Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

Project background and context

This capstone project focuses on the commercial space industry, specifically on SpaceX's cost-effective rocket launches. I assumed the role of a data scientist working for SpaceY, a new competitor founded by billionaire Allon Mask The primary objective of this project is to determine the price of each rocket launch and predict whether SpaceX will reuse the first stage of its rockets.

Problems to find answers to

Predicting First Stage Reuse

- What factors influence whether the first stage of the rocket will be reused?
- Can a machine learning model accurately predict the likelihood of first-stage reuse based on these factos?

Data Gathering and Cleaning

- How can data to answer these questions be found and what steps need to be taken to make it useable by ML algorithms?

Presentation

- How can Findings be presented in a concise and easy to understand way?

Section 1

Methodology

Methodology

Executive Summary

- 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

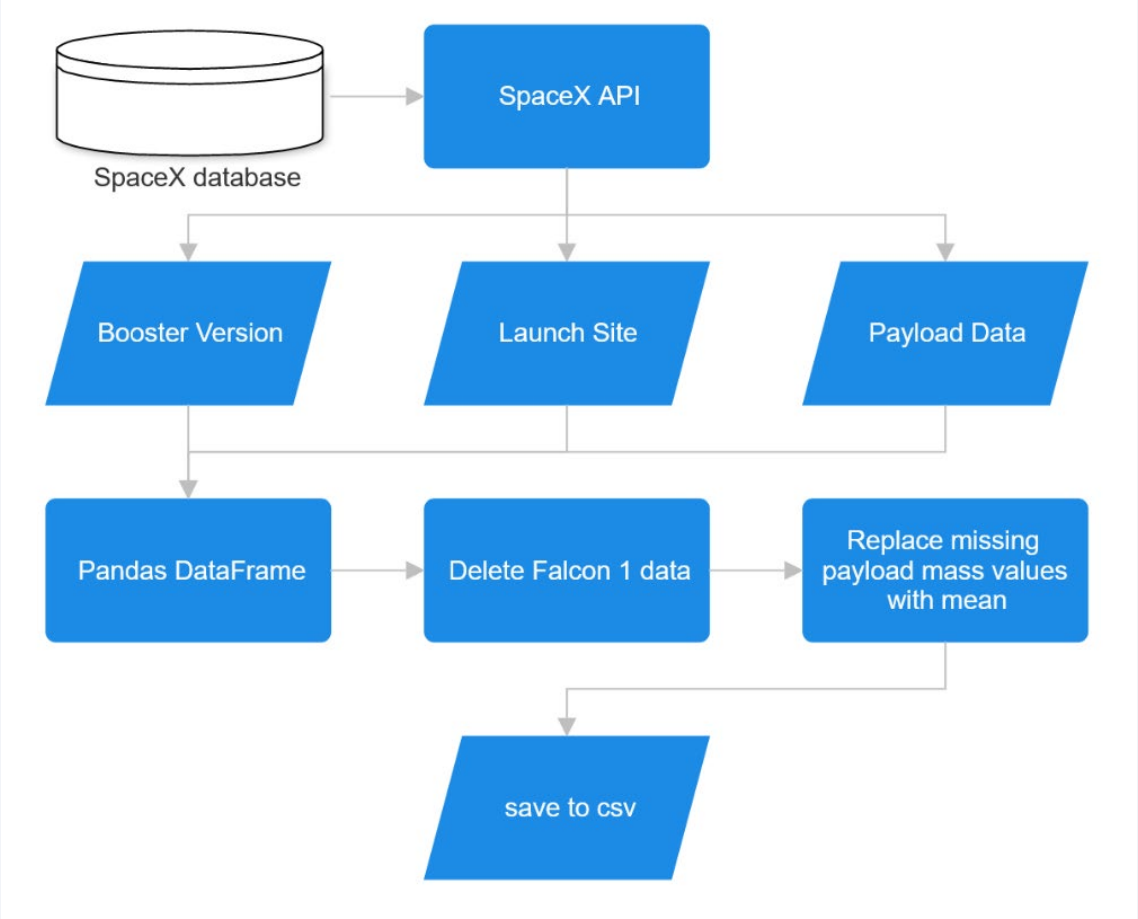
Data Collection

Data was collected in two ways

- Through the spacexdata.com api, cleaned, and saved to csv
  source: https://api.spacexdata.com/v4/rockets/
- Through webscraping with BeautifulSoup from Wikipedia
  source: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches

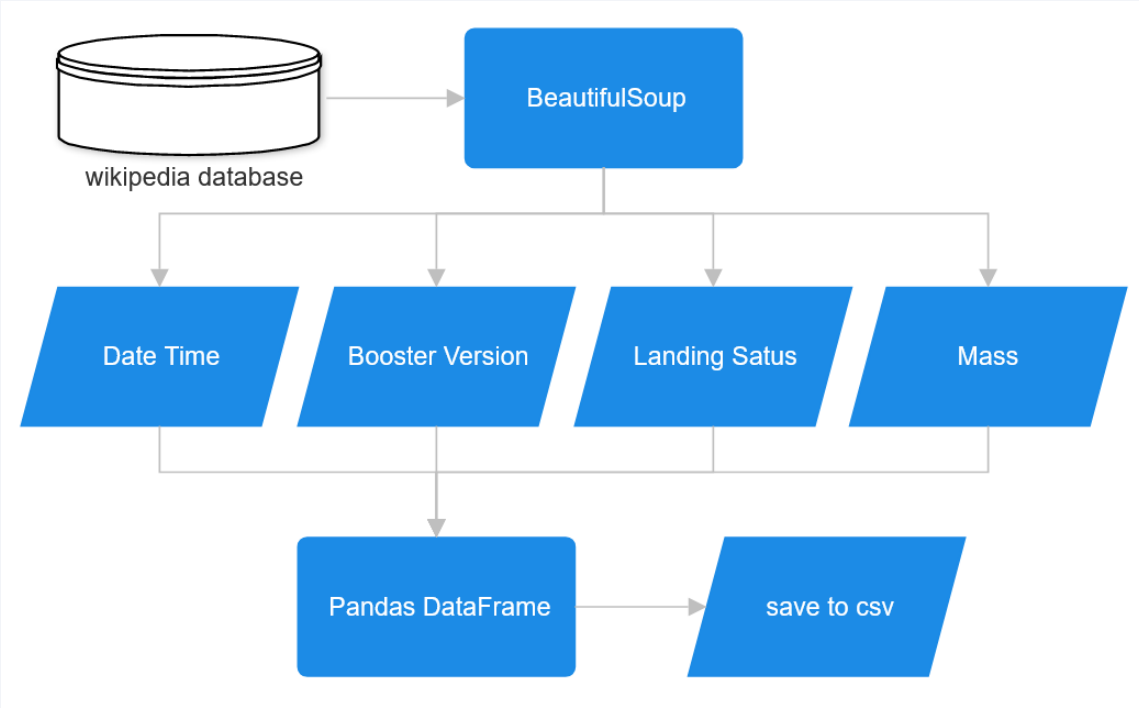
Data Collection – SpaceX API

- Data can be obtained via the public SpaceX API
- Collect and clean data like shown in flowchart
- GitHub URL of the completed SpaceX API calls notebook https://github.com/elmi3/ibm_capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Alternatively, data can be obtained from wikipedia.org
- Data is downloaded and processed via BeautifulSoup like shown in flowchart
- GitHub URL of the completed webscraping notebook https://github.com/elmi3/ibm_capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

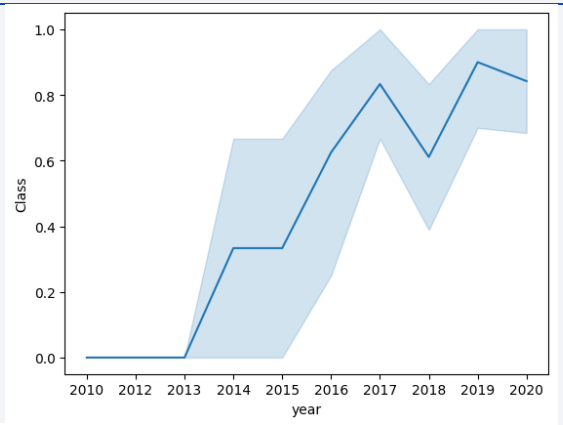
- Some initial EDA was performed, consisting of:
  - Calculating the number of launches on each site
  - Calculating the number and occurrence of each orbit
  - Calculating the number and occurrence of mission outcome of the orbits
- Then a landing outcome label was created and added to the DataFrame this is necessary for ML training in the following chapters

