IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anod Pun
29/04/25

Outline

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

Executive Summary

Summary of Methodologies

- Data collection via SpaceX API and web scraping Wikipedia.
- Data Wrangling and data cleaning was performed. Filtering for Falcon 9 missions, and engineered features using one-hot encoding.
- Exploratory Data Analysis(EDA) was performed using SQL queries and interactive visualisations through Folium and Plotly Dashboard.
- Predictive Modelling and analysis was then applied by training four classification models – Logistic Regression, SVM, Decision Tree and KNN – to predict landing success.

Summary of Results

- KSC LC-39A had the highest launch success rate.
- Payloads between 2000-4000 kg had the best success rates.
- F9 B5 boosters were the strongest performers.
- Our Logistic Regression model performed best on the test set, recording an accuracy of 0.833.

Introduction

Project background and context

With this report we are analyzing historical SpaceX launch data and using it to predict the likelihood of SpaceX's Falcon 9 rocket's first stage successfully landing after a launch. Our goal is to identify key influencing factors and provide data-driven insights that support better decision-making and predict modeling for future launches.

Specifically, in this project we aim to answer the following questions:

- "Which launch sites have the highest success rates?"
- "What payload ranges are associated with higher or lower success rates?"
- "Which booster versions perform best?"
- "Can we build a machine learning model to predict the outcome of a launch?"

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We first gathered the launch data from SpaceX's Public API. This collection contained structured information on missions, launch sites, booster versions and payloads.
  - We then gathered additional information on launch outcomes from SpaceX's Wikipedia page using BeautifulSoup and requests to web scrape the page.
- Perform data wrangling
  - We cleaned and filtered the data to include only Falcon 9 launches. Missing values and irrelevant columns were omitted. Also applied one-hot encoding to the categorical variables.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We trained classification models to predict launch success, using key features such as site, payload and booster version.

Data Collection

How Data Sets Were Collected

To build a comprehensive dataset for analysis, we gathered launch data from two main sources: the official SpaceX API and supplementary information was gathered from Wikipedia using web scraping.

SpaceX API Data Collection

- First we sent a GET request to a static JSON URL that simulated the SpaceX REST API response
- We then parsed and flattened the JSON repsonse using pd.json_normalize() to handle the nested structure of the data
- Next we filtered for relevant columns, cleaned the entries, and then stored the results in a Pandas dataframe for further analysis

Wikipedia Web Scraping

- First we used the requests library to retrieve the HTML content of the SpaceX launch history page from Wikipedia
- We then parsed the page using BeautifulSoup to locate the first HTML table that contained Falcon 9 launch records
- Next we looped through the rows of the parsed table to extract key variables such as booster version, payload mass, orbit, launch site, and landing outcome
- Finally the extracted data was stored in a python list and then organised into an empty dictionary, then converted into a Pandas dataframe. This data was then merged with the API dataset to enrich the launch records with landing outcomes and additional booster details.

Data Collection – SpaceX API

Flowchart of SpaceX REST API Data collection

API URL
- Send a GET request to SpaceX API using:
