IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Janice Chan
31 Oct 2025

Outline

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

Executive Summary

- Space Y, new rocket launch provider, aims to compete with Space X.
- Space X - Falcon 9 rocket launches with a cost of $62M
  - Very inexpensive → reason: reuse 1st stage
- Using mission parameters (payload, orbit, etc.) to predict successful 1st stage landings
  - Accuracy rate of developed predictive models reached 83.3%
- Data-driven approach determines cost of launch
  - Enable more accurate cost assessments and competitive bidding strategies.

Introduction - Background

- This presentation is a part of the Applied Data Science Capstone Course.
- In this capstone, I take the role of a data scientist working for a new rocket company, Space Y.
- Leveraging data science insights and predictive models from this report, Space Y can make evidence-based bids for rocket launches, enhancing its competitiveness against SpaceX.

Introduction – Business Issues

- Space X benefits from low-cost rocket launches.
- Space X advertises Falcon 9 rocket launches on its website with a cost of $62M when the 1st stage can be reused.
- Space X will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.
- Therefore, this report aims to use first-stage landing predictions as a proxy to estimate launch costs more accurately.

Section 1

Methodology

Methodology

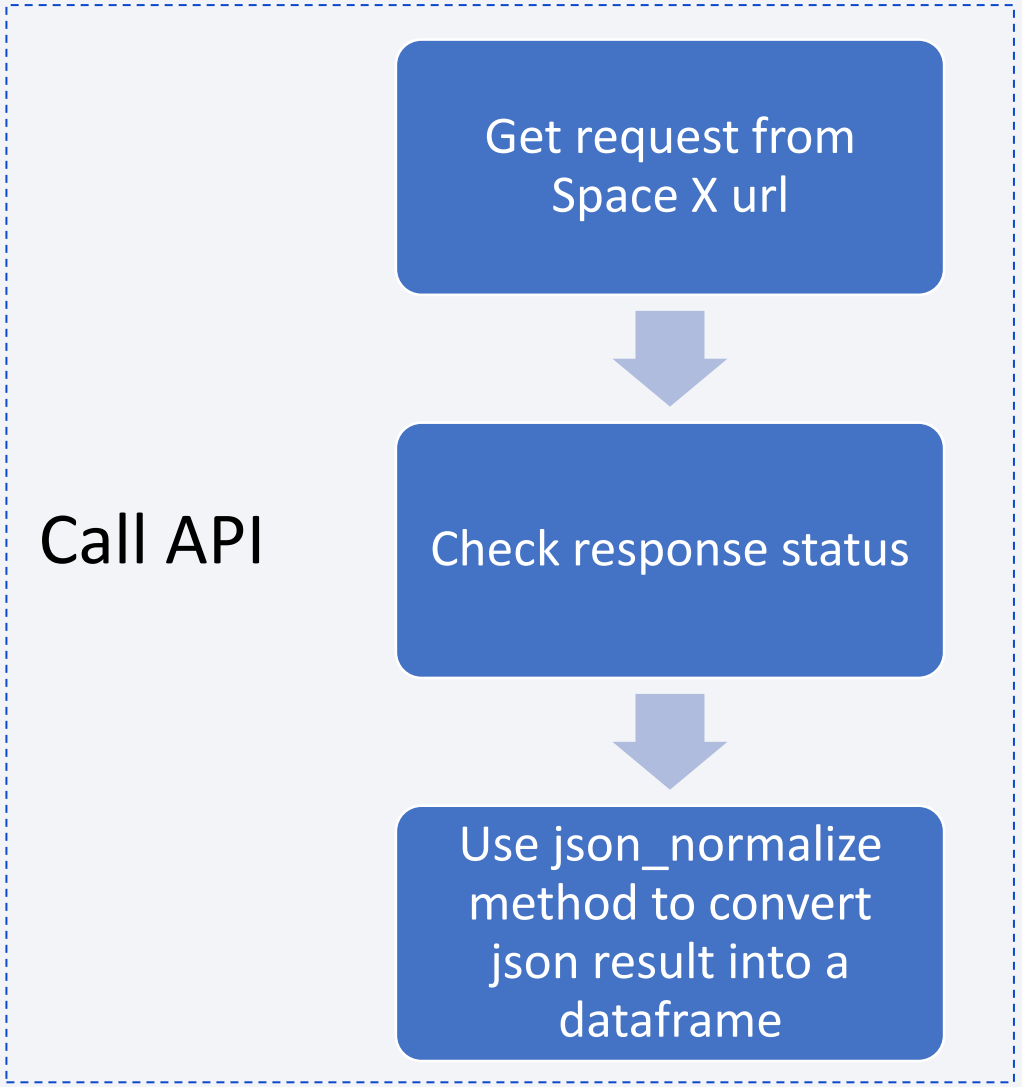
- Data Collection
- Data Wrangling
- Exploratory Data Analysis (EDA) using SQL and Visualization
- Interactive Visual Analytics using Folium and Plotly Dash
- Predictive Analysis using Classification models

