IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdullah Hel Asif
28/12/2023

Outline

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

Executive Summary

Summary of methodologies

- Data collection by web scraping and SpaceX REST API
- EDA using SQL and plotting graphs, Data Wrangling, Data Visualization (Static and Interactive)
- Building a Plotly Dashboard to analyze records
- Using Machine Learning to predict success rates

Summary of all results

- Able to collect public data using Web Scraping and from API
- Able to understand data from visualization techniques and have clean data from data wrangling
- Able to interact with all data from Dashboard and Interactive Maps
- Able to predict which features are responsible for success and which ML model is the best suit

Introduction

- The goal is to predict if the Falcon 9 will land successfully for company Y from the data of SpaceX.
- Find out which conditions or criteria are needed to make the new company successful

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collected from the SpaceX API: https://api.spacexdata.com/v4/rockets/
  - Collected from Wikipedia using Web Scraping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)
- Perform data wrangling
  - Handled missing values from the collected Data
  - Create new column to represent success or failure
- Perform exploratory data analysis (EDA) using visualization and SQL

Methodology

Executive Summary

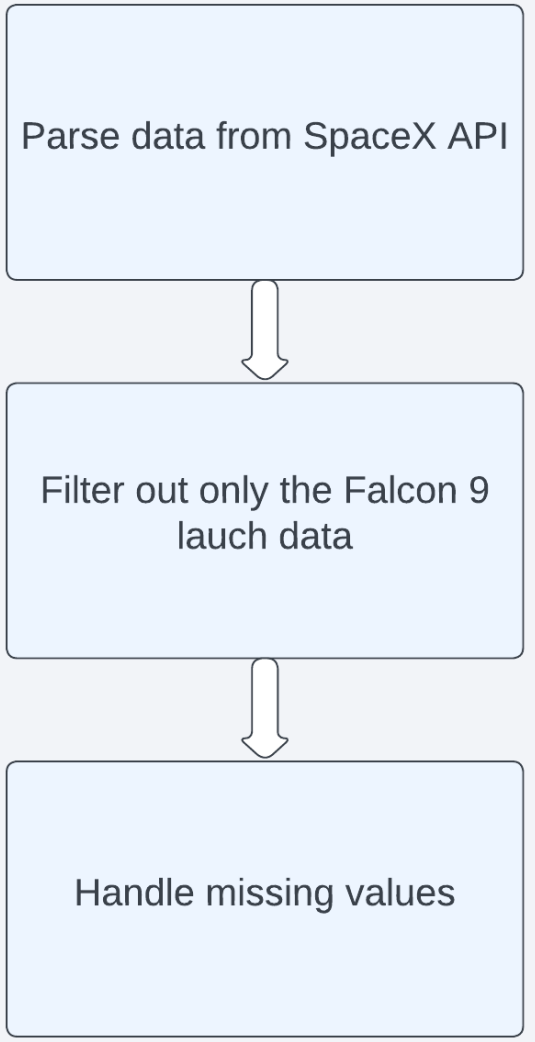
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Collected data was normalized
  - Split into training and testing set
  - Trained on 4 models
  - Evaluated each model and hyperparameters

Data Collection

- The data was collected from SpaceX REST API which is https://api.spacexdata.com/v4/rockets/
- The rest of the data was collected from Wikipedia using Web Scraping from this webpage https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

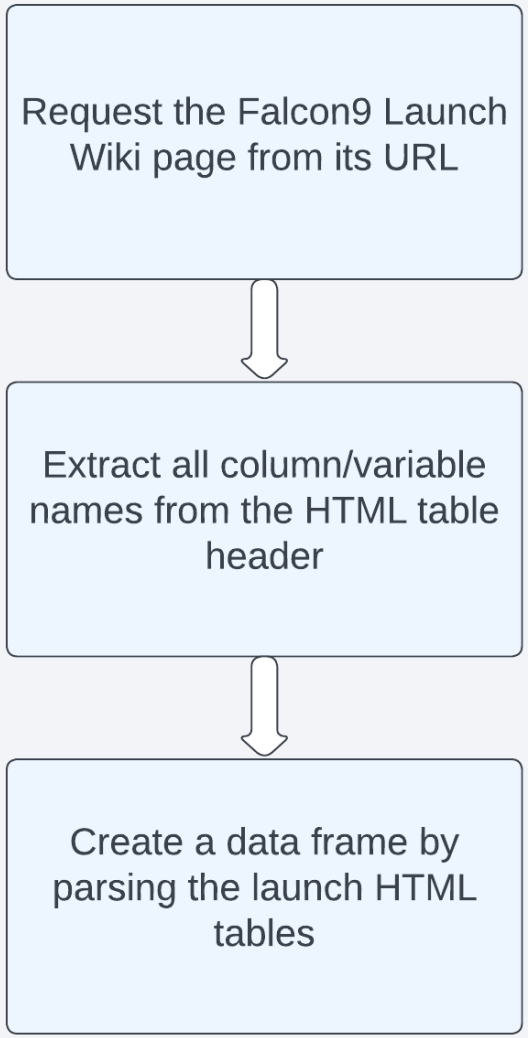
Data Collection – SpaceX API

- GET request to the API was made to parse the data
- Only Falcon 9 launch data was filtered
- Missing values were filled by the mean value of the column
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- HTTP GET was used to request the HTML data from the Wikipedia page
- All columns were collected from HTML table header
- A dictionary was made from the collected column names and then converted into a dataframe
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Firstly, Exploratory Data Analysis was done over the dataset.
- Launches per site, Occurrences of each orbit and Occurrences of mission outcome per orbit was derived from the EDA
- New column was made which represents the outcome of each launch
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Catplot was used to represent Flight Number and Payload Mass, visualize the relationship between Flight Number and Launch Site with the overlay of the outcome of the launch. Catplot is best to represent a relationship between categorical and numerical variables.
- Scatterplot was used to visualize the relationship between Payload and Launch Site and between Flight Number and Orbit type.
- Barplot was used to visualize the relationship between success rate of each orbit type,
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

SQL queries performed:

- The names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Display 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.
- The records of the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium Maps objects used: markers, circles, lines and marker clusters
  - Markers: Added to pinpoint specific locations or events on the map.
  - Circles: Used to indicate a specific radius or area around a point of interest.
  - Lines: Drawn to represent paths, routes, or connections between locations.
  - Marker Clusters: Employed to group nearby markers at high zoom levels for better visibility and organization.Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - Dropdown for Launch Site Selection:
  - Pie Chart (Success Rate):
  - Payload Range Slider:
  - Scatter Plot (Payload vs. Launch Success):
- Adding plots and interactions to a dashboard enhances data visualization, promotes user engagement through interactivity, and facilitates informed, data-driven decision-making by transforming complex information into intuitive visual representations.
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- 4 ML models: Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbors were used to predict the success outcomes
- Grid search was used to find out the best Hyperparameters
- Score method was used to evaluate the models on a test set
- Confusion Matrix was used to evaluate the test results
- Accuracy of each model was compared

Predictive Analysis (Classification)

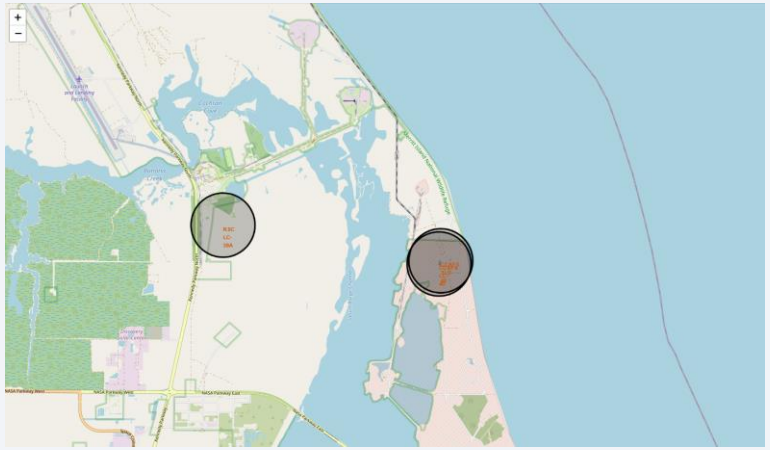
- https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