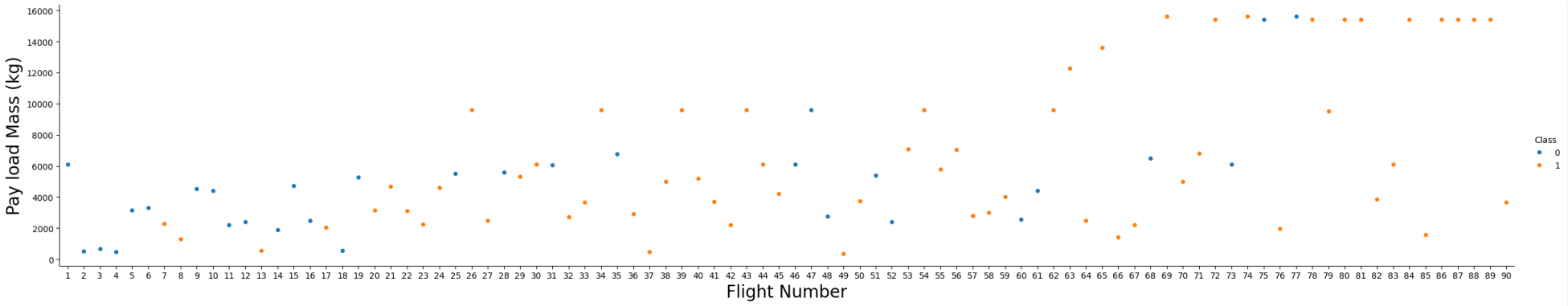
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

- GitHub URL of the completed data wrangling notebook https://github.com/elmi3/ibm_capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Graphical EDA consisted of several visualizations, like:
  - Increase of successful launches since 2010 (right)
  - Increase in Payload Mass over the program's duration (bottom)





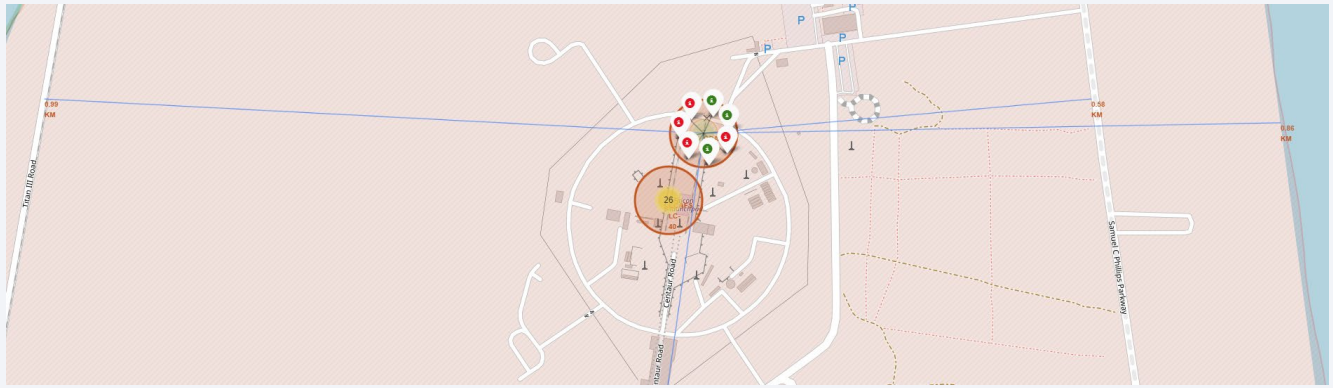
- GitHub URL of the completed data visualization notebook https://github.com/elmi3/ibm_capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- List of performed SQL queries:
  - names of the unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'CCA'
  - total payload mass carried by boosters launched by NASA (CRS)
  - average payload mass carried by booster version F9 v1.1
  - date when the first successful landing outcome in ground pad was achieved
  - names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - total number of successful and failure mission outcomes
  - names of the booster_versions which have carried the maximum payload mass
  - month names, failure landing_outcomes in drone ship, booster versions, launch_site for 2015
  - landing outcomes between 2010-06-04 and 2017-03-20 in descending order
- GitHub URL of the completed EDA with SQL notebook https://github.com/elmi3/ibm_capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

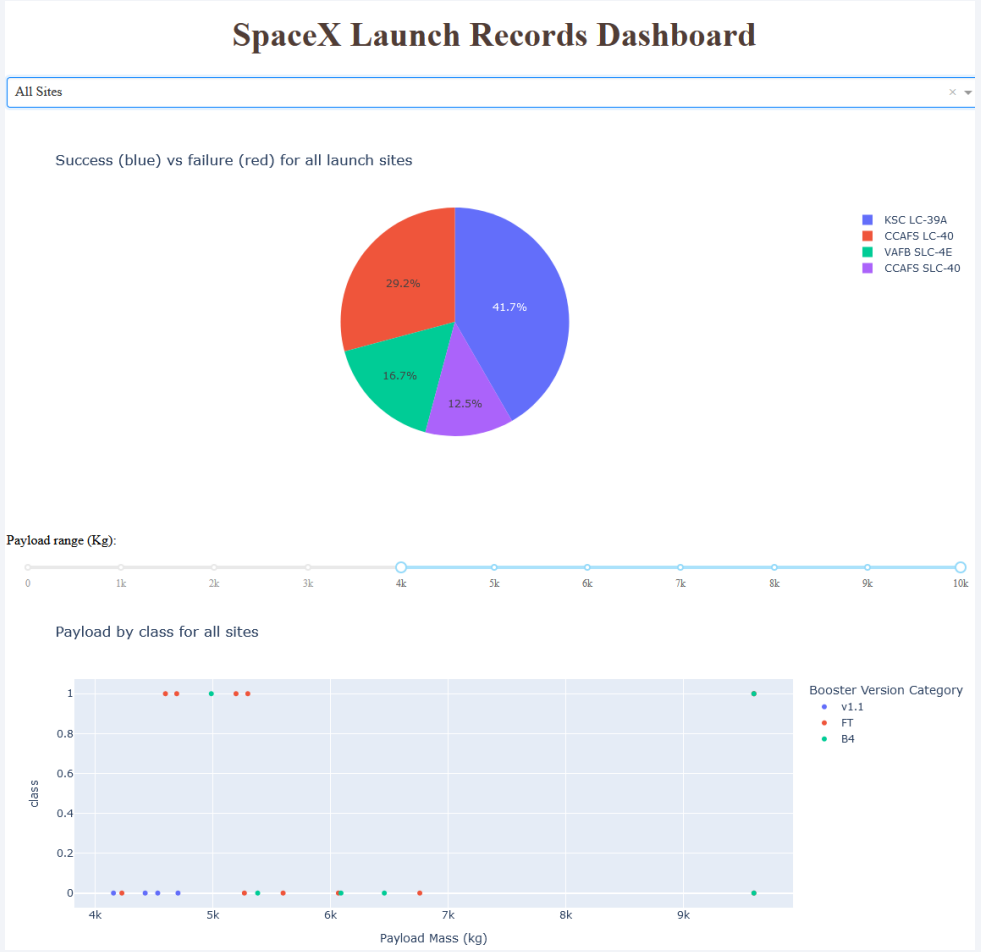
- Folium maps was used, including markers, circles, lines, and marker clusters to show
  - The area of launch sites (circles)
  - Launch sites (markers)
  - Launches (marker clusters)
  - Distances between points (lines)



- GitHub URL of the completed interactive map notebook https://github.com/elmi3/ibm_capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was built with Plotly dash to show
  - Pie charts showing the total launches by specific sites
  - Scatter plots showing the relationship between payload mass and outcome for specific booster versions
- GitHub URL of the completed dashboard code https://github.com/elmi3/ibm_capstone/blob/main/spacex_dash_app.py



