22, 2015

```
q5 = """
    SELECT
    min(Date)
    FROM spacextbl
    Where Date is not null AND Landing_Outcome = 'Success (ground pad)'"""
cur.execute(q5)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('2015-12-22',)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
q6 = """
    SELECT DISTINCT Booster_Version
    FROM spacextbl
    Where Date is not null AND Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ between 4000 and 6000"""
cur.execute(q6)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('F9 FT B1022',)

('F9 FT B1026',)

('F9 FT  B1021.2',)

('F9 FT  B1031.2',)
```

Total Number of Successful and Failure Mission Outcomes

- This query calculates the total number of successful and failure mission outcomes
  - Failure: 1
  - Success: 100

```
q7 = """
    SELECT
    substr(Mission_Outcome, 1, 7) as Mission_Outcome,
    count(*)
    FROM spacextbl
    Where Date is not null
    Group by 1"""
cur.execute(q7)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('Failure', 1)

('Success', 100)
```

Boosters Carried Maximum Payload

- This query lists the names of the booster which have carried the maximum payload mass

```
q8 = """
    SELECT DISTINCT Booster_Version from spacextbl
    where Date is not null AND PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl) """
cur.execute(q8)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('F9 B5 B1048.4',)

('F9 B5 B1049.4',)

('F9 B5 B1051.3',)

('F9 B5 B1056.4',)

('F9 B5 B1048.5',)

('F9 B5 B1051.4',)

('F9 B5 B1049.5',)

('F9 B5 B1060.2 ',)

('F9 B5 B1058.3 ',)

('F9 B5 B1051.6',)

('F9 B5 B1060.3',)

('F9 B5 B1049.7 ',)
```

2015 Launch Records

- This query lists the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 and returns only two entries in 2015

```
q9 = """
    SELECT Date,
    substr(Date, 6, 2),
    launch_site,
    booster_version,
    "Landing_Outcome"
    FROM spacextbl
    Where substr(Date, 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'
    """
cur.execute(q9)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('2015-01-10', '01', 'CCAFS LC-40', 'F9 v1.1 B1012', 'Failure (drone ship)')

('2015-04-14', '04', 'CCAFS LC-40', 'F9 v1.1 B1015', 'Failure (drone ship)')
```

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

- This query ranks the count of landing outcomes between the dates 2010-06-04 and 2017-03-20, in descending order
- The highest count show no attempt for landing

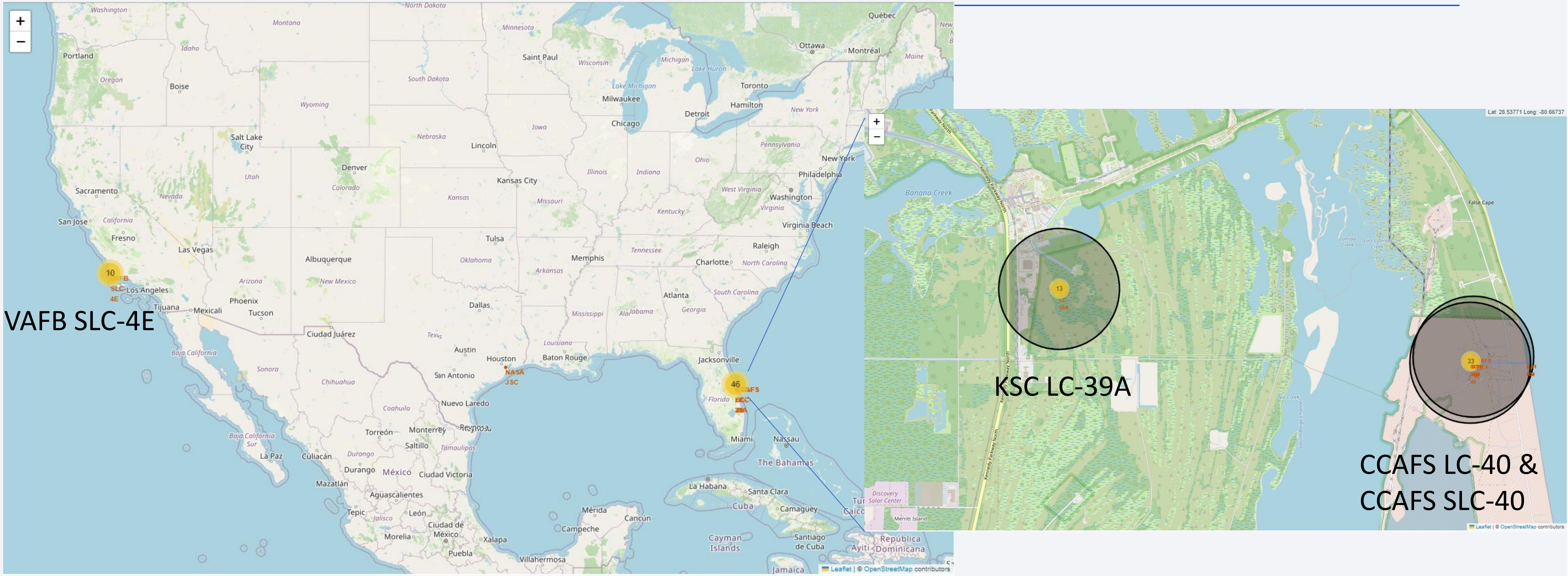
```
q10 = """
    SELECT
    Landing_Outcome,
    COUNT(*) as Outcome_Count
    FROM spacextbl
    Where Date between '2010-06-04' and '2017-03-20'
    Group by Landing_Outcome
    ORDER BY Outcome_Count Desc
    """
cur.execute(q10)
items = cur.fetchall()
for i in items:
    print(i)
```

```
('No attempt', 10)

('Success (drone ship)', 5)

('Failure (drone ship)', 5)

('Success (ground pad)', 3)

('Controlled (ocean)', 3)

('Uncontrolled (ocean)', 2)

('Failure (parachute)', 2)

('Precluded (drone ship)', 1)
```