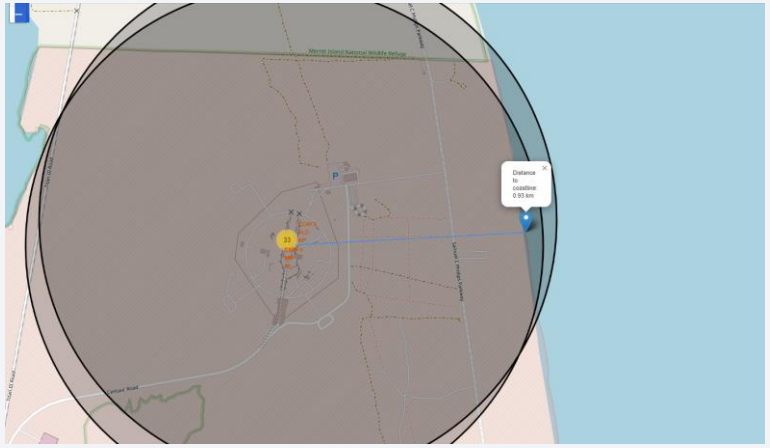
Results

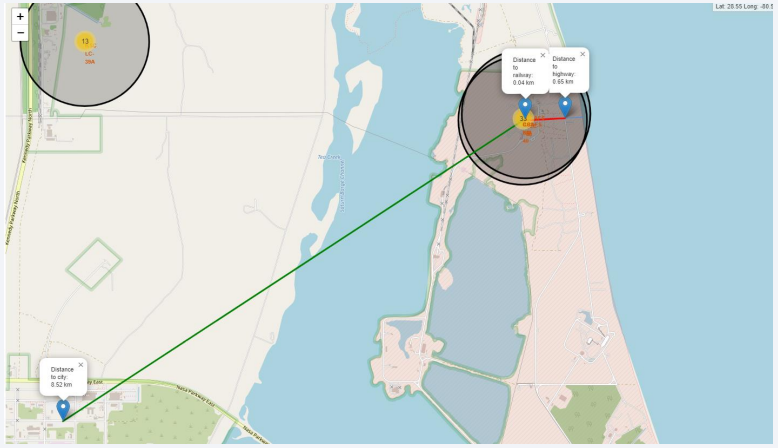
- Exploratory data analysis results:
  - 3 different launch sites have different success rates
  - For the VAFB-SLC launchsite, no rockets launched for heavy payload mass (greater than 10000)
  - 4 orbits have 100% success rate, 1 has 0% and others are between 50%-85% success rates
  - no relationship between flight number when in GTO orbit
  - With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS
  - The success rate since 2013 kept increasing till 2020
- Interactive analytics demo in screenshots
- Predictive analysis results

Results

- Interactive analytics demo in screenshots



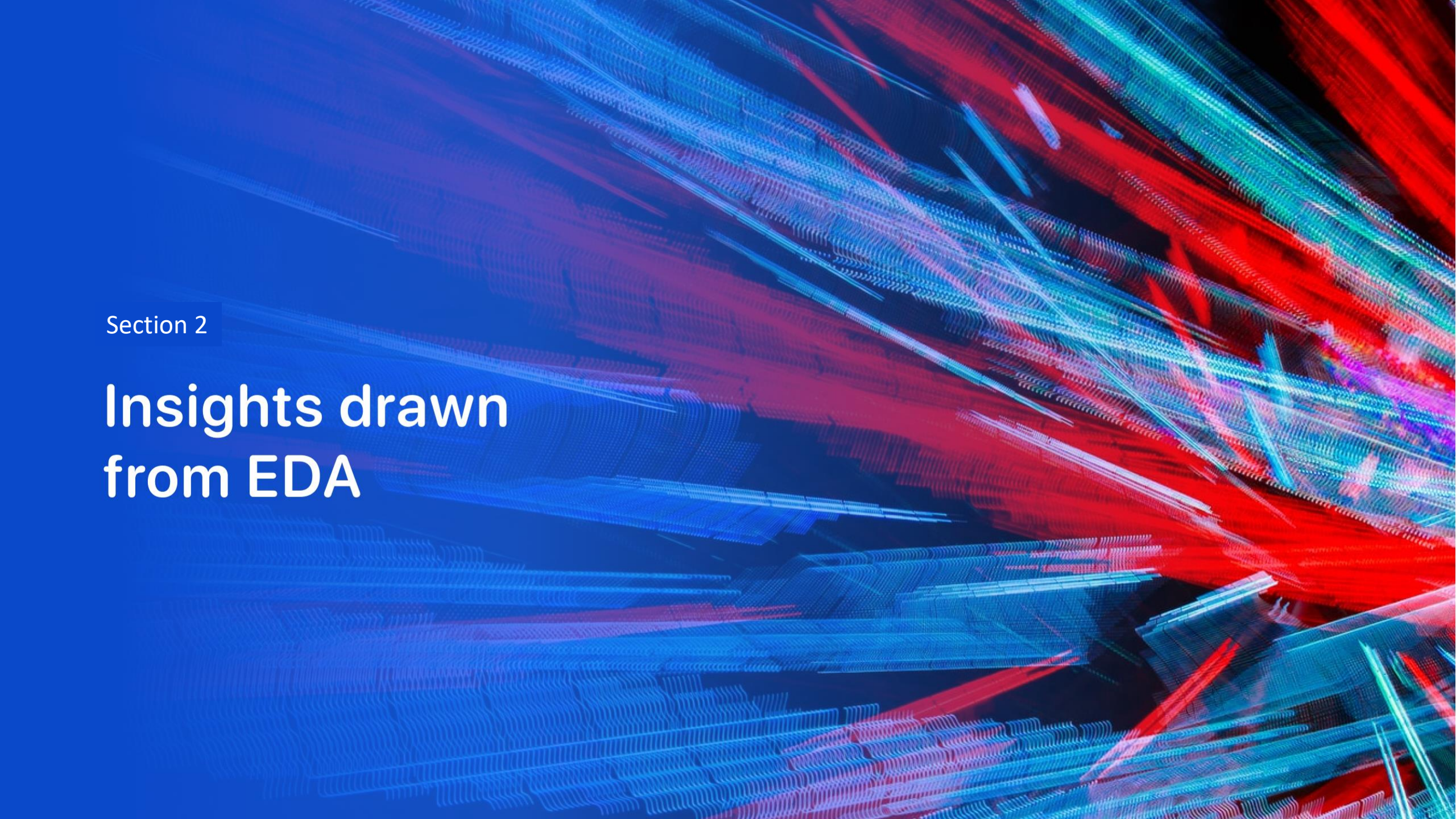




Results

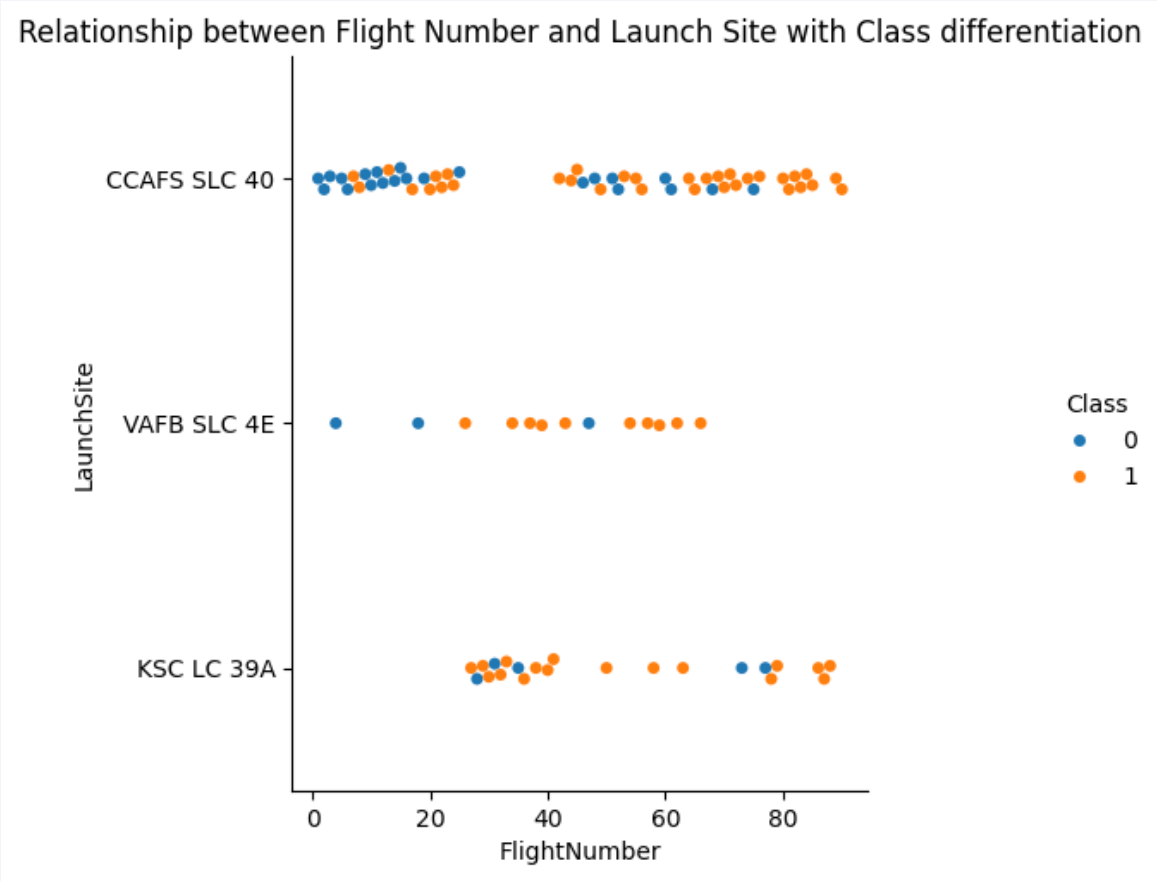
- Predictive analysis results conclude that Decision Tree Classifier was the best to use in this case.