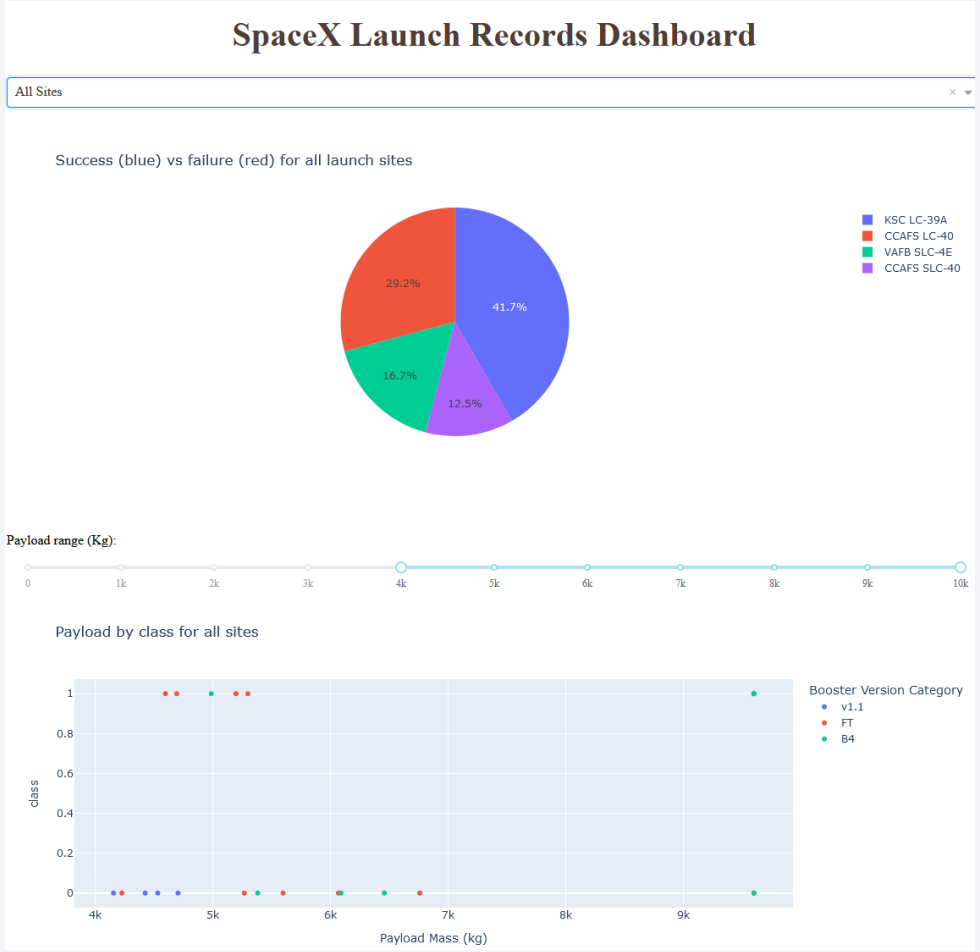
Predictive Analysis (Classification)

Using NumPy, Pandas, and scikit-learn, I

- Preprocessed the data to a useable form
  (including classification and splitting into training and test sets)
- Used the data to train several different ML models, including
  - Logistic Regression, Support Vector Machine, Decision Tree Classifier, Nearest Neighbor
- Evaluated the models using
  - Confusion Matrices, scikit's score() function
- GitHub URL of the completed interactive map notebook
  https://github.com/elmi3/ibm_capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Result

- Exploratory data analysis results include
  - Number of launch sites: 4
  - Launch sites significant distance away from cities but close to logistic infrastructure and coasts
  - Improvement in mission success and payload mass over time
  - First mission: 2010
  - First successful stage 1 landing: 2015
  - Most frequent orbits: ISS, GTO & VLEO
  - Average F9 v1.1 payload mass: 2928kg
- Interactive analytics demo ->
- Predictive analysis results
  - Models built and evaluated

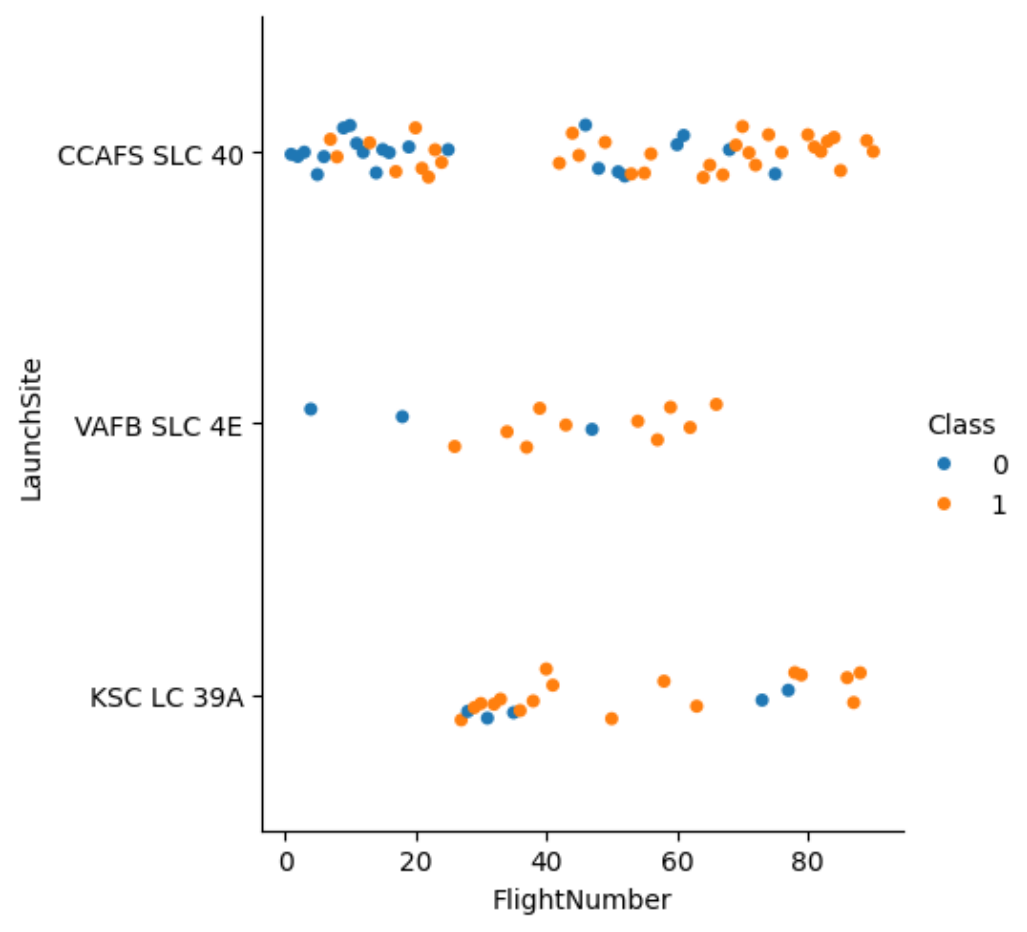


Section 2

Insights drawn from EDA

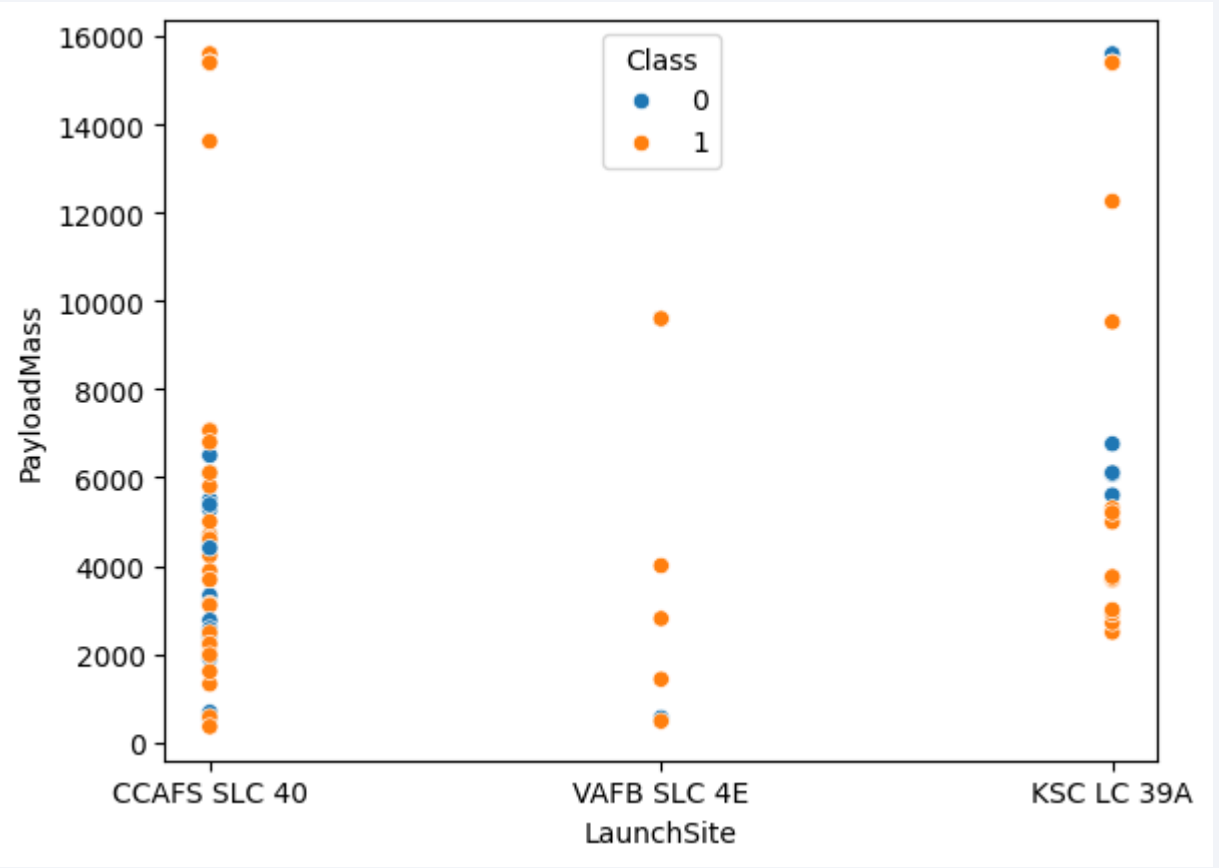
Flight Number vs. Launch Site

- Findings:
  - Most launches from SLC 40
  - Successful missions (Class: 1) increase with increase in flight number independent of launch site
  - Launch site SLC 4E is no longer in use



