IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Satvik Reddy Konda
28 September 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection Using API
  - Data Collection Using Web Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA with Visualization
  - Interactive Visual Analysis with Folium & Plotly Dash
  - Prediction Using Machine Learning
- Summary of all results
  - Understanding Variable Relationships Through EDA Plots
  - Understanding Location Based Relationships through Folium
  - Interactive Analysis for Further Understanding
  - Exploring Machine Learning Models for Predicting Landing Outcomes

Introduction

- SpaceX has revolutionized space travel by returning a space craft from outer space back to Earth in 2010
- Advertises that the Falcon 9 rocket launcher costs 62 million while other companies spend up to 165 million
- Most of the savings are due to SpaceX's ability to reuse the first stage
- Therefore, knowing if the first stage can land leads to the understanding of launch costs
- Problem: SpaceY is a new company that wants to compete with SpaceX but does not having sufficient information
- Solution: Understand SpaceX's already conducted launches and successfully beat SpaceX for a rocket launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Requesting SpaceX API for multiple sets of data
  - Webscraping launch data from Wikipedia
- Perform data wrangling
  - Filtering data through SpaceX API data
  - Creating “Class” training labels which determine launch outcome & used later for models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Experiment with multiple machine learning models while using GridSearchCV and accuracy scores

Data Collection

- Data sets collected using SpaceX's API & Webscraping Falcon 9 Tabular Data
  - Multiple endpoints used to collect various types of data
    - https://api.spacexdata.com/v4/launches/past
    - v4/rockets/
    - v4/launchpads/
    - v4/payloads/
    - v4/cores/
    - json_normalize() used to convert Json file into a data frame
    - Data cleaned to only keep relevant data
  - BeautifulSoup used to web scrape Falcon 9 launch data
    - Relevant information loaded onto a separate data frame

Data Collection – SpaceX API

- v4/launches/past used to gather previous launch data of Falcon 9's
- Rockets, launchpads, payloads, and cores API endpoints data requested and parsed into a data frame

- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/data_collection.ipynb

•**Start**
•↓
•**API Request** (to endpoints like /v4/launches/past, /v4/rockets/, etc.)
•↓
•**Receive JSON Response**
•↓
•**Parse JSON to Data Frame** (using json_normalize())
•↓
•**Data Cleaning** (keep only relevant data)
•↓
•**Data Set Collected**
•↓
•**End**

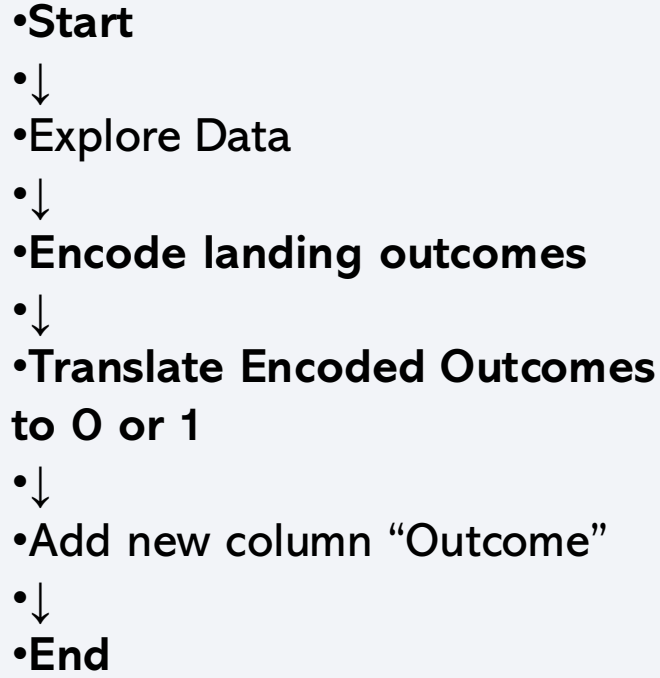
Data Collection - Scraping

- Created a BeautifulSoup object and passed Wikipedia link
- Sent a HTTP GET request to the page to receive HTML content
- HTML response parsed and loaded into a data frame
- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/webscraping.ipynb

•Start
•↓
•Create BeautifulSoup object
•↓
•Send GET Request via HTTP (searching for only <th>)
•↓
•Receive Response and find Column Names
•↓
•Extract Column Names from tables
•↓
•Fill Data Frame
•↓
•End

Data Wrangling

- Used EDA to understand variables in the data
  - Finding percentage of missing values per column
  - Identified data types of each column
  - Found number of launches per unique launch station
  - Found type of orbit for each launch
  - Determined type and number of landing outcomes of all launches
- Determined the training labels and creating a new column for it
  - Encoded all ”False” and “True” landing outcomes
  - Determined final “Outcome” of launch as either 0 (failure) or 1 (successful)
  - This data added into column as training label for later ML models



https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/data_wrangling.ipynb

EDA with Data Visualization

- Plotted Charts (with hue as Class -> 0 or 1)
  - Flight Number vs. Payload (if payload weight influences landing outcome)
  - Flight Number vs. Launch Site (if location of launch effected landing outcome)
  - Launch Site vs. Payload (if a pattern between launch location and payload weight exists)
  - Success Rate vs. Orbit (if any specific orbit has a higher/lower success rate than others)
  - Flight Number vs. Orbit (if any specific orbit has better landing outcomes than others)
  - Payload vs. Orbit (if weight of payload and type of orbit shows any patterns)
  - Year vs. Success Rate (to check yearly success rate of launch outcomes and see if patterns exist)
- All features then one hot encoded and changed into numerical values
- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/eda_viz.ipynb

EDA with SQL

- SQL Queries Performed to Gather Further Insights
  - Unique Launch Sites
  - 5 records beginning with 'CCA'
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Date of first successful landing outcome in ground pad
  - Boosters success in drone ship of payload mass greater than 4000 & less than 6000
  - Total number of successful & failure mission outcomes
  - Booster versions that carries max payload mass (using subqueries)
  - Month, failure landing outcomes in drone ship, booster version, launch site in the year 2015
  - Number of occurrences rank of landing outcomes between 2016/06/04 and 2017/03/20
- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/eda_sql.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Circles with markers plotted to identify location of launch and landing outcome
- Lines with distances plotted to show distance between launch site and nearby landmarks (railways, airports, coastlines)
- These objects added to visually check for any relationship between launch locations, its outcomes, and nearby landmarks (to check if they may have had an effect)
- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/folium.ipynb

Build a Dashboard with Plotly Dash

- To allow for interactive analysis of the data, a dashboard was created using Plotly
  - Pie chart
    - Depending on selected data subset (all or single landing location), a pie chart is shown based on successful landing locations for all sites and success vs failure if a single location is chosen
  - Scatter Plot
    - Depending of selected data subset, a scatter plot of payload mass vs booster versions are shown
    - A range selector of payload mass is also shown to filter the scatter plot furthermore

- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

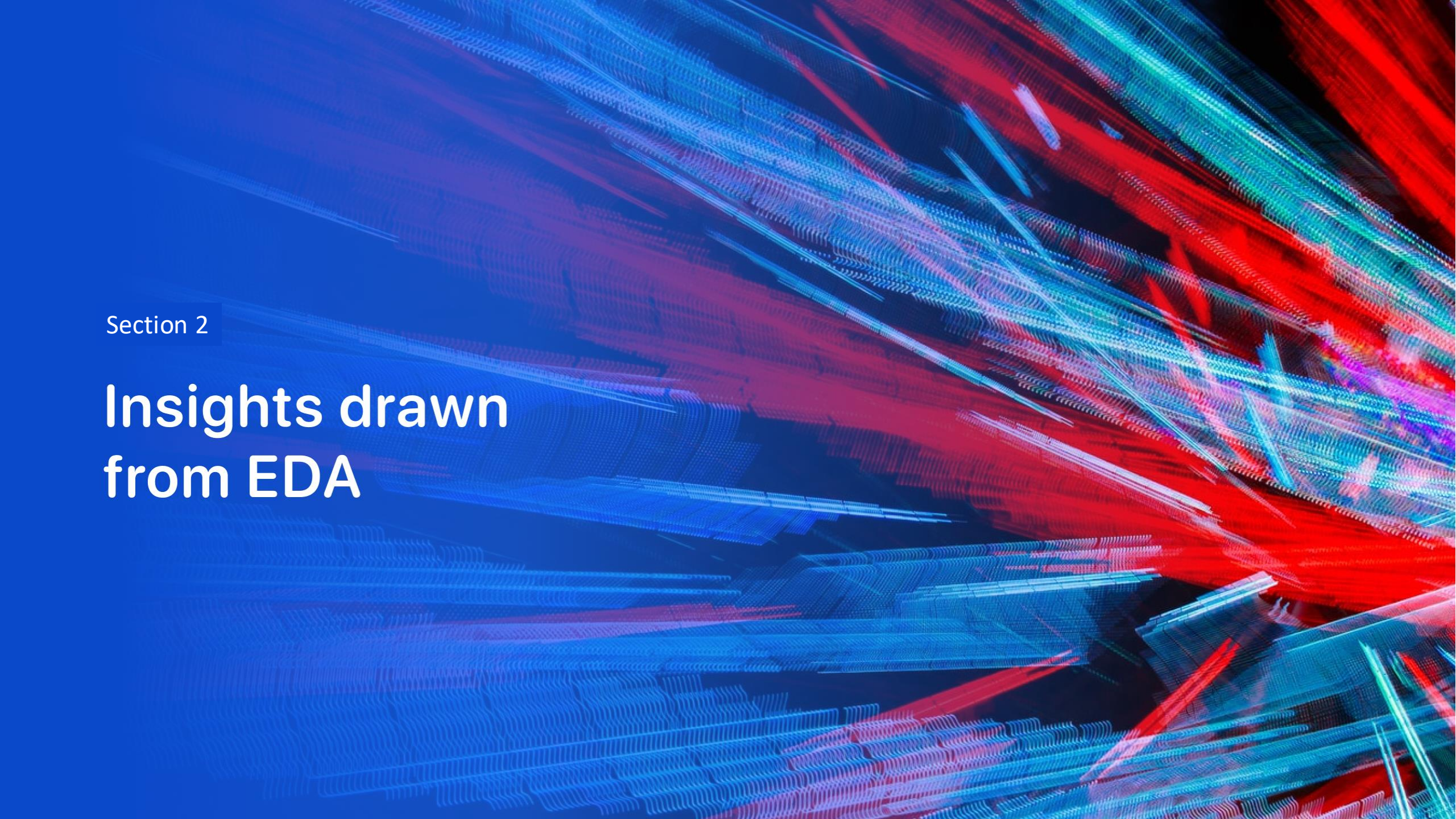
1. Load data frame
2. Assign column “Class” to Y and convert to NumPy array
3. Apply StandardScaler() on X
4. Train/Test split the data with 20% for testing
5. Use GridSearchCV to fit multiple different models, check accuracy, and plot confusion matrices
    1. Logistic Regression
    2. SVC
    3. Decision Tree Classifier
    4. KNN Classifier

- https://github.com/stavik31/SpaceX-Landing-Prediction/blob/main/prediction.ipynb

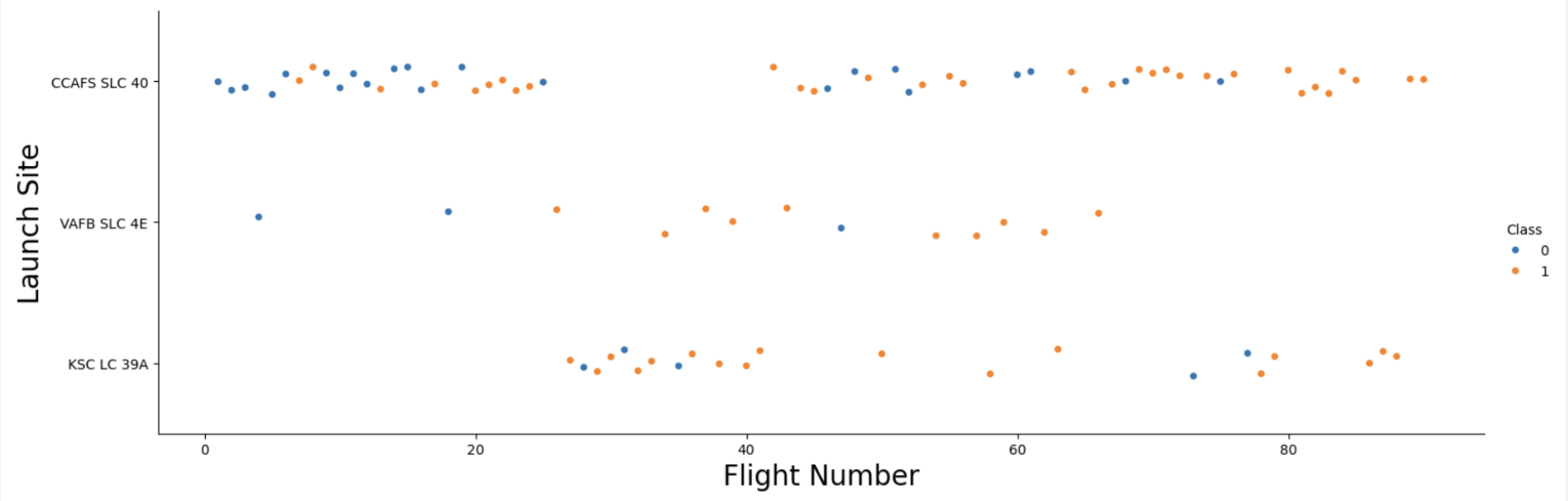
- **Start**
- ↓
- **Load Data Frame**
- ↓
- **Assign X and Y (convert Y to Numpy array)**
- ↓
- **Standardize X**
- ↓
- Train/Test Split
- ↓
- **GridSearchCV for multiple algorithms**
- ↓
- **Evaluate Using accuracy & confusion matrices**
- ↓
- End

