Winning Space Race with Data Science

Kay Sun
May 21, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection methodology
  - Perform data wrangling
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly Dash
  - Perform predictive analysis using classification models
- Summary of all results
  - Exploratory data analysis
  - Launch site proximities analysis
  - Interactive analytics demo
  - Predictive analysis

Introduction

- Project background and context
  - SpaceX charges less than other providers for rocket launches since it can reuse the first stage.
  - For SpaceX to determine how much to charge, it needs to be able to predict if the Falcon 9 first stage will land successfully, allowing for it to be used.
  - This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - Relationships between rocket launch parameters and launch success rates based on historical data.
  - Predict launch success from rocket launch parameters using machine learning modeling.

Section 1

Methodology

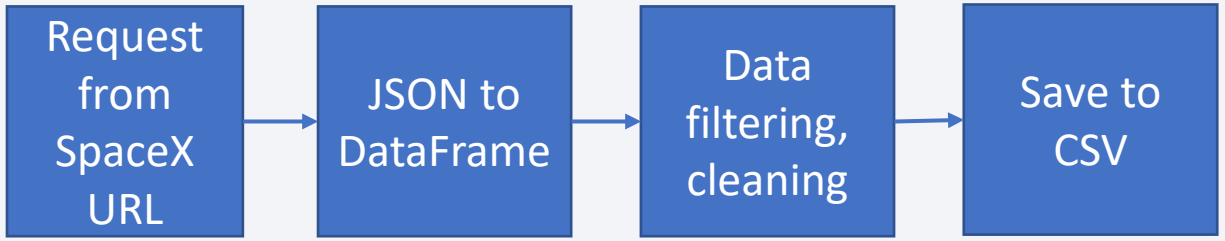
Methodology

Executive Summary

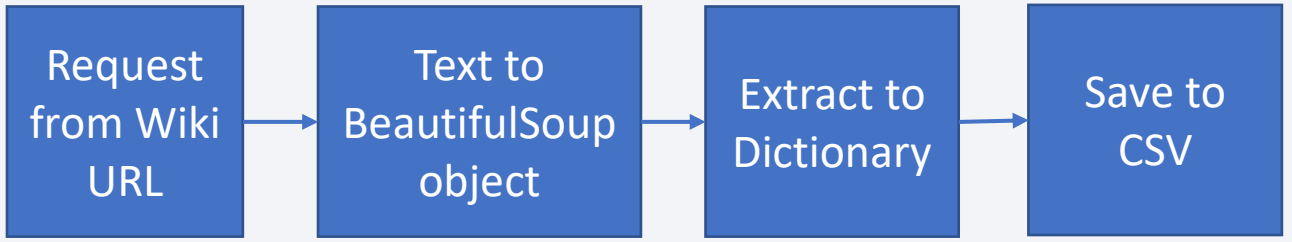
- Data collection methodology:
  - From SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - Convert outcome by success or failure
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Hyperparameter tunning and model evaluation

Data Collection

- Data Sources
  - SpaceX API (https://api.spacexdata.com/v4/launches/past)



  - Wikipedia page of SpaceX launches (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)



Data Collection – SpaceX API

1. GET request from URL

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

response = requests.get(spacex_url)
```

2. Convert JSON to DataFrame

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

3. Functions to extract data

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Populate extracted data into Dictionary.

```
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. Convert Dictionary to DataFrame

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

6. Filter and impute missing values

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion'] == 'Falcon 9']
```

```
# Calculate the mean value of PayloadMass column
mean_payloadmass = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean_payloadmass)
```

7. Save to CSV

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


Github link

Data Collection - Scraping

1. GET request from URL

```
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
# assign the response to a object
response = requests.get(static_url).text
```

2. Convert to BeautifulSoup object

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html5lib')
```

3. Find elements in object

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4. Populate extracted data into 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 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']=[]
```

5. Convert Dictionary to DataFrame

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

6. Save to CSV

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


Github link

Data Wrangling

- Convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.

1. Explore ‘Outcomes’

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

2. Assign ‘Class’ based on ‘Outcomes’

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
```

3. Calculate average success rate

```
df["Class"].mean()
```

4. Save to CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```


Github link

EDA with Data Visualization

- Flight Number vs Launch Site by Launch Success
  - Visualize in scatterplot relationship between flight number and launch site, colored based on launch success.
- Payload Mass vs Launch Site
  - Visualize in scatterplot relationship between payload mass and launch site, colored based on launch success.
- Success Rate vs Orbit
  - Visualize in horizontal bar chart relationship between success rate and orbit.
- Flight Number vs Orbit
  - Visualize in scatterplot relationship between flight number and orbit, colored based on launch success.
- Payload Mass vs Orbit
  - Visualize in scatterplot relationship between payload mass and orbit, colored based on launch success.
- Year vs Success Rate
  - Visualize in line relationship between year and success rate.

Github link

EDA with SQL

- SQL queries
  - Display the names of the unique launch sites in the space mission
  - 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
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

Github link

Build an Interactive Map with Folium

- Added to map
  - Markers to show launch sites
  - Markers to show success or failure for each launch site
  - Lines to show distances between launch site and proximities.
- Graphical illustrations show geographical positions, launch successes and distances to proximities for easy understanding.

Github link

Build a Dashboard with Plotly Dash

- Pie chart to show launch success by sites and success/failure per site.
  - User can select launch site.
  - Illustrate success rate for each site or in total.
- Scatter plot to show relationship between launch success and payload mass per booster version category.
  - User can select launch site and payload mass levels from the slider.
  - Illustrate success rate by payload, booster version and site.

Github link

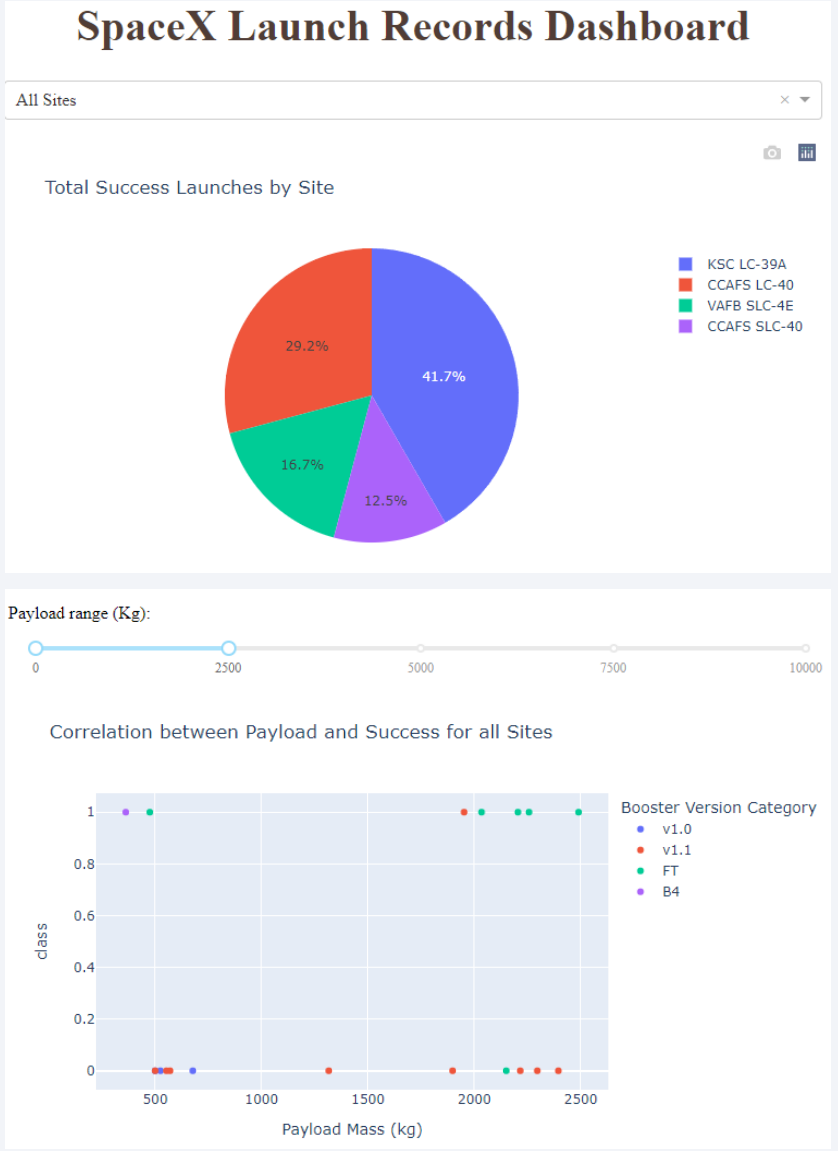
Predictive Analysis (Classification)

