IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nivedita K. Naresh
03/30/2022

Outline

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

Executive Summary

- Methodologies:
  - Data collection, Data wrangling, Exploratory Data Analysis, Data visualization, Predictive analysis
- Summary of all results
  - Launch outcomes improved over time
  - Launches with payloads between 3000 kg and 5250 kg were successful
  - Orbits with 100% success rate: ES-L1, GEO, HEO, SSO
  - Launch site KSC LC 39A had the highest launch success ratio of 76.9%

Introduction

- SpaceX has gained worldwide attention for a series of historic milestones.
- It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars wheras other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.
- We want to use data science tools to predict the success outcomes for SpaceX launches

Section 1

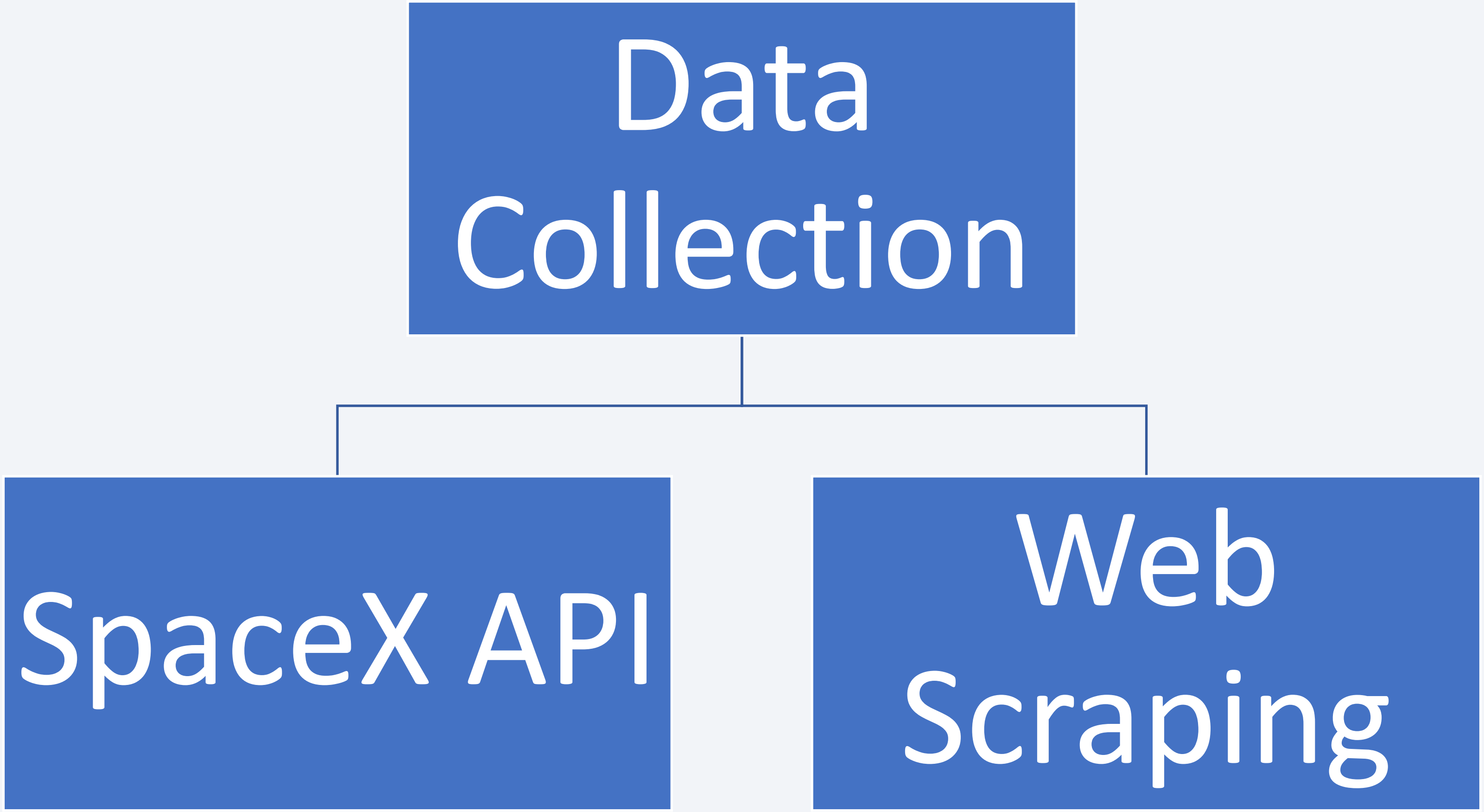
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using web scraping with Wikipedia entry as the source
  - Data was also collected using data wrangling with Spacex API
- Perform data wrangling
  - Data was processed using data wrangling and data filtering
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Models: Logistic Regression, Support Vector Machine, Decision Tree Classifier, k-nearest neighbor

Data Collection

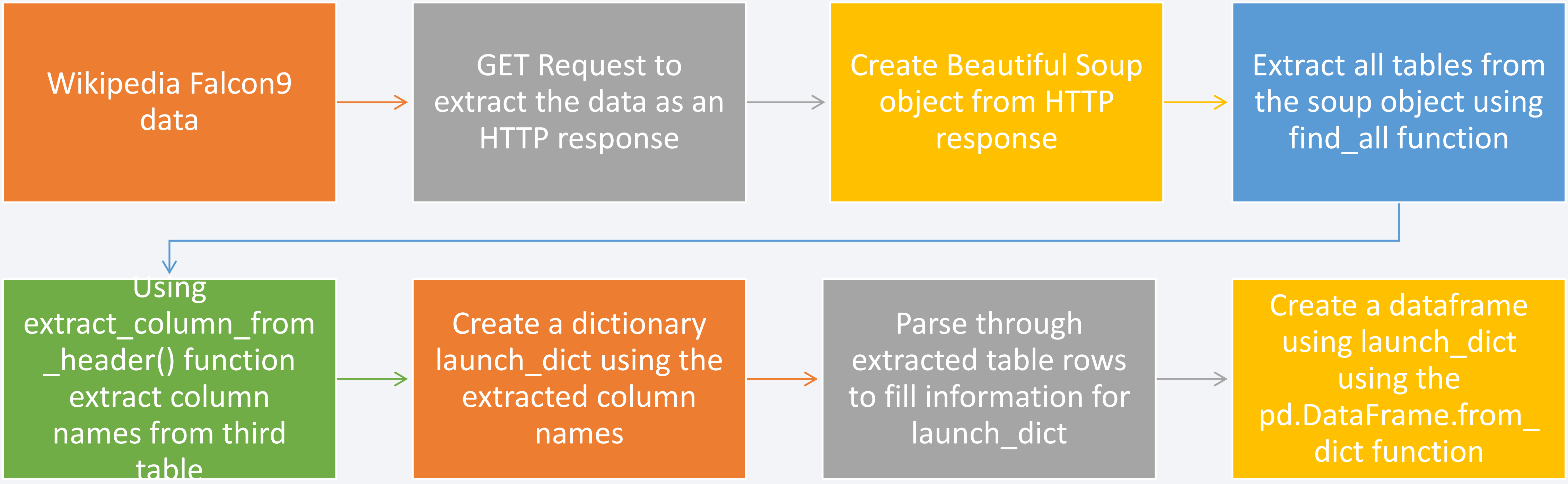


Data Collection – SpaceX API

SpaceX API	url = https://api.spacexdata.com/v4/launches/past
Data collection using GET request	response = requests.get(url)
Convert response content to JSON format	response.json()
Convert to a Dataframe after JSON normalization	pd.json_normalize(response_json)
Dataframe filtering	data[['rocket', 'payloads'…]]
Create a new dataframe with filtered data	`data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]`

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W1_DataWrangling.ipynb

Data Collection - Scraping



https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W1_WebScraping.ipynb

Data Wrangling

Data Wrangling	Used for Data cleaning
Find missing values in the dataset	data.isnull().sum()
Replace missing values with mean value for the chosen category	data[‘Category’].replace(np.nan, mean_value, inplace=True)