Model	Train Accuracy	Test Accuracy
Logistic Regression	0.846	0.833
Support Vector Machines	0.848	0.833
Decision Tree	0.901	0.888
K-Nearest Neighbors	0.848	0.833


Section 2
Insights drawn from EDA

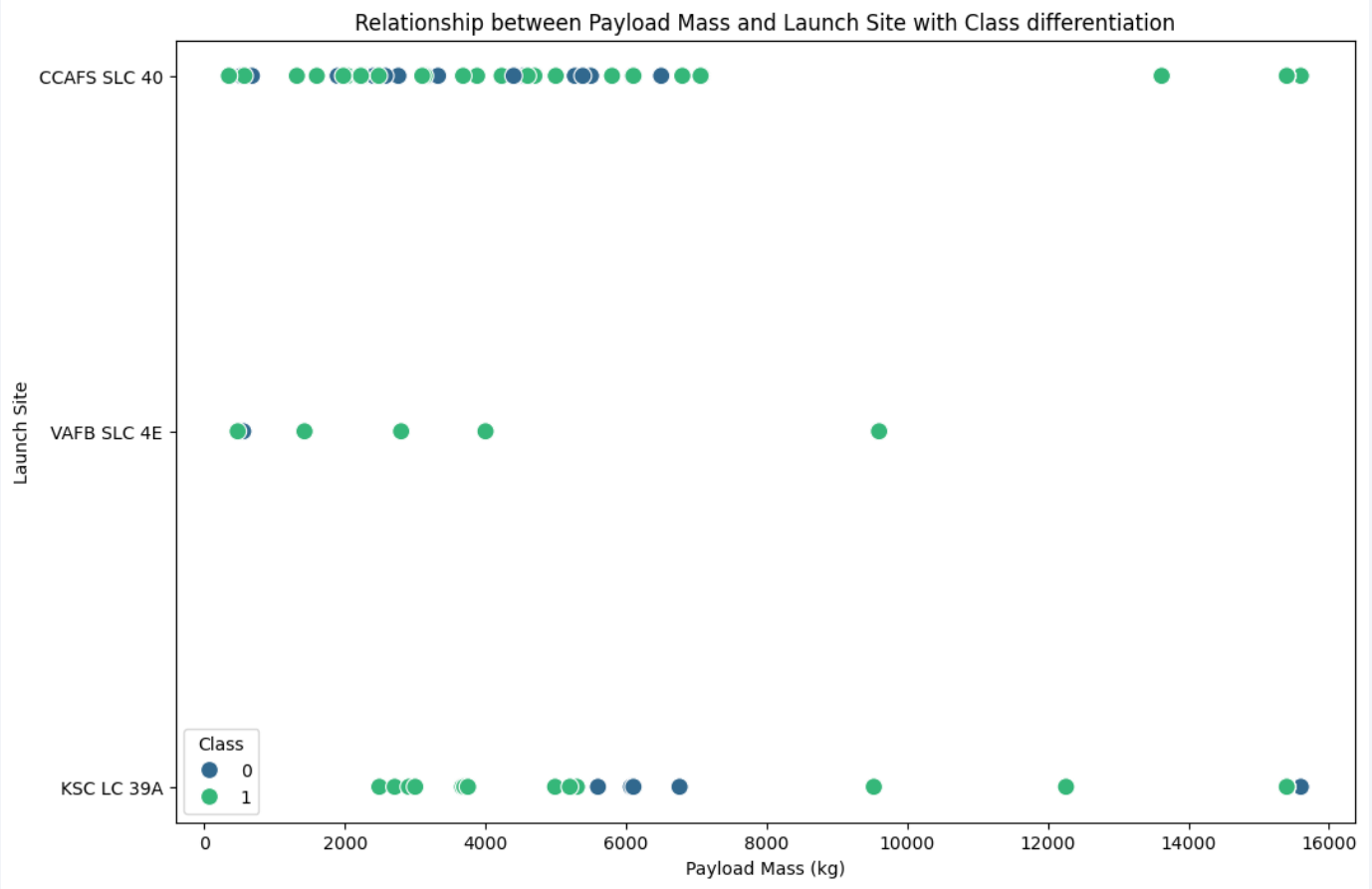
Flight Number vs. Launch Site

- The best launch site is CCAF5 SLC 40
- In second place VAFB SLC 4E and third place KSC LC 39A
- Success rate increased with flight numbers over time
- VAFB SLC 4E has the lowest flight numbers



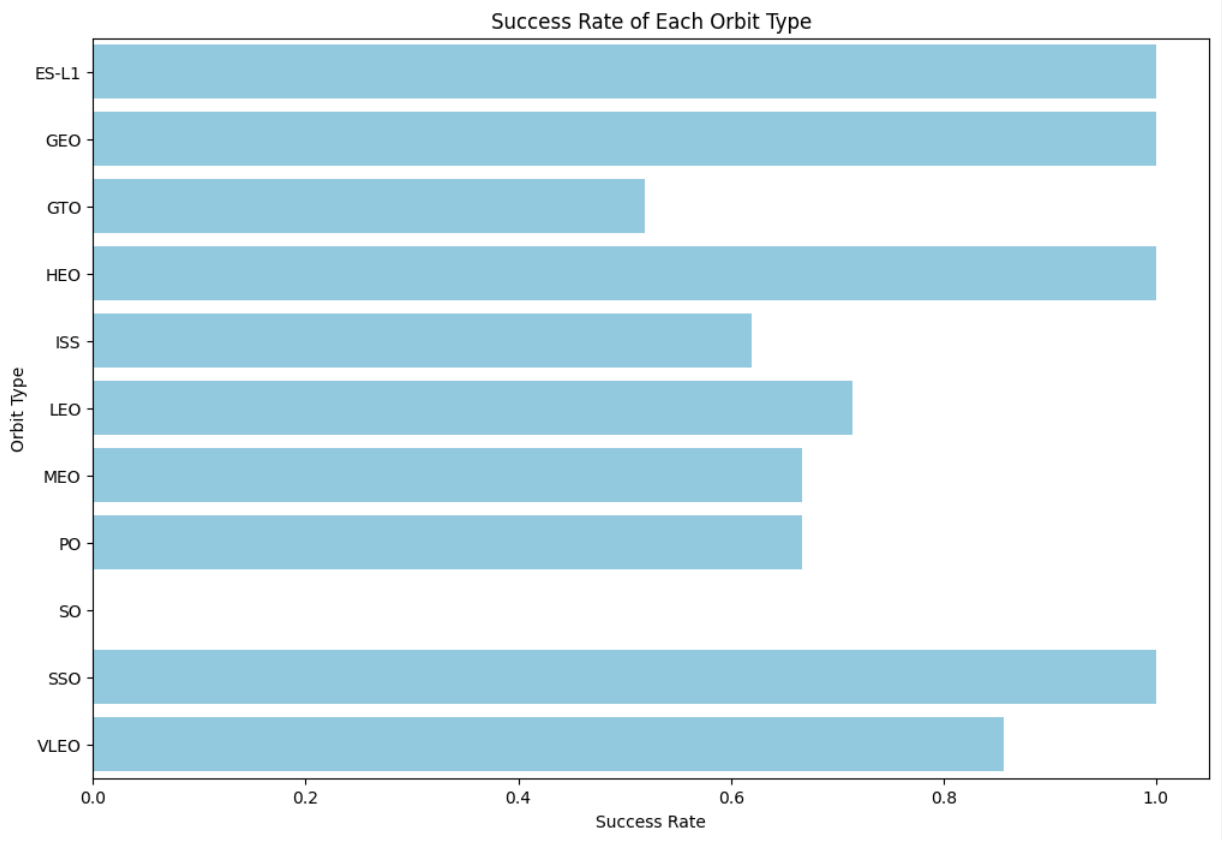
Payload vs. Launch Site

- CCAFS SLC 40 and KSC LC 39A can send payloads over 10000kg
- Payloads over 8000kg have higher success rates
- KSC LC 39A and VAFB SLC 4E have very high success rates for payloads below 6000kg