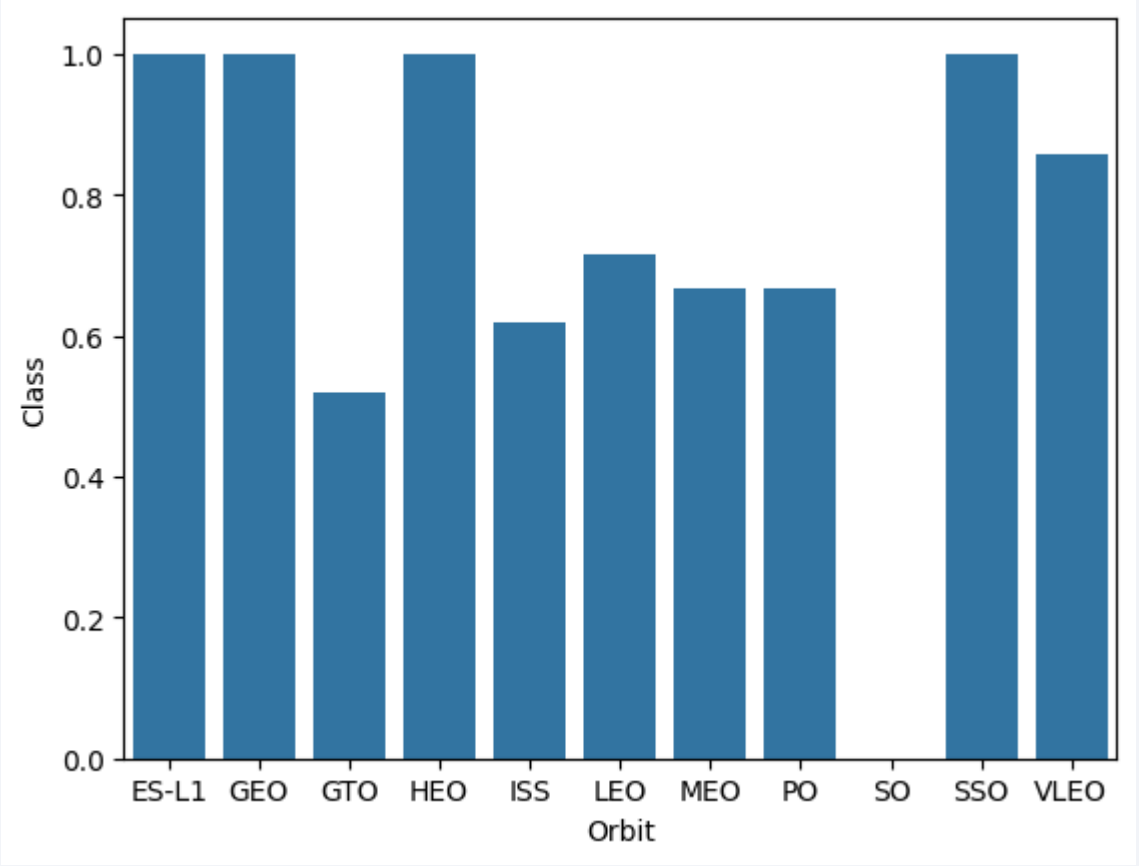
Payload vs. Launch Site

- Findings:
  - Success increased with increased payload mass
  - Site SLC 4E is not used for payloads over 10t



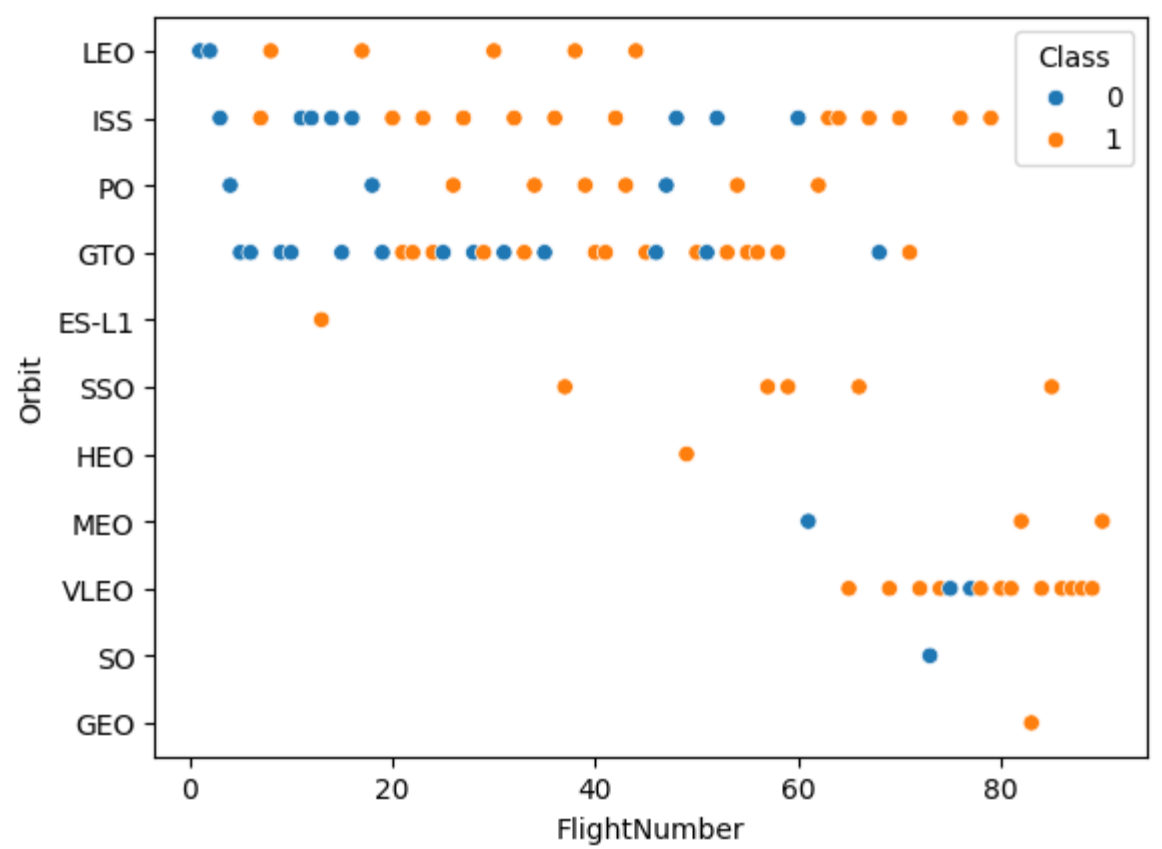
Success Rate vs. Orbit Type

- Findings:
  - ES-L1, GEO, HEO, and SSO have a 100% success rate
  - SO is the only orbit with 100% failure rate (1 of 1 launches)