Section 3

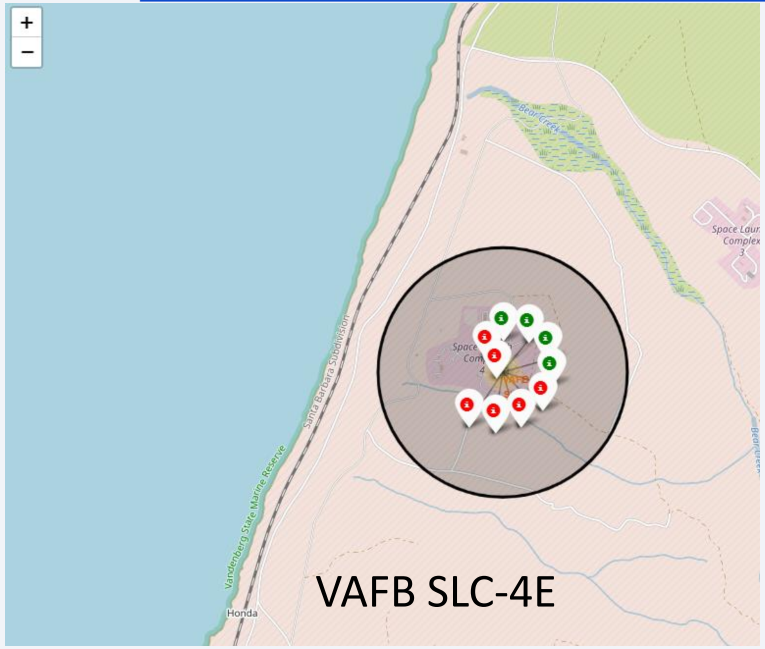
Launch Sites Proximities Analysis

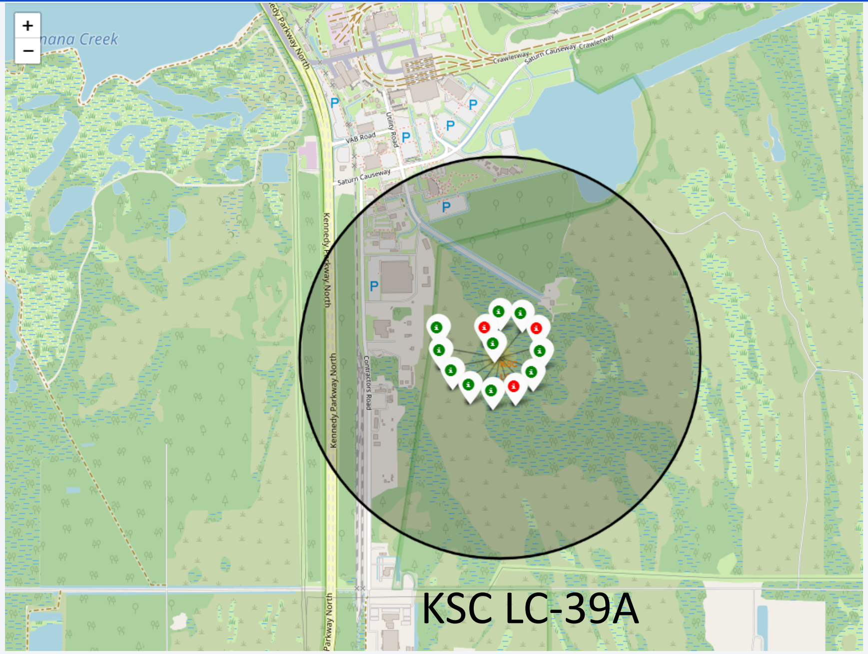
Launch Site Overview

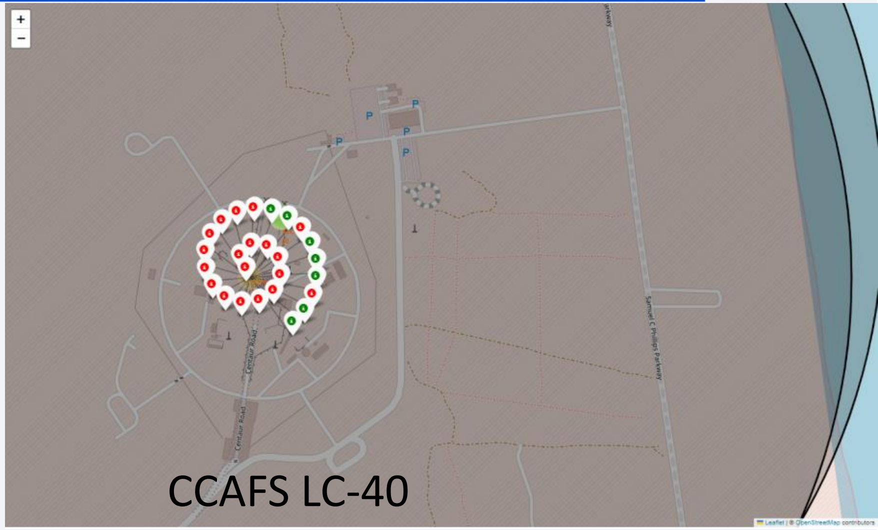


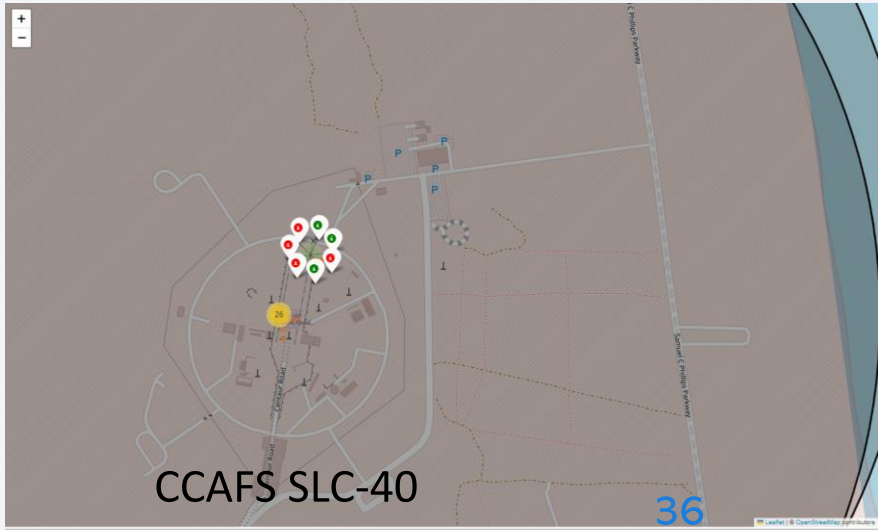
- One site is located on the California coast (VAFB SLC-4E), while the other three sites are located on the Florida coast
- The closeup view shows the location of sites present in Florida
- NASA has a site in Houston, which was not included in this study