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W1_DataWrangling.ipynb

EDA with Data Visualization

Plots	Purpose
Scatter plot (catplot)	To compare how the flight number and payload mass affected the launch outcomes
Scatter plot (catplot)	To visualize relationship between flight numbers and launch site and how it affected the launch outcomes
Scatter plot	To visualize relationship between payload mass and launch site
Bar plot	To visualize relationship between orbit type and success rate
Scatter plot	To visualize relationship between payload mass and orbit type along with launch outcomes
Line chart	To visualize launch success yearly trend

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W2_jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W2_EDA-SQL.ipynb

Build an Interactive Map with Folium

Map Object Templates	Purpose
Circle	Used to indicate coordinates for a site whose name is indicated in the pop-up label
Marker	Used to indicate coordinates for a site whose name is indicated in the pop-up label
MarkerCluster	Used to cluster multiple markers which have the same coordinates, where the number indicates the number of markers clustered together
Line	Used to indicate distance between coast and launch site

Map Objects	Label
Circle	Nasa Johnson Space Center
Marker	Nasa Johnson Space Center
Circle	Launch Sites
Marker	Launch Sites
MarkerCluster	Clustering sites for all launches
Mouse position	Provides coordinates for your mouse position on the map
Marker	Coastline
Line	Distance between coastline and launch site

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W3_DataVizFolium.ipynb

Build a Dashboard with Plotly Dash

Dashboard Elements	Purpose
Dropdown	To select either a specific launch site or to select all launch sites
Pie chart	To visualize the success outcomes for launch sites
Range slider	To select the range of payload mass
Scatter plot	To visualize the correlation between payload mass and success outcomes for launch sites

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W3_SpaceXDash_app.ipynb

Predictive Analysis (Classification)

Classification Models Built	Logistic regression
	K-nearest neighbor
	Classification tree
	Support vector machine
Model Evaluation	GridSearch to find the best hyperparameters
	Model Fitting
	Confusion matrix
	Accuracy score
Best classification model	Comparison of accuracy scores

Building classification models

- lr=LogisticRegression()

Model evaluation

- logreg_cv = GridSearchCV(lr, parameters, cv=10, refit=True)
- logreg_cv.fit(X_train, Y_train)
- yhat=logreg_cv.predict(X_test)
- plot_confusion_matrix(Y_test,yhat)
- print("tuned hyPerparameters :(best parameters) ",logreg_cv.best_params_)
- print("accuracy :",logreg_cv.best_score_)

Finding the best classification model

- accuracy_dict = dict({'log_reg': accuracy, 'svm': accuracy_svm, 'knn': accuracy_knn, 'tree': accuracy_tree})
- for key, val in list(accuracy_dict.items()):
- print(key, val)

https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W4_PredictionPipeline.ipynb

Results

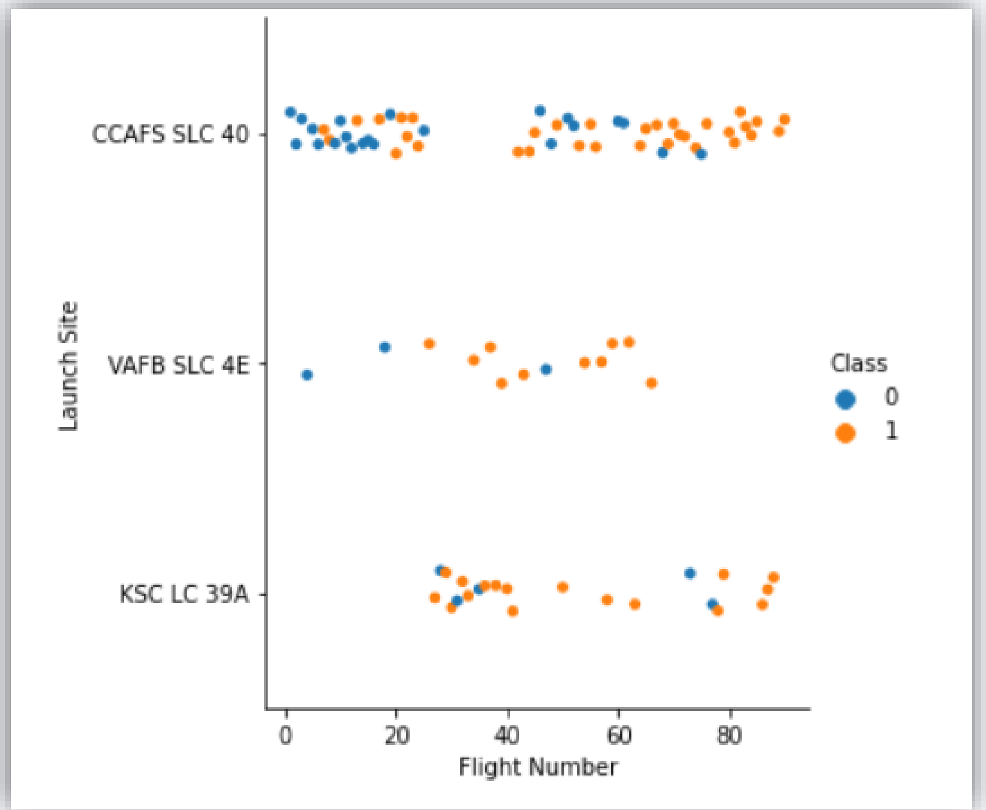
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

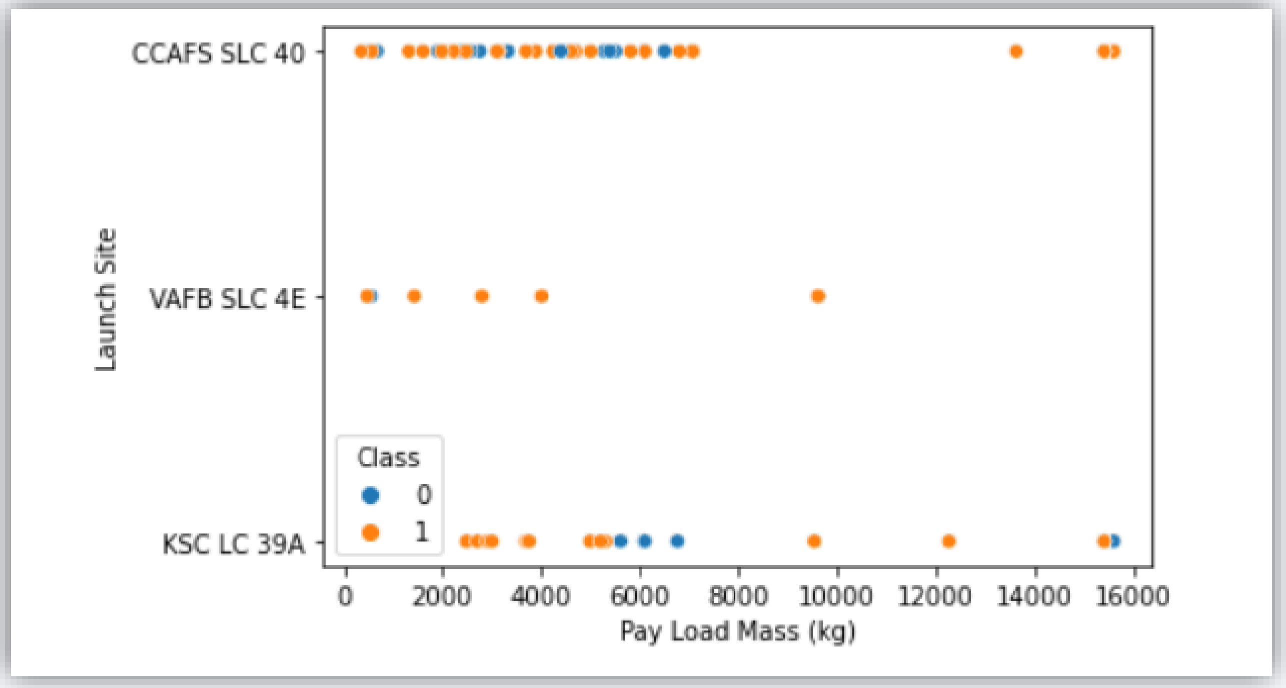
Flight Number vs. Launch Site

- Launch sites with launch outcomes where
  - class=0: failed outcome
  - Class=1: successful outcome
- A higher flight number refers to later flight in time
- For all the launch sites, success outcomes improved over time