Data Collection

- Two types of data collection
  - API
  - Web scraping

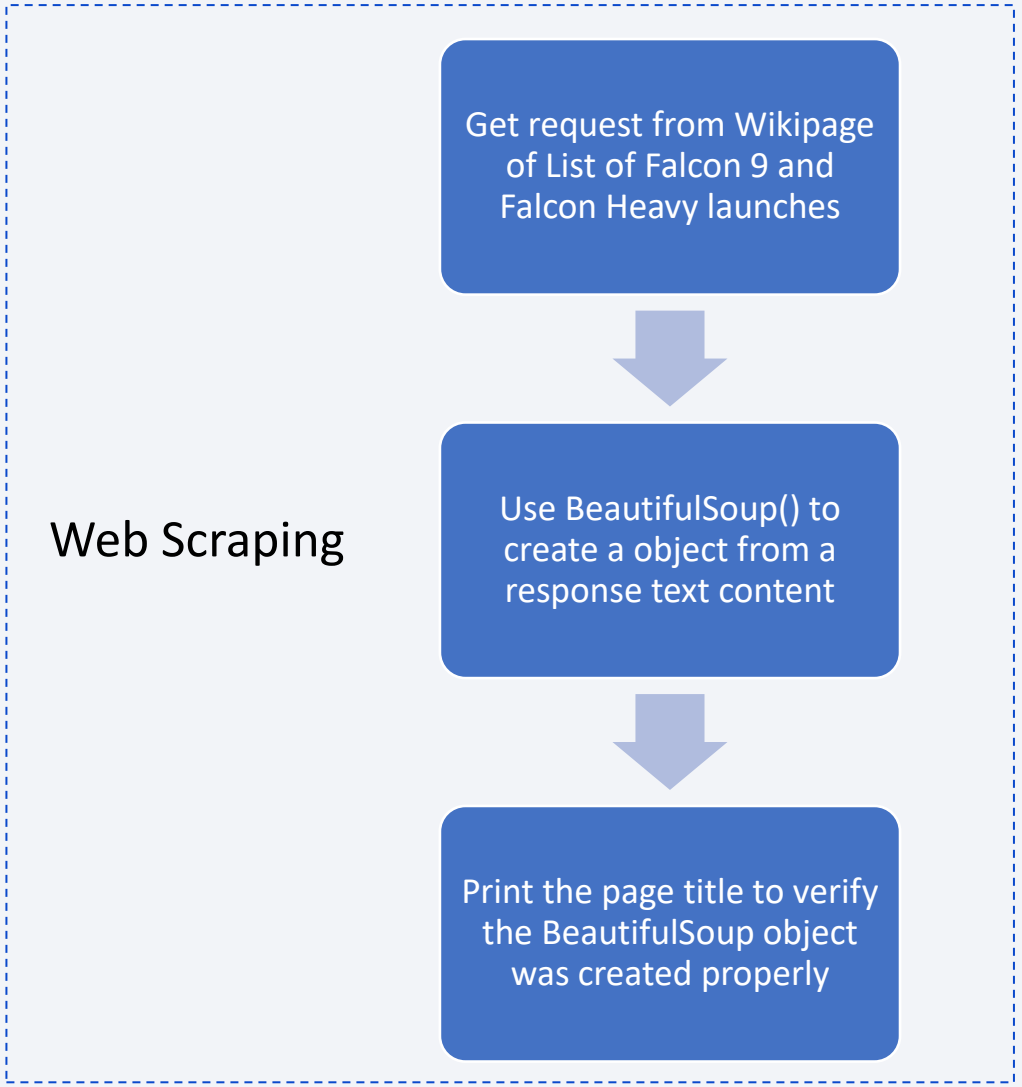
Data Collection – SpaceX API

- Retrieved SpaceX launch history using an open-source REST API
- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches
- Dealing with Missing Values



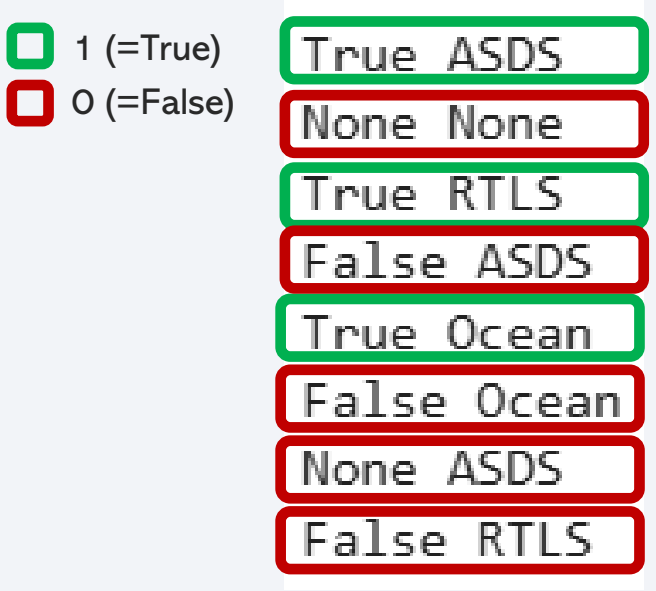
Data Collection – Web Scraping

- Retrieved Falcon 9 and Falcon Heavy Launches history from Wikipedia
- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
- Create a dataframe by parsing the launch HTML tables



Data Wrangling

- Conducted Exploratory Data Analysis (EDA) to identify data patterns
  - Apply value_counts() on columns
  - Number of launches on each site
  - Number and occurrence of each orbit
  - Number and occurence of mission outcome of the orbits with for loop
- Define labels for supervised learning models
  - Training label is: 'Class'
  - landing_class = 0 if bad outcome
    - False Ocean: failure to land a specific region of ocean
    - False RTLS: failure to to land a ground pad
    - False ASDS: failure to to land a drone ship
    - None ASDS and None None: failure to land
  - landing_class = 1 if good outcome
    - True Ocean: success to land a specific region of ocean
    - True RTLS: success to land a ground pad
    - True ASDS: success to land a drone ship



Exploratory Data Analysis (EDA)

- EDA with SQL
  - Load data into database called ‘SPACEXTBL’ table

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

  - Run SQL queries to retrieve and learn data about Launch site, Payload mass (kg), etc.
- EDA with Data Visualization
  - Read the SpaceX dataset into a Pandas dataframe
  - Plot graphs using Matplotlib and Seaborn visualization libraries
  - Perform Data Features Engineering
    - Apply OneHotEncoder to the categorical columns

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	\
0	1	6104.959412	1	False	False	False	1.0	
1	2	525.000000	1	False	False	False	1.0	
2	3	677.000000	1	False	False	False	1.0	
3	4	500.000000	1	False	False	False	1.0	
4	5	3170.000000	1	False	False	False	1.0	

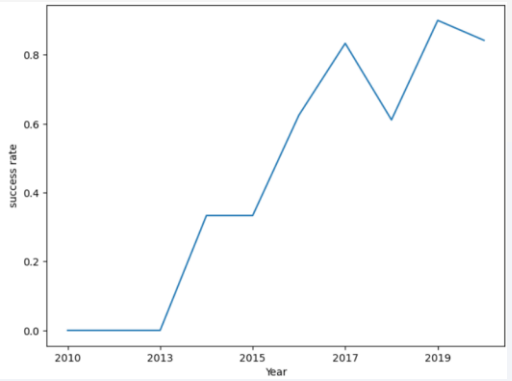
	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Serial_B1049	\
0	0	False	False	...	False	False	
1	0	False	False	...	False	False	
2	0	False	False	...	False	False	
3	0	False	False	...	False	False	

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
# Create the line chart
success_rate = df.groupby('Date')['Class'].mean()

# Create the line chart
plt.figure(figsize=(8, 6)) # Optional: set the figure size
success_rate.plot(x='Date', y=success_rate, kind='line')