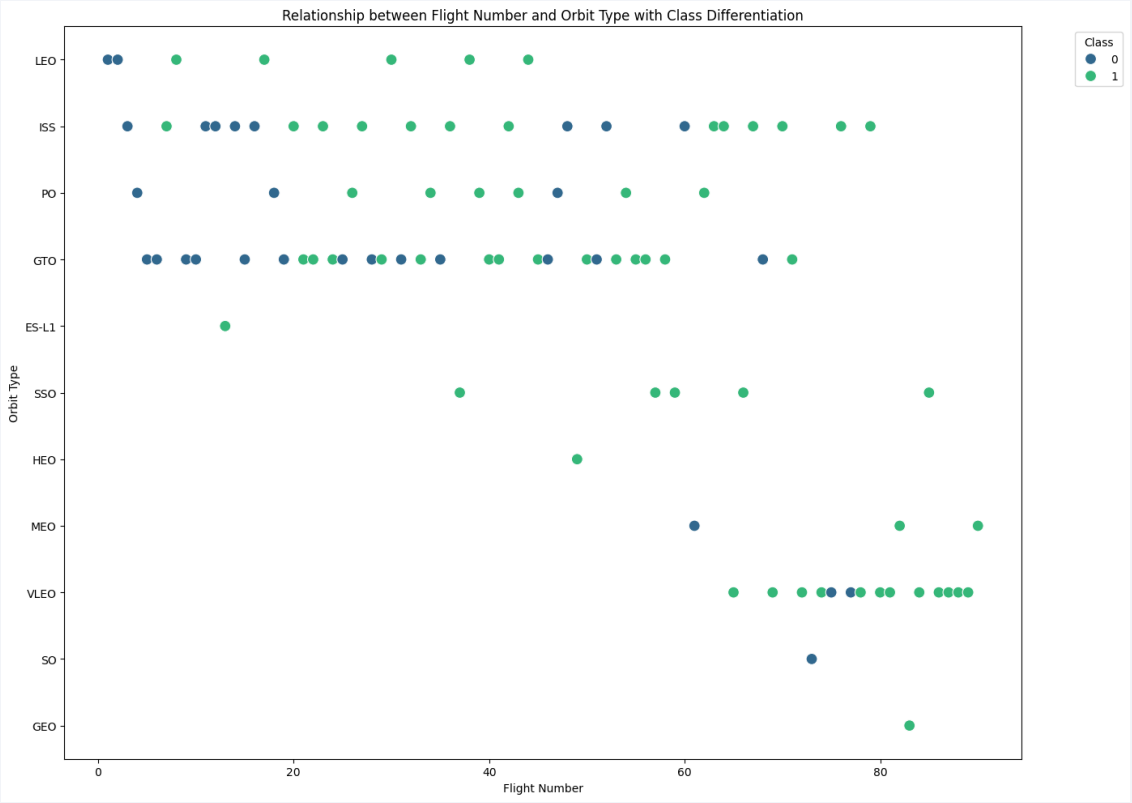
Success Rate vs. Orbit Type

- 4 orbits have 100% success rates: ES-L1, GEO, HEO and SSO
- SO has 0% success rate
- Other sites have success rates between 50% to 85%



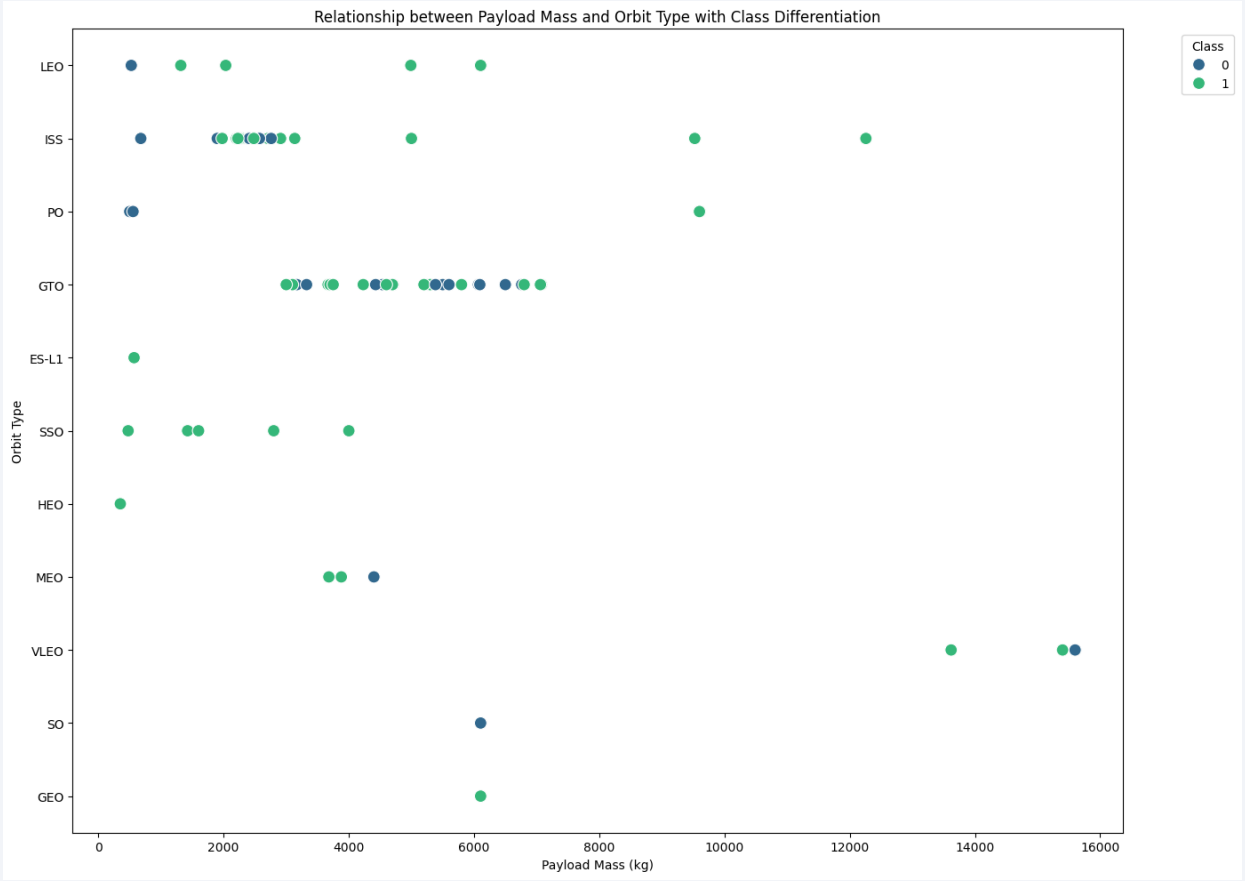
Flight Number vs. Orbit Type

- No certain relationships between flight numbers and orbit type
- Success rates increased as flight numbers increased
- Newer launches were in VLEO orbit
- ES-L1 and SO orbits are not preferred for some reason