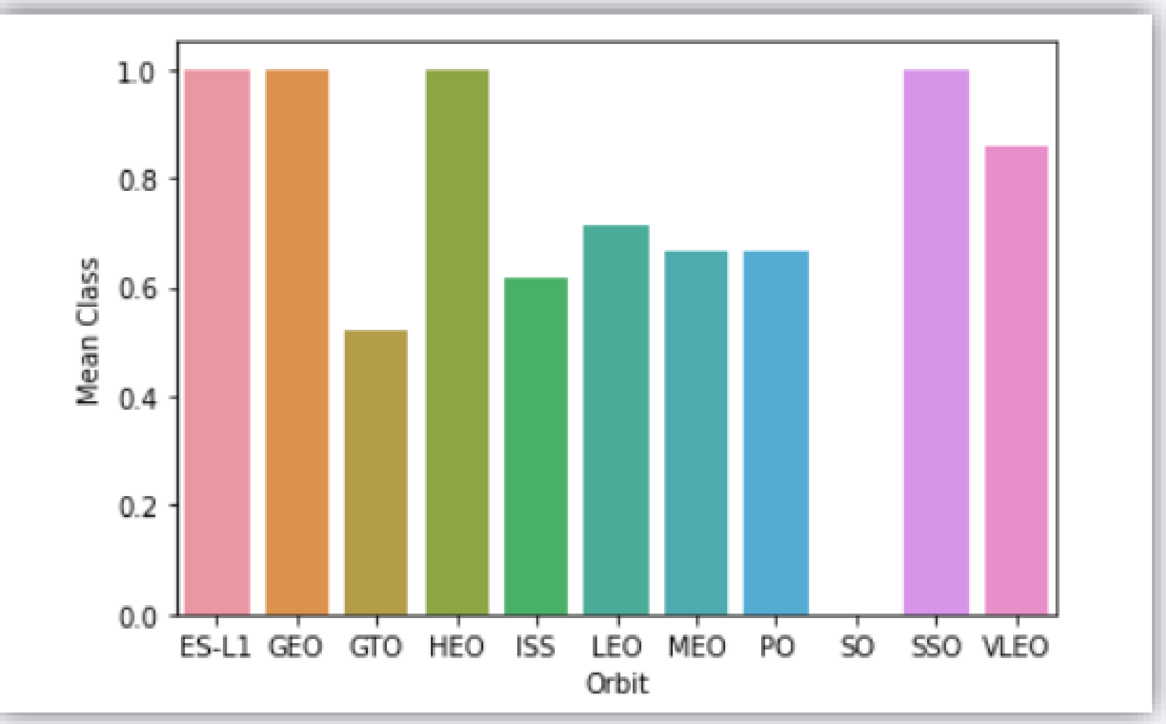
Payload vs. Launch Site

- Increasing the payload mass improved the success outcomes for all launch sites



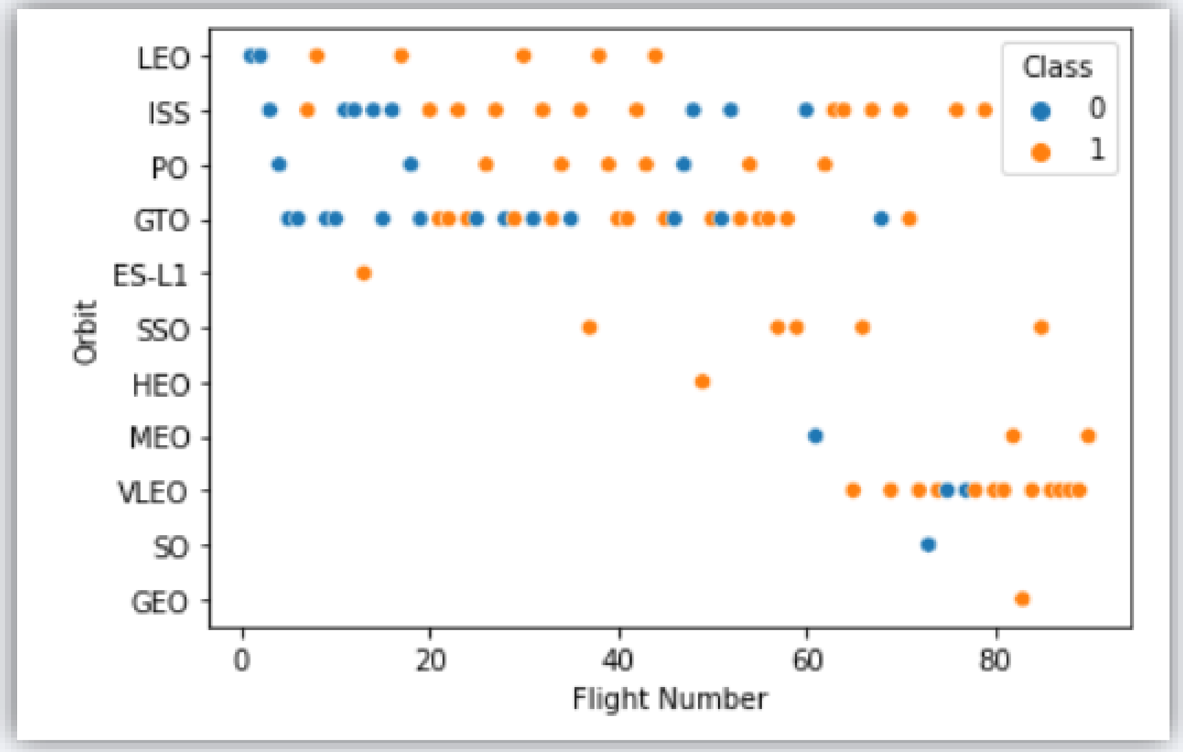
Success Rate vs. Orbit Type

- A higher mean class value refers to improved success outcomes
- The following orbits had a 100% success rate:
  - ES-L1
  - GEO
  - HEO
  - SSO
- The success rate for orbit VLEO was 85%