Results

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

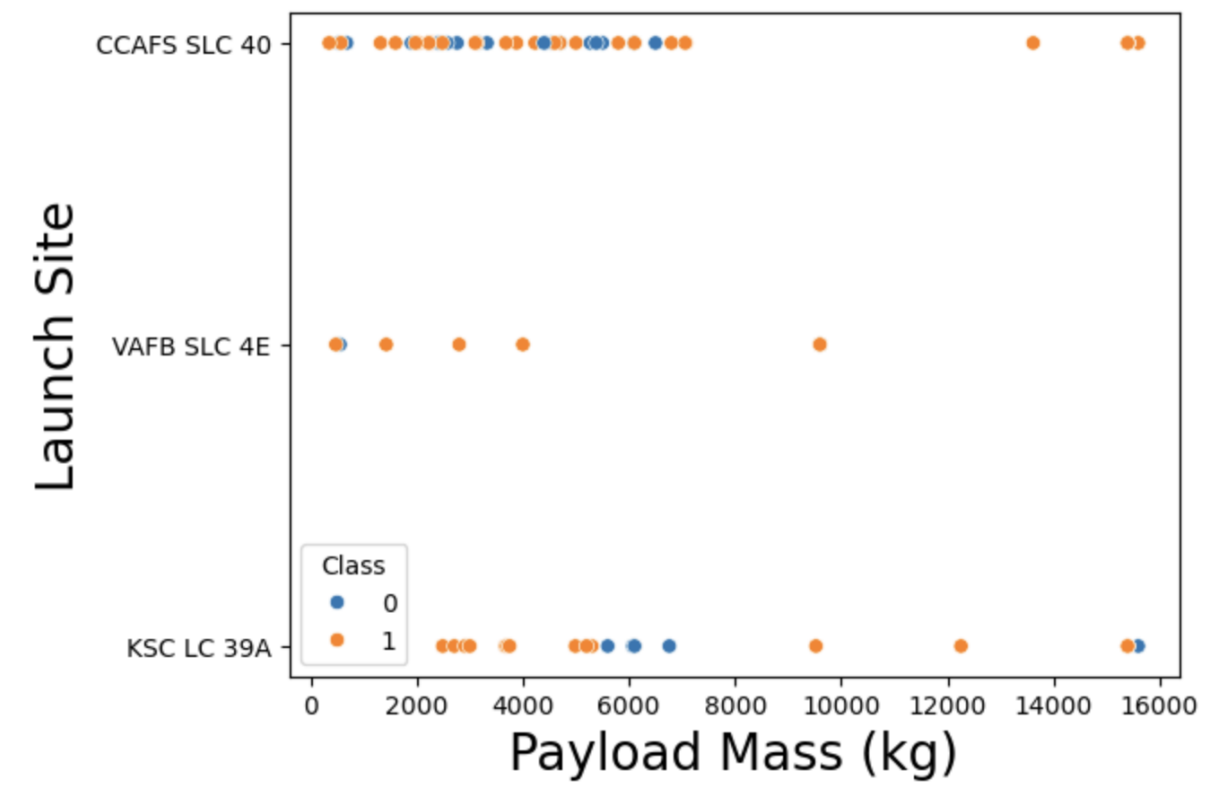

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



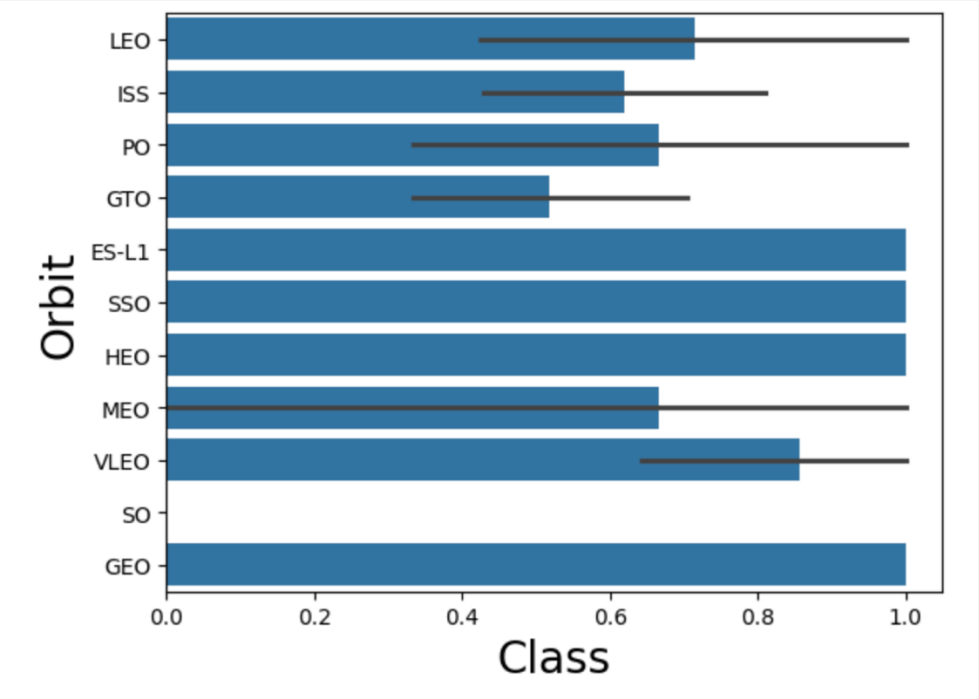
- Initial launches show that the majority are failures
- Regardless of launch site, as the number of flights increased, frequency of successful launches increased
- Possibly indicates that better technology was implemented or improved methods were implemented

Payload vs. Launch Site



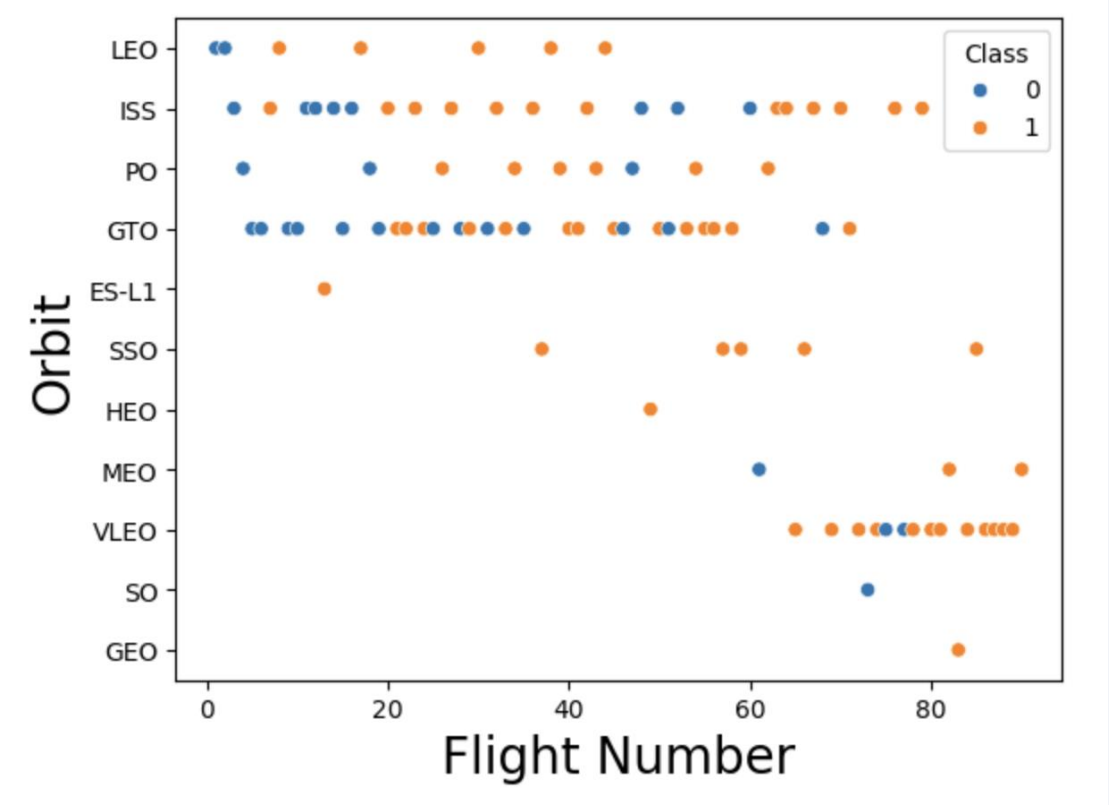
- All launch sites have varying payload mass
- SLC 40 shows the most frequency of launches with success and failure mixed
- Although limited, heavier payloads show more success than failures
- Suggests that most launches did not require high payload masses but when required, failures were minimal