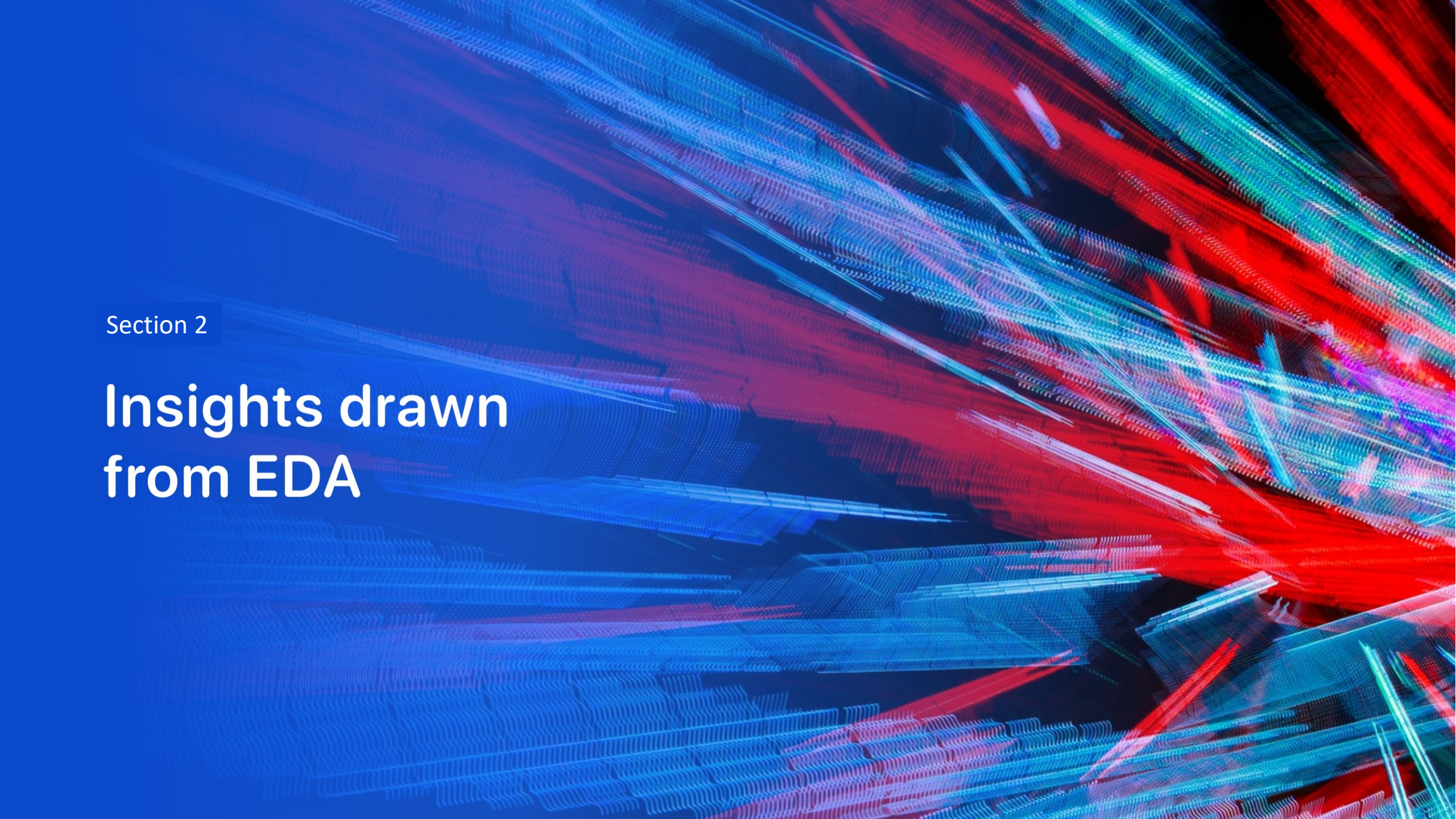
- Extract target from data
- Standardization of data
- Train/test dataset split
- Models (Logistic Regression, SVM, Decision Tree, KNN)
- Hyperparameter tunning and cross validation
- Evaluate performance using accuracy as metric
- Select best performing model and parameters

Github link

Results

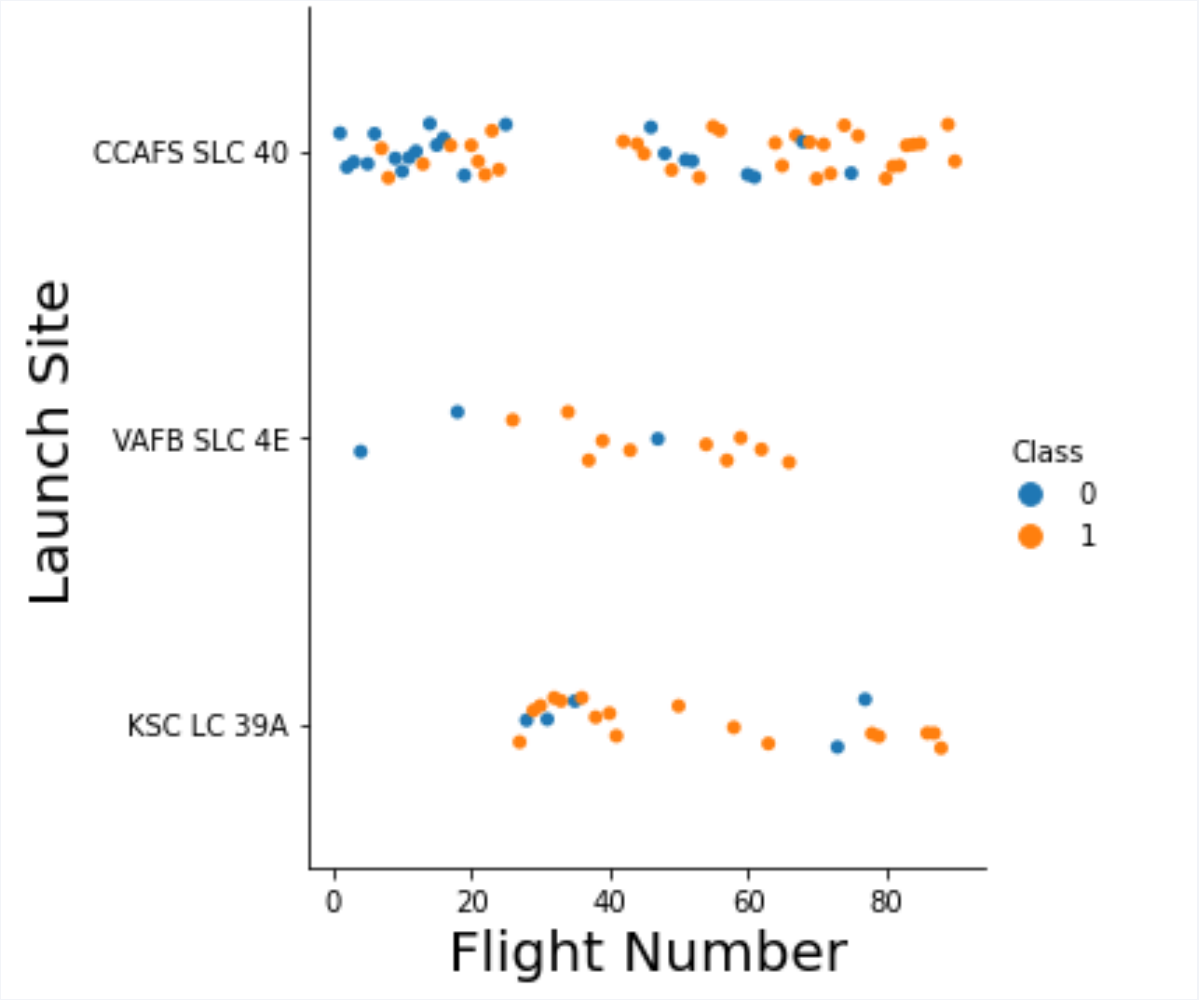
- Data collected from SpaceX API and web crawling
  - Filtered and cleaned.
  - Relationships between flight number, launch site, payload mass, orbit, and success rate.
  - Visualized geographically on map.
- Screenshots of interactive analytics demo illustrated on the right
  - Total success launches by site
  - Correlation between payload and success launches for all sites
- All the 4 models performed similarly with accuracy of around 0.83.