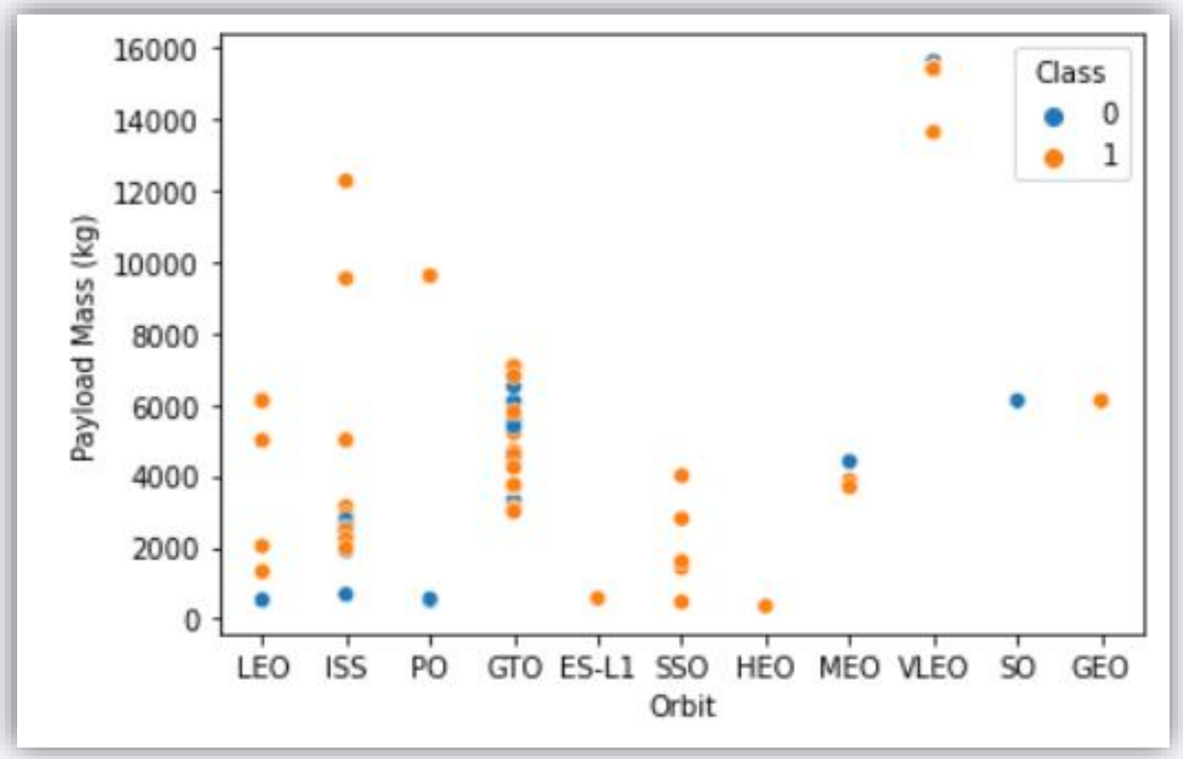
Flight Number vs. Orbit Type

- The success outcomes for all orbits improved with time



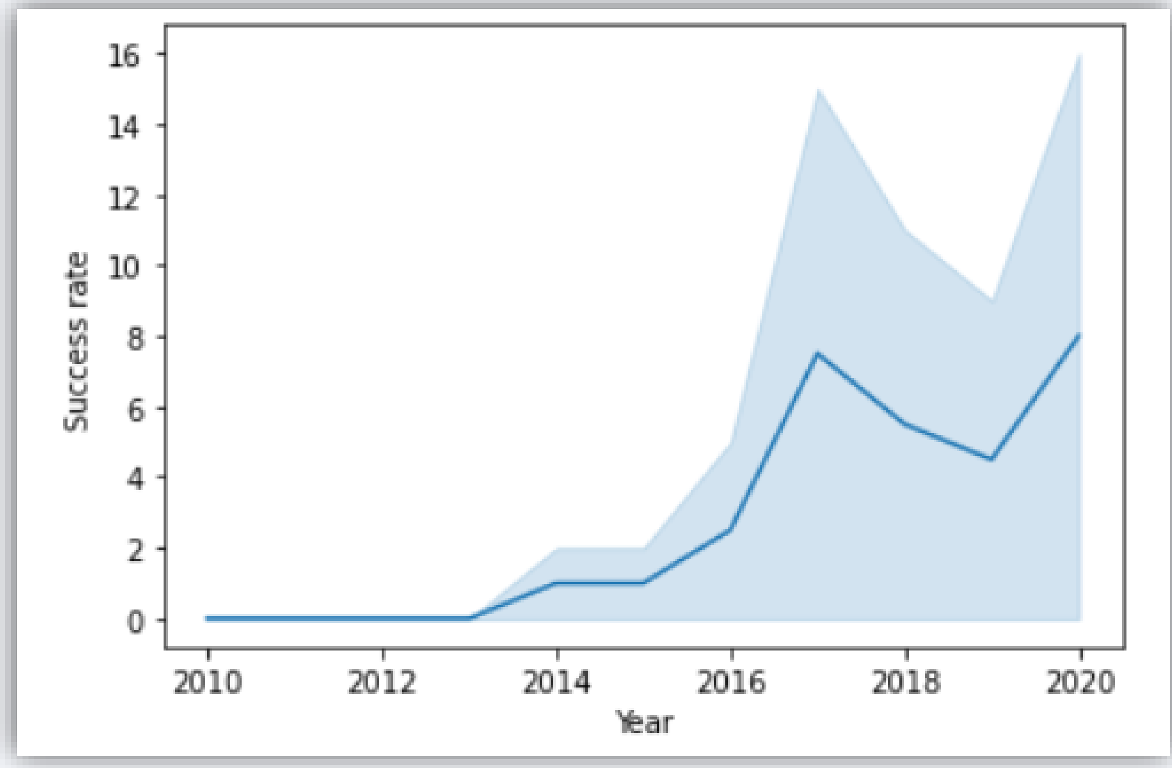
Payload vs. Orbit Type

- For most of the orbits, the success outcomes were improved by increasing the payload mass
- The following orbits were exceptions where a higher payload mass reduced the success outcome:
  - MEO
  - VLEO



Launch Success Yearly Trend

- The following graph illustrates the improvement in launch success rate over time



All Launch Site Names

- There were four main launch sites:
  - CCAFS LC-40
  - CCAFS SLC 40
  - VAFB SLC 4E
  - KSC LC 39A

```
# With SQL
cur.execute('SELECT DISTINCT(Launch_Site) FROM spacex')
for row in cur:
    print("Launch sites: ", row)
```

```
Launch sites:  ('CCAFS LC-40',)
Launch sites:  ('VAFB SLC-4E',)
Launch sites:  ('KSC LC-39A',)
Launch sites:  ('CCAFS SLC-40',)
```

Launch Site Names Begin with 'CCA'

- An SQL query was used to extract launch records for launch sites starting with ‘CCA’
- The image on the bottom shows the result for the query

```
# Filter and find the rows that correspond to launch sites starting with 'CCA'
idx = []
spacex_filtered = pd.DataFrame()
for site in find_sites:
    cur.execute('SELECT * FROM spacex WHERE Launch_Site = ? LIMIT 5', [site])
    for row in cur:
        idx.append(row[0])
        print(row[0])
        spacex_filtered = spacex_filtered.append(spacex.iloc[row[0]])

# Question 2
spacex_filtered.head(5)
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The NASA customers were identified using numpy functionality
- An SQL query was used to extract the total payload mass for NASA customers
- The total payload carried by boosters from NASA was 99980 kg

```python
customers = np.unique(spacex[['Customer']])
find_cust = []
j = 0
for i in range(0,len(customers)):
    if(str(customers[i]).startswith('NASA')):
        find_cust.append(customers[i])
```

```python
#Question 3 with SQL

total_payload = 0
for cust in find_cust:
    cur.execute('SELECT sum(PAYLOAD_MASS__KG_) FROM spacex WHERE CUSTOMER = ?', [cust])
    for row in cur:
        total_payload = total_payload + row[0]
print("Total Payload: ", total_payload)
```

```
Total Payload:  99980
```

Total Payload Mass: NASA (CRS)

- An SQL query was used to extract the total payload mass for launches by NASA (CRS)
- The total payload carried by boosters from NASA (CRS) was 45596 kg

```
cur.execute('SELECT sum(PAYLOAD_MASS__KG_) FROM spacex WHERE CUSTOMER = "NASA (CRS)"')
for row in cur:
    print('Total Payload for NASA (CRS): ', row)
```

```
Total Payload for NASA (CRS):  (45596,)
```

Average Payload Mass by F9 v1.1

- The average payload mass for booster version F9 v1.1 was calculated using both SQL and Pandas
- The average payload mass carried by booster version F9 v1.1 was 2928.4 kg

```
# Question 4 with Dataframe
temp = spacex[spacex[spacex_cols[2]]=='F9 v1.1']
avg_payloadmass_F9v1 = temp[spacex_cols[5]].mean()
print("Average Payload for F9 v1.1: ", avg_payloadmass_F9v1)

# With SQL
cur.execute('SELECT AVG(PAYLOAD_MASS__KG_) FROM spacex WHERE BOOSTER_VERSION = ?', ['F9 v1.1'])
for row in cur:
    print("Average Payload for F9 v1.1: ", row)
```