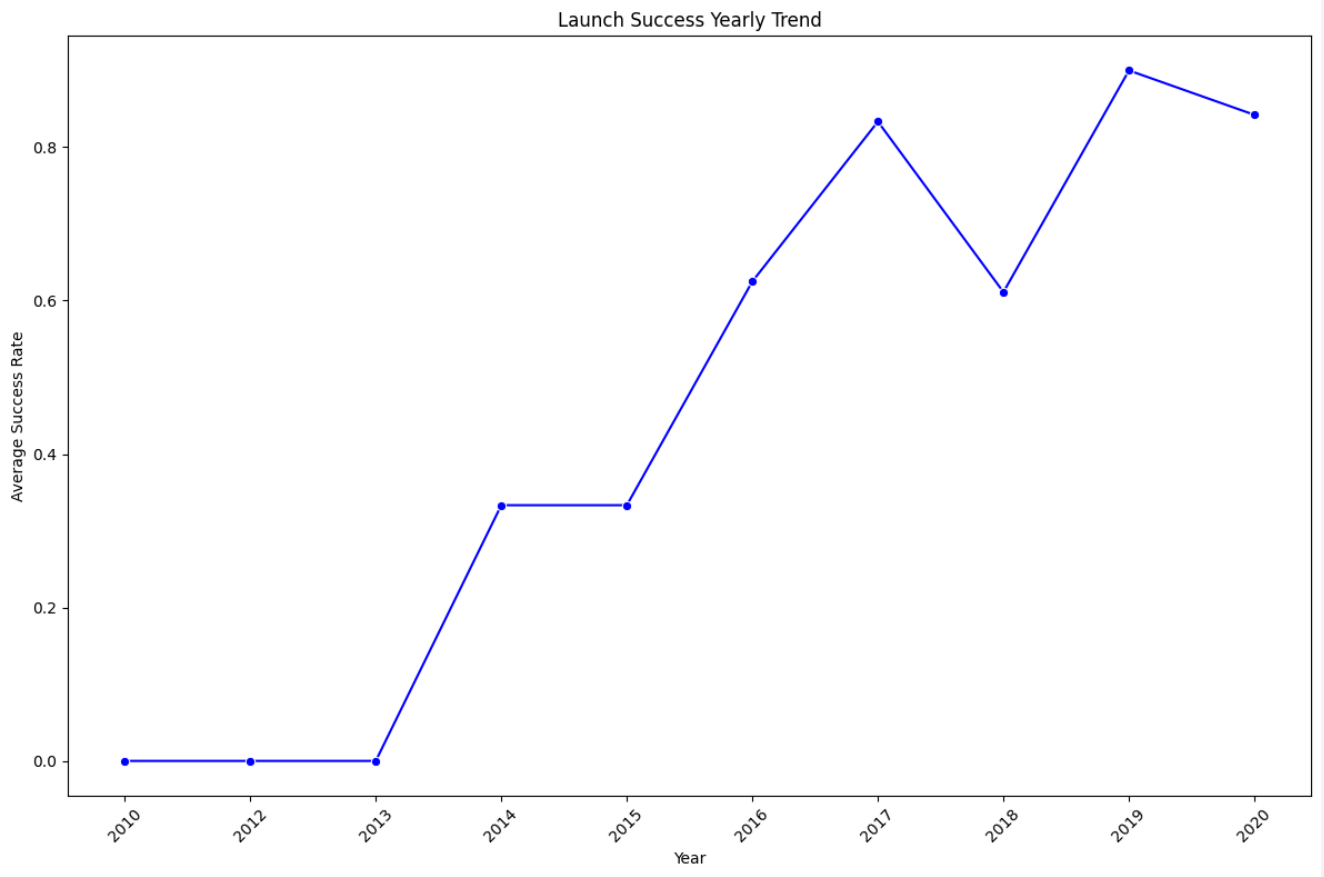
Payload vs. Orbit Type

- For heavy payloads the success rates are more for Polar, LEO and ISS.
- For GTO we cannot distinguish any kind of success/failure relations due to its high frequency
- ISS is most preferred for all kinds of payload mass
- GTO has no payload of higher mass (above 8000kg)



Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020
- The success rate had a drop between 2017-2018
- The success rate had a high boost between 2013-2017
- The first 3 years were mostly trials and tests which might be the reason of low success rate



All Launch Site Names

- Query: SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
- Explanation: Lists distinct launch sites from the

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

Launch Site Names Begin with 'CCA'

- Query: SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
- Explanation: Retrieves the first 5 records with launch sites starting 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

Total Payload Mass

- Query: SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass_kg FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)'
- Explanation: Calculates the sum of payload mass for launches by NASA (CRS).

Total_Payload_Mass_kg
45596

Average Payload Mass by F9 v1.1

- Query: SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass_kg FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1'
- Explanation: Computes the average payload mass for booster version F9 v1.1.

Average_Payload_Mass_kg
2928.4

First Successful Ground Landing Date

- Query: SELECT MIN("Date") AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success'
- Explanation: Determines the earliest date when a successful landing on the ground pad occurred.

First_Successful_Landing_Date
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query: SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success' AND "Orbit" = 'GTO' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000
- Explanation: Lists booster versions that successfully landed on a drone ship with payload mass between 4000 and 6000 kg in GTO orbit.

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1058.2

Total Number of Successful and Failure Mission Outcomes

- Query: SELECT "Mission_Outcome", COUNT(*) AS Total_Outcomes FROM SPACEXTABLE GROUP BY "Mission_Outcome"
- Explanation: Counts the number of missions categorized by their outcomes, either successful or failed

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

Boosters Carried Maximum Payload

- Query: SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
- Explanation: Uses a subquery to find booster versions that carried the maximum payload mass.

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

2015 Launch Records

- Query: SELECT substr("Date", 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" LIKE '%Failure%' AND "Landing_Outcome" LIKE '%drone ship%'
- Explanation: Retrieves records for the year 2015 that have a landing outcome of failure on a drone ship, displaying month, landing outcome, booster version, and launch site.

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

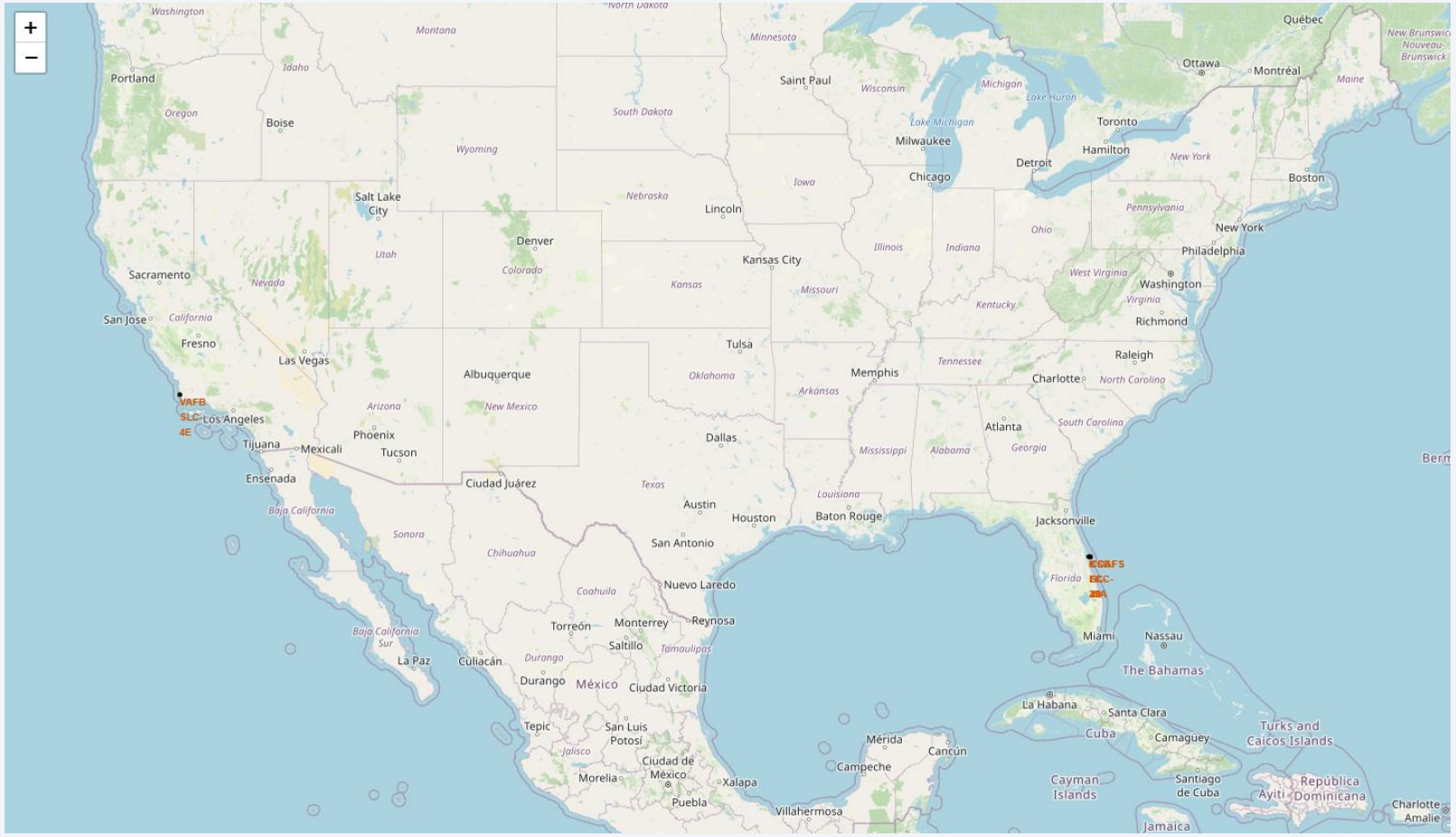
- Query: SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Outcome_Count DESC
- Explanation: Ranks the count of landing outcomes between specific dates in descending order, showcasing outcomes like failure on a drone ship or success on a ground pad.