Section 2
Insights drawn from EDA

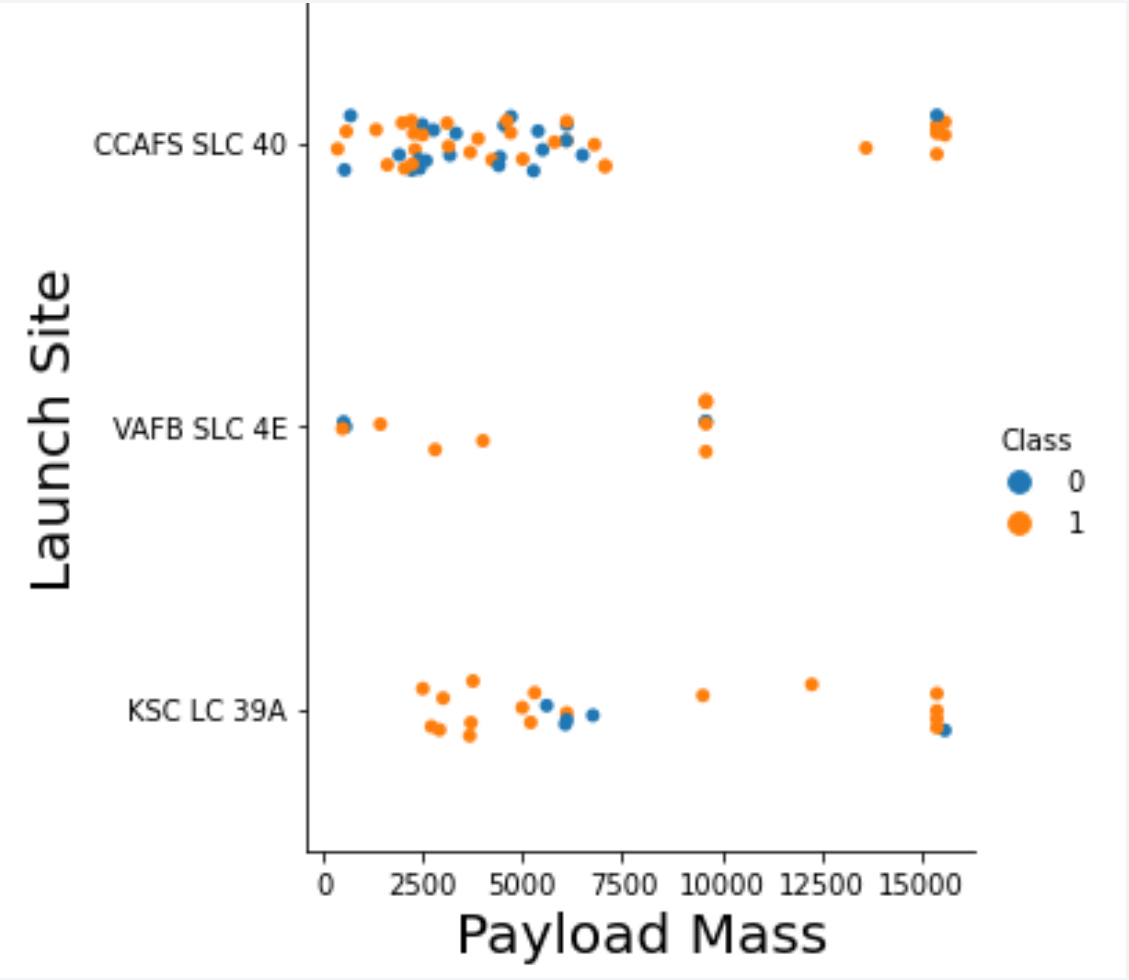
Flight Number vs. Launch Site

- Blue indicates failure, while orange indicates successes.
- More failure launches at the start of the program (flight numbers less than 40).
- More successes with more flights and experience.
- Most flights are at CCAFS SLC 40.



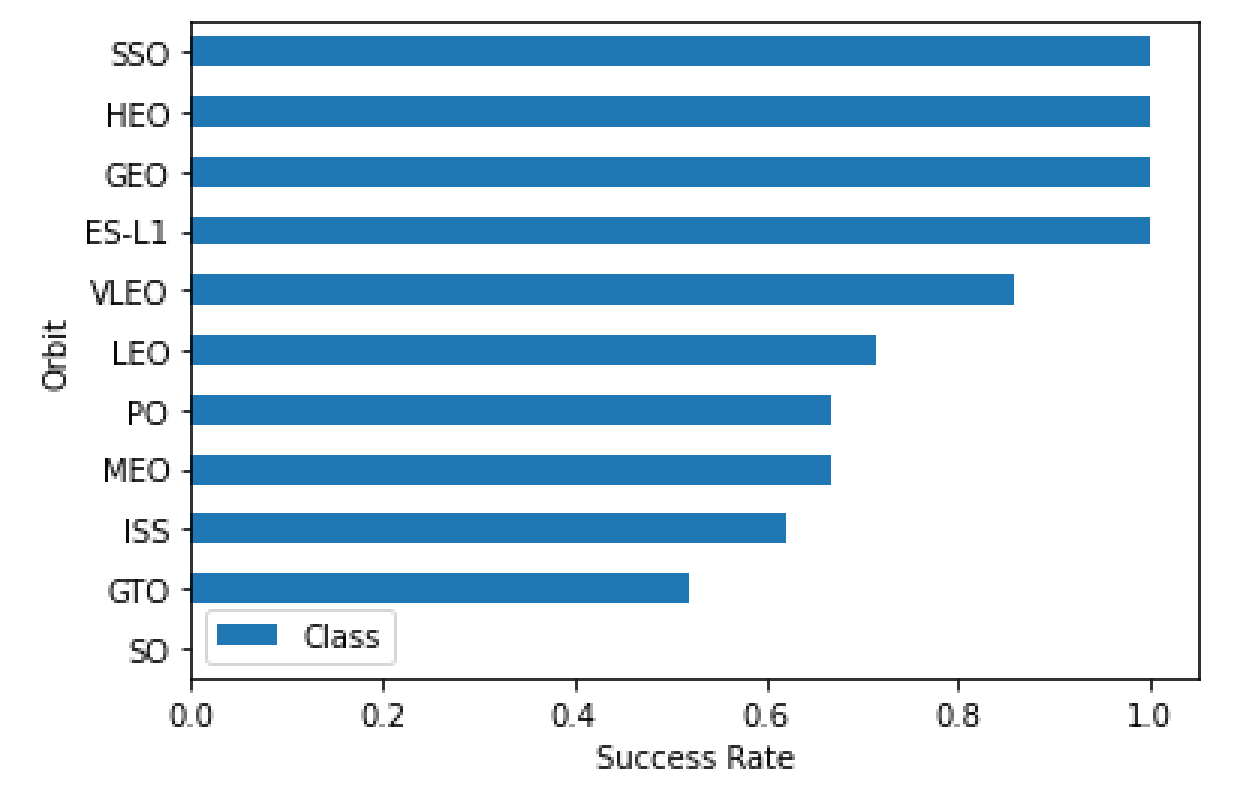
Payload vs. Launch Site

- Blue indicates failure, while orange indicates successes.
- Mix of success and failures across the payload mass.
- There have been both successes and failures at high, middle and low payloads.
- Most flights are at CCAFS SLC 40.



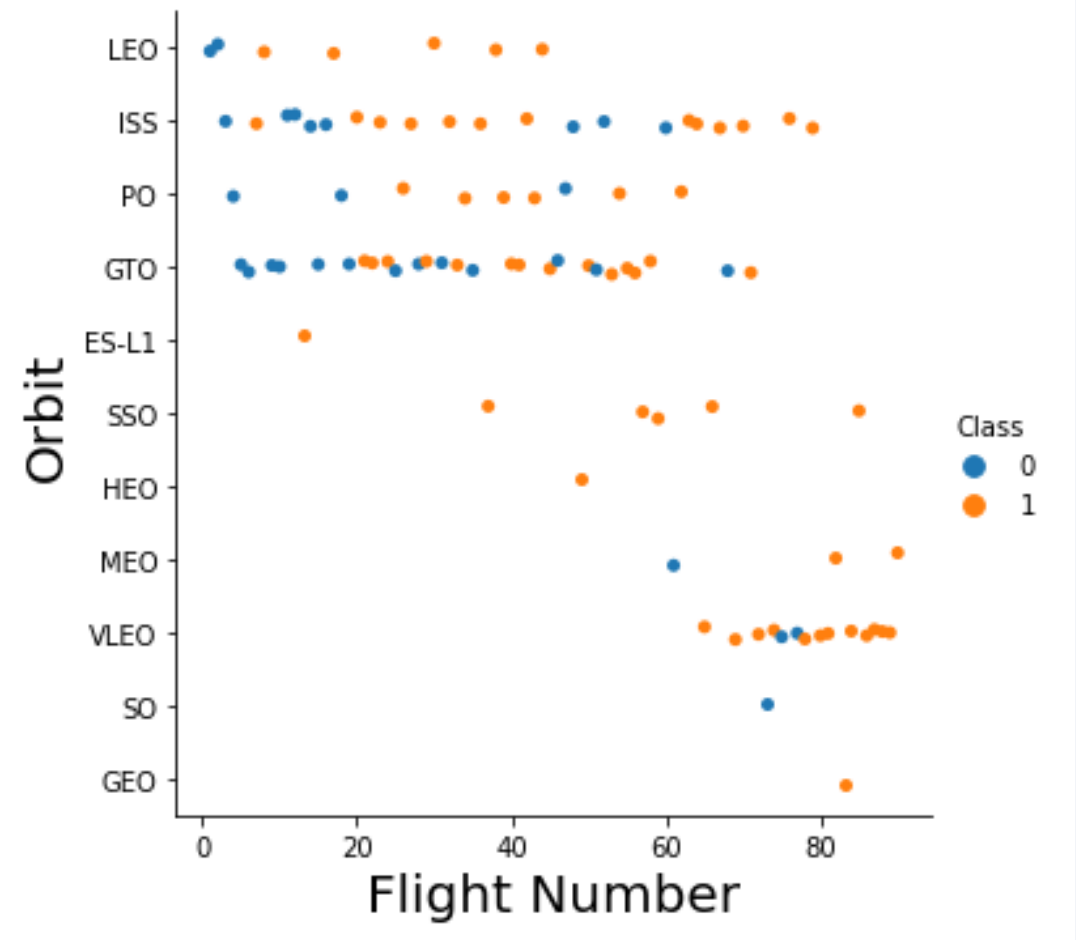
Success Rate vs. Orbit Type

- Highest success rate at SSO, HEO, GEO and ES-L1.
- SO has not had a success.
- GTO is only 0.5 success rate.