Launch Site Details



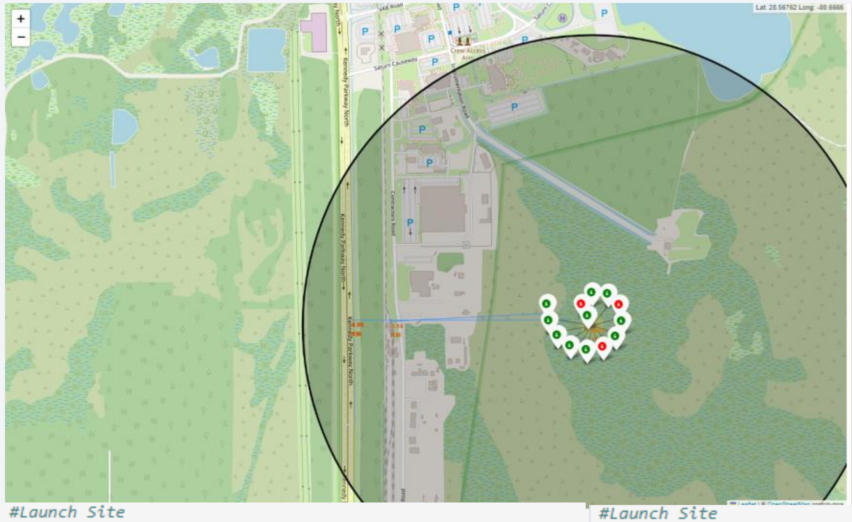


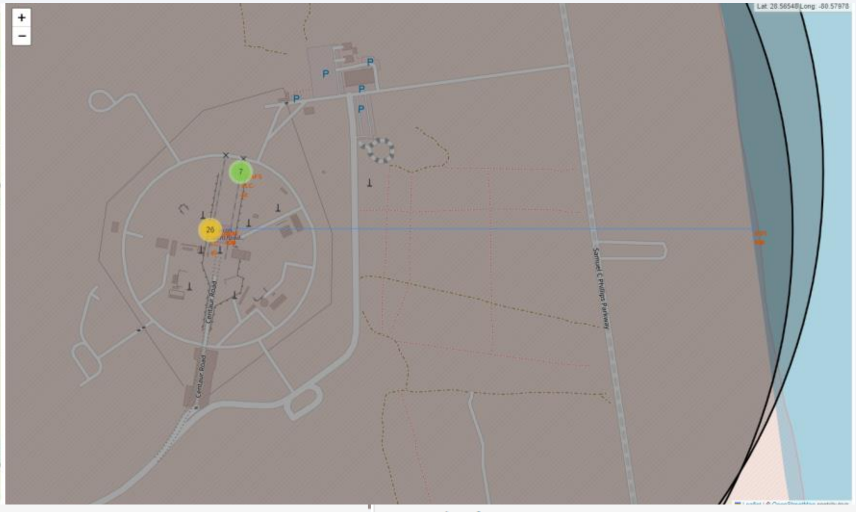


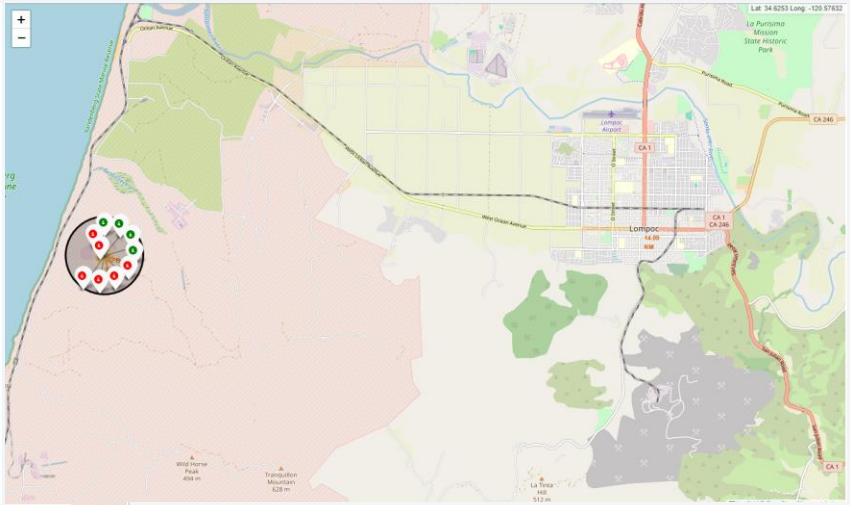


- Closeup views of each launch site show the number of successful launches (green) and failed launches (red).
- KSC LC-39A in Florida has the highest percentage of successful launches
- CCAFS LC-40 in Florida has the highest percentage of failed launches

Proximity to Launch Sites







```
#Launch Site
lat5=28.57363
lon5=-80.64849

#railway Site
lat6=28.57339
lon6=-80.654

distance_rail = calculate_distance(lat5, lon5, lat6, lon6)
distance_rail
```

```
0.5388922178211951
```

```
#Launch Site
lat3=28.57343
lon3=-80.6477

#highway Site
lat4=28.57343
lon4=-80.65539

distance_highway = calculate_distance(lat3, lon3, lat4, lon4)
distance_highway
```

```
0.7511790467160326
```

```
#Launch Site
lat1=28.56232
lon1=-80.57708

#Coastline Site
lat2=28.56232
lon2=-80.56779

distance = calculate_distance(lat1, lon1, lat2, lon2)
distance
```

```
0.9075669918924812
```

```
#Launch Site
lat7=34.63321
lon7=-120.61169

#rcity Site
lat8=34.63801
lon8=-120.45788

distance_city = calculate_distance(lat7, lon7, lat8, lon8)
distance_city
```