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

Section 3

Launch Sites Proximities Analysis

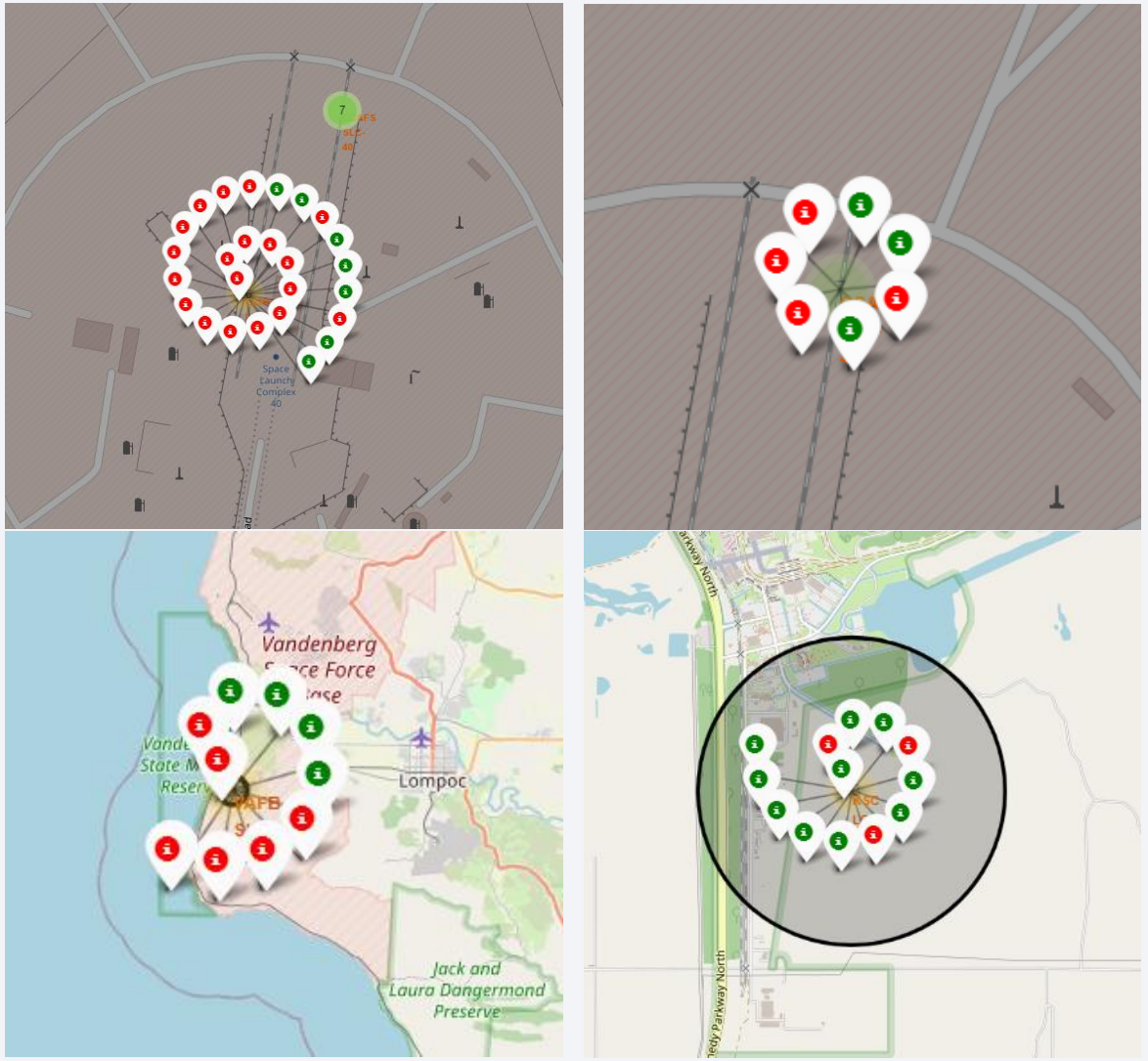
All Launch Sites of SpaceX Falcon 9



- All launch sites are near the sea

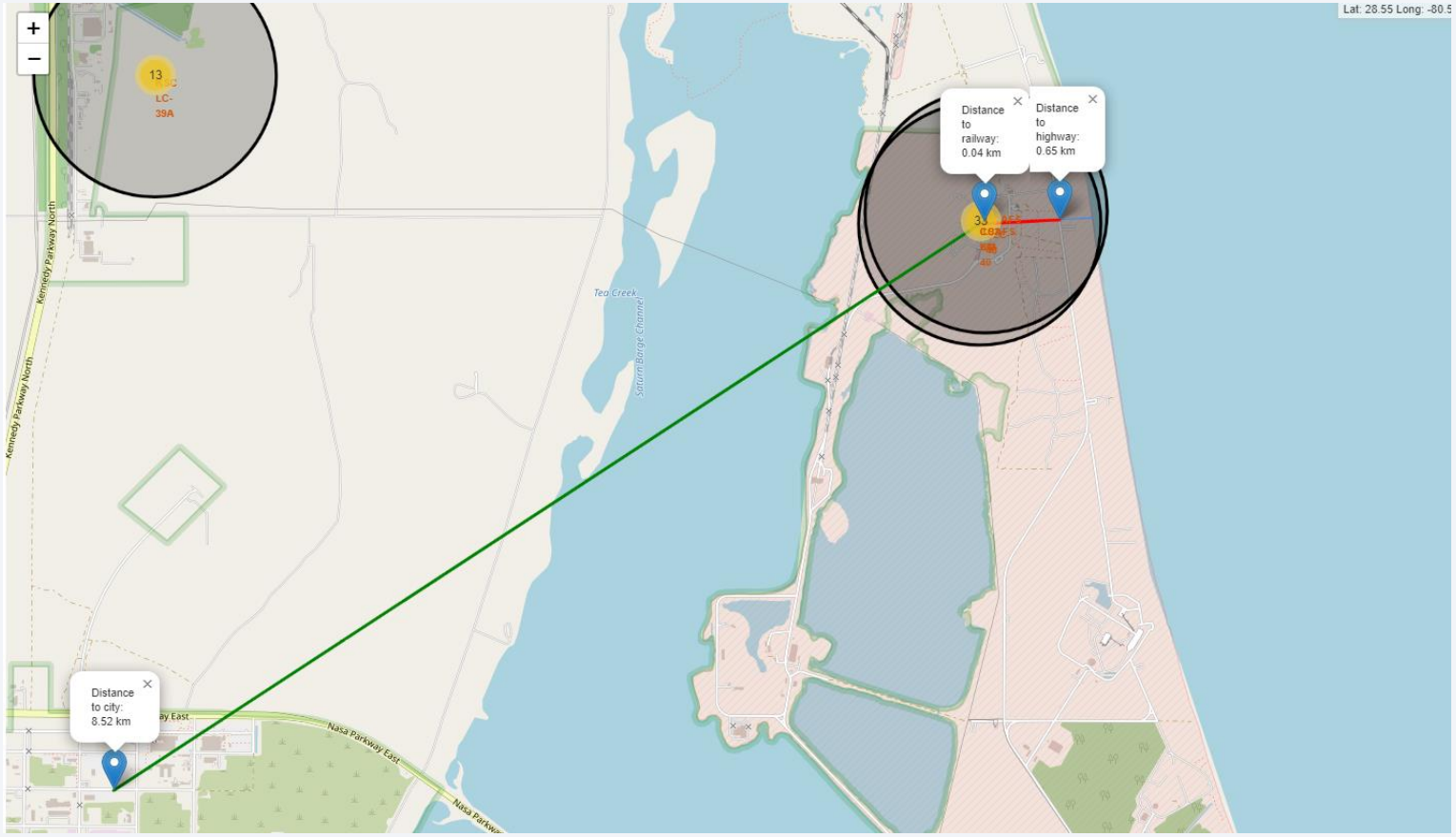
Launch outcomes of all sites

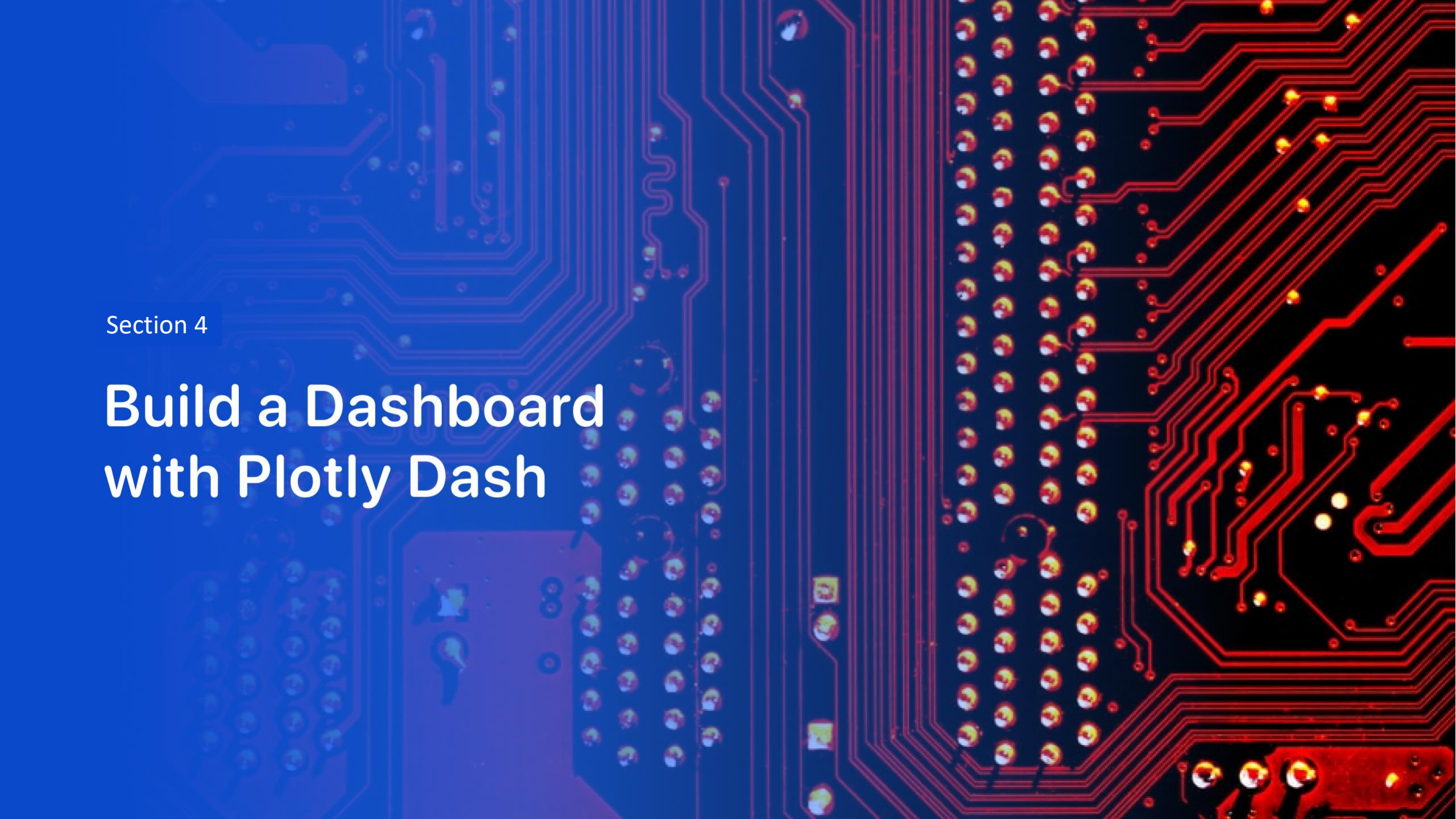
- Green markers show successful landings
- Red markers show failure landings