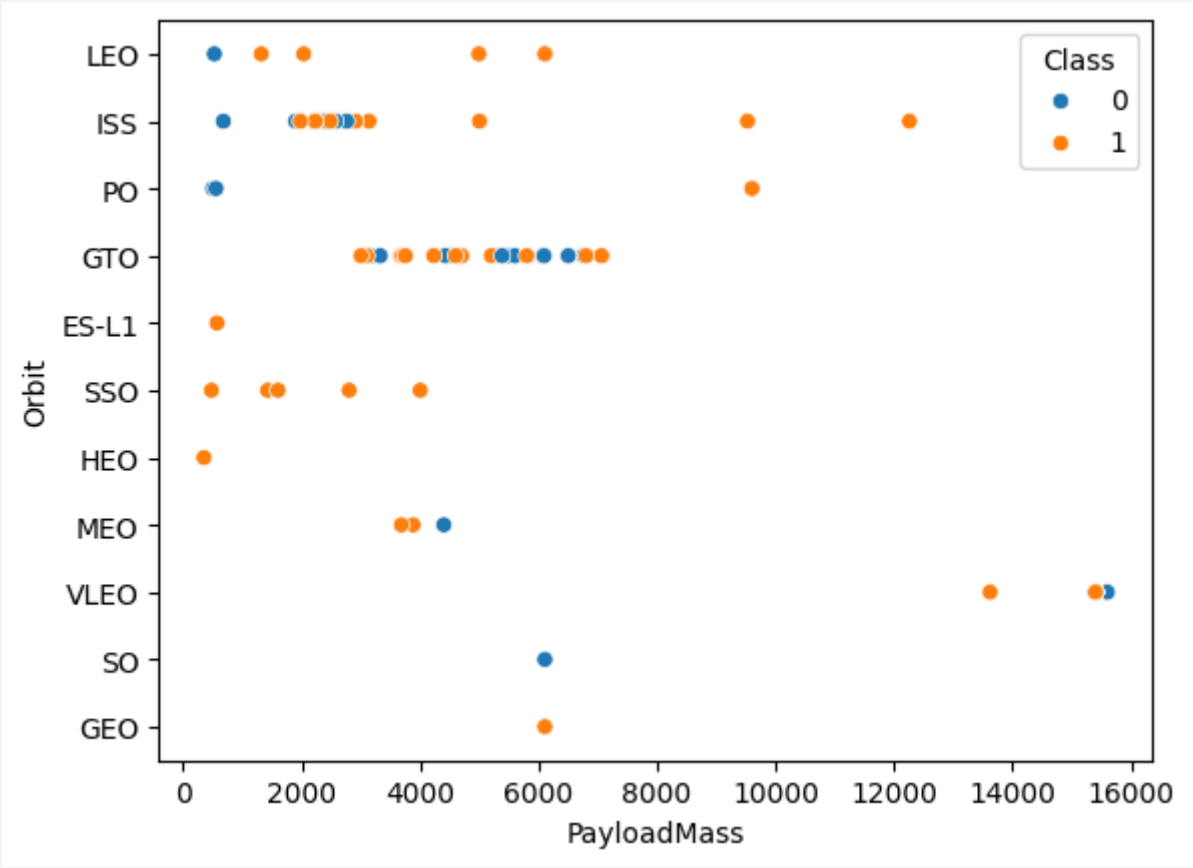
Flight Number vs. Orbit Type

- Findings:
  - Success rate tendentially increases with flight number
  - VLEO has become a very popular orbit recently
  - ISS and GTO are the most frequently used orbits



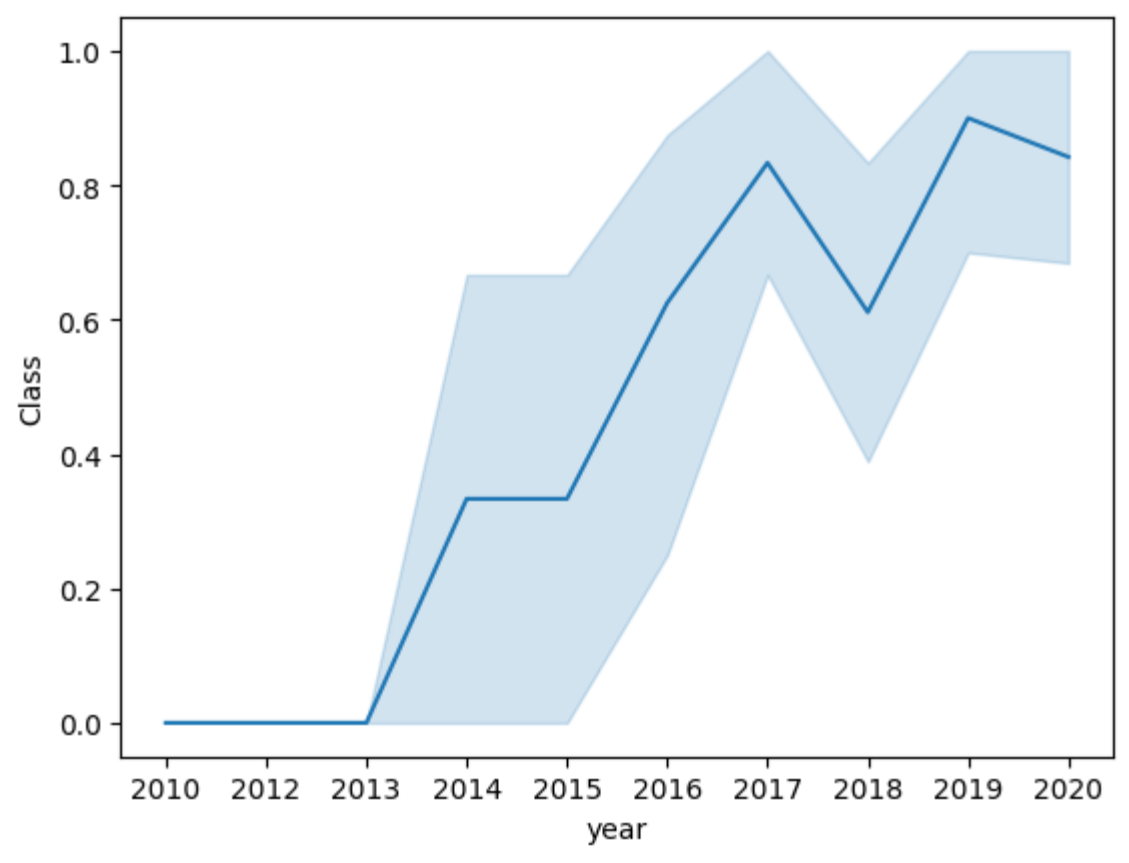
Payload vs. Orbit Type

- Findings:
  - ISS and VLEO are the only orbits with payloads over 7t
  - Big clusters:
    - 2 – 3t for ISS orbit
    - 3 – 7t for GTO orbit



Launch Success Yearly Trend

- Findings:
  - Steep increase in successful missions between 2013 and 2017
  - Relatively steady high success rate between 2017 and 2020



All Launch Site Names

- Explanation:
  - Display the names for all launch sites
- Findings:
  - 4 launch sites found:
    - CCAFS LC-40 (Cape Canaveral AFS Launch Complex 40)
    - VAFB SLC-4E (Vandenberg Space Launch Complex 4E)
    - KSC LC-39A (Kennedy Space Center Launch Complex 39A)
    - CCAFS SLC-40 (Cape Canaveral Space Launch Complex 40)

```
1  %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
✓ 0.0s

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

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

Launch Site Names Begin with 'CCA'

- Explanation:
  - Display 5 records where launch sites begin with the string ‘CCA’

```
1  %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
2
```

✓ 0.0s

```
* 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 Mass

- Explanation:
  - Display the total payload mass carried by boosters launched by NASA (CRS)

```
1  %sql SELECT SUM(PAYLOAD_MASS__KG_) as 'total payload mass (CRS)' FROM SPACEXTABLE WHERE Customer LIKE '%CRS%'
✓ 0.0s

 * sqlite:///my_data1.db
Done.

total payload mass (CRS)
                   48213
```

Average Payload Mass by F9 v1.1

- Explanation:
  - Display average payload mass carried by booster version F9 v1.1

```
1  %sql SELECT ROUND(AVG(PAYLOAD_MASS__KG_),2) as 'average payload mass (F9 v1.1)' FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
✓ 0.0s

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

average payload mass (F9 v1.1)
2534.67

First Successful Ground Landing Date

- Explanation:
  - List the date when the first successful landing outcome in ground pad was achieved

```
1  %sql SELECT MIN(Date) as 'first successful landing on ground pad' FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'
✓ 0.0s

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

first successful landing on ground pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Explanation:
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
1  %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 AND Landing_Outcome = 'Success (drone ship)'
✓  0.0s

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Explanation:
  - List the total number of successful and failure mission outcomes

```
1  %sql SELECT COUNT(CASE WHEN Landing_Outcome LIKE 'Success%' THEN 1 END) as success, COUNT(CASE WHEN Landing_Outcome LIKE 'Failure%' THEN 1 END) as failure FROM SPACEXTABLE
```

✓ 0.0s

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

success	failure
61	10

Boosters Carried Maximum Payload

- Explanation:
  - List the names of the booster_versions which have carried the maximum payload mass.

```
1  %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

✓ 0.0s

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

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

2015 Launch Records

- Explanation:
  - List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015

```
1  %sql SELECT SUBSTR(Date, 6, 2) as Month, Landing_Outcome as 'failure landing_outcomes in drone ship', Booster_Version as 'booster versions', Launch_Site as 'launch_site' FROM SPACEXTABLE WHERE SUBSTR(Date, 0,5) = '2015'
```

✓ 0.0s

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

Month	failure landing_outcomes in drone ship	booster versions	launch_site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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

- Explanation:
  - 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