# Add labels and a title (optional)
plt.xlabel("Year")
plt.ylabel("success rate")

# Display the chart
plt.show()
```



Data Visualization with Folium

- Analysis on Launch Sites Location
  - Create interactive maps using the Folium Python library
  - Marked all launch sites on a map
  - Marked the successful/ failed launches for each site on map
  - Calculated the distances between a launch site to its proximities
    - Railways
    - Highways
    - Coastlines
    - Cities

Data Visualization with Plotly Dash

- Launch Record Dashboard
  - Easier for stakeholders to explore interactive real-time data using Plotly Dash
  - Dropdown can choose all sites or specific sites
  - Pie chart show the success rate
    - Different launch site represented by different color
  - Scatter chart show the correlation between payload mass and landing outcome
    - Different booster version category represented by different color
    - Range slider for limiting the payload amount

Predictive Analysis (Classification)

Import libraries

Load dataset to dataframe

Create training label 'Class'

Standardize data using StandardScalar() and transform

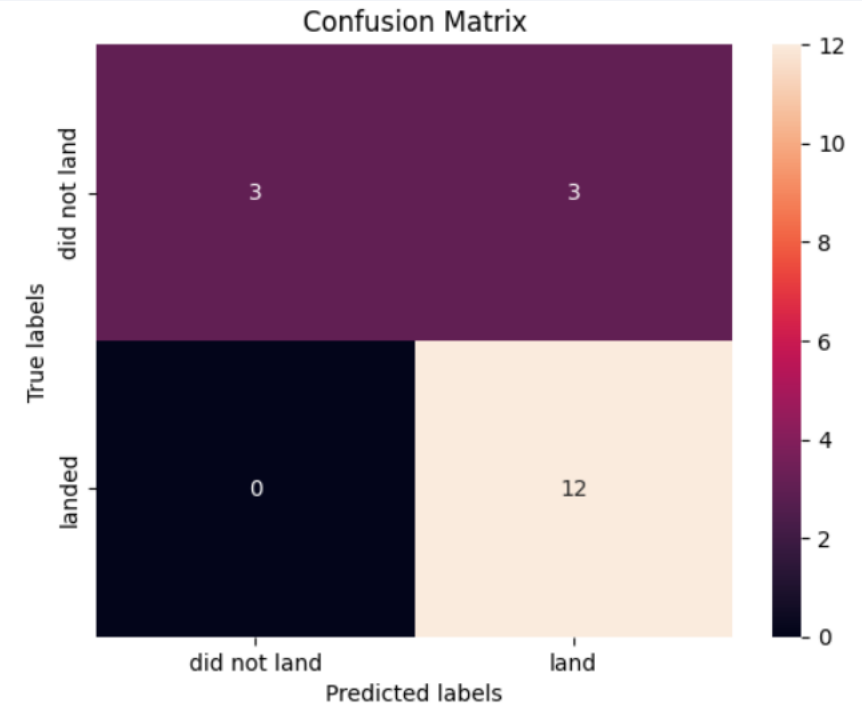
Use train_test_split function to split train dataset and test dataset

Fit training data to different training models

- Logistic Regression
- Support Vector Machine
- Decision Tree Classifier
- K-Nearest Neighbors Classifier

Use cross-validation and grid search to identify the best hyperparameter set for each model

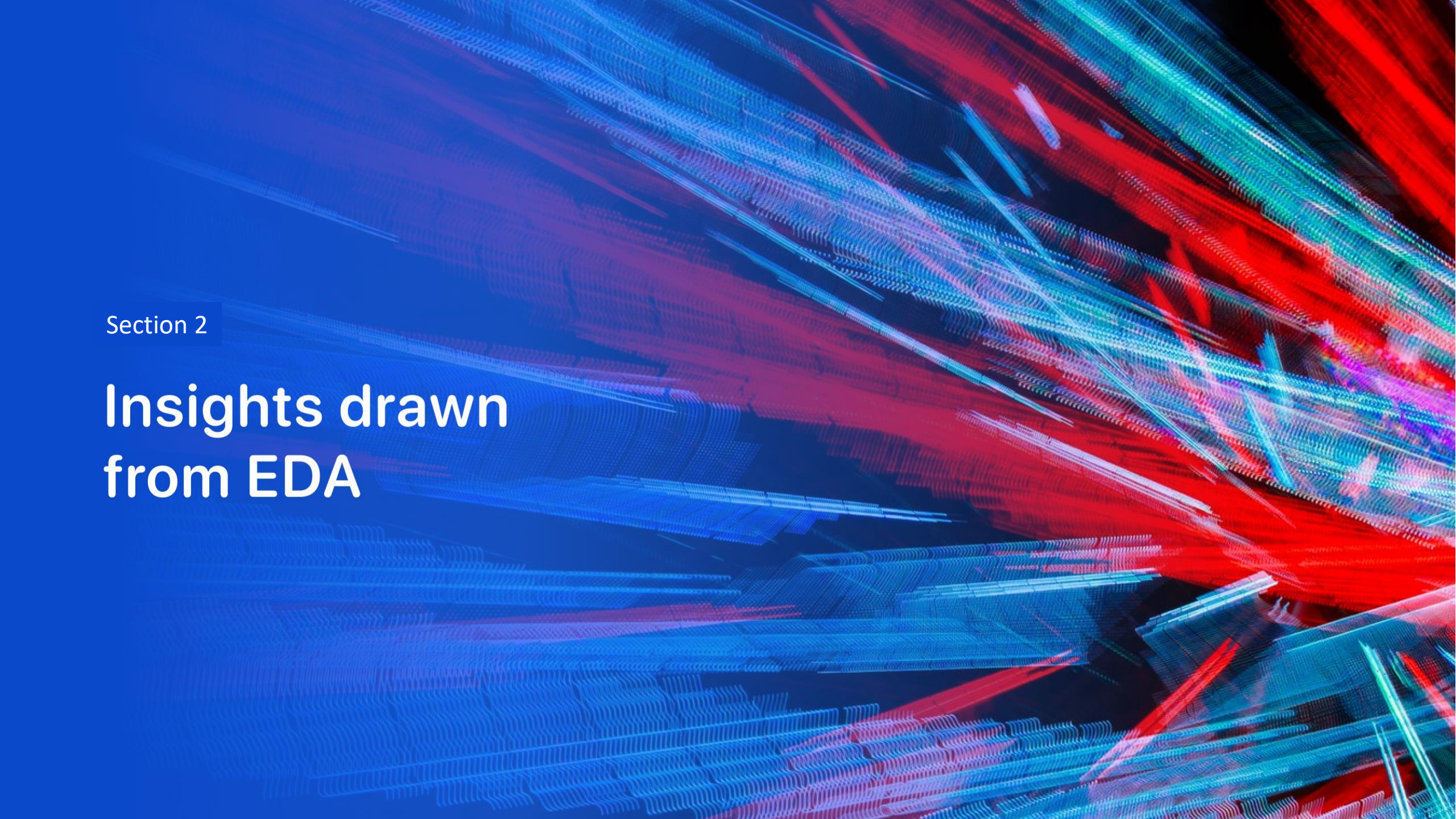
Evaluate accuracy score on validation set to select best model



Confusion matrix shows 15 true positive and 3 false positive in logistic regression model prediction

Results

- Results will show in the followings:
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results


Section 2
Insights drawn from EDA

Results: EDA with SQL

- Name of all launch sites
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
%sql select distinct Launch_Site from SPACEXTBL
```

- Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

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	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 carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

sum(PAYLOAD_MASS__KG_)
45596

- Average payload mass carried by booster version F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version='F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

- Dates of the first successful landing outcome on ground pad

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

min(Date)
2015-12-22

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql select distinct 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

Results: EDA with SQL - Continued

- Total number of successful and failure mission outcomes

```
%sql select distinct Mission_Outcome, count(1) from SPACEXTBL group by Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	count(1)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Names of the booster which have carried the maximum payload mass

```
%sql select distinct Booster_Version from (select distinct *, max(PAYLOAD_MASS__KG_) over(partition by 1) as max_payload from SPACEXTBL) a
        where PAYLOAD_MASS__KG_ = max_payload
```

Booster_Version
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

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql select distinct substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL
where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5) = '2015'
```

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

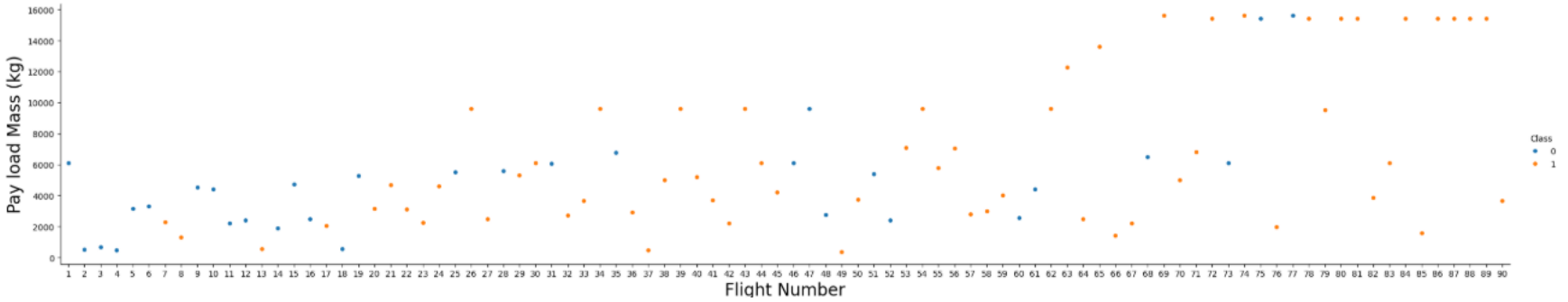
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

```
%sql select distinct Landing_Outcome, count(1) as outcome_rank from SPACEXTBL where Date between '2010-06-04' and '2017-03-20'
 group by Landing_Outcome order by count(1) desc
```