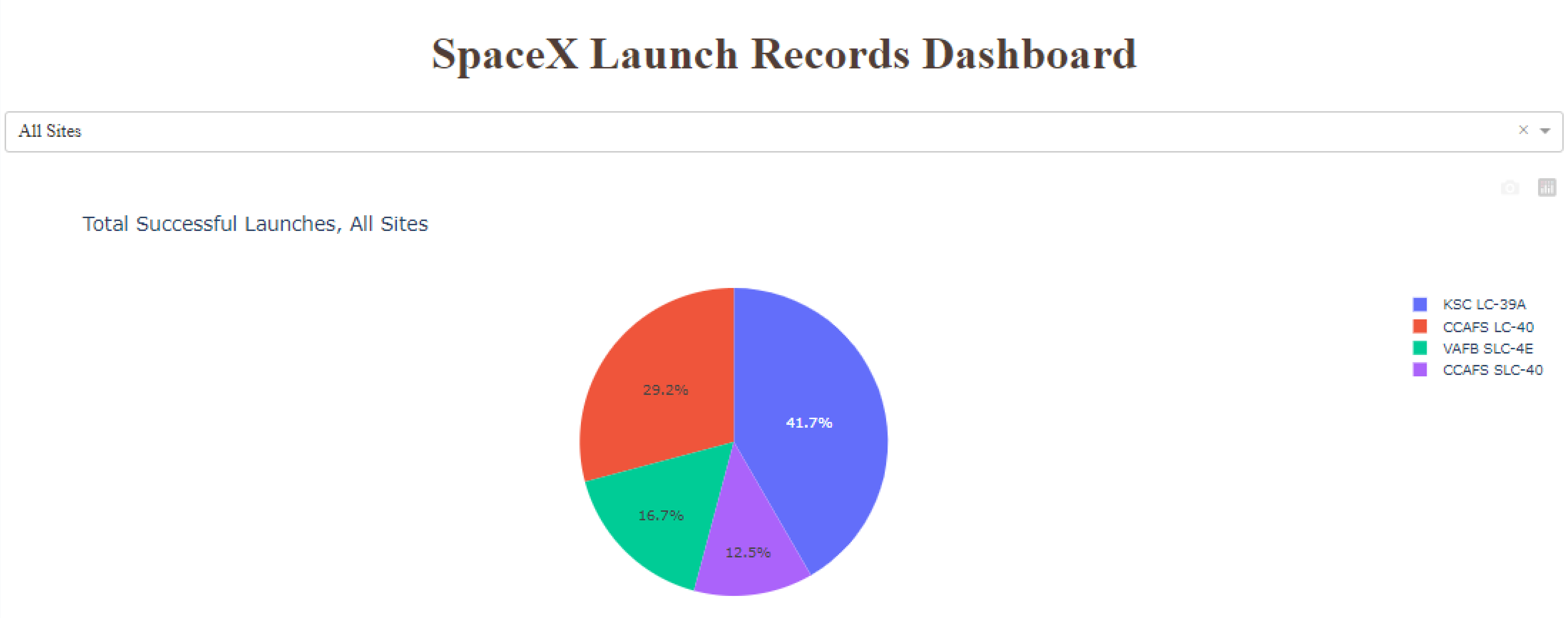
```
14.086511138634757
```

- Each launch site was evaluated for proximity to major map features
- It’s common for launch sites to be within one mile of coastlines, railways, and major highways
- Cities are typically over 14 miles away from launch sites