Distance of a close proximities for a launch site

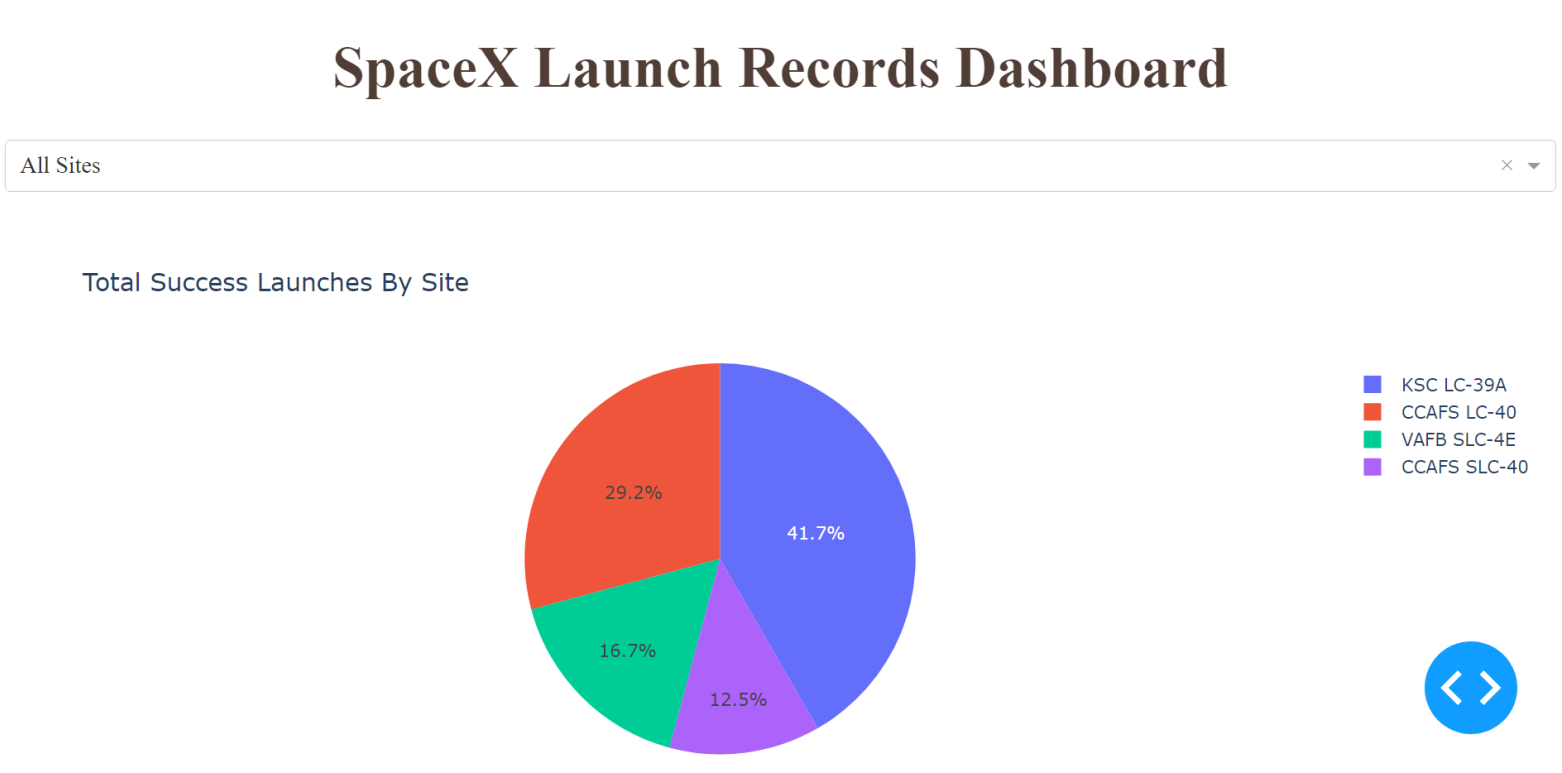
- Launch site is pretty farm from cities (For safety)
- Launch site is close to coastline (For safety)
- Launch site is close to railways (To move goods)




Section 4
Build a Dashboard with Plotly Dash

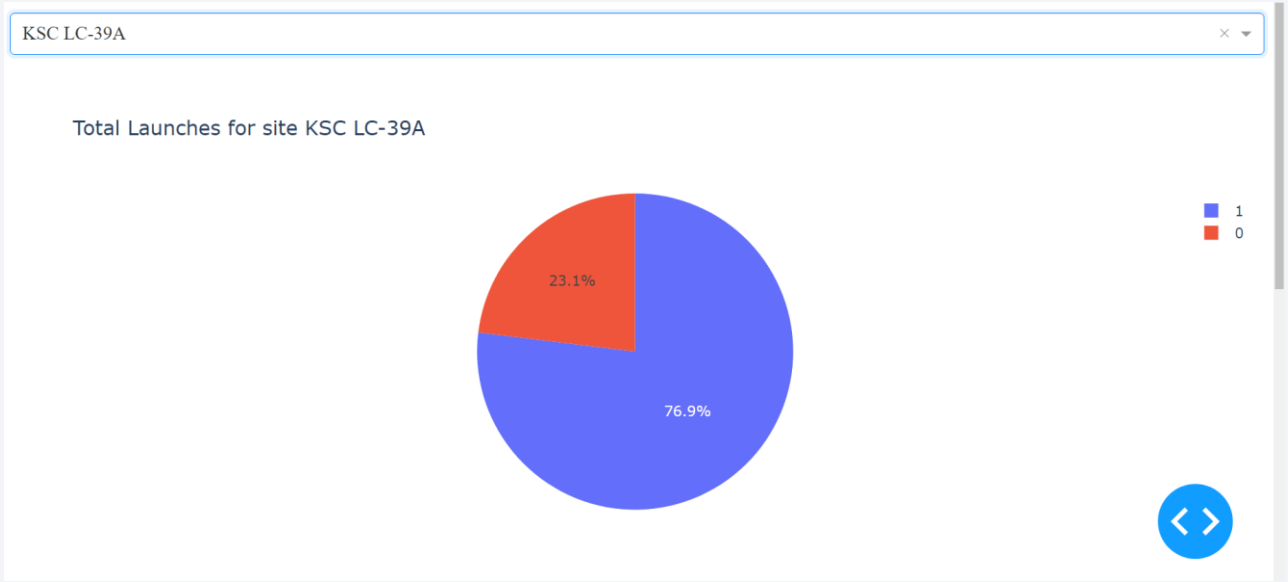
Piechart of successful launches for all sites

- KSC LC-39A is the most successful launch site
- CCAFS SLC-40 is the least successful launch site