Success Rate vs. Orbit Type



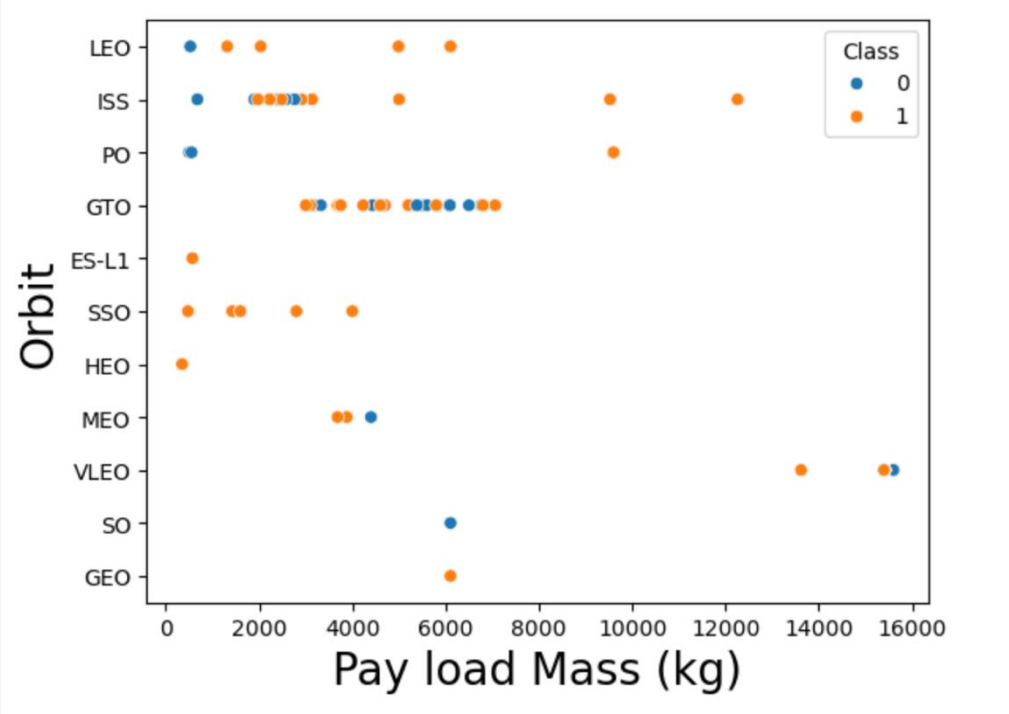
- Close to half of the orbit types have a success rate of over 80%
- Other half have lower success rates with SO having no data or 100% failure
- Suggests that some launches may have more advanced launch methodologies and some may not have enough operational development yet

Flight Number vs. Orbit Type



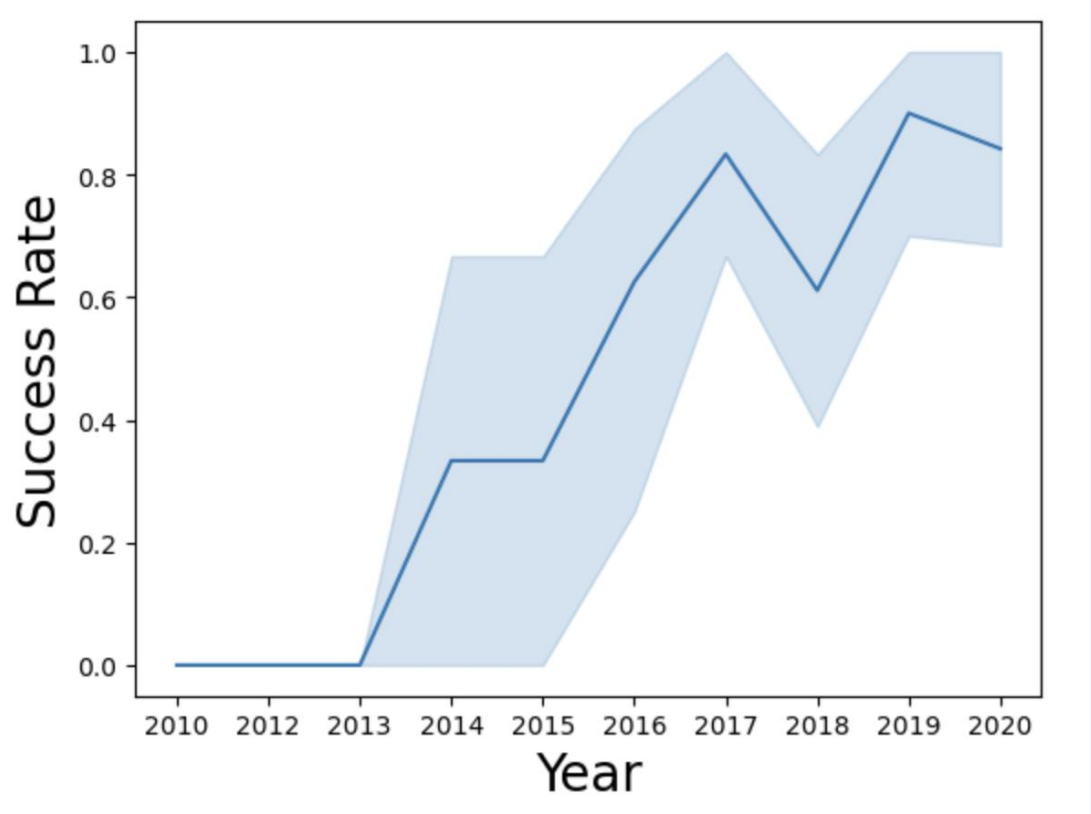
- Lauches vary is many different orbits, even as flight number increases
- Some orbits attempted much later than others suggesting that technological or operational improvements were needed
- Earlier flight numbers consist of many failures but as they increase, success also followed

Payload vs. Orbit Type



- Wide range of payload mass between all orbits but specific orbits have a range
- There is no direct relationship between payload mass, orbit, and if the launch is successful or not

Launch Success Yearly Trend



- Overall, a consistent improvement in success rates can be seen
- Significant increase in 2013 and 2015 with a minor decrease in 2018 and 2020
- Suggests a upward trend in the future from a purely visual perspective

All Launch Site Names

- A total of 4 unique launch sites:
  - CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, & CCAFS SLC-40

- select distinct Launch_Site from SPACEXTABLE
  - “distinct” takes only unique instances of “Launch_Site” that appear in the data frame

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

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

- select * from SPACEXTABLE where Launch_Site like '%CCA%' limit 5
  - ”like” alongside ”%” allows for any string that contain “CCA” to be retrieved (with a limit of 5 entries)

Total Payload Mass

- Total payload carried by boosters from NASA (CRS)
  - 45596 kg
- select sum(PAYLOAD_MASS__KG_) as 'NASA (CRS) total payload' from SPACEXTABLE where Customer = 'NASA (CRS)'
  - “sum” allows for every numerical column to be added together, resulting in a total

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1
  - 2534.66 kg
- select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like '%F9 v1.1%'
  - Alongside “like”, “avg” allows for finding the average of all entries, resulting in the average

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad
  - 2015-12-22
- select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
  - “min” retrieves the lowest value possible given the entries, resulting in the minimum date

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - F9 FT B1022, F9 FT B1026, F9 FT B1012.2, F9 FT B1032.2

- select distinct Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
  - “between” allows for a shorter query than using ”<“ or “>”
  - When used, retrieves entries between two values

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
  - Successful Outcome: 61
  - Failed Outcome: 10
- select sum(Landing_Outcome like '%Success%') as successful_outcome, sum(Landing_Outcome like '%Failure%') as failed_outcome from SPACEXTABLE
  - “sum” & “like” were used along side each other to sum any values where a column had either the word “Success” or “Failure”

Boosters Carried Maximum Payload

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

- select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
  - Subquery used by setting PAYLOAD_MASS_KG to the filtered data frame where required data exists

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

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

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

- SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome LIKE '%Failure%' AND substr(Date, 1, 4) = '2015' AND Landing_Outcome LIKE '%drone ship%';
  - "substr" used to refer to retrieve a specifc year in the "Date" column