Section 4

Build a Dashboard with Plotly Dash

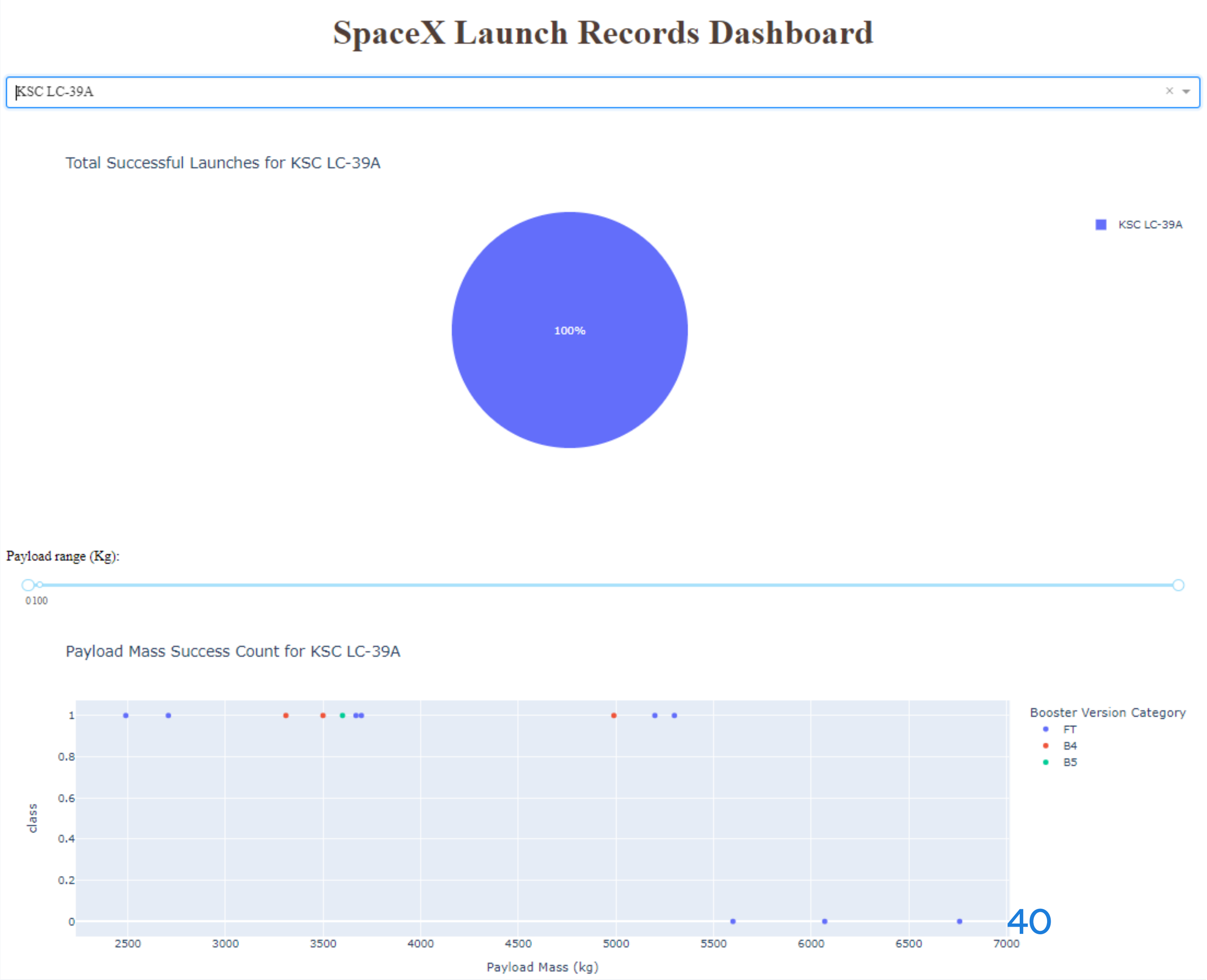
SpaceX Launch Records Dashboard



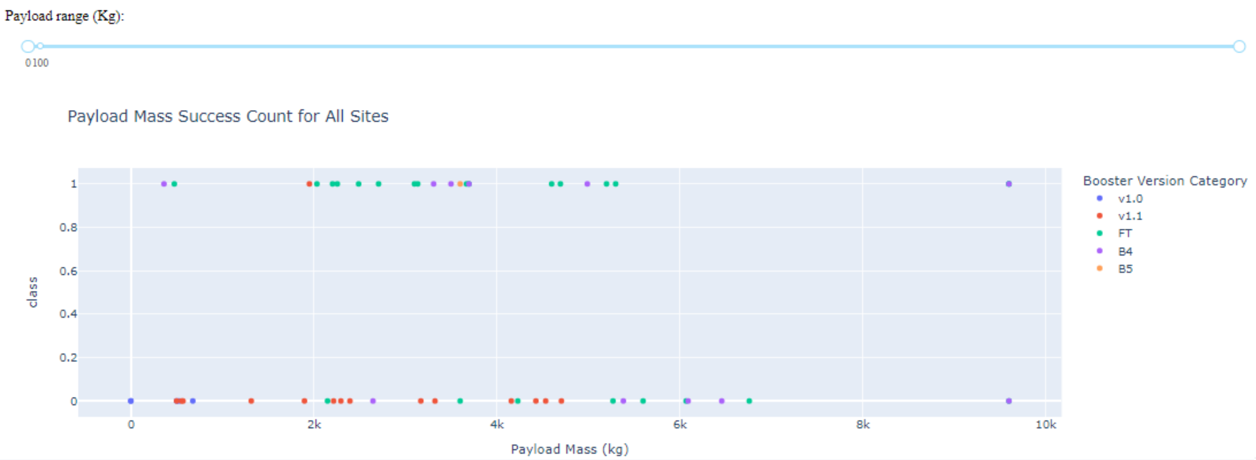
- A dashboard was developed which shows the success rate for all launch sites
- KSC LC-39A has the highest number of successful launches