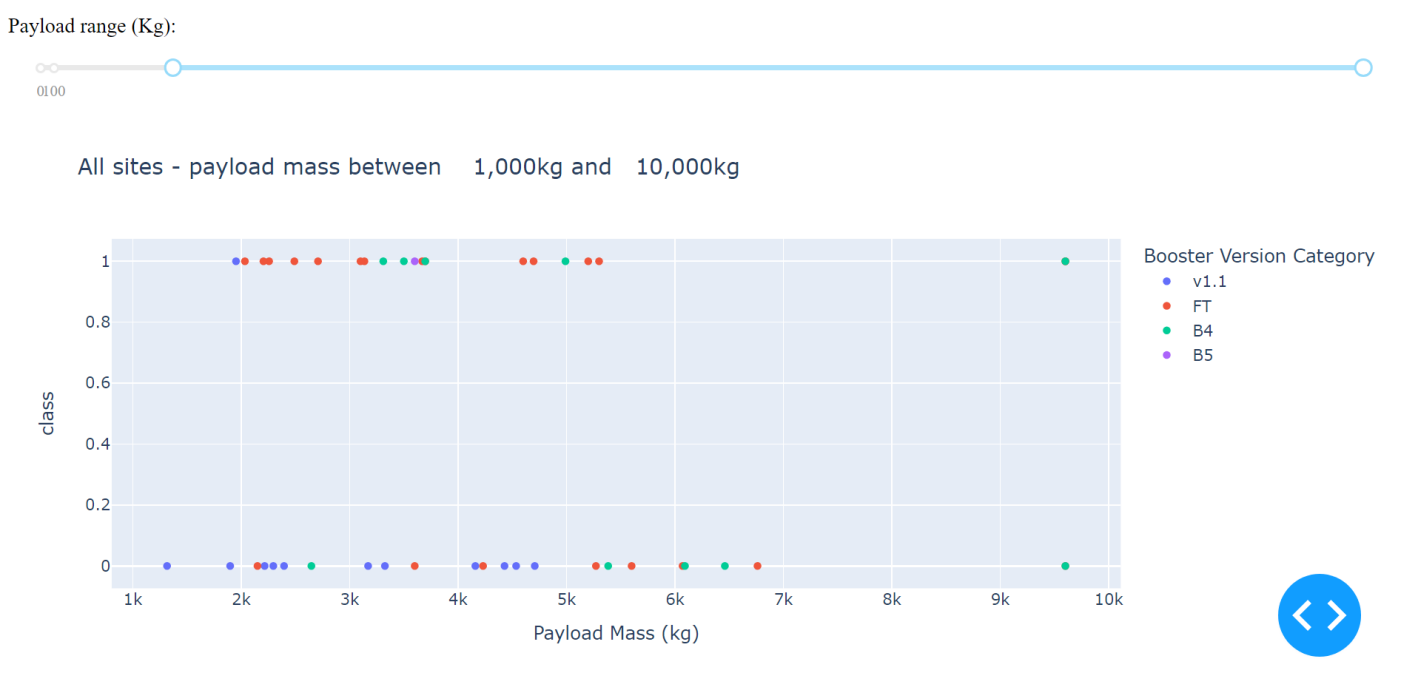
Success ration for launches of site KSC LC-39A

- More than three-fourth of the launches are successful for this site
- The exact percentage is 76.9%



Payload vs Launch Outcomes

- FT boosters are most successful
- V1.1 boosters are least successful

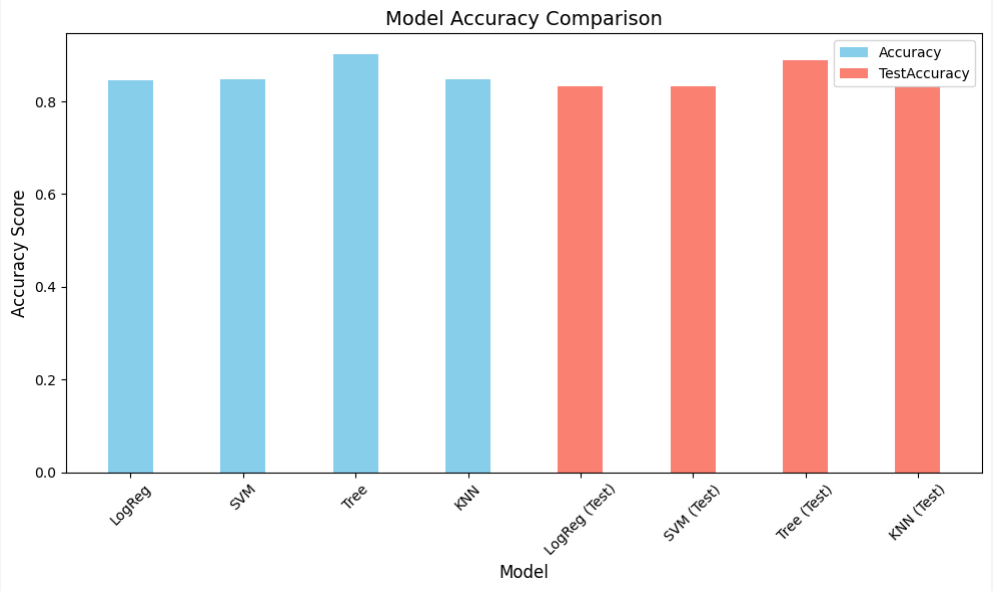


Section 5

Predictive Analysis (Classification)

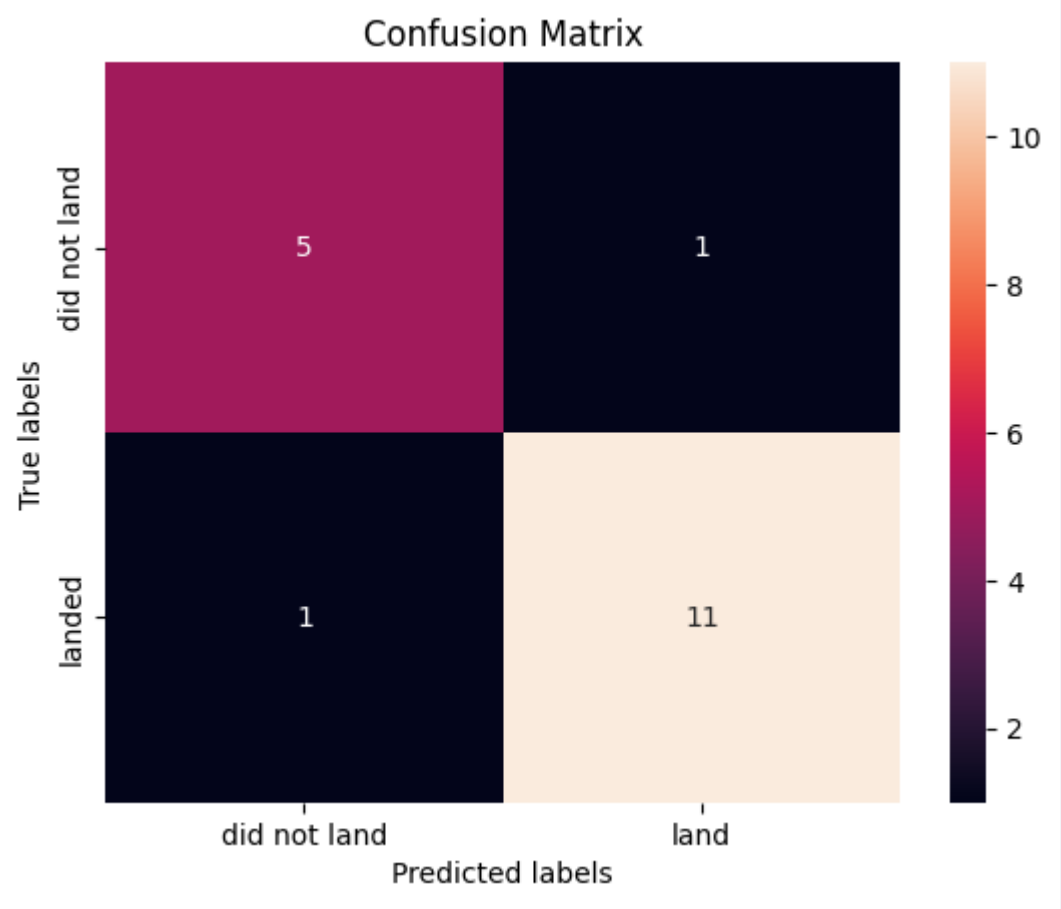
Classification Accuracy

- Decision Tree has the highest accuracy for both training and testing set
- It has accuracy of almost 88% on testing set



Confusion Matrix

- Out of 18 test samples 16 True Positive and True Negative predictions were made
- Out of 18 test samples only 2 False Positive and False Negative predictions were made
- Approximately 90% True predictions and 10% False predictions



Conclusions

- Data was collected from two different sources
- Out of 4 launch sites, the best one was KSC LC-39A
- Launches below 8000kg are less successful which indicates heavy payloads are more successful
- Success for landings increases as flight numbers increased
- Launch sites are near coastal lines and farther from cities for safety purposes
- Decision Tree was the best model to predict the success of the launches

Appendix

- For bar charts of accuracy, instead of running the whole notebook, lists of accuracy scores were made from previous results.
- Github repo: https://github.com/blackbolttt/Applied-Data-Science-Capstone-Coursera

```
# Data
models = ['LogReg', 'SVM', 'Tree', 'KNN']
accuracy = [0.8464, 0.8482, 0.9018, 0.8482]
test_accuracy = [0.8333, 0.8333, 0.8888, 0.8333]

# Create bar plots
fig, ax = plt.subplots(figsize=(10, 6))

# Plotting bars for Accuracy
ax.bar(models, accuracy, width=0.4, label='Accuracy', align='center', color='skyblue')

# Plotting bars for TestAccuracy
ax.bar([model + ' (Test)' for model in models], test_accuracy, width=0.4, label='TestAccuracy', align='center', color='salmon')
```

Thank you!