```
Average Payload for F9 v1.1:  2928.4
Average Payload for F9 v1.1:  (2928.4,)
```

First Successful Ground Landing Date

- The dates for the first successful landing outcome on ground pad was calculated using both Pandas and SQL
- The first successful landing outcome on ground pad was on 5th May 2017

```
# Question 5
success_outcomes = np.unique(spacex[[spacex_cols[9]]])
print(success_outcomes[9])
temp = spacex[spacex[spacex_cols[9]]==success_outcomes[9]]
first_date_success_ground = temp[spacex_cols[0]].min()
print("First successful Ground Landing Date: ", first_date_success_ground)

# With SQL
cur.execute('SELECT MIN(Date) FROM spacex WHERE "Landing _Outcome" = ?', [success_outcomes[9]])
for row in cur:
    print("First successful Ground Landing Date: ", row)
```

```
Success (ground pad)
First successful Ground Landing Date:  01-05-2017
First successful Ground Landing Date:  ('01-05-2017',)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

```
# With SQL
cur.execute('''SELECT BOOSTER_VERSION FROM spacex WHERE "LANDING _OUTCOME"= ? AND
            PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000''', [success_outcomes[8]])
for row in cur:
    print("Names of boosters which have success in drone ship with 4000<Payload<6000: ", row)
```

```
Names of boosters which have success in drone ship with 4000<Payload<6000:  ('F9 FT B1022',)
Names of boosters which have success in drone ship with 4000<Payload<6000:  ('F9 FT B1026',)
Names of boosters which have success in drone ship with 4000<Payload<6000:  ('F9 FT  B1021.2',)
Names of boosters which have success in drone ship with 4000<Payload<6000:  ('F9 FT  B1031.2',)
```

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 kg but less than 6000 kg was extracted using SQL queries:
    - F9 FT B1022
    - F9 FT B1026
    - F9 FT B1021.2
    - F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes was estimated using SQL query to be 100
- The total number of failure mission outcomes was estimated to be 1

```
# With SQL
cur.execute('''SELECT count(MISSION_OUTCOME) FROM spacex WHERE MISSION_OUTCOME = ? OR MISSION_OUTCOME = ?
OR MISSION_OUTCOME = ?''', [mission_outcomes[1], mission_outcomes[2] , mission_outcomes[3]])
for row in cur:
    print('Number of successful outcomes is ', row)

cur.execute('''SELECT count(MISSION_OUTCOME) FROM spacex WHERE MISSION_OUTCOME = ?''',
            [mission_outcomes[0]])
for row in cur:
    print('Number of failed outcomes is ', row)
```

```
Number of successful outcomes is  (100,)
Number of failed outcomes is  (1,)
```

Boosters Carried Maximum Payload

```
# With SQL
cur.execute('SELECT BOOSTER_VERSION FROM spacex WHERE PAYLOAD_MASS__KG_ = (SELECT max(PAYLOAD_MASS__KG_) FROM spacex)')
for row in cur:
    print('Booster version with max payload is', row)
```

- The names of the boosters which have carried the maximum payload mass are:

F9 B5 B1048.4	F9 B5 B1049.5
F9 B5 1049.4	F9 B5 1060.2
F9 B5 B1051.3	F9 B5 B1058.3
F9 B5 1056.4	F9 B5 1051.6
F9 B5 B1048.5	F9 B5 B1060.3
F9 B5 1051.4	F9 B5 1049.7

```
Booster version with max payload is ('F9 B5 B1048.4',)
Booster version with max payload is ('F9 B5 B1049.4',)
Booster version with max payload is ('F9 B5 B1051.3',)
Booster version with max payload is ('F9 B5 B1056.4',)
Booster version with max payload is ('F9 B5 B1048.5',)
Booster version with max payload is ('F9 B5 B1051.4',)
Booster version with max payload is ('F9 B5 B1049.5',)
Booster version with max payload is ('F9 B5 B1060.2 ',)
Booster version with max payload is ('F9 B5 B1058.3 ',)
Booster version with max payload is ('F9 B5 B1051.6',)
Booster version with max payload is ('F9 B5 B1060.3',)
Booster version with max payload is ('F9 B5 B1049.7 ',)
```

2015 Launch Records

```
cur.execute('''SELECT Booster_Version, Launch_Site FROM temp
WHERE "Landing _Outcome" = ? AND Year = 2015''', [success_outcomes[2]] )
for row in cur:
    print('Failed landing outcomes in drone ship with Booster version ', row[0], 'and Launch Site ', row[1])
```

```
Failed landing outcomes in drone ship with Booster version  F9 v1.1 B1012 and Launch Site  CCAFS LC-40
Failed landing outcomes in drone ship with Booster version  F9 v1.1 B1015 and Launch Site  CCAFS LC-40
```

- The launch site CCAFS LC-40 failed in their attempt to land on a drone ship in 2015 using booster versions F9 V1.1 B1012 and F9 v1.1 B1015

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

```
cur.execute('''SELECT  DISTINCT "LANDING _OUTCOME" , COUNT(MISSION_OUTCOME) FROM spacex
GROUP BY "LANDING _OUTCOME"
ORDER BY  COUNT(MISSION_OUTCOME) DESC''')
for row in cur:
    print(row)
```

```
('Success', 38)
('No attempt', 21)
('Success (drone ship)', 14)
('Success (ground pad)', 9)
('Failure (drone ship)', 5)
('Controlled (ocean)', 5)
('Failure', 3)
('Uncontrolled (ocean)', 2)
('Failure (parachute)', 2)
('Precluded (drone ship)', 1)
('No attempt ', 1)
```

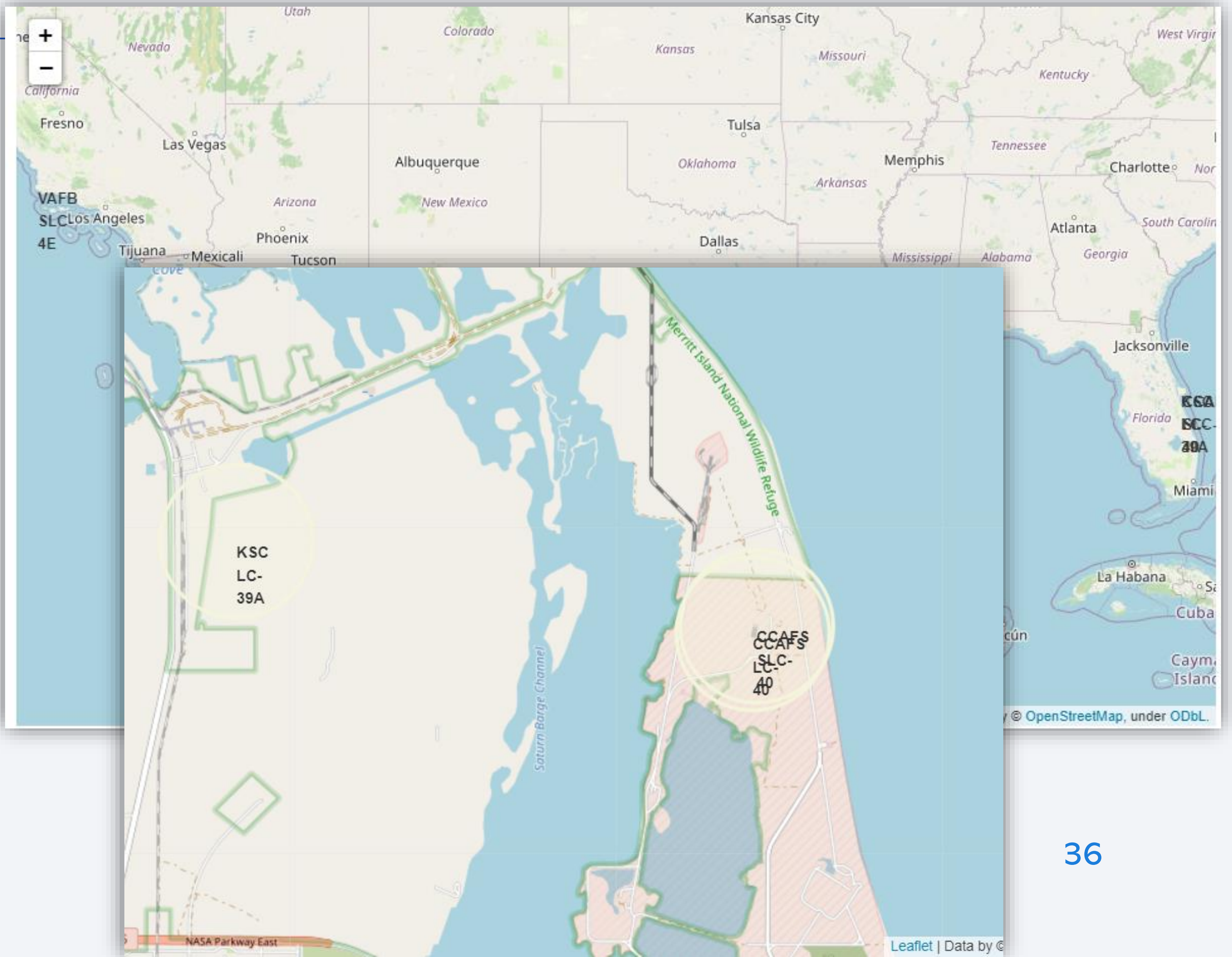
- There were 11 distinct landing outcomes between the date 2010-06-04 and 2017-03-20, ranked here in descending order of landing outcomes