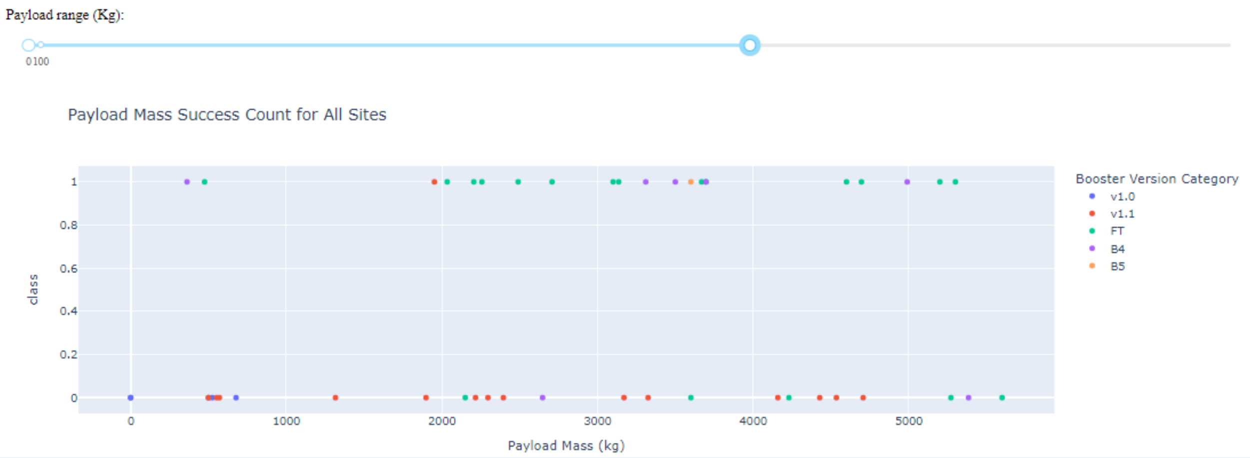
SpaceX Launch Records Dashboard

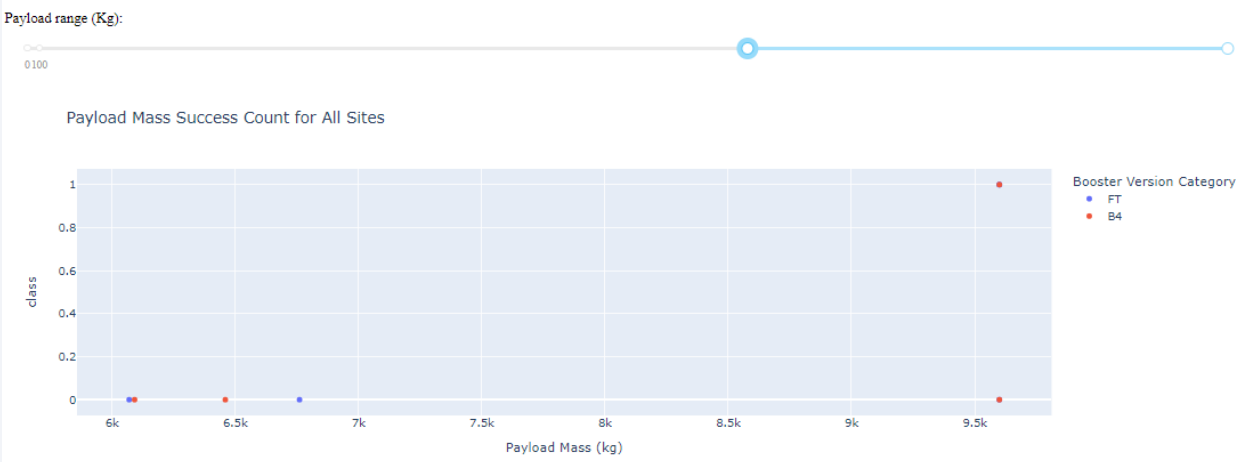
- The dashboard also shows successful and failed launches by payload. The site with the most successful launches is KSC LC-39A



Payload Mass and Booster Versions







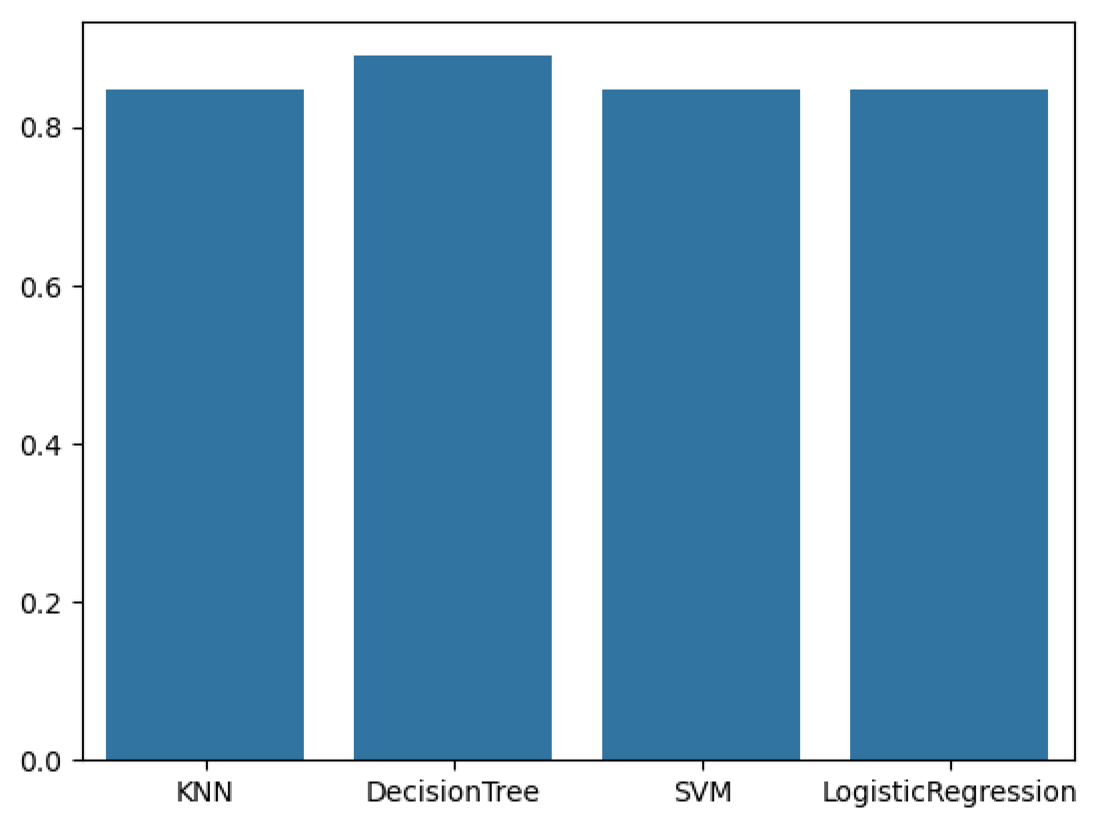
- For all payloads, the most successes occur with the FT Booster Version
- For payloads under 6000 kg, the FT Booster Version is also the most successful, with v1.1 and B5 performing poorly
- For payloads above 6000 kg, the B4 Booster Version is the only one to have success, but the success rate is 1/4