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

- Ranking 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

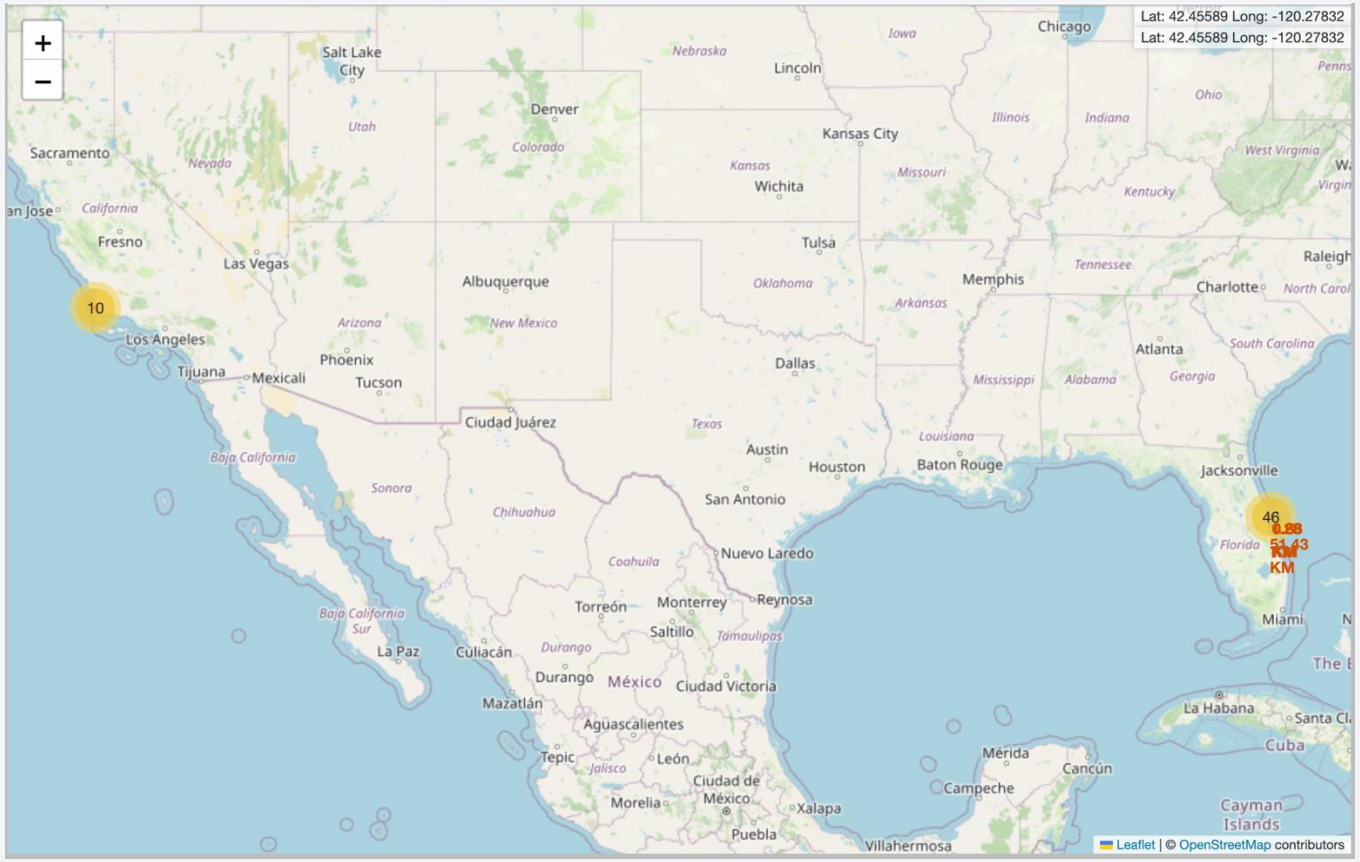
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC;
  - "ORDER BY" & "DESC" used to order entries by a specific column and in decreasing order