Section 3

Launch Sites Proximities Analysis

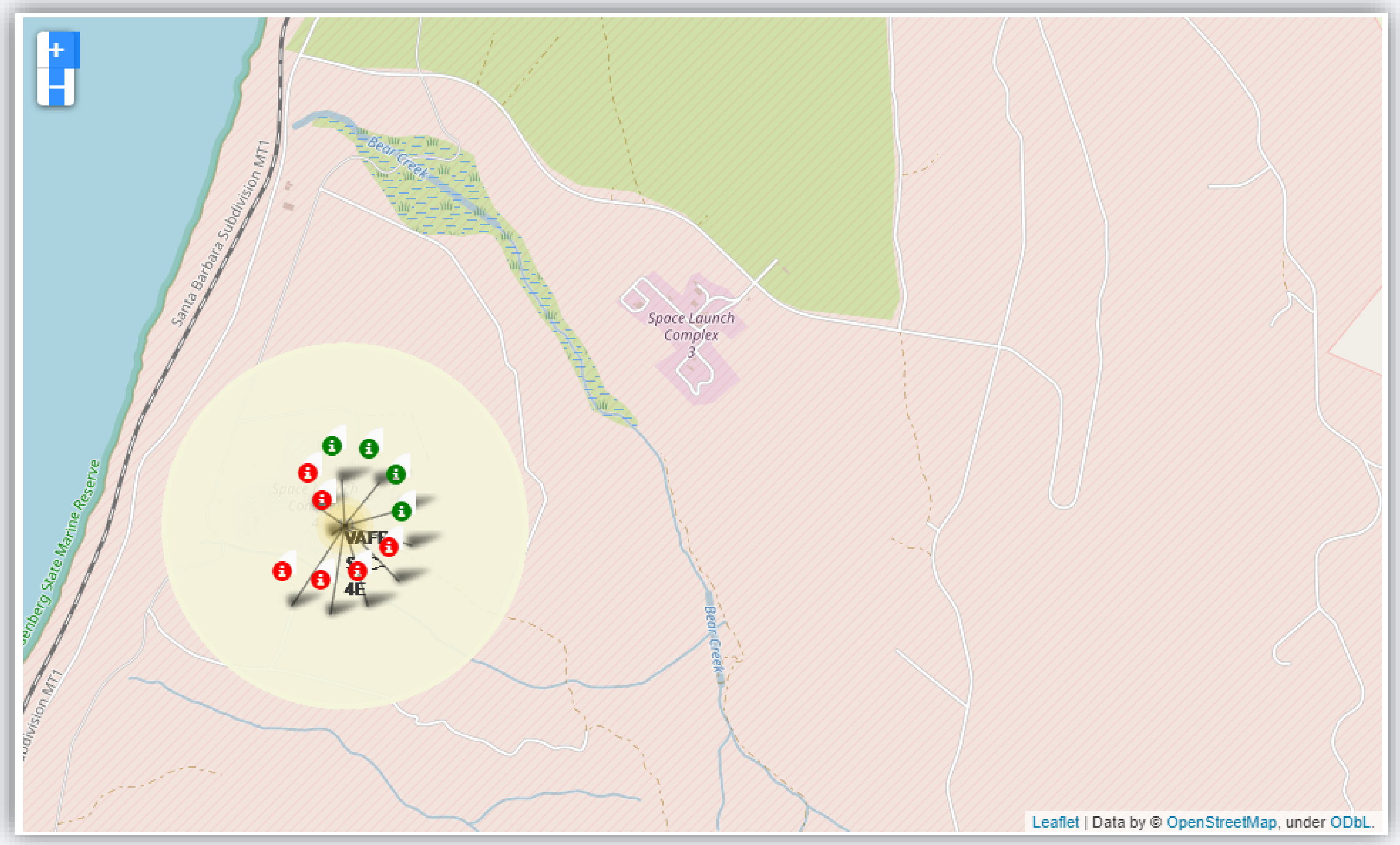
Launch Site Location Markers

- The adjacent Folium map shows the location for all the launch sites
- Launch site VAFB SLC 4E is located on the west coast.
- Launch sites CCAFS SLC 40, CCAFS LC-40 and KSC LC 39A are located on the east coast