```
1  %sql SELECT DISTINCT Landing_Outcome as 'outcome', COUNT(*) as '#' FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC
```

✓ 0.0s

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

outcome	#
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

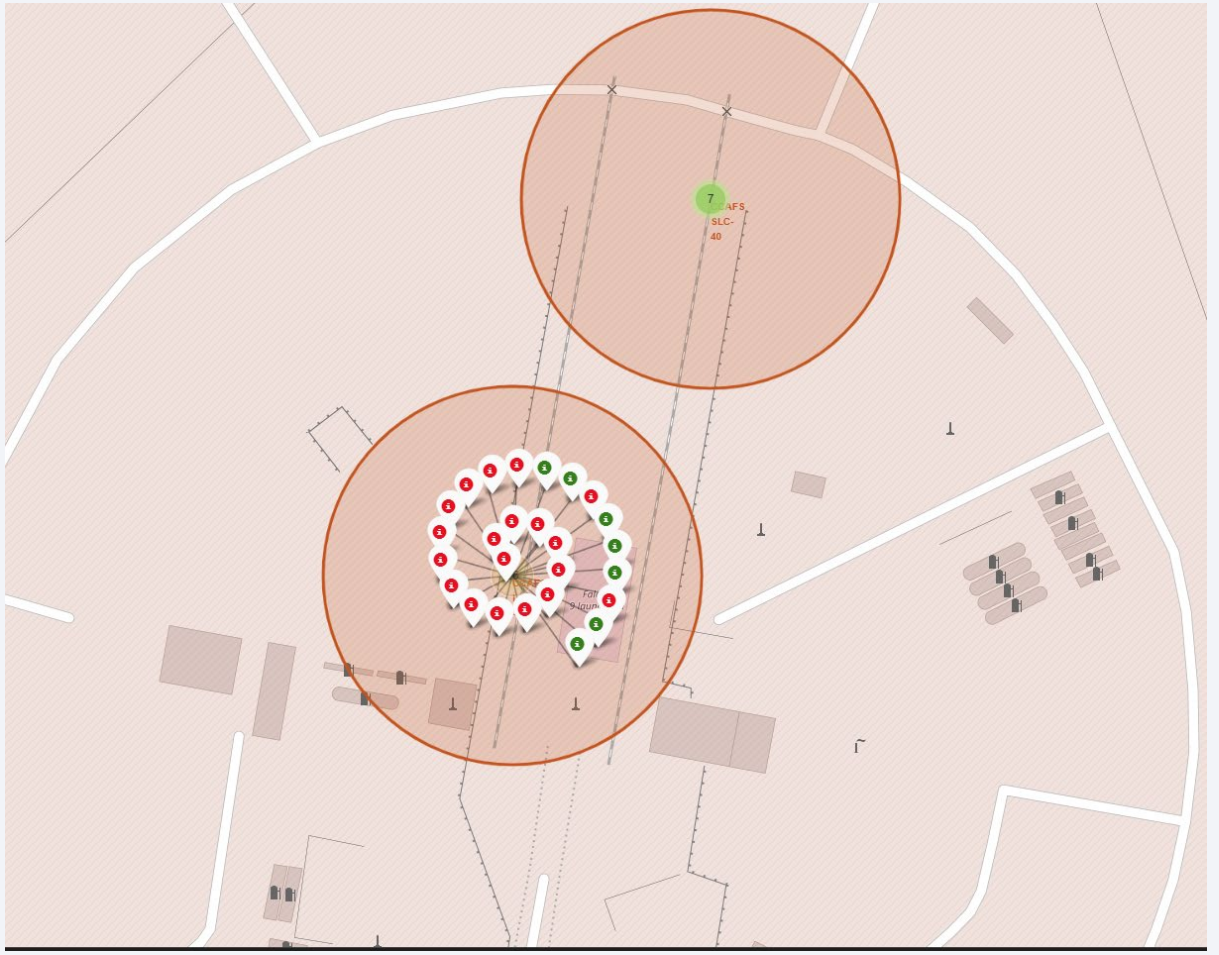

Section 3
Launch Sites
Proximities Analysis

SpaceX launch sites

- Findings:
  - All locations close to the ocean and in the USA

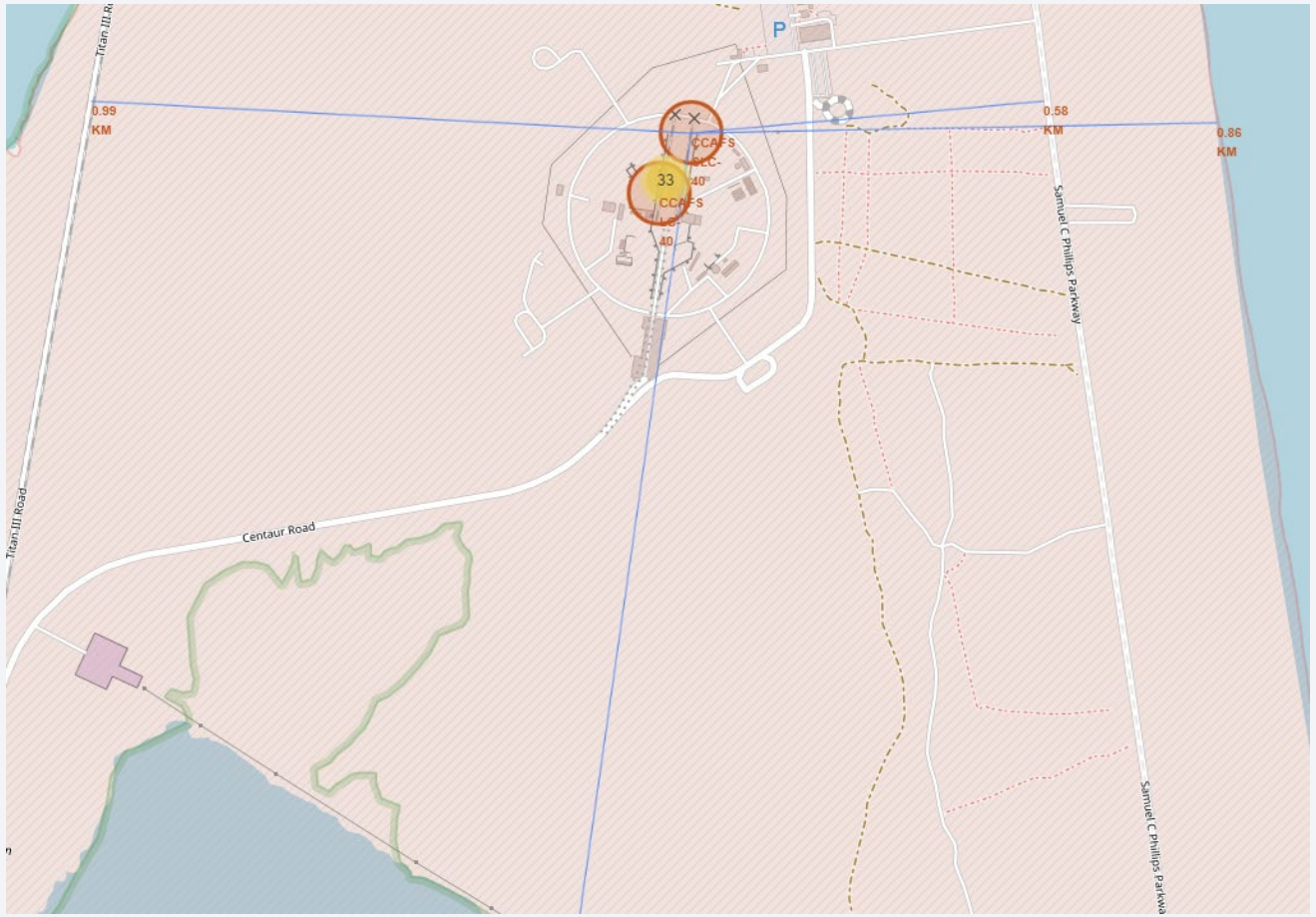
Launch outcome color labels

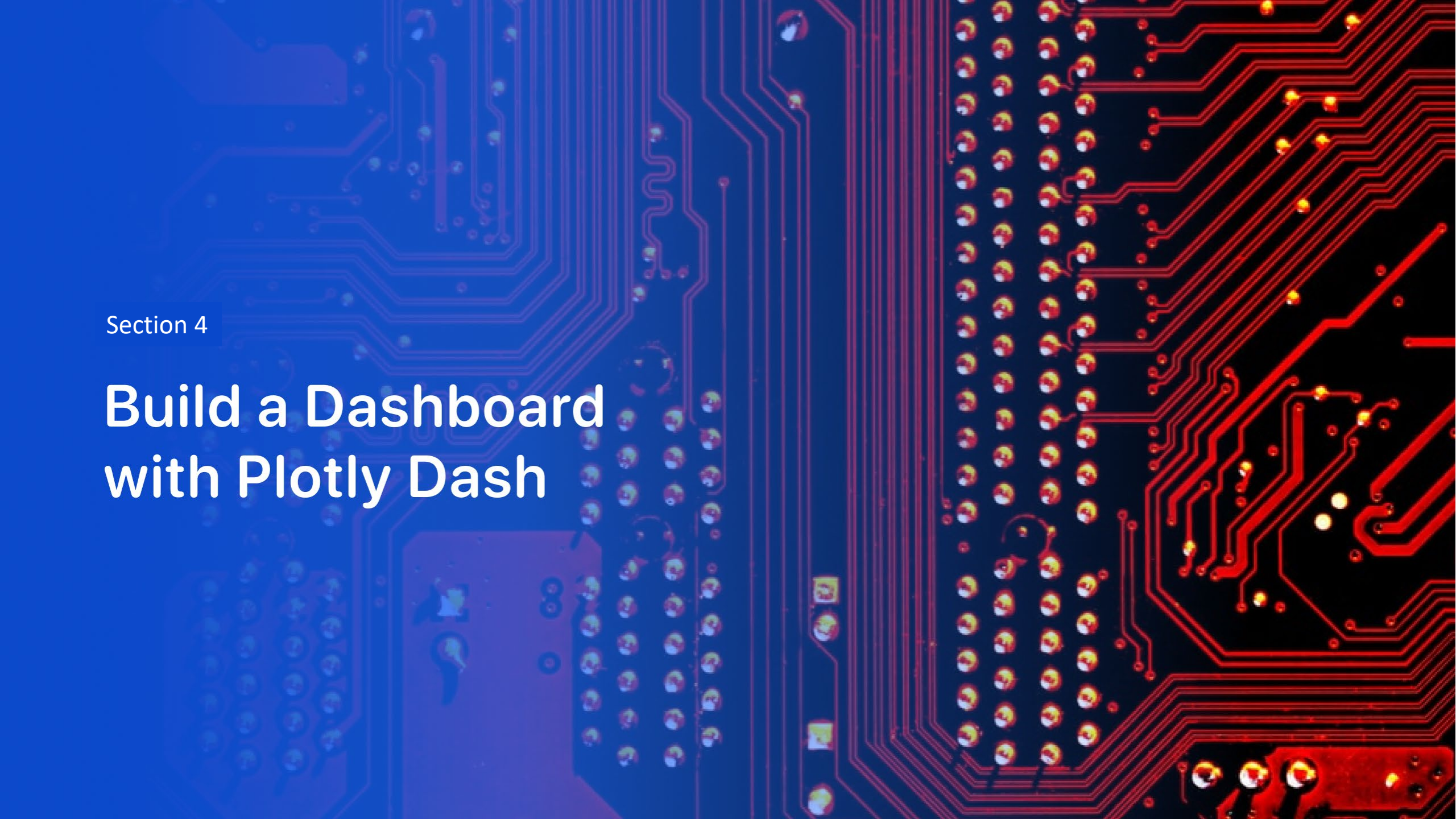
- Red labels for failures, green for successes
- Shown for CCAFS LC-40



Distances to Infrastructure

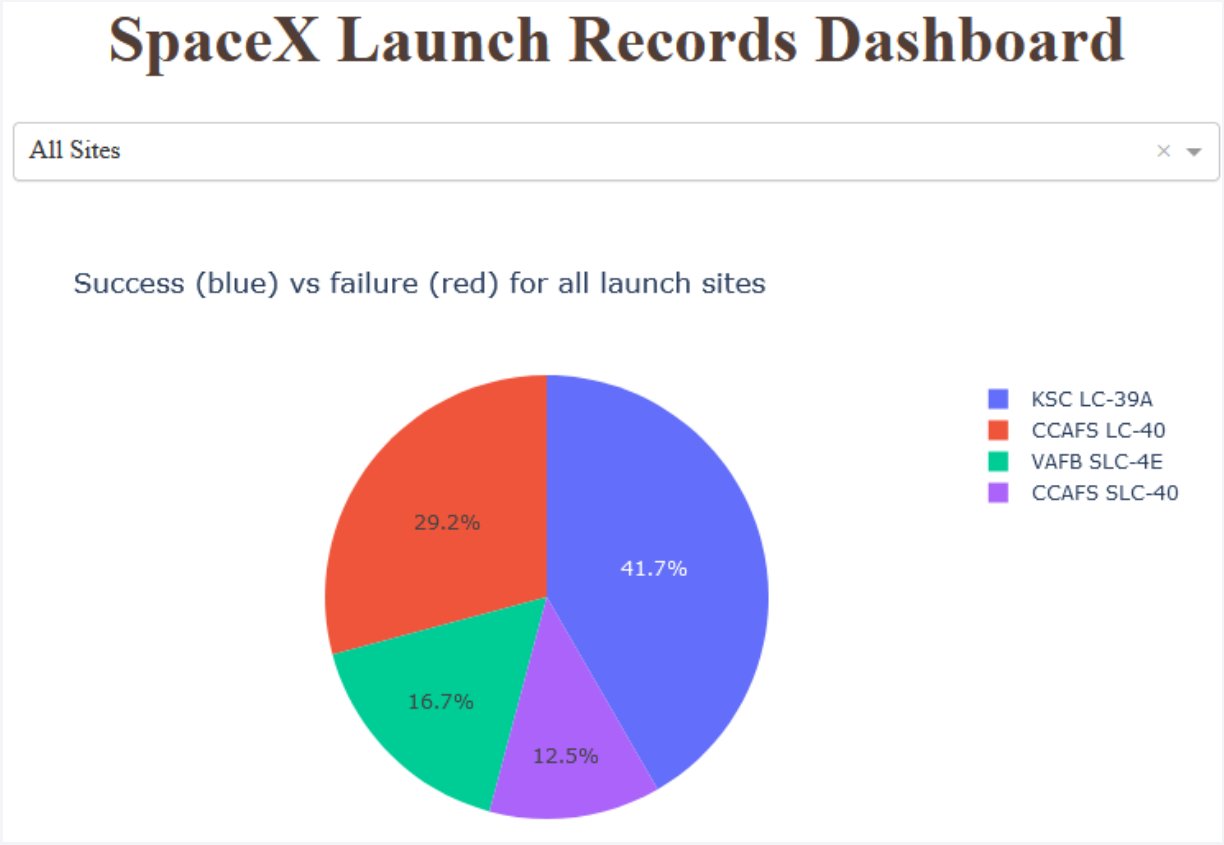
- Lines with distance labels to relevant infrastructure:
  - Left: railroad