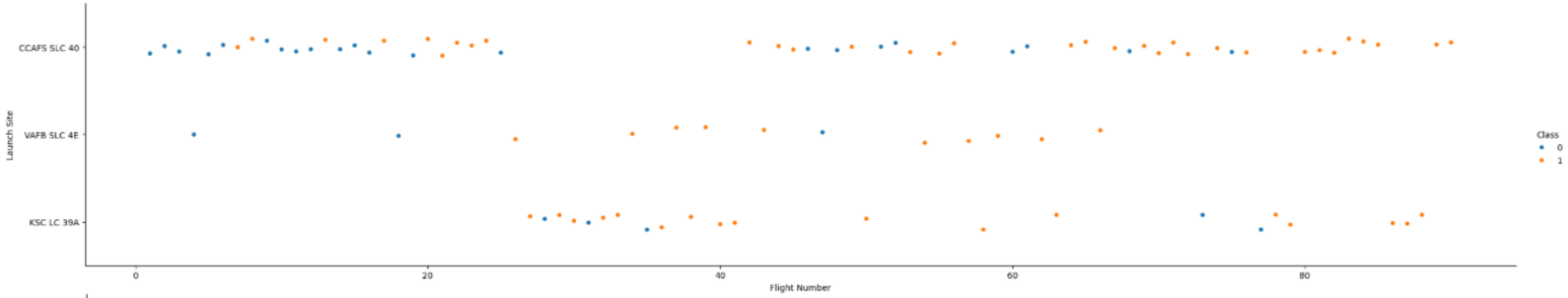
Landing_Outcome	outcome_rank
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Results: EDA with Visualization

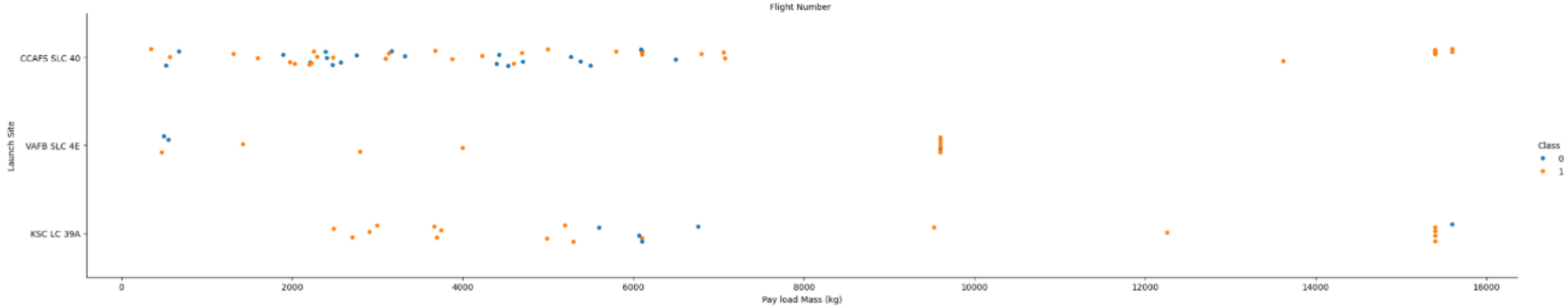
- Flight Number vs. Payload Mass (kg)
  - 1st stage successful land is positively correlated to flight number but tends to negatively related to payload mass. 1st stage failure land has no pattern.



- Flight Number vs. Launch Site
  - CCAFS SLC 40 has the most failure land in 1st stage, while others have more successful land in 1st stage.

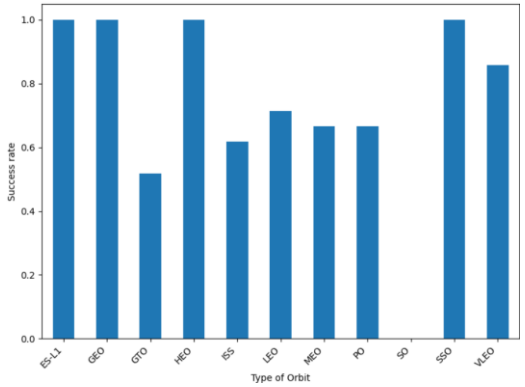


- Payload Mass (kg) vs. Launch Site
  - CCAFS SLC 40 has more 1st stage successful land at heavier payload mass, while VAFB SLC 4E and KSC LC 39A are more 1st stage successful land in different payload mass.

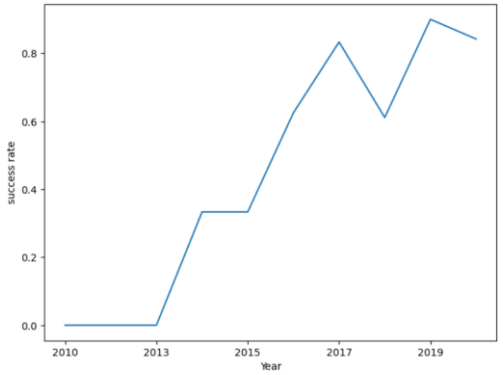


Results: EDA with Visualization

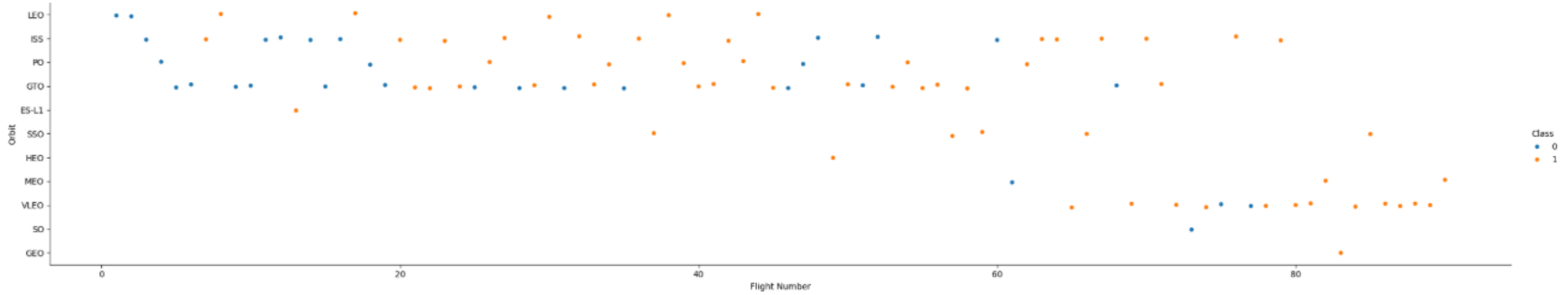
- Type of Orbit vs. Success rate
  - Only 'SO' did not have successful land in 1st stage, while all others type of orbits have successful land in 1st stage.