Section 3

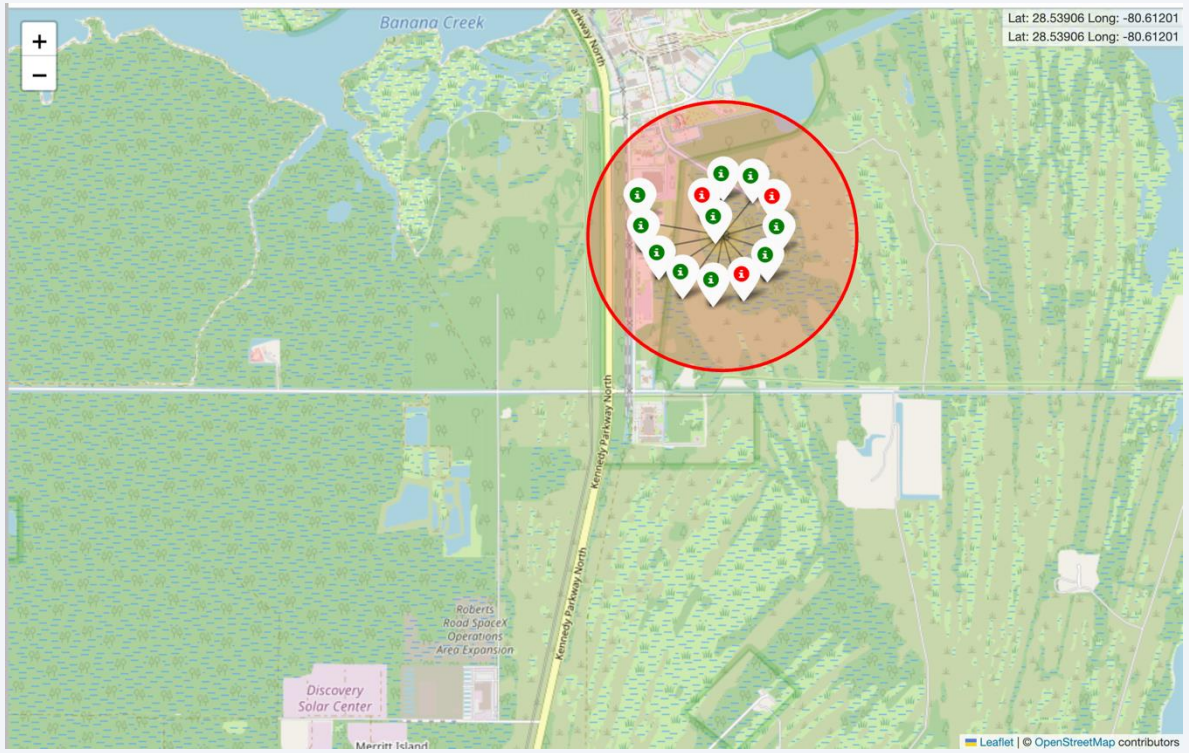
Launch Sites Proximities Analysis

Launch Locations



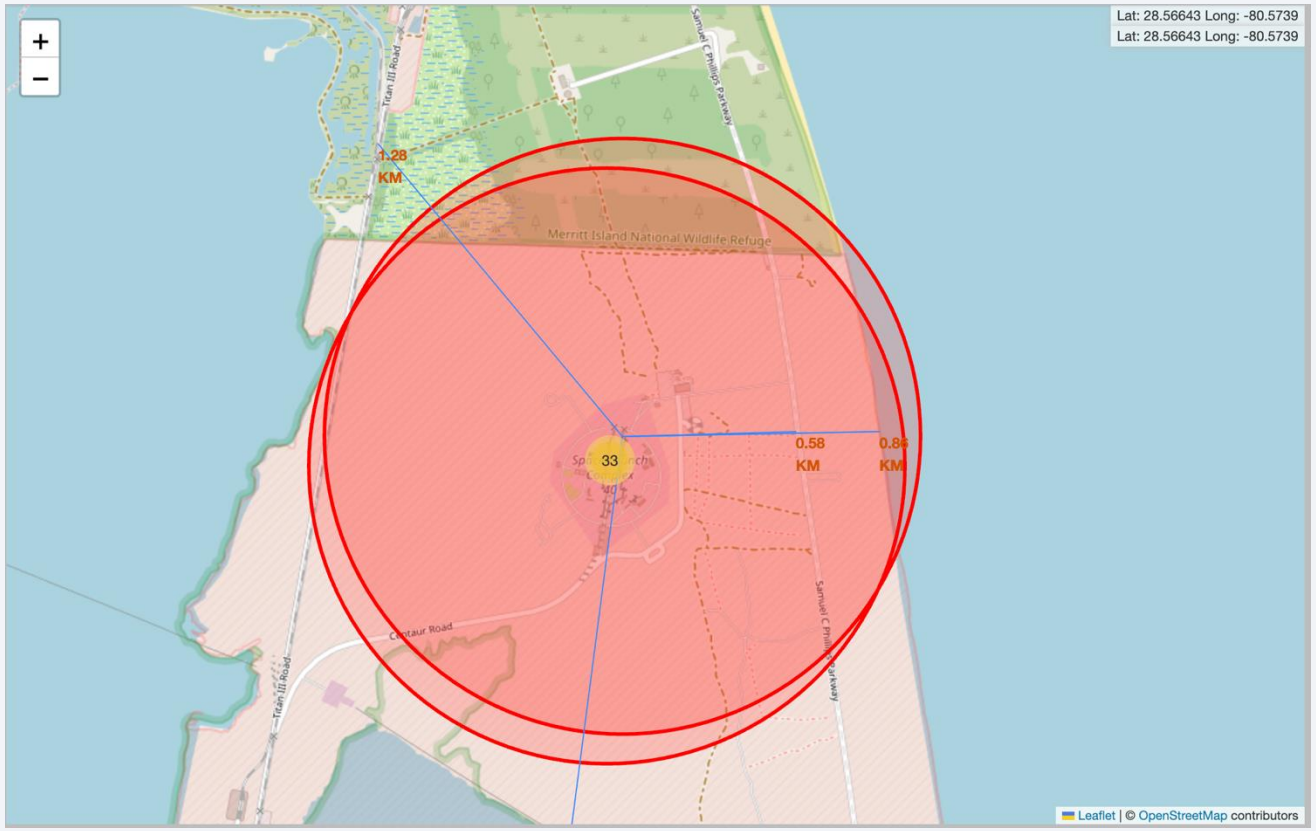
- All launch locations are placed at either coasts (California & Florida)
- Possible reasons for such placement could include to avoid human and rocket related accidents

<Folium Map Screenshot 2>

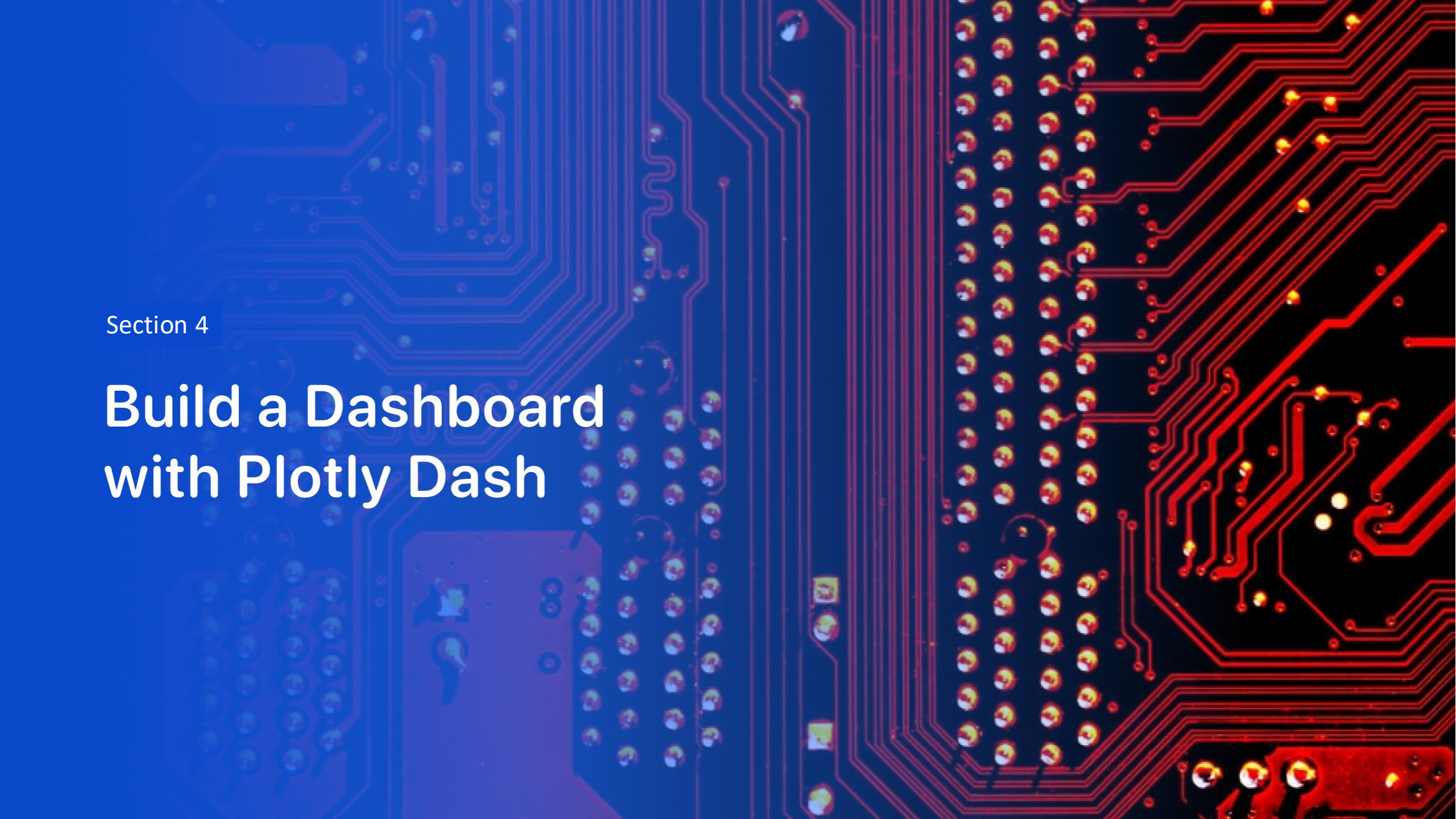


- Within each launch location, the outcome is also shown
- Red = Failed Launch
- Green = Successful Launch