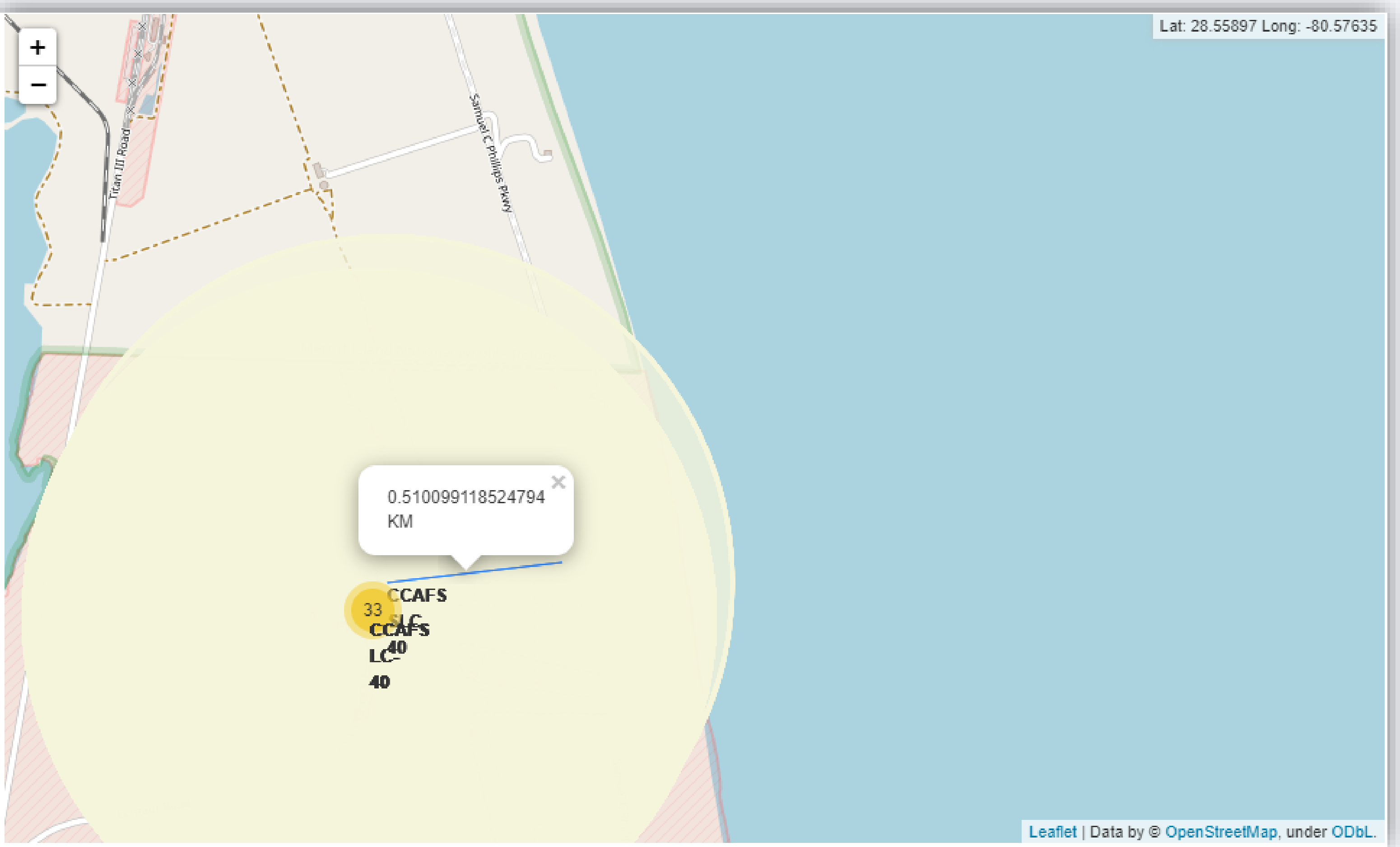
Launch Outcomes for VAFB SLC-4E

- The adjoining image shows the launch outcomes for launch site VAFB SLC-4E
  - Green: Successful
  - Red: Unsuccessful



Distance Between Coastline and Launch Site CCAFS-SLC-40

- The generated folium map shows the screenshot of launch site CCAFS SLC 40 and its proximity to coastline.
    - The line indicates the distance between the launch site and coastine
- CCAFS SLC 40 is located 0.51 km away from the coastline

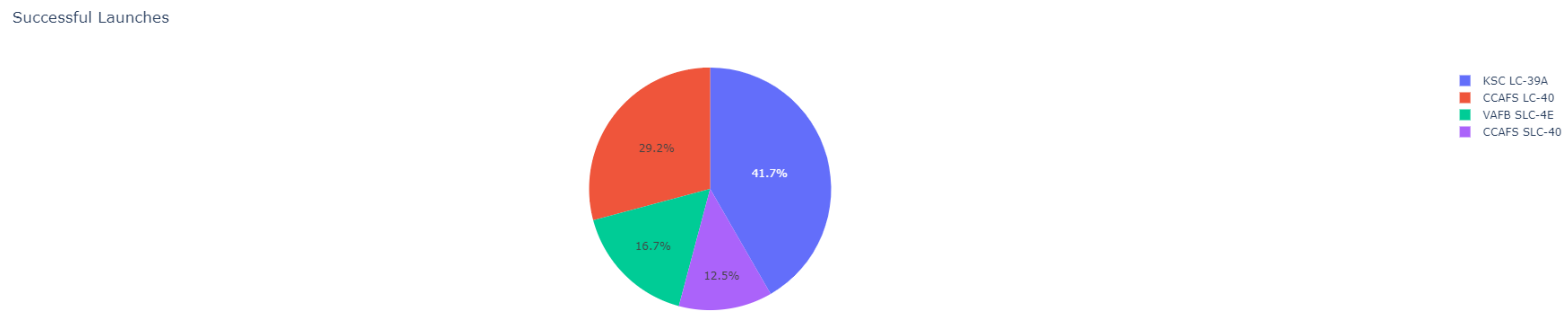


Section 4

Build a Dashboard with Plotly Dash

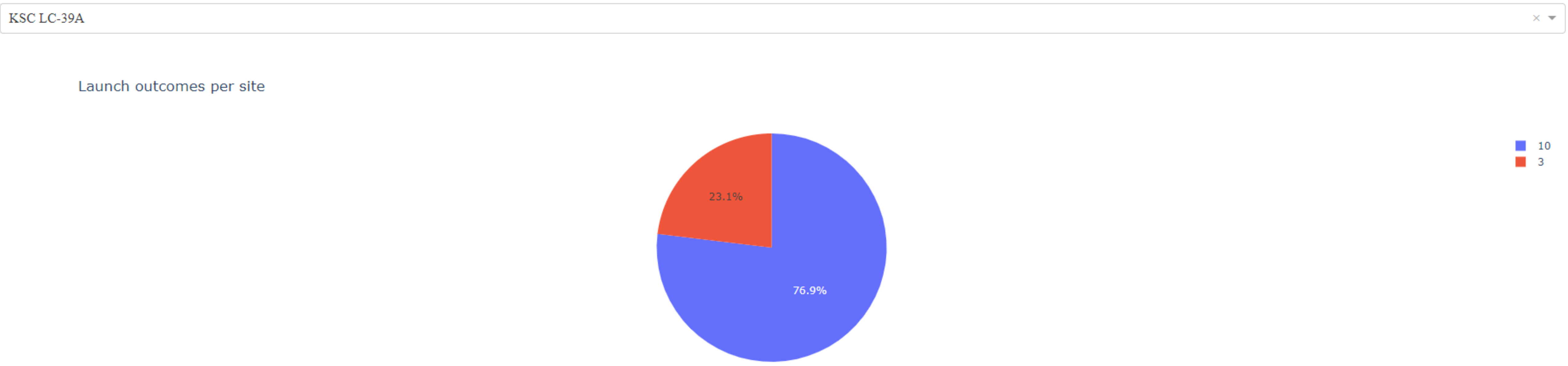
Successful Launches For All Sites

- The success outcomes for the various launch sites are as follows"
    - KSC LC-39A: 41.7%
    - CCAFS SLC-40: 12.5%
    - VAFB SLC-4E: 16.7%
    - CCAFS LC-40: 29.2%



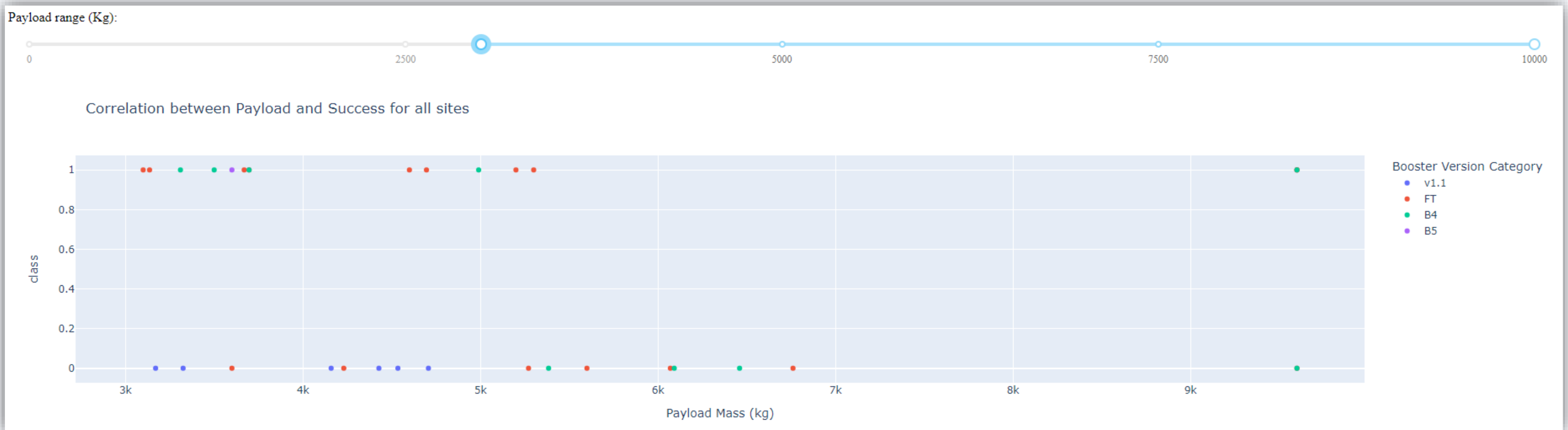
Launch Outcomes For Launch Site with Highest Launch Success Ratio