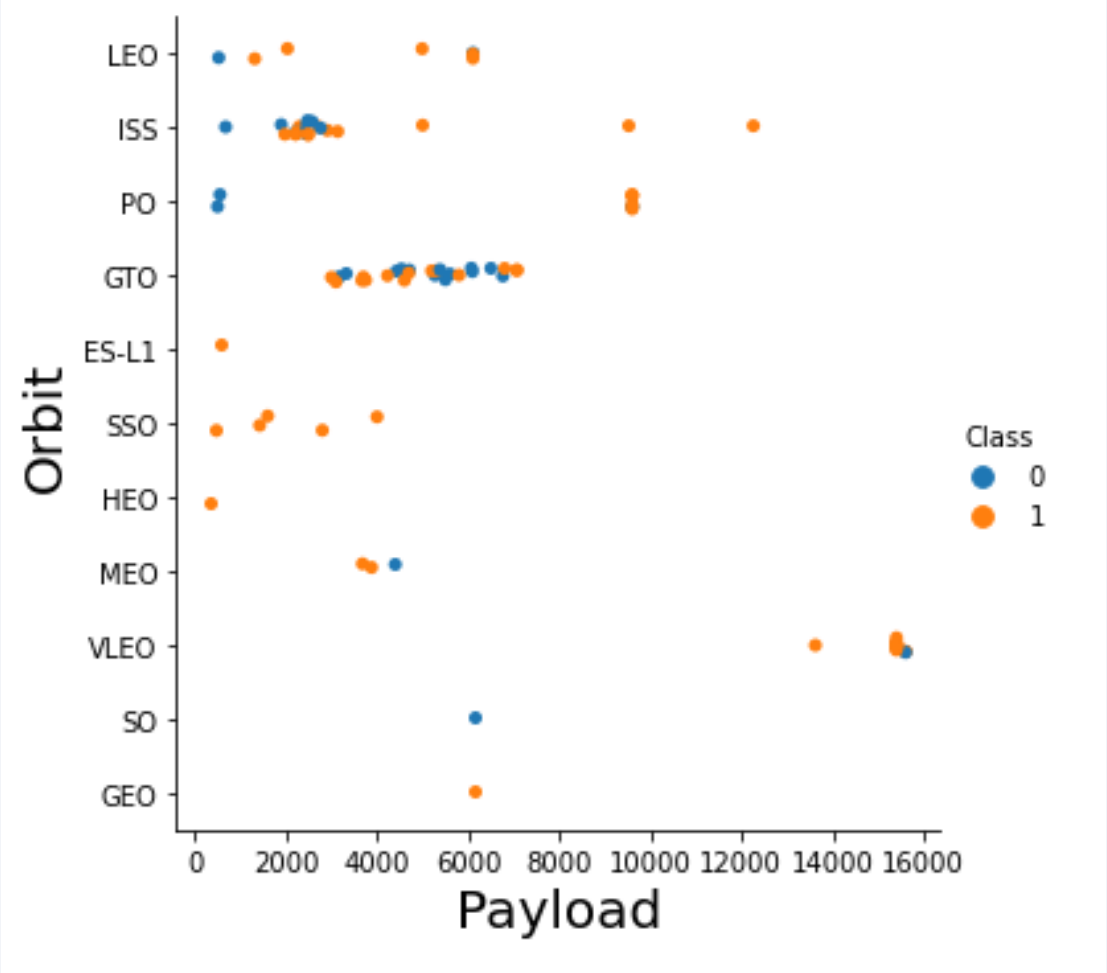
Flight Number vs. Orbit Type

- Blue indicates failure, while orange indicates successes.
- Most flights have been for LEO, ISS, PO, GTO, and VLEO with mixed to majority successes.
- SSO has had few flights but all have been successful.
- SO sole flight was a failure.



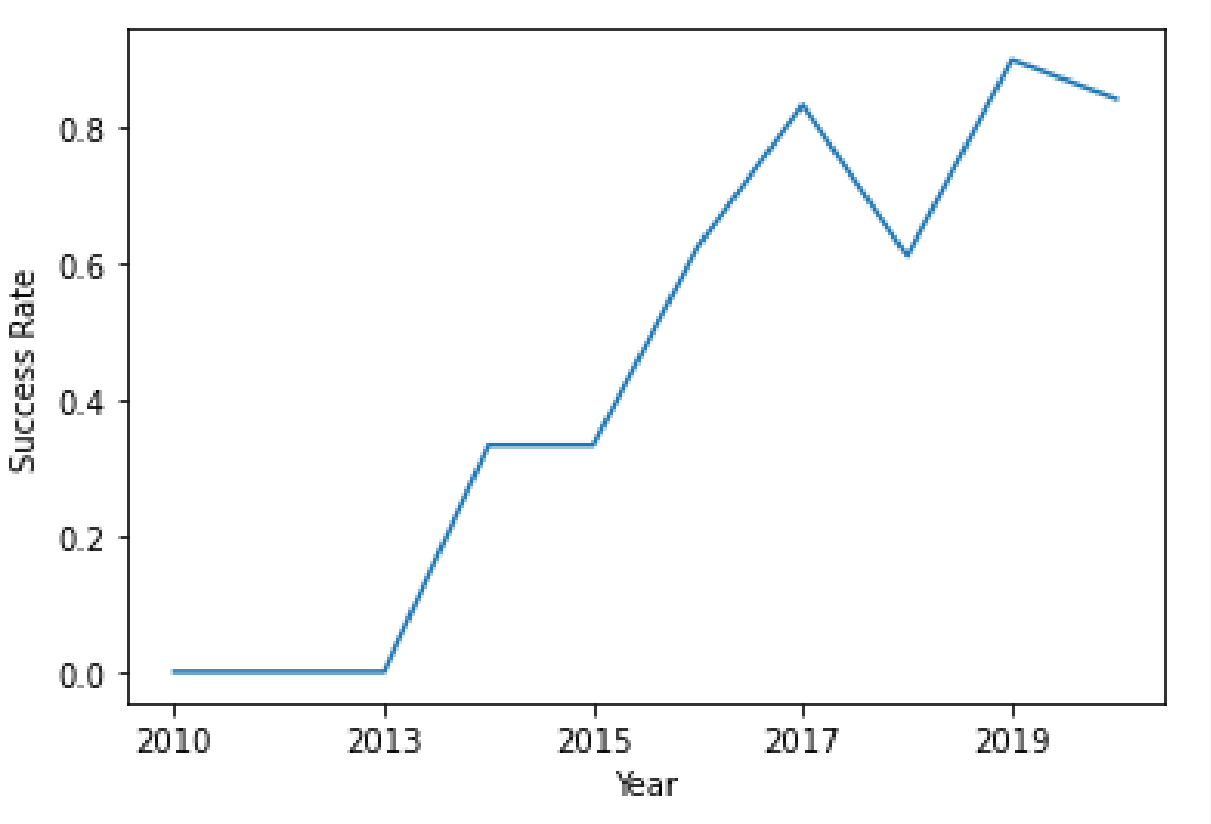
Payload vs. Orbit Type

- Blue indicates failure, while orange indicates successes.
- Lower payloads had greater success for ES-L1, SSO, HEO.
- Greater successes for higher payloads for LEO, ISS and PO.
- GTO has mixed success across all their payloads.



Launch Success Yearly Trend

- General trend of increasing success rate since 2013.
- Large drop in 2018.
- Latest success rate at 0.84 in 2020.



All Launch Site Names

- Query ‘launch_site’ from ‘spacex’ table.
- Only unique launch sites with ‘distinct’.

```
%sql select distinct launch_site from spacex;
```

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

Launch Site Names Begin with 'CCA'

- Query all from ‘spacex’ table.
- Condition of ‘launch_site’ that starts with ‘CCA’. % is wild card for trailing characters.
- Limit to 5 results.

```
%sql select * from spacex \
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

- Query ‘payload_mass__kg_’ from ‘spacex’ table.
- Sum the ‘payload_mass__kg_’.
- Condition of ‘customer’ is ‘NASA (CRS)’.

```
%sql select sum(payload_mass__kg_) from spacex \
where customer = 'NASA (CRS)';
```

1
45596

Average Payload Mass by F9 v1.1

- Query ‘payload_mass__kg_’ from ‘spacex’ table.
- Average the ‘payload_mass__kg_’.
- Condition of ‘booster_version’ is ‘F9 v1.1’.

```
%sql select avg(payload_mass__kg_) from spacex \
where booster_version = 'F9 v1.1';
```

1
2928

First Successful Ground Landing Date

- Query ‘date’ from ‘spacex’ table.
- Find minimum ‘date’.
- Condition of ‘landing_outcome’ is ‘Success (ground pad)’.

```
%sql select min(date) from spacex \
where landing__outcome = 'Success (ground pad)';
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query ‘booster_version’ from ‘spacex’ table.
- Condition of ‘landing_outcome’ is ‘Success (ground pad)’ and
- Condition of ‘payload_mass__kg_’ is between 4000 and 6000 kg.

```
%sql select booster_version from spacex \
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

- Query ‘mission_outcome’ from ‘spacex’ table.
- Count the number of ‘mission_outcome’ .
- Grouping by the mission_outcome.