  - Right: highway and ocean
  - Bottom: city (cut off)




Section 4
Build a Dashboard with Plotly Dash

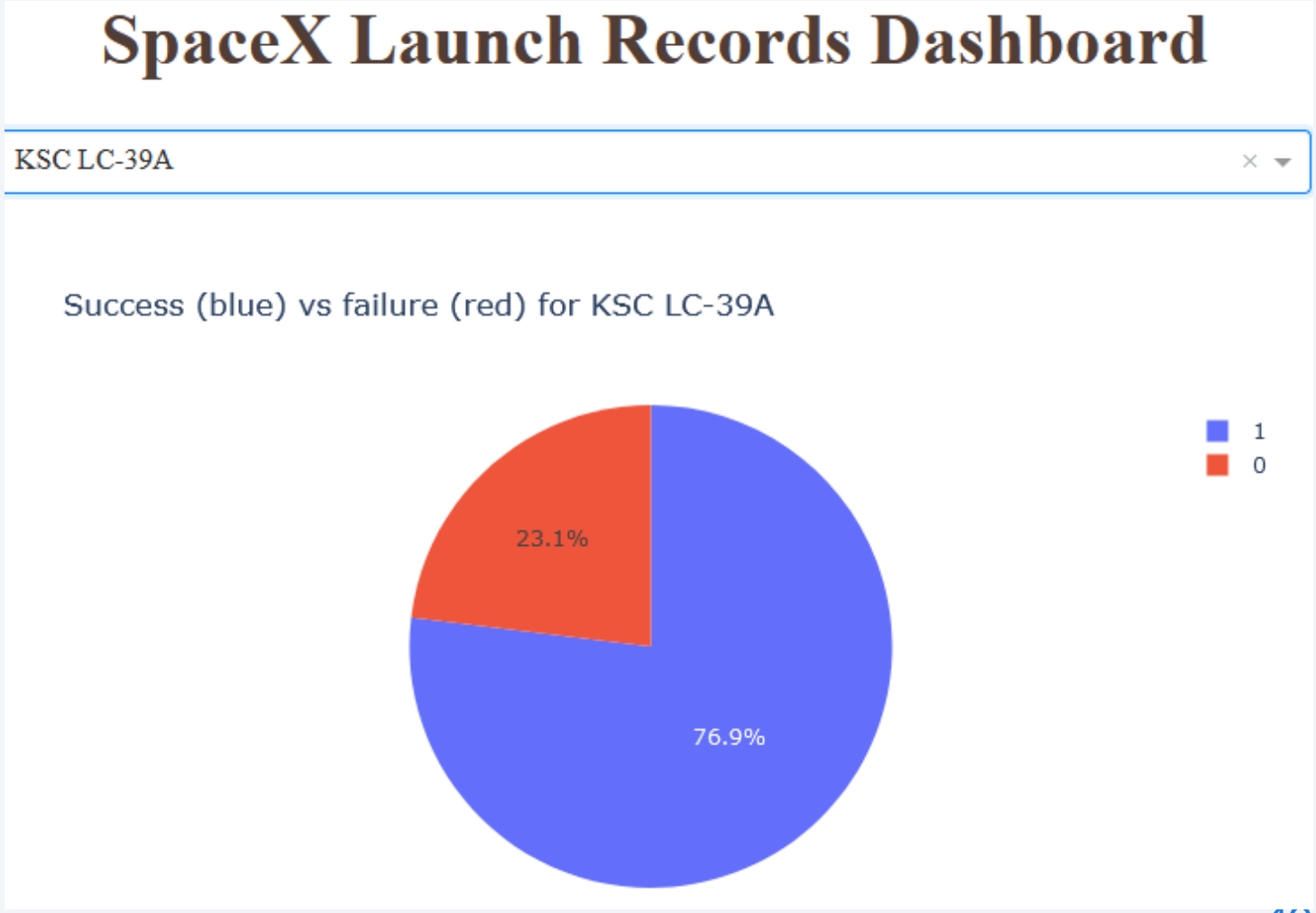
Dashboard: Launch success for all sites

- Launch successes for all 4 sites shown.
- KSC LC-39A and CCAFS LC-40 launched most successes



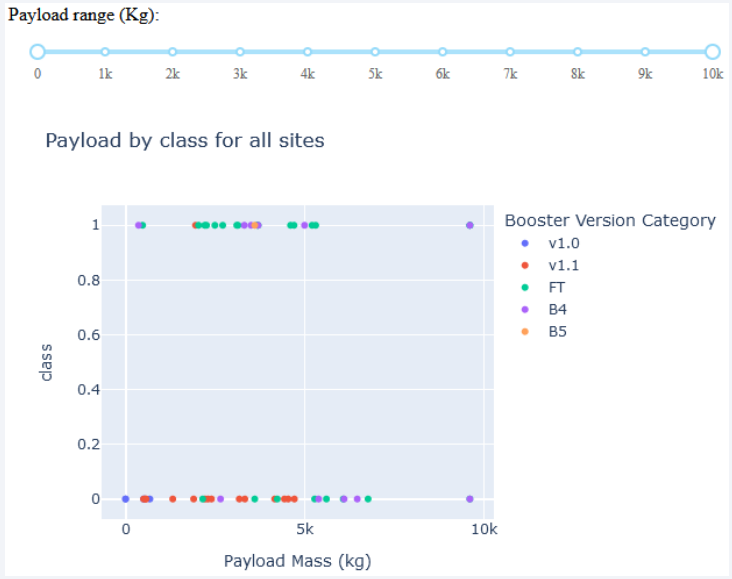
Dashboard: KSC LC-39A details

- Pie chart for KSC LC-39A
  - 10 successes (76.9%)
  - 3 failures (23.1%)

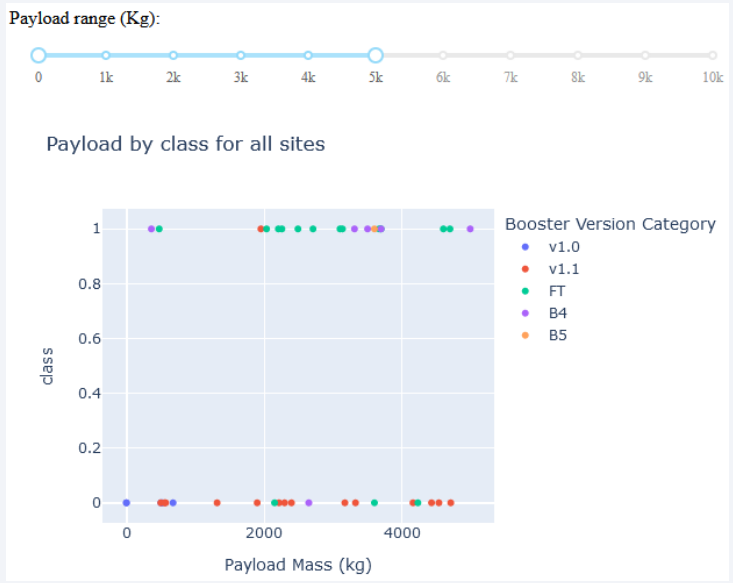


Dashboard: Different payloads for all sites

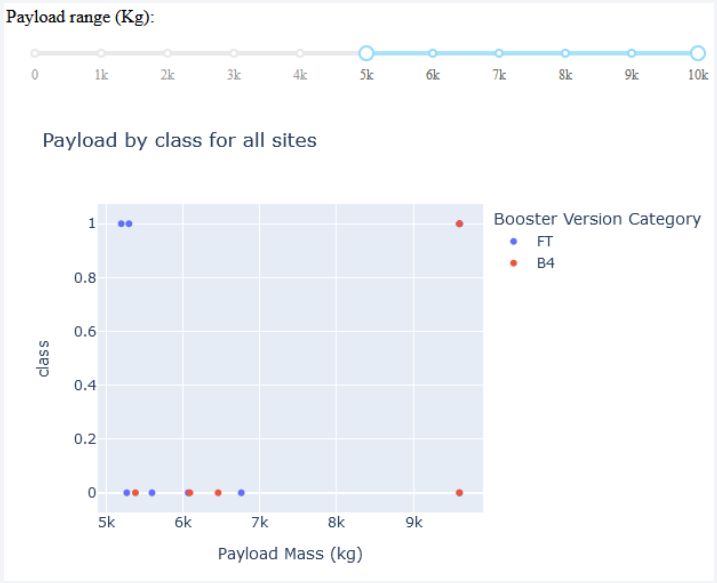
Total range (0 – 10t)



lower range (0 – 5t)



upper range (5 – 10t)



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All models have identical accuracy

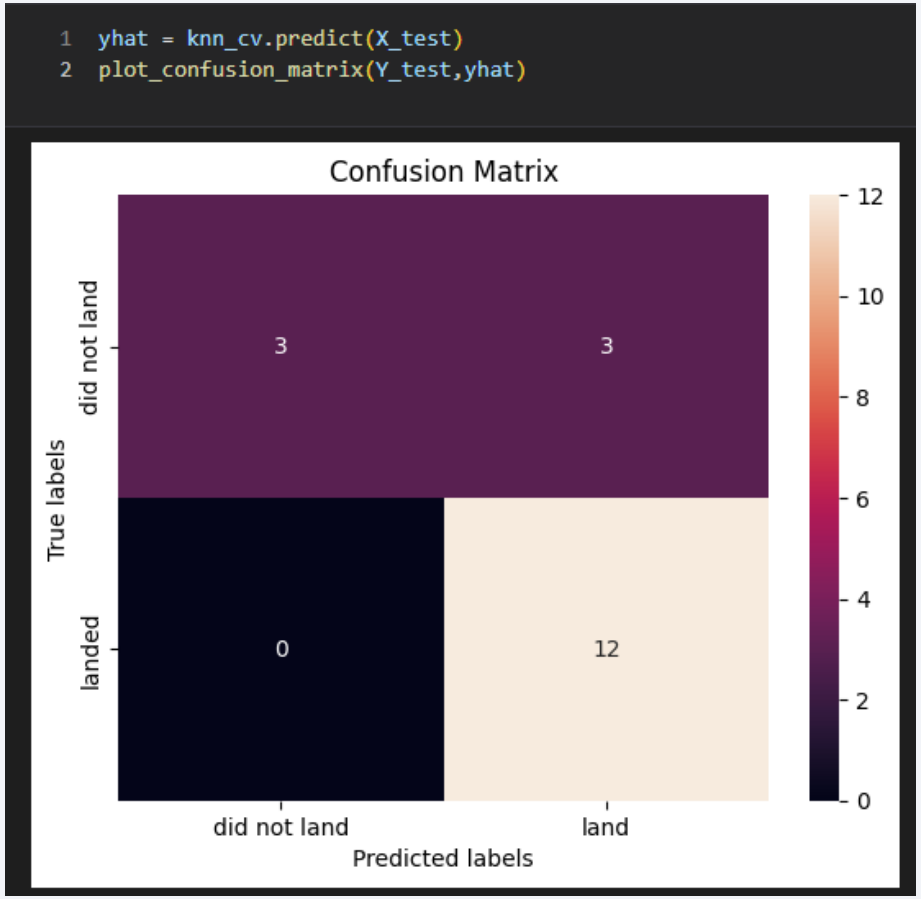
```
print(f'LogReg: {logreg_cv.score(X_test,Y_test)}'
print(f'SVM: {svm_cv.score(X_test,Y_test)}')
print(f'Tree: {tree_cv.score(X_test,Y_test)}')
print(f'KNN: {knn_cv.score(X_test,Y_test)}')
```

```
LogReg: 0.8333333333333334
SVM: 0.8333333333333334
Tree: 0.8333333333333334
KNN: 0.8333333333333334
```

Confusion Matrix

- Confusion matrix for KNN
- 15 hits
- 3 false positives
- 0 false negatives

```
1  yhat = knn_cv.predict(X_test)
2  plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Conclusions for Classification
  - ML models can be used to classify launches with decent accuracy
  - The chosen model is not relevant

Appendix

- Link to entire github repository (all used files)
  https://github.com/elmi3/ibm_capstone

Thank you!