- KSC LC 39A had the highest launch success ratio of 76.9%



Payload vs Launch Outcome for All Sites

- The following plot shows the correlation between payload mass and success outcomes for all sites with payload range between 2750 kg and 10000 kg
- Generally, the sites had successful outcomes for payloads between 3000 kg and 5250 kg



Payload vs Launch Outcome for All Sites

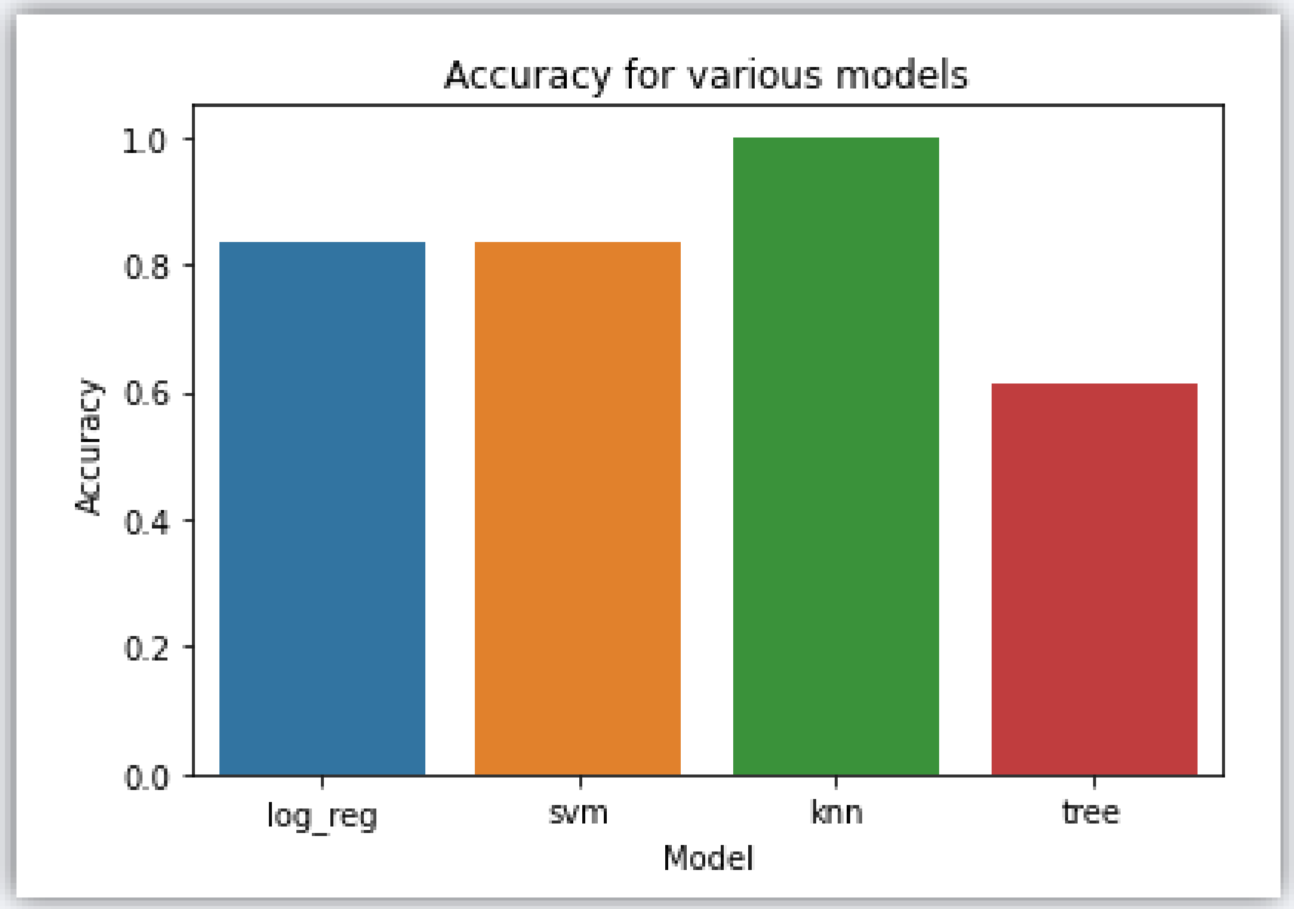
- The following plot shows the correlation between payload mass and success outcomes for all sites with payload range between 5750 kg and 10000 kg
- Generally, sites had unsuccessful outcomes for chosen payload range

Section 5

Predictive Analysis (Classification)

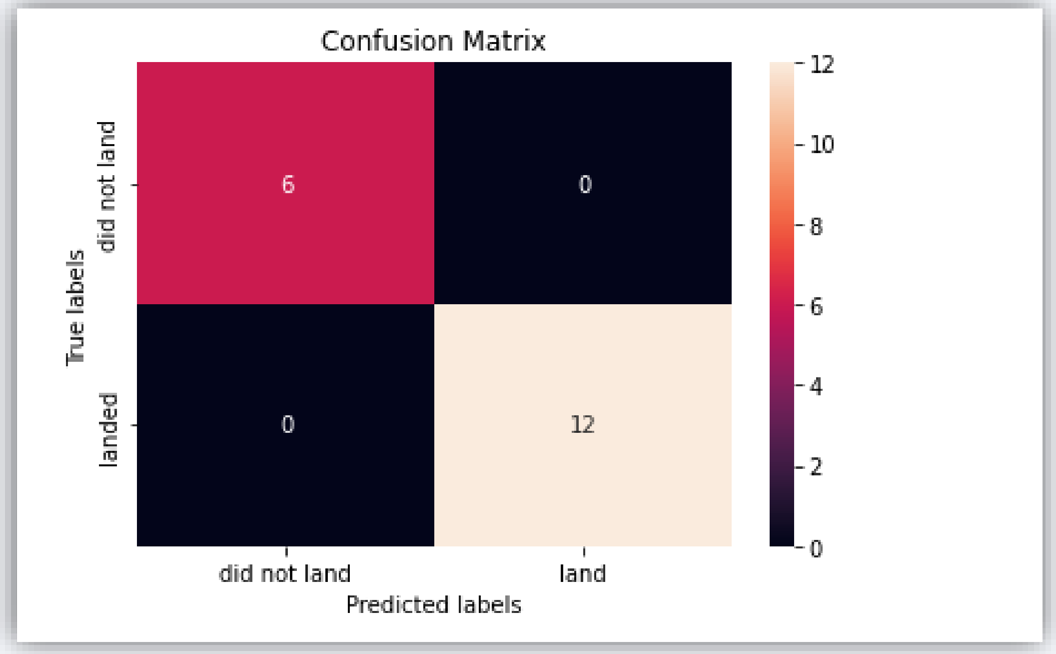
Classification Accuracy

- Models were compared in their accuracy to correctly classify success outcomes
- Knn model has the highest classification accuracy



Confusion Matrix

- Shown here is the confusion matrix for knn model
- The classification accuracy was 1.0 since it accurately classified the successful and unsuccessful outcomes



Conclusions

Launch success outcomes improved over time

Overall, launches with payloads in the range of 3000 - 5250 kg were successful

Orbits ES-L1, GEO, HEO, SSO had a 100% success rate

Overall, the success rate in terms of mission outcomes was 99%

Launch site KSC LC 39A had the highest launch success ratio of 76.9%

Knn model had 100% classification accuracy

Appendix

- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W1_DataWrangling.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W1_WebScraping.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W2_jupyter-labs-eda-dataviz.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W2_EDA-SQL.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W3_DataVizFolium.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W3_SpaceXDash_app.ipynb
- https://github.com/learner-nkn/datasciencecoursera/blob/main/C10_W4_PredictionPipeline.ipynb

Thank you!