```
%sql select mission_outcome, count(*) from spacex \
group by mission_outcome;
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Query ‘booster_version’ and ‘payload_mass__kg_’ from ‘spacex’ table.
- Only distinct ‘booster_version’
- Sub-query for maximum ‘payload_mass__kg_’ from ‘spacex’ table.

```
%sql select distinct booster_version, payload_mass__kg_ from spacex \
where payload_mass__kg_ = ( \
select max(payload_mass__kg_) from spacex \
);
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- Query ‘landing__outcome’, ‘booster_version’ and ‘launch_site’ from ‘spacex’ table.
- Condition of ‘landing__outcome’ is ‘Failure (drone ship)’
- And condition of year of date is 2015.

```
%sql select landing__outcome, booster_version, launch_site from spacex \
where landing__outcome = 'Failure (drone ship)' \
and year(date) = '2015' \
;
```

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

- Query ‘landing__outcome’ from ‘spacex’ table.
- Count the number of ‘landing__outcome’ as total
- Condition of ‘date’ is between 2010-06-04 and 2017-03-20
- Grouping by different landing__outcome
- Sorting by the total landing outcome.

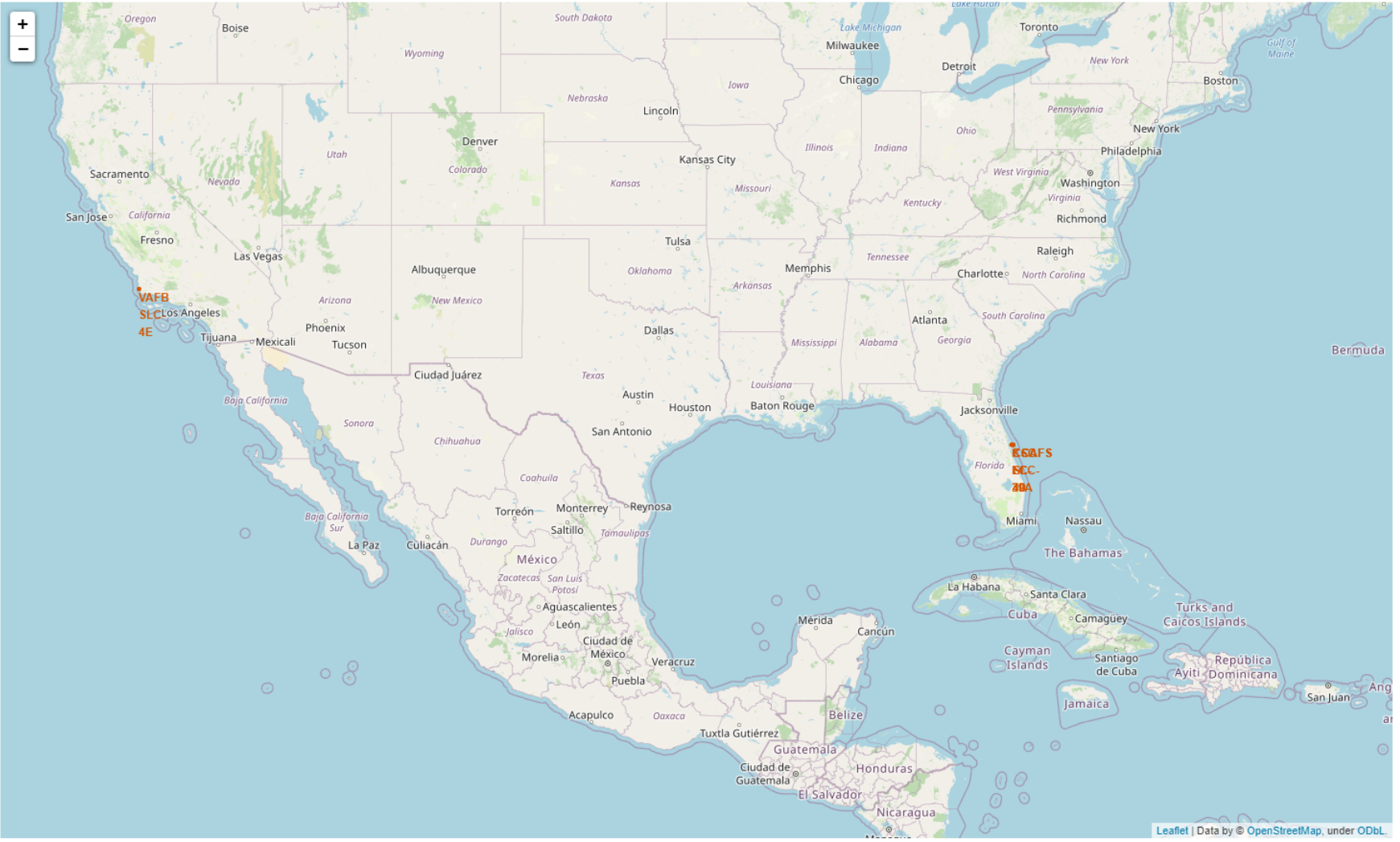
```
%sql select landing__outcome, count(landing__outcome) as total from spacex \
where date between '2010-06-04' and '2017-03-20' \
group by landing__outcome \
order by total desc \
;
```

landing_outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

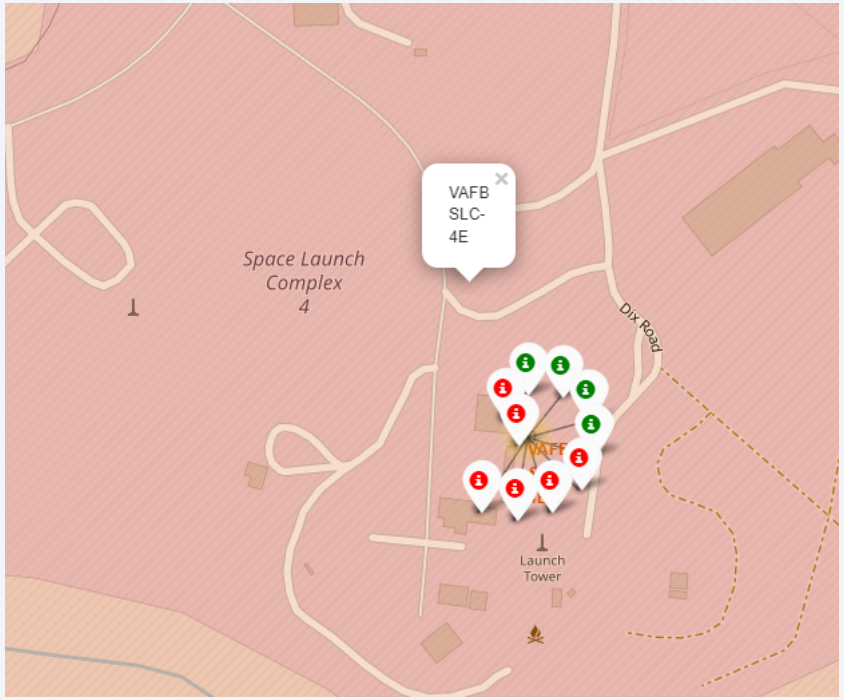
Launch Sites Proximities Analysis

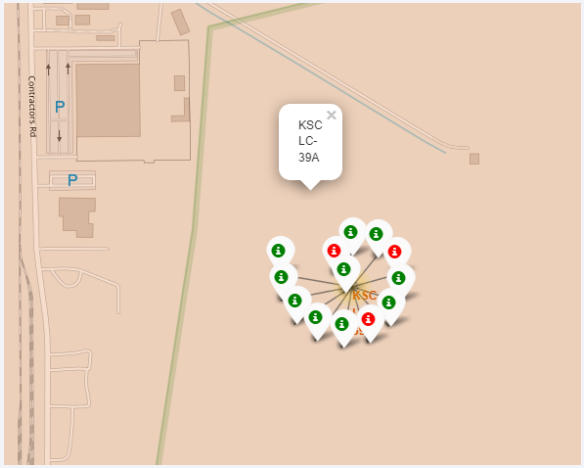
All Launch Sites Locations

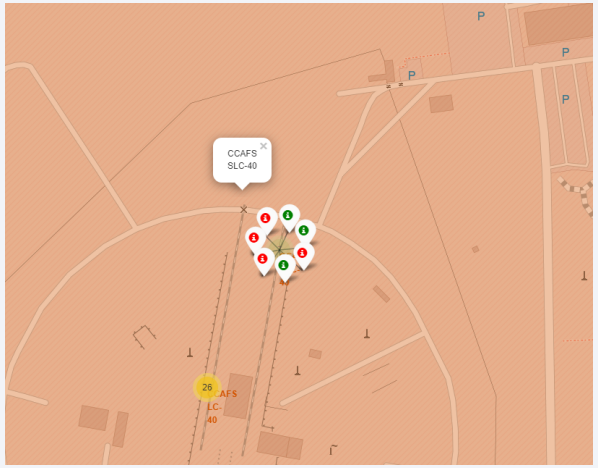


- Launch sites are in California and Florida.
- Both are along the coast to avoid land in the event of failure.

Color-labeled Launch Outcomes

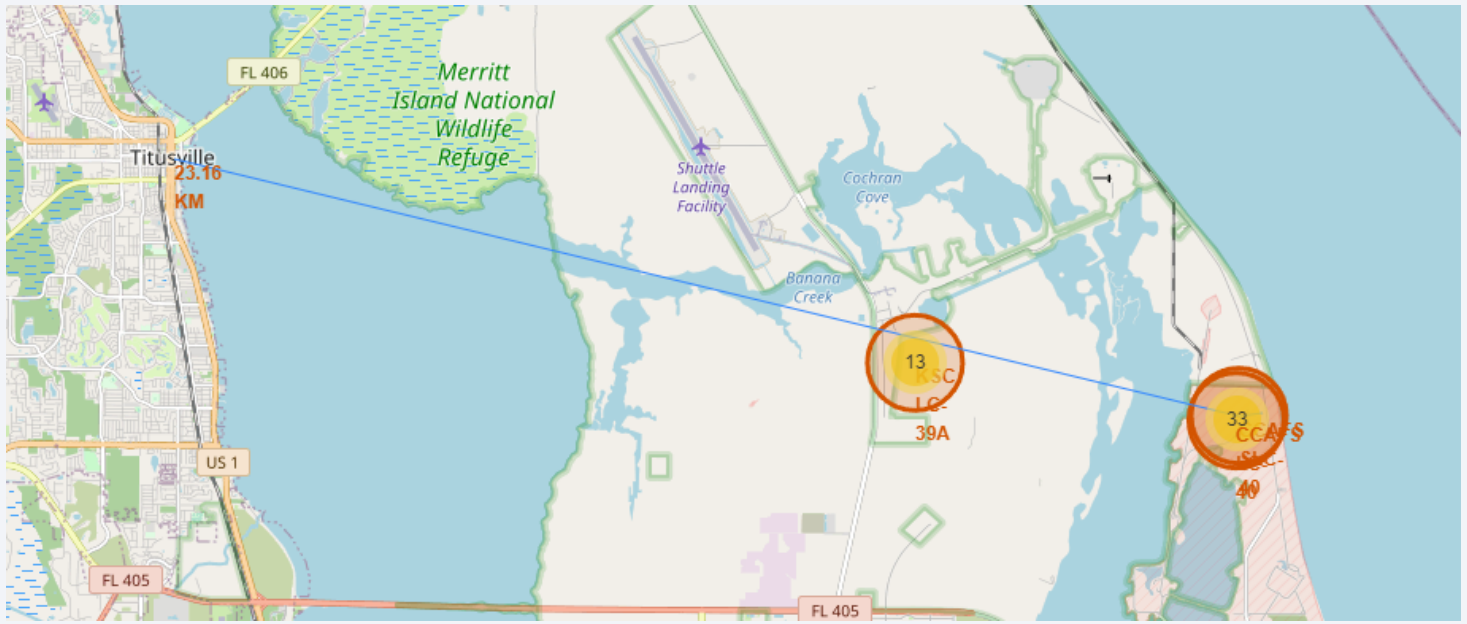


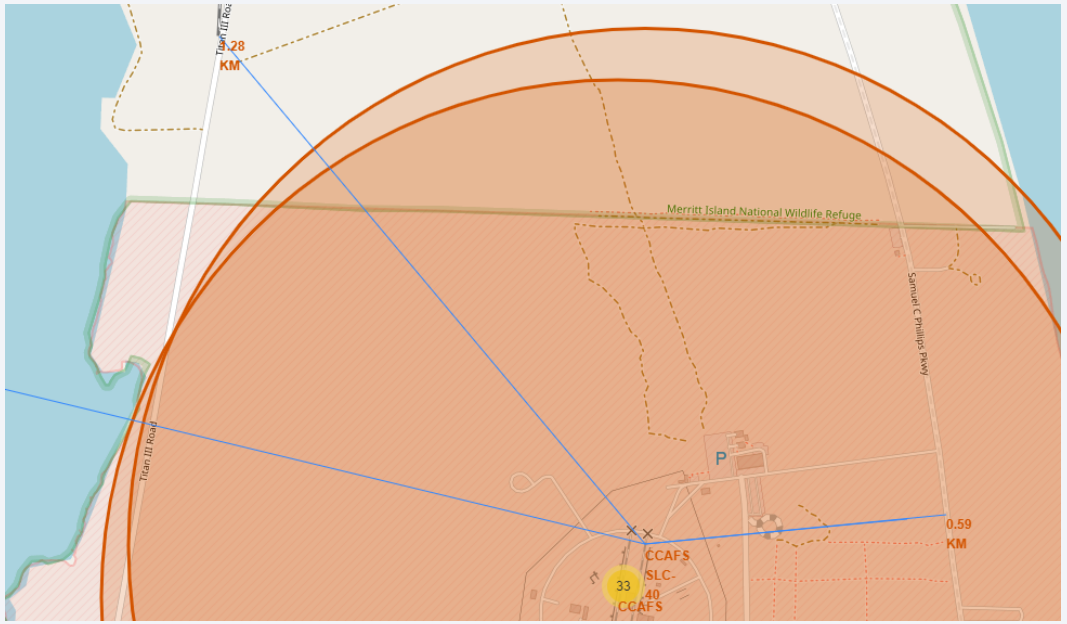




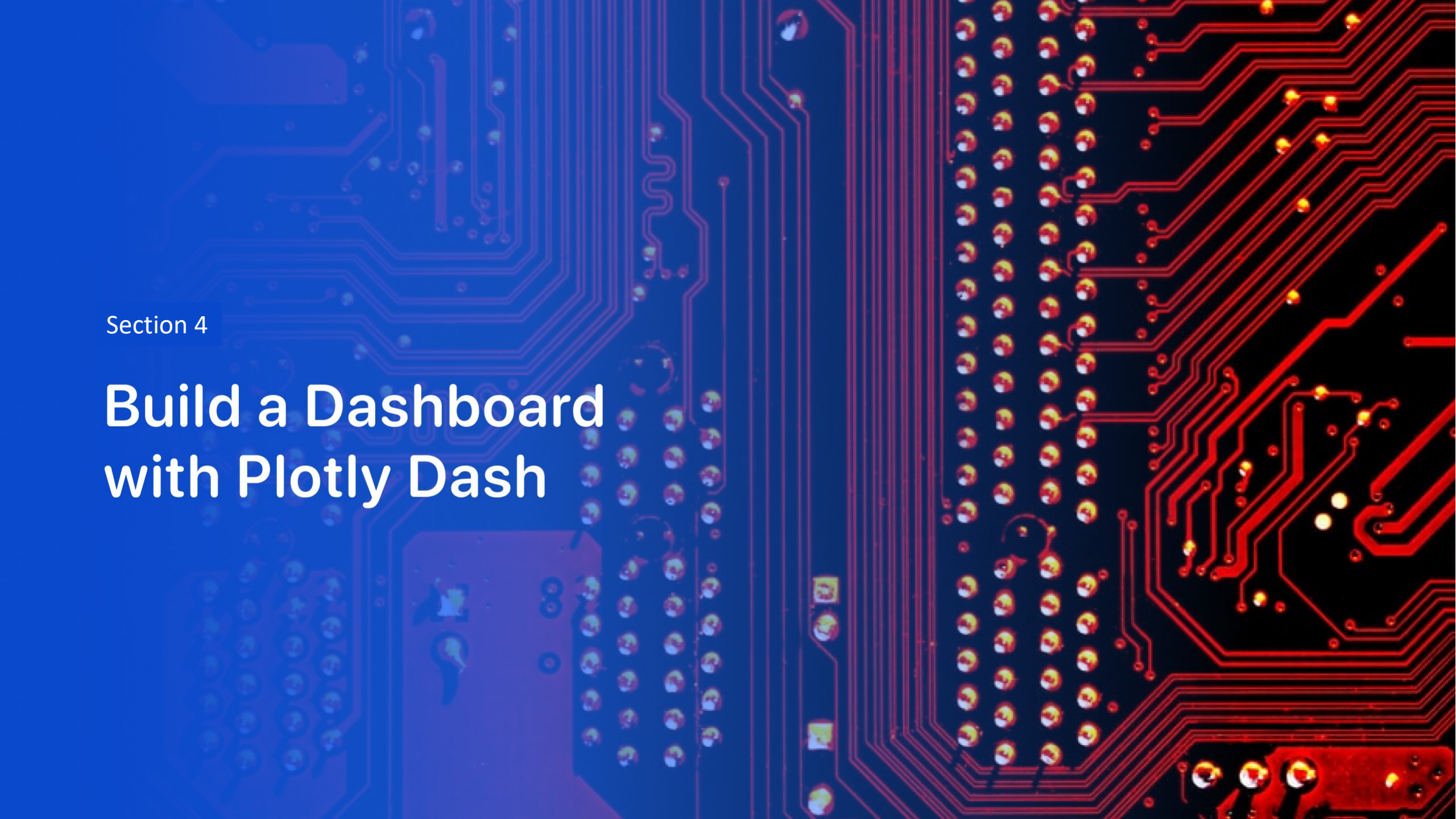
- California site is on the left, while the 3 Florida sites are on the right.
- Red markers indicate failures, while green markers indicate success.

Launch Site to Proximities



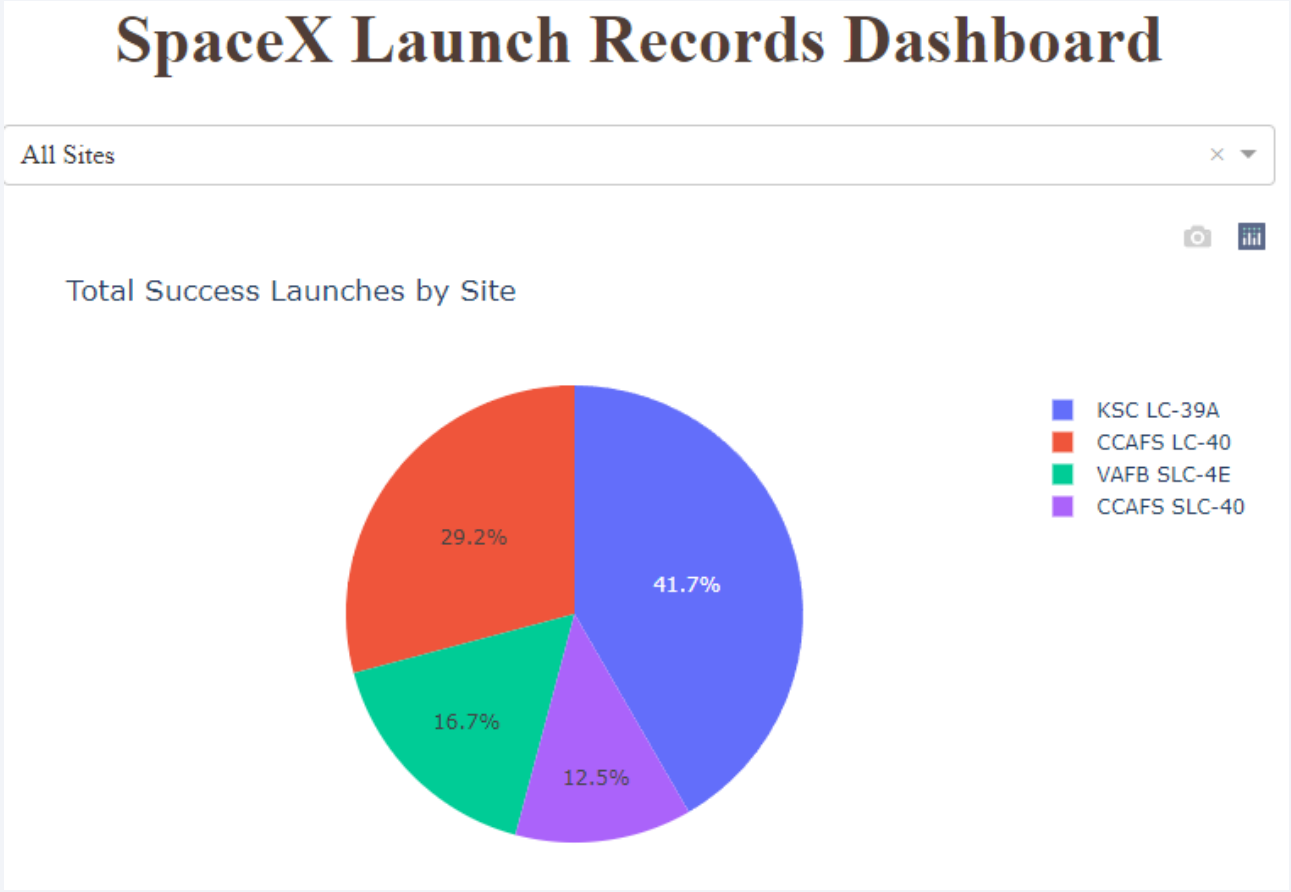


- Distance from CCAFS SLC-40 to Titusville is 23.16 km.
- Distance from CCAFS SLC-40 to railway is 1.28 km.