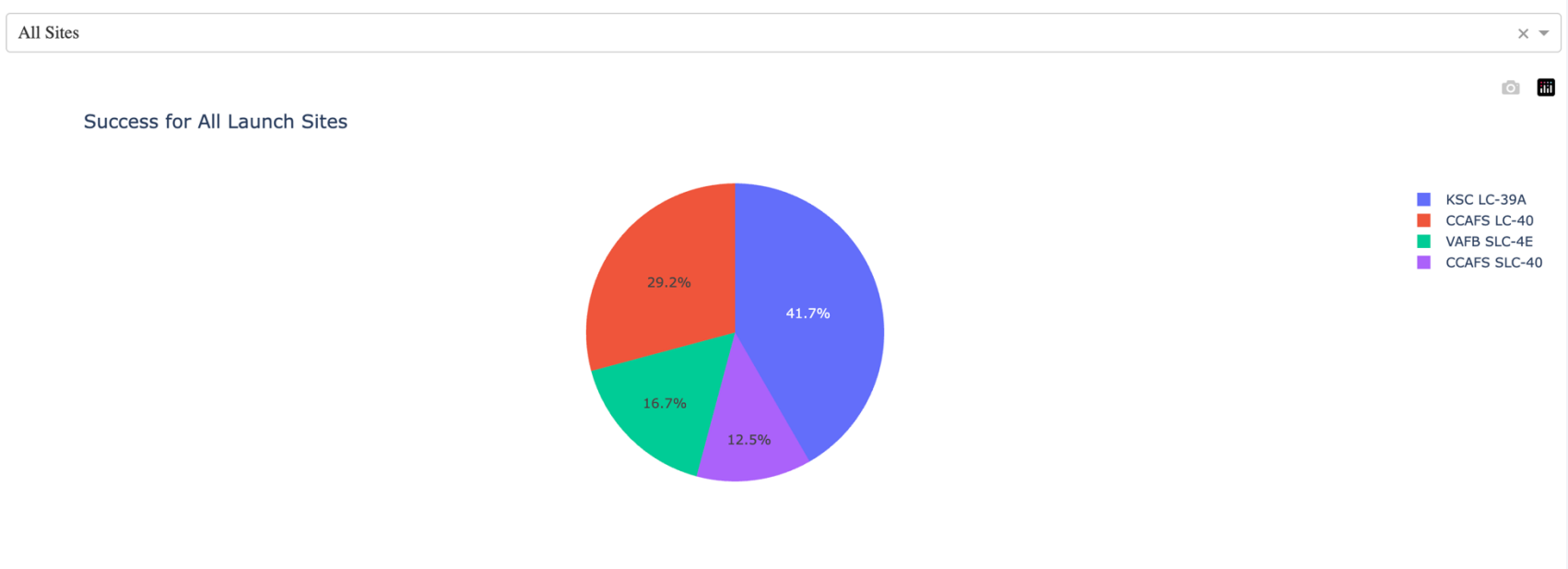
<Folium Map Screenshot 3>



- Blue line from the center of the launch locations indicate distances to landmarks nearby
- For Example…
  - 0.86km to the coast
  - 0.58km to the highway
  - 1.28km to the railway

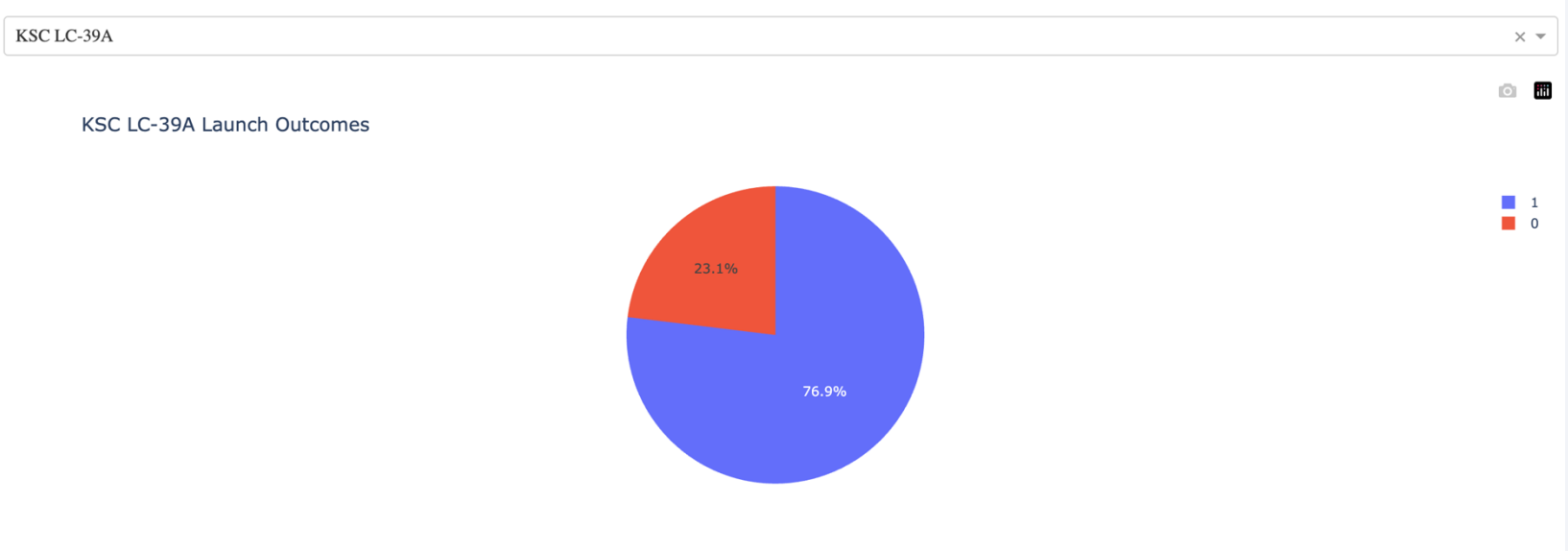

Section 4
Build a Dashboard with Plotly Dash

Success Rates for All Launch Sites



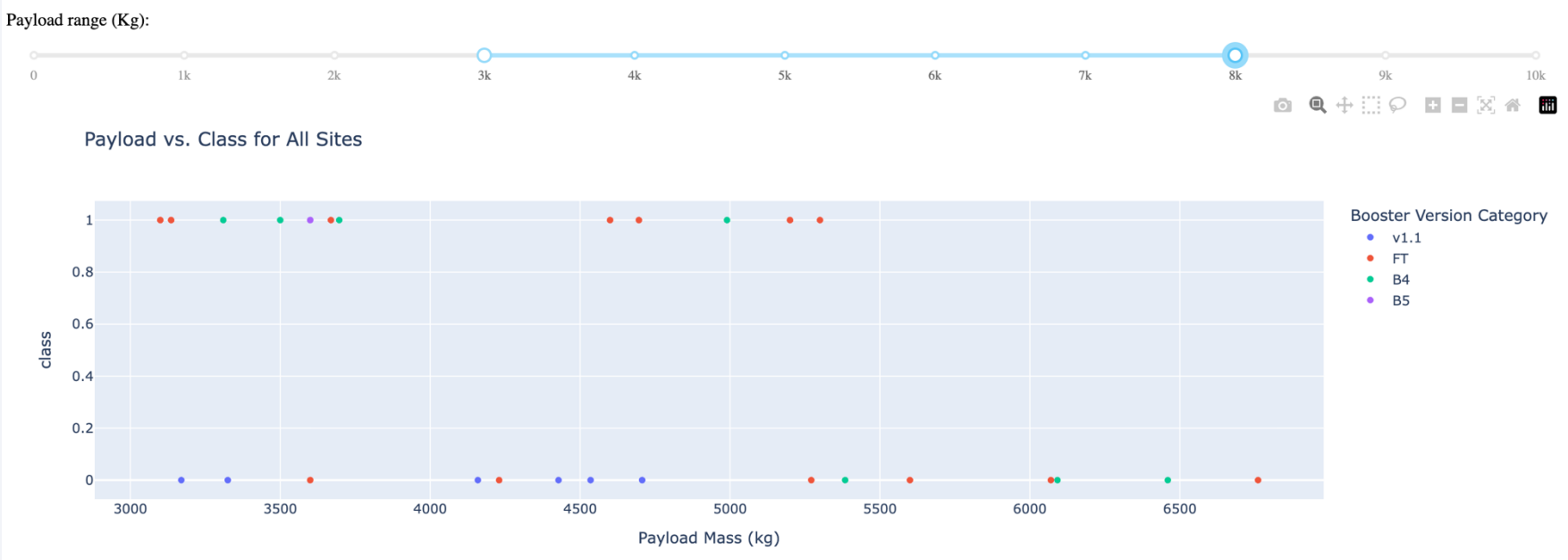
- KSC LC-39A had the highest proportion of successful landings
- Followed by CCAFS LC-40 with just over ¼ of all successful landings
- VAFB SLC-4E & CCAFS SLC-40 had the lowest proportion of successful landings

KSC LC-39A Launch Outcome



- Within all launches of KSC LC-39A:
  - Just over ¾ of all launches were successful
  - Just under ¼ of all launches were failures