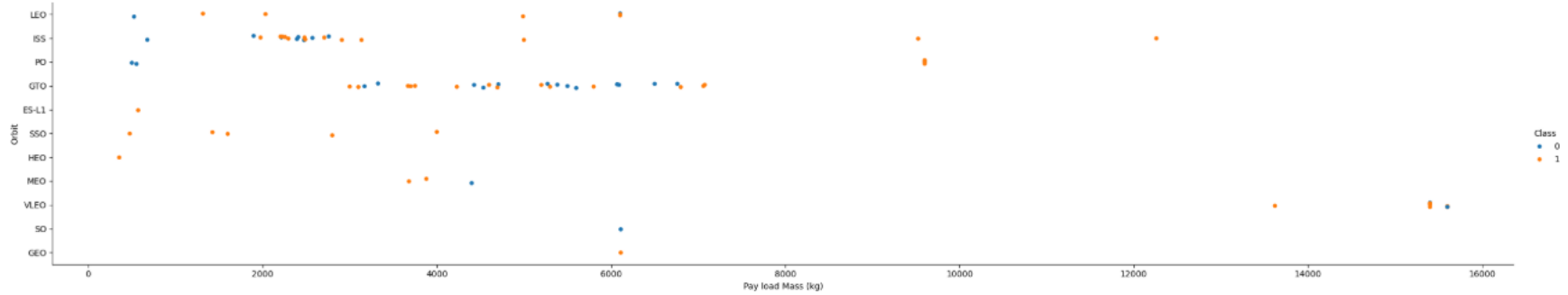
- Launch Success Yearly Trend
  - Launch year is positively correlated to the success rate since 2013.



- Flight Number vs. Type of Orbit
  - Flight number is positively correlated to all type of orbit, which have more successful land in 1st stage with larger flight number.



- Payload Mass (kg) vs. Type of Orbit
  - Lighter payload mass has positive influence on SSO, heavier payloads have a negative influence on GTO, but positive influence on ISS.



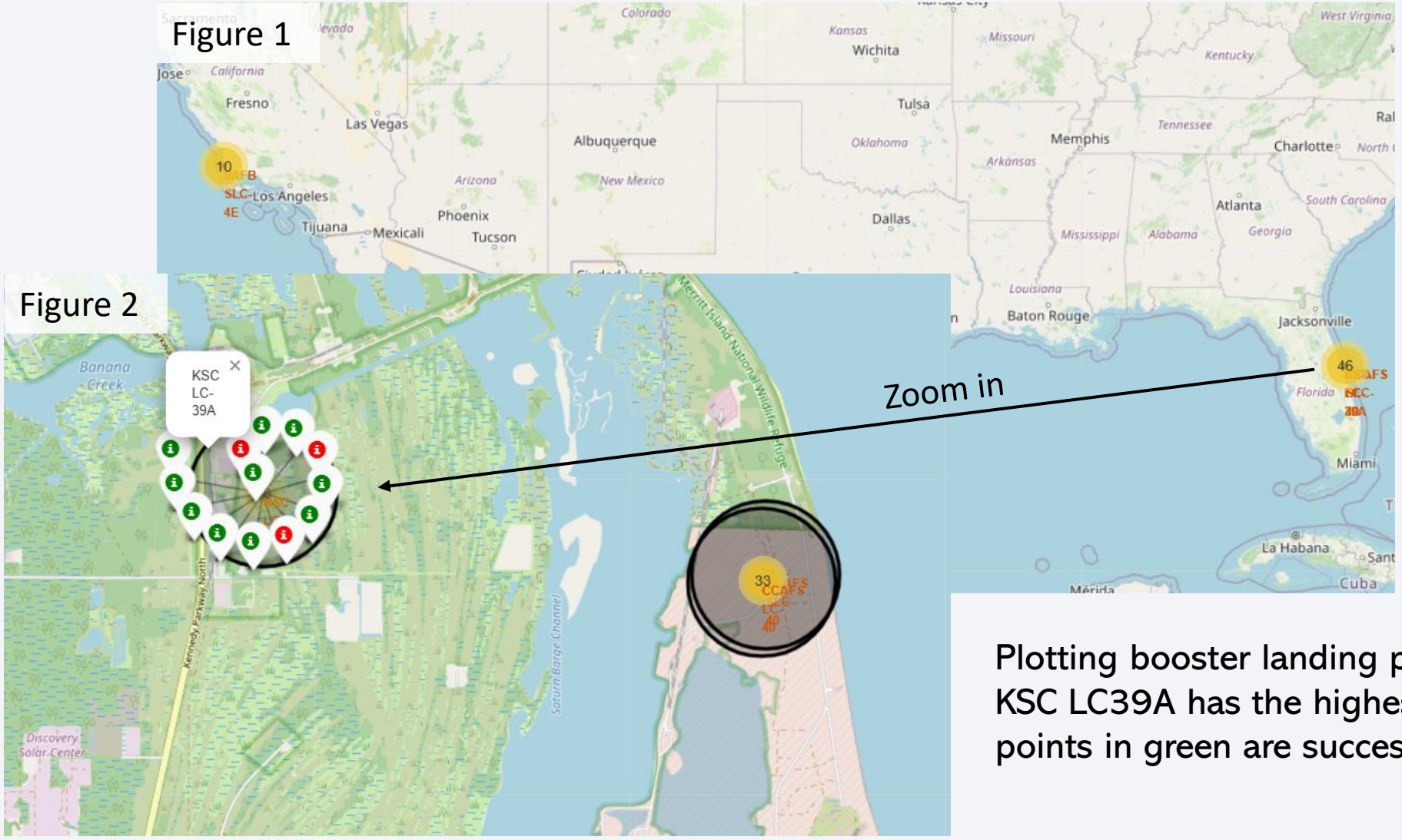
Section 3

Launch Sites Proximities Analysis

Results: All launch sites on a map

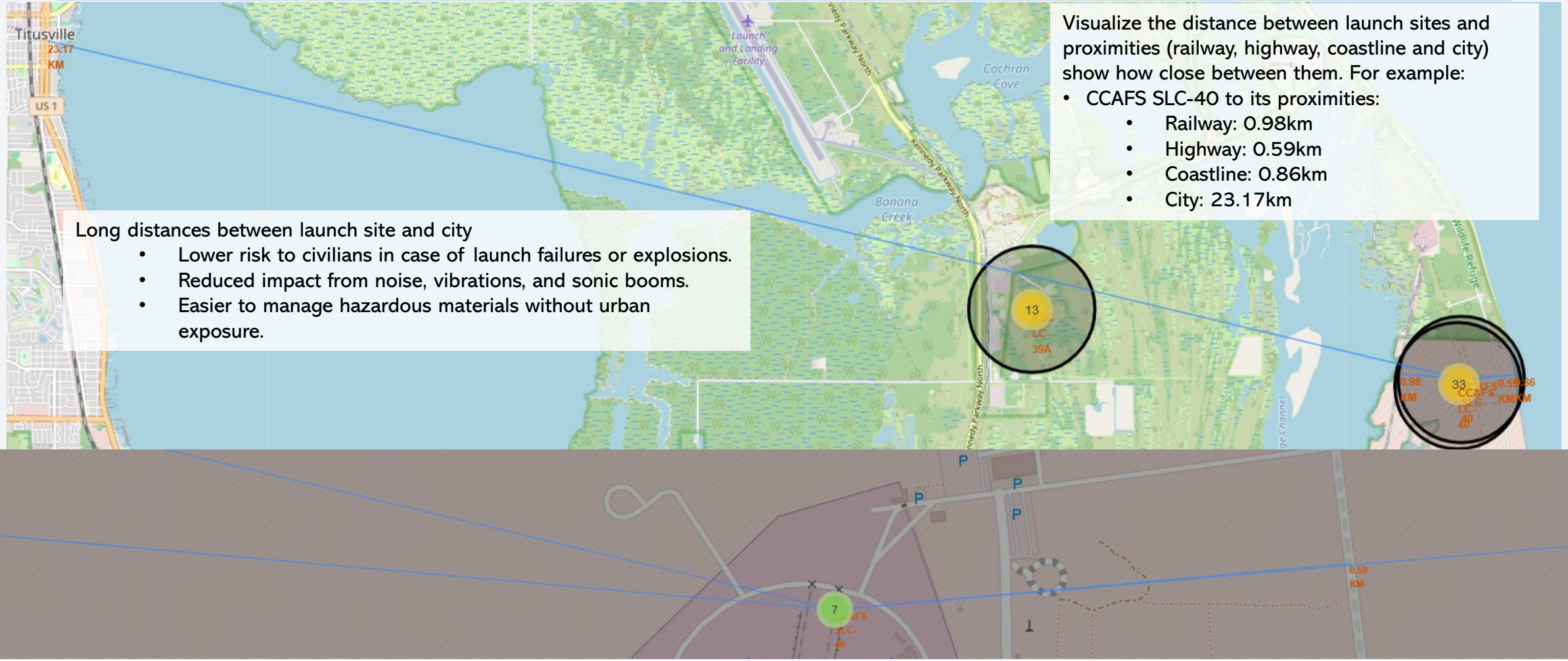
Plotting launch sites emphasizes their proximity to the coast and equator.