Section 5

Predictive Analysis (Classification)

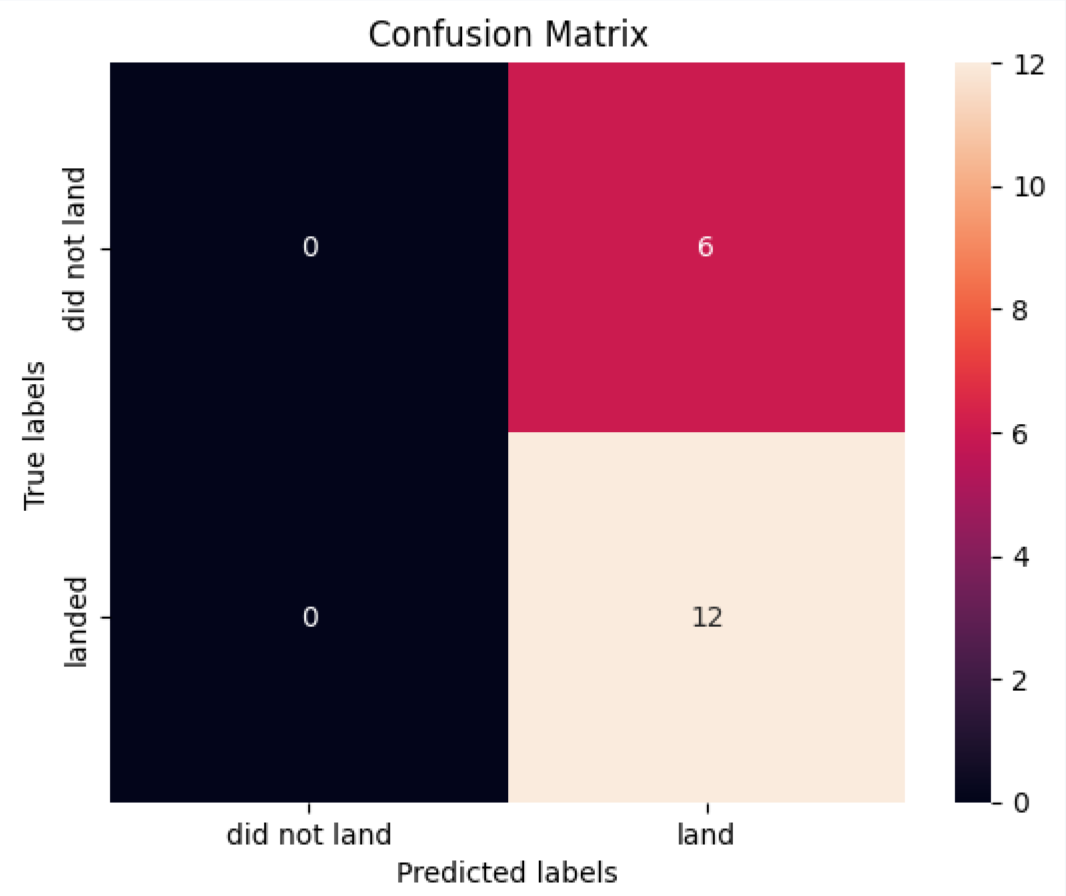
Classification Accuracy

- Several models were used to analyze the SpaceX data: Decision Tree, K-Nearest Neighbor (KNN), Logistic Regression (LR), and Support Vector Machines (SVM).
- Each model was compared to find the highest classification accuracy.
- For this study, Decision Tree analysis yields the highest classification accuracy



Confusion Matrix

- The confusion matrix for Decision Tree analysis show no false positives and correctly sorts the data by class (successes and failures)



Conclusions

- Decision Tree Analysis is the best model for this data set
- The first successful landing for SpaceX was in December of 2015, with successful launch numbers increasing with experience
- The F9 Booster has success in drone ship and where the payload mass is between 4000 and 6000 kg
- The F9 Booster has also been able to carry the maximum payload mass
- Overall, success is higher when payload mass is lighter
- For payloads above 6000 kg, the B4 Booster Version is the only one to have success, but the success rate is low with one out of four launches being successful
- The site KSC LC-39A has the most successful launches
- The F9 Booster type FT has the most successes at payloads below 6000 kg
- ES-L1, GEO, HEO, and SSO orbits are the most successful

Thank you!