<Dashboard Screenshot 3>



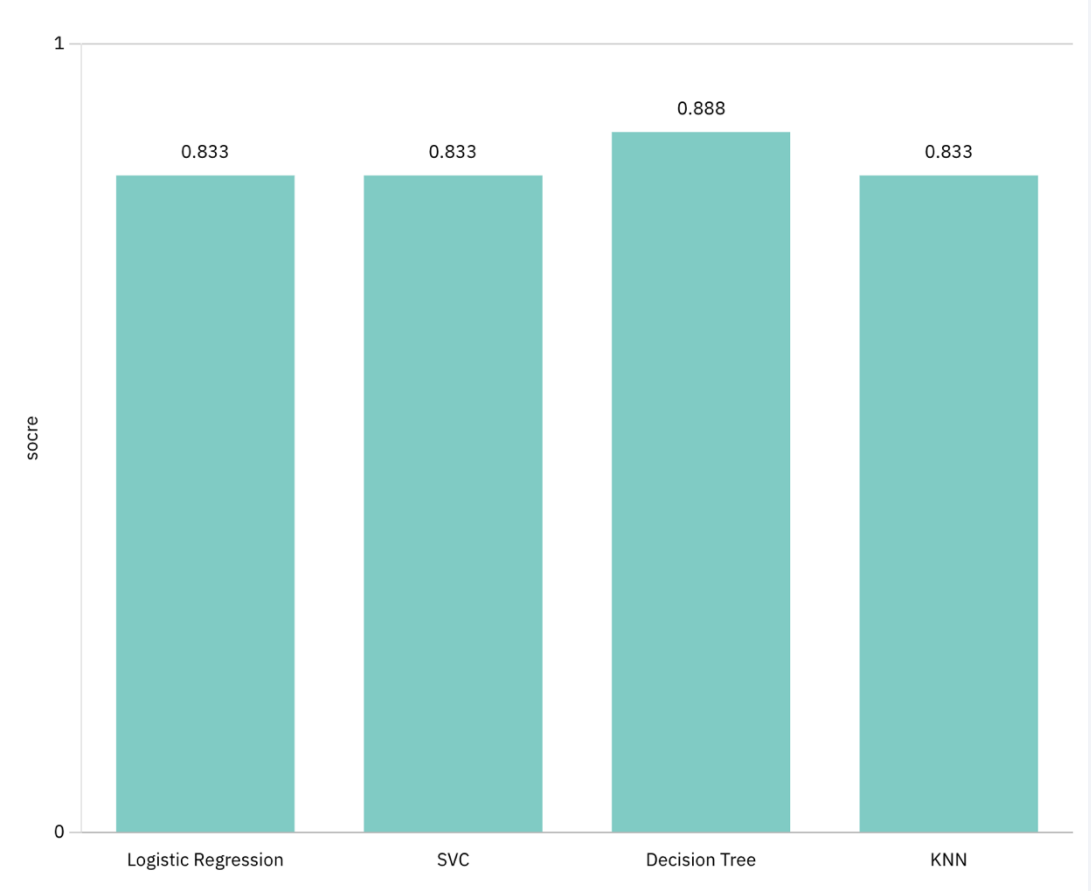
- For Payload ranges of 3000kg to 8000kg:
  - Booster B5 had the highest ratio of successful launches but only had 1 recorded launch
  - Booster v1.1 had the highest number of failed launches
  - Overall, Booster FT and B4 had the highest total number of successful launches between these payload masses

Section 5

Predictive Analysis (Classification)

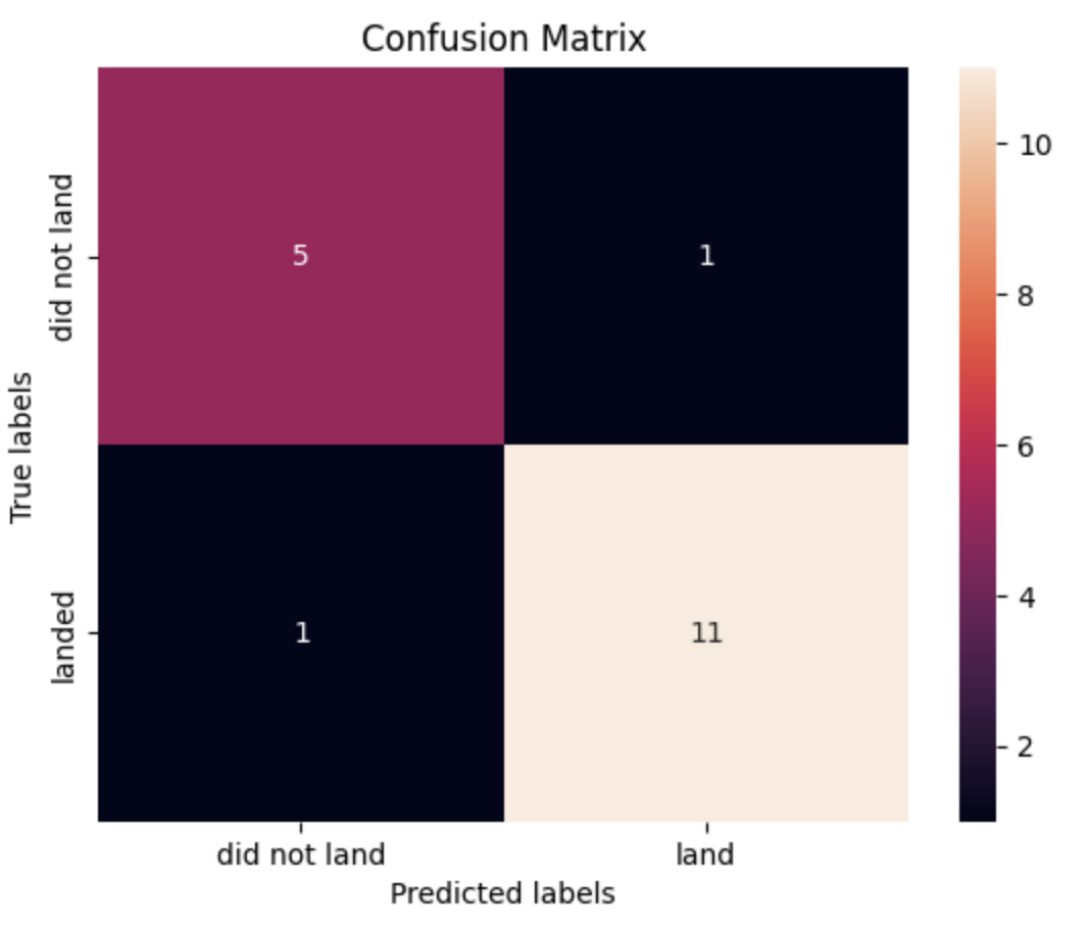
Classification Accuracy

- Out of all algorithms tested, 3 of them had identical performances
- The best performing algorithm was the Decision Tree Classifier at around 88% accuracy



Confusion Matrix

- Results of Decision Tree Confusion Matrix
  - Model only identified 2 labels wrong
  - A “landed” label identified as “did not land”
  - A “did not land” label identified as “landed”
- Overall, the model predicted the labels very well with minimal mistakes
- Can optimize model to have no “False Positive Rate” as a successful launch results in millions of dollars being used



Conclusions

- Success rates increased as number of flights increased. This may suggest that improved operational planning and technological improvements have increased the number of successful launches.
- Orbits of GEO, HEO, SSO, & ES-L1 had perfect successful launch rates of out all orbits
- Future launches may need to consider launch site KSC LC-39A as it had the highest level of successful launches
- The Decision Tree Classifier performed the best out of all algorithms. Future work may include using deep learning approaches to involve better understanding of nonlinear relationships

Thank you!