Results: Success/failed launches for each site on the map



Plotting booster landing performance by site shows KSC LC39A has the highest success rate, which points in green are success, red are fail.

Results: Distances between launch site to its proximities



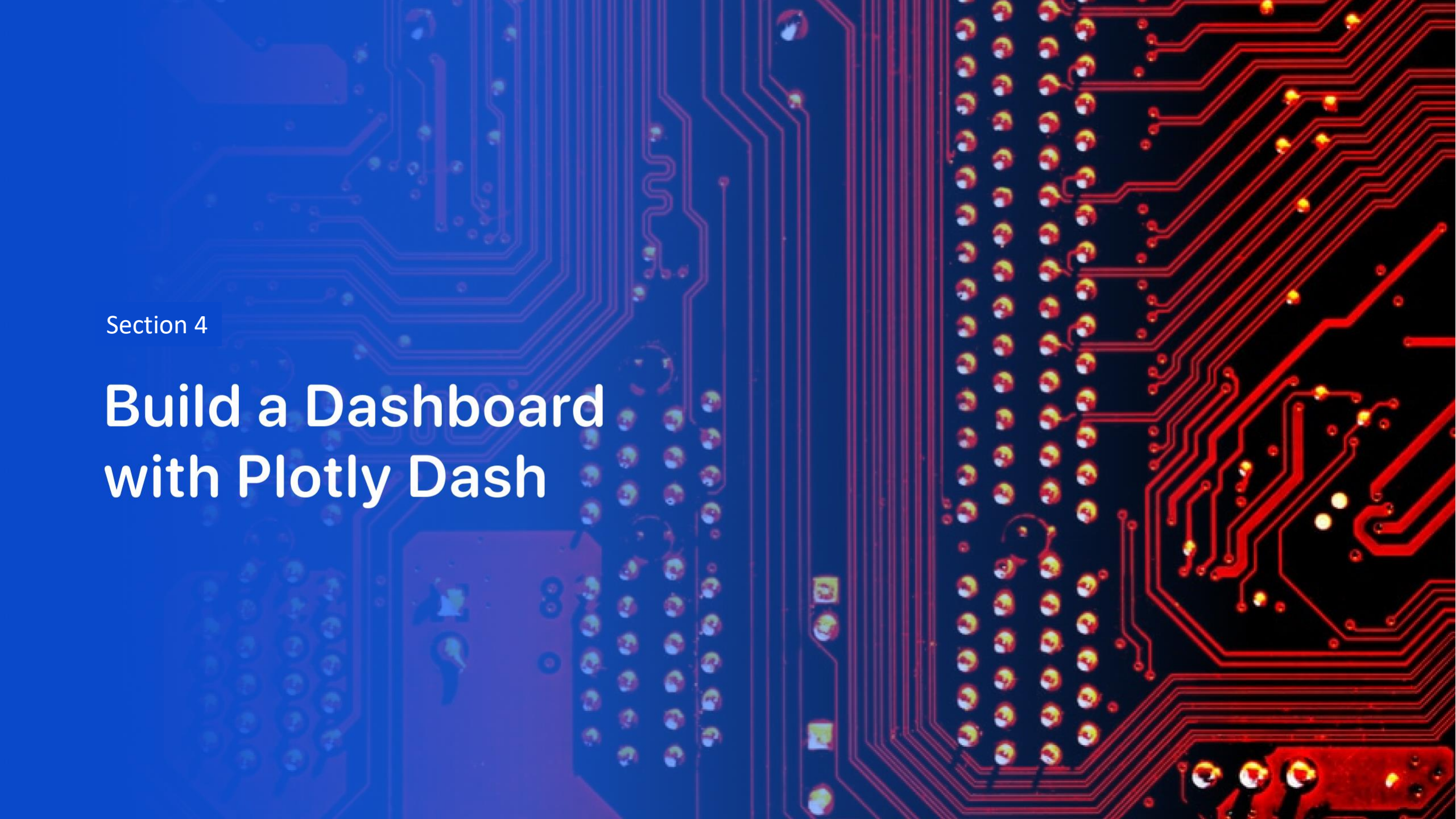
Visualize the distance between launch sites and proximities (railway, highway, coastline and city) show how close between them. For example:

- CCAFS SLC-40 to its proximities:
  - Railway: 0.98km
  - Highway: 0.59km
  - Coastline: 0.86km
  - City: 23.17km