- Distance from CCAFS SLC-40 to highway is 0.59 km.
- These close proximities to cities and transportations make availability of equipment and products as well as personnel accessible.


Section 4
Build a Dashboard with Plotly Dash

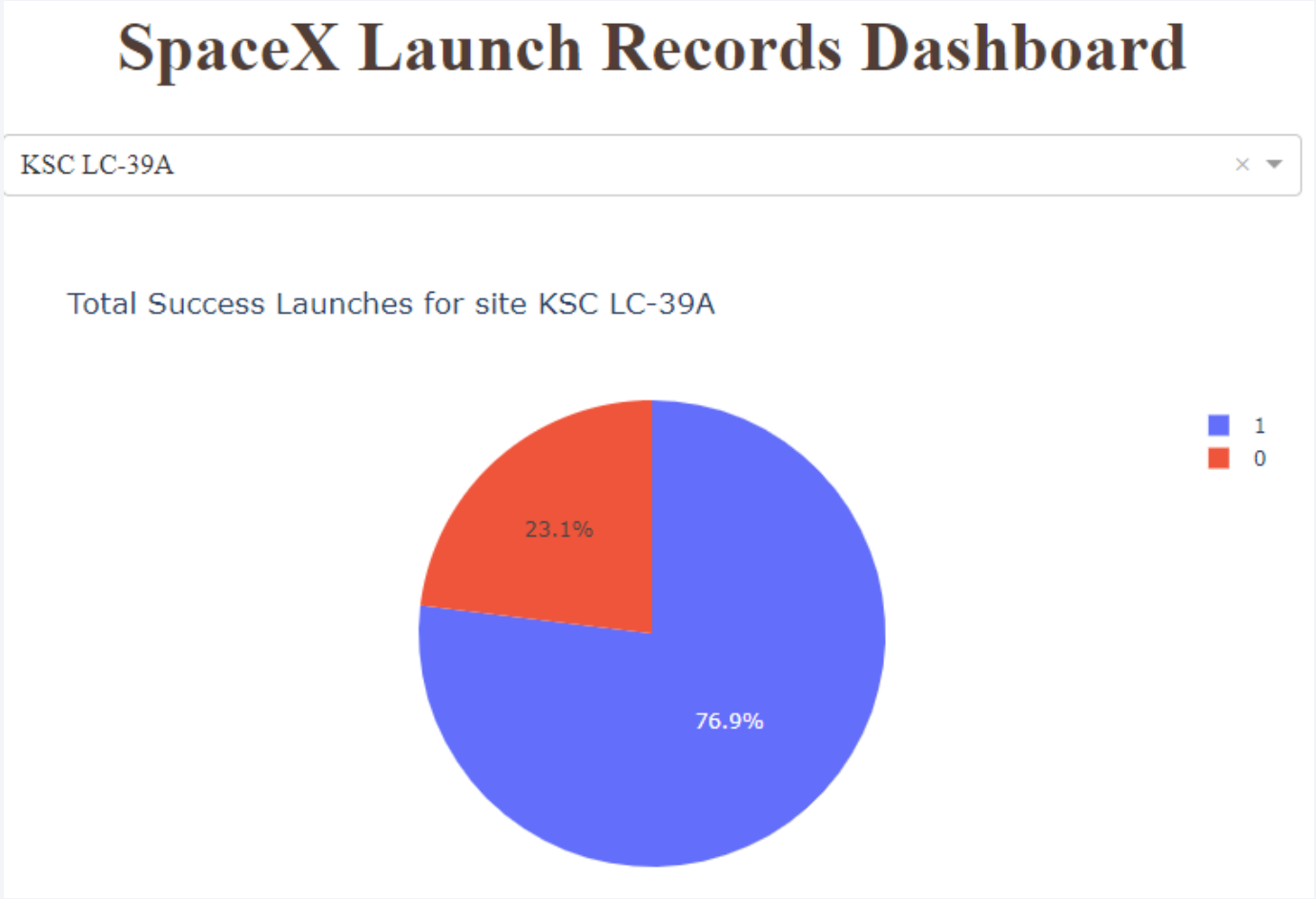
Total Success Launches by All Sites

- Most successes at 41.7% is at KSC LC-39A.
- Least successes at 12.5% is at CCAFS SLC-40

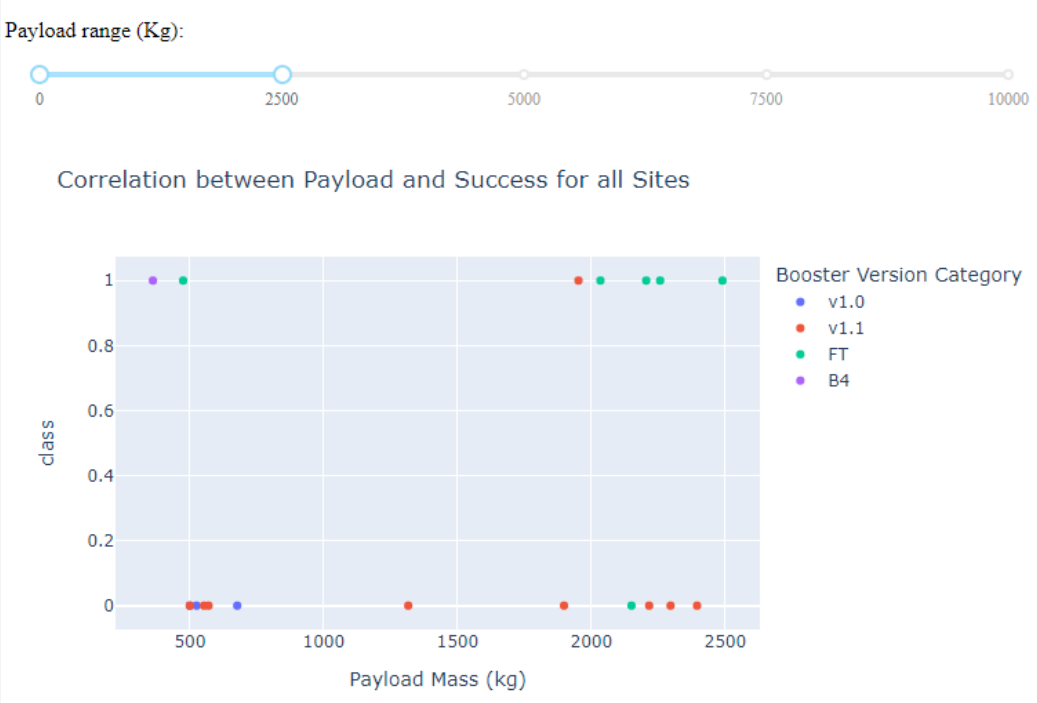


Total Success Launches for KSC LC-39A

- Highest success launches is at KSC LC-39A.
- 76.9% success.
- 23.1% failure.



Payload vs. Launch Outcome for All Sites



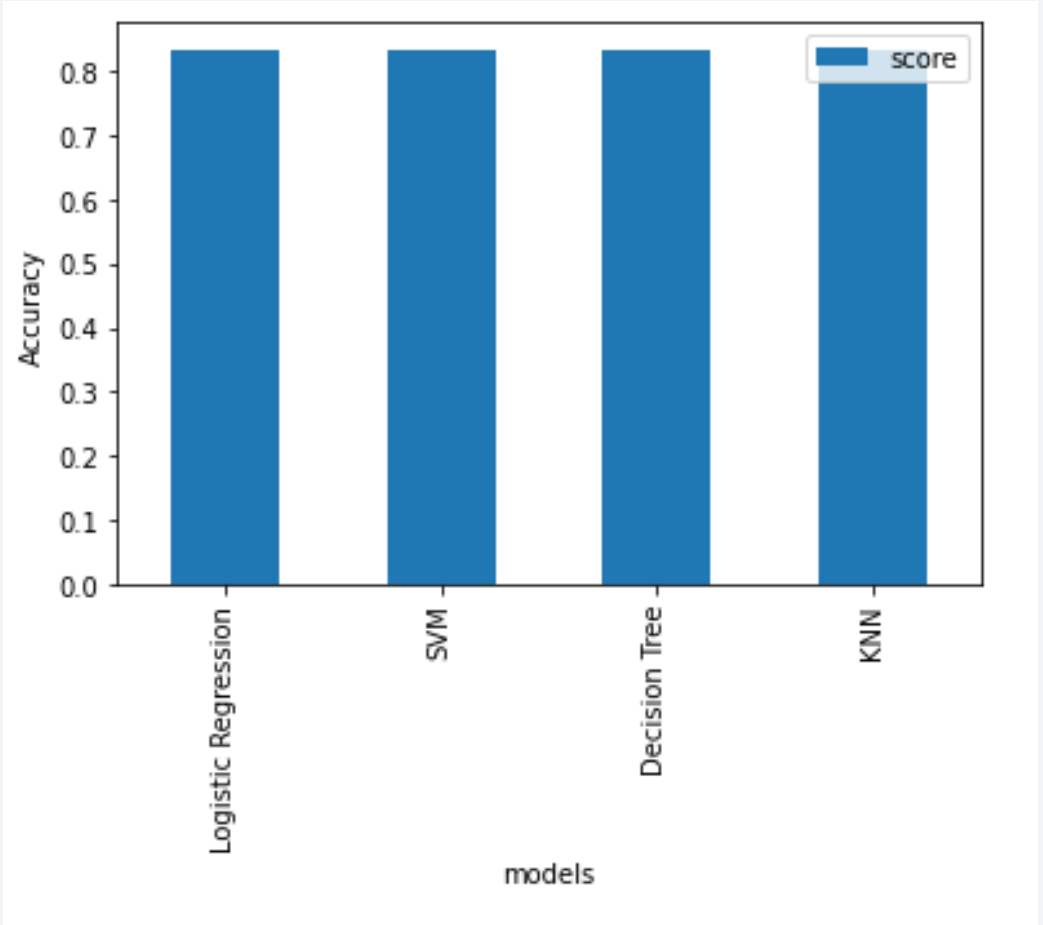
- Success is defined class = 1, Failure is defined class = 0.
- Higher success rate for low end of payload (less than 2500 kg) than high end of payload (greater than 7500 kg).

Section 5

Predictive Analysis (Classification)

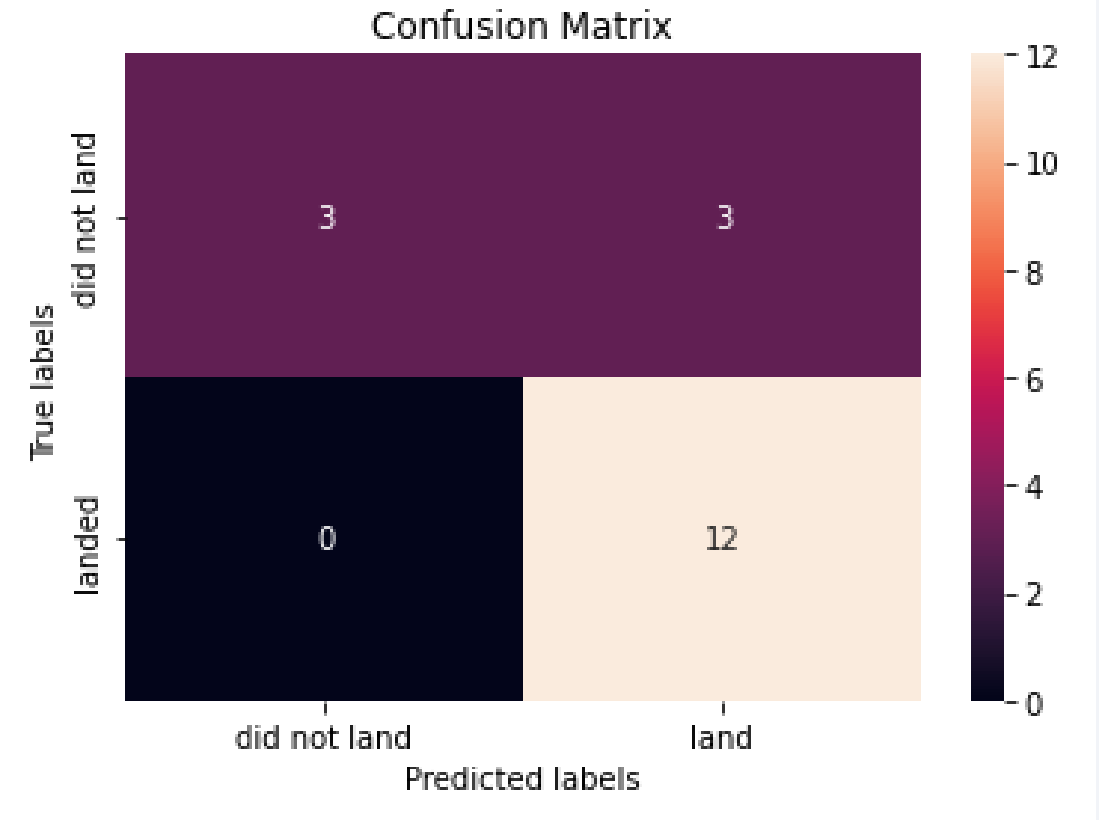
Classification Accuracy

- All 4 models showed similar accuracies at around 83%.
- There is overfitting of the data to each model due to the low number of test size (n=18).
- Even more data may lead to one model performing better than the rest.



Confusion Matrix

- The confusion matrices for all 4 models are identical.
- 12 True Positives
- 3 True Negatives
- 0 False Negatives
- 3 False Positives



Conclusions

- Increasing success rate over the years since 2013. Rate in 2020 is 0.84.
- Highest success rate with SSO, HEO, GEO, ES-LI.
- Launch sites are near the coast in California and Florida in case of rocket failures, near railway and highway for transport of equipment and products, and near a city for personnel to live nearby.
- Most successful launch site is KSC LC-39A, least is CCAFS SLC-40.
- All 4 models (KNN, Decision tree, SVM and Logistic Regression) performed similarly.
- All 4 tunned models had accuracy of 0.83.
- Models are overfitted with just 18 test data points. More data may result in truly identifying a best performing model.

Appendix

- Coursera IBM Data Science
  - https://www.coursera.org/learn/applied-data-science-capstone?specialization=ibm-data-science
- Github
  - https://github.com/kaysunphd/coursera/tree/main/IBM_Data_Science/Capstone

Thank you!