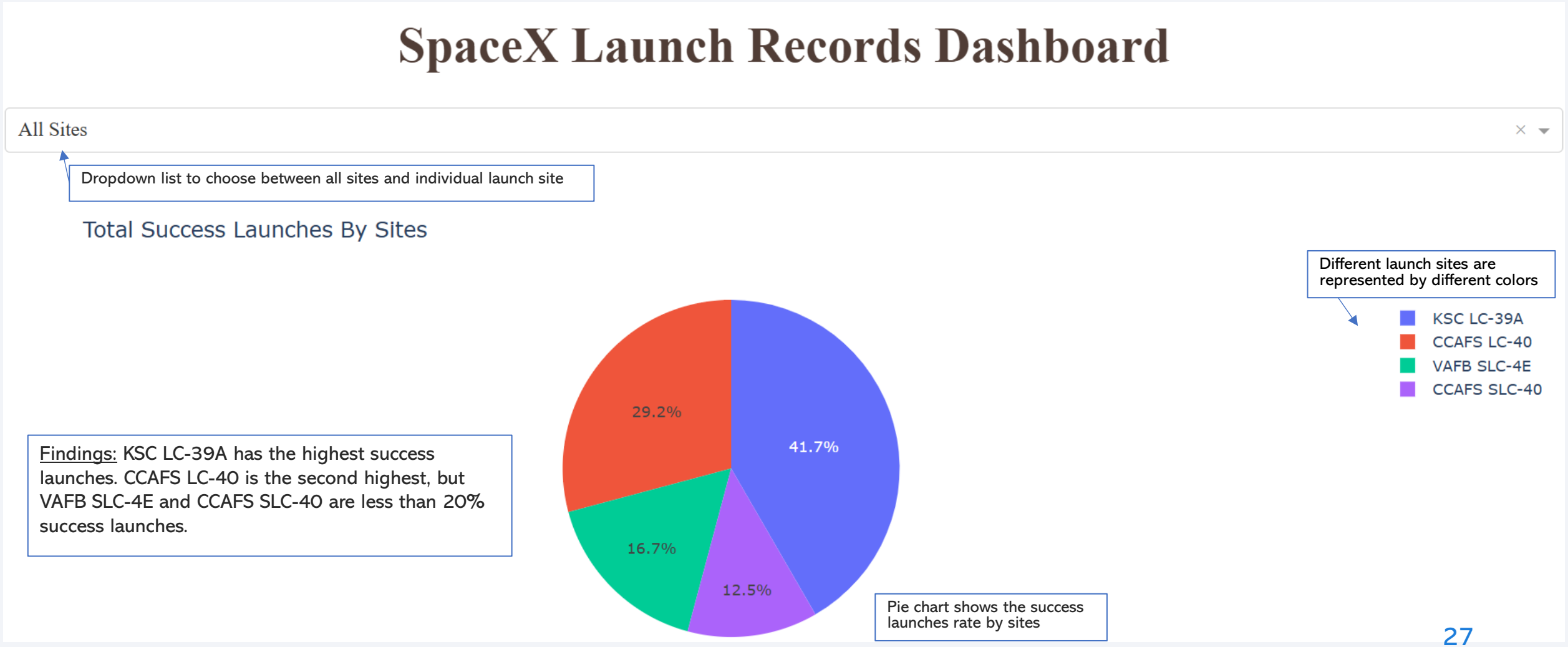
Long distances between launch site and city

- Lower risk to civilians in case of launch failures or explosions.
- Reduced impact from noise, vibrations, and sonic booms.
- Easier to manage hazardous materials without urban exposure.

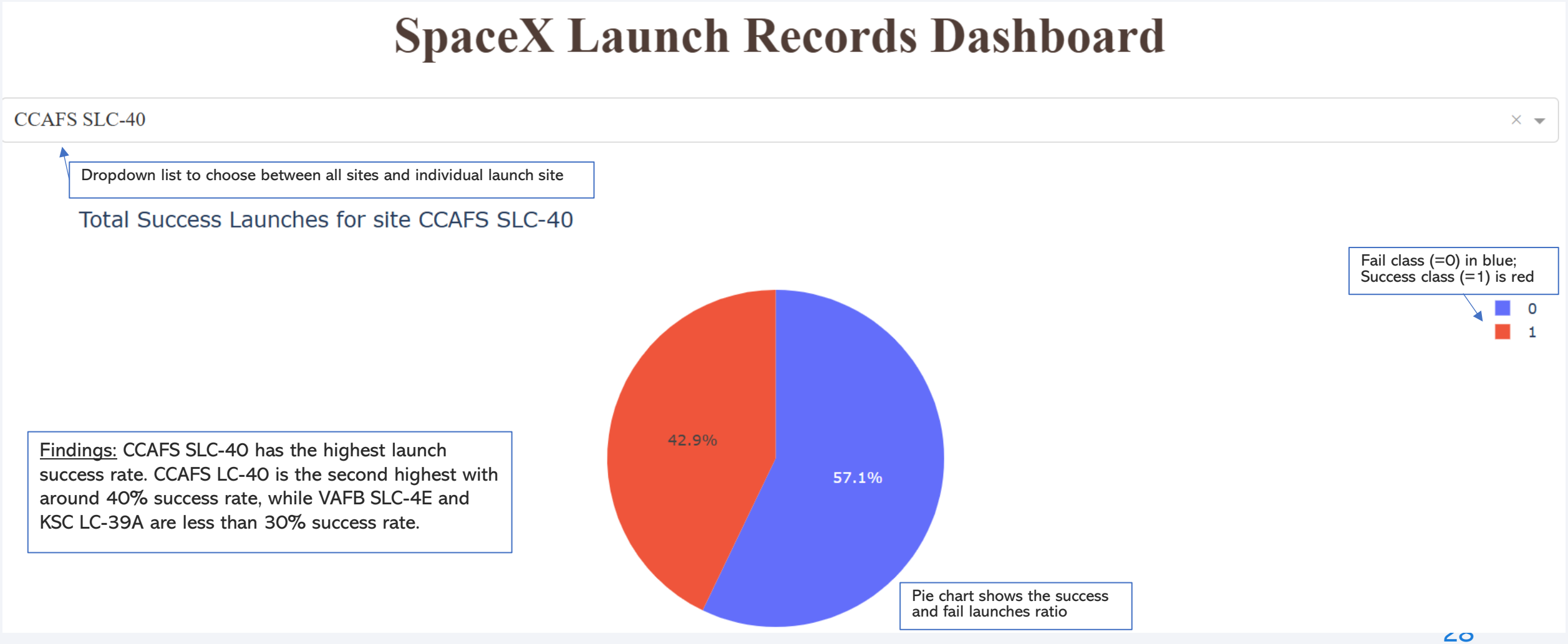
- Short distances between launch site and railway, highway, coastline
  - faster and cheaper transportation of heavy equipment, and fuel.
  - Easier movement of oversized components like boosters and payloads
- Launches over the ocean, minimizing risk to people and property.
- Easier recovery of reusable rocket stages and capsules from the sea.


Section 4
Build a Dashboard with Plotly Dash

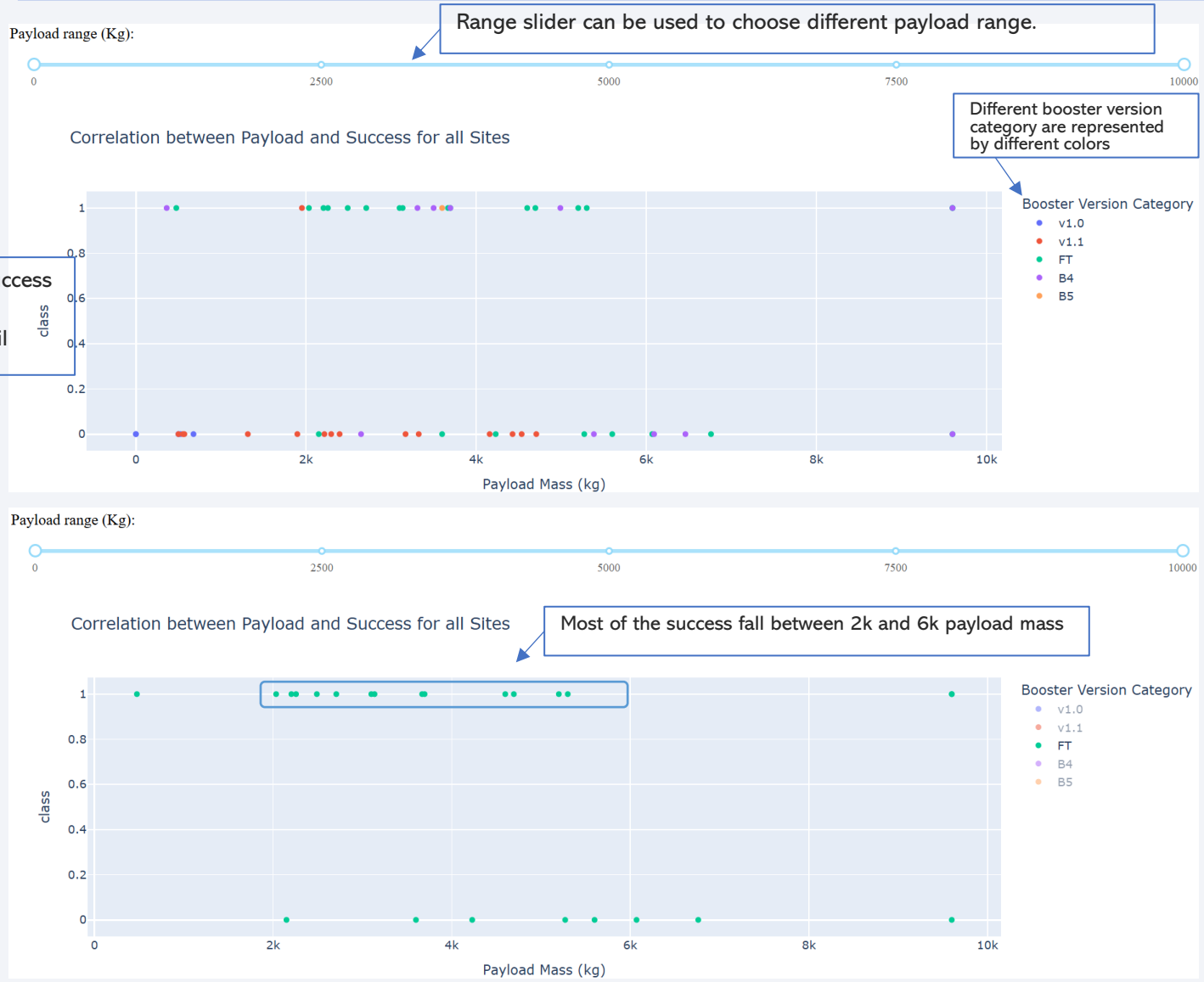
Results: Launch Records Dashboard – All Sites



Results: Launch Records Dashboard – Sites with highest launch success ratio



<Dashboard Screenshot 3>



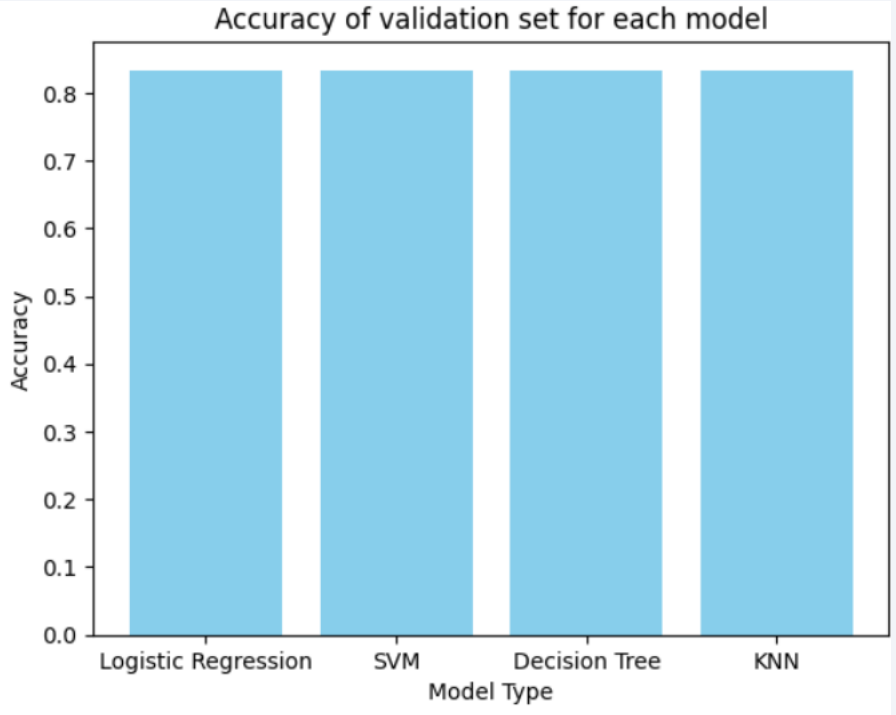
Findings:

- Most of the success are fall into the payload range between 2000kg and 6000kg.
- The highest success rate of booster version category is FT with less fail rate.
- Other booster version category are tended to more failure.

Section 5

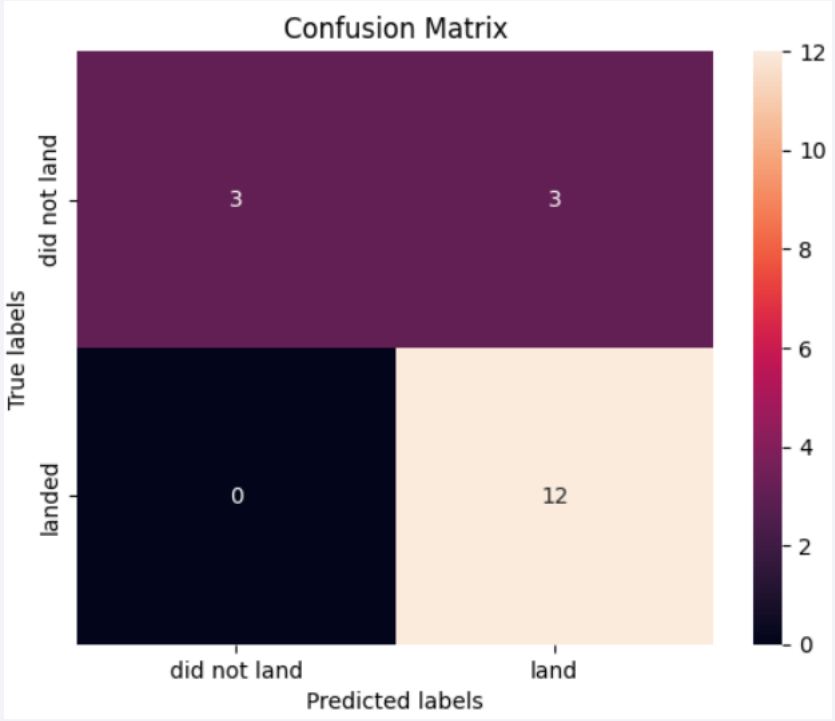
Predictive Analysis (Classification)

Classification Accuracy



All models have the same accuracy which is 83.33%

Confusion Matrix



All models have the same confusion matrix.

There are 15 true positive and 3 false positive out of 18 samples in validation set at the 1st stage booster.

Conclusions

- Step-by-step to determine the cost of launch
  - Data collection through Space X API and Web scraping about Falcon 9 and Falcon Heavy Launches history
  - Data wrangling to clean and transform raw data to usable format for analysis
  - Exploratory Data Analysis (EDA) using SQL and Visualization to find the patterns in data
  - Interactive Visual Analytics using Folium to drill down and analyze launch site locations and Plotly Dash to explore interactive real-time data to show the correlation between parameters.
  - Predictive Analysis using Classification models can predict Space X would land the 1st stage booster successfully with 83.3% accuracy.
- Improvement to have better competitiveness against Space X
  - Add more useful information as dataset
  - Use other predictive models to train
  - Fine-tune the models with feature engineering and re-fit with whole training dataset

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project
- Data collection
  - Space X API - https://github.com/r-spacex/SpaceX-API
  - Falcon 9 and Falcon Heavy launches - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Notebook outputs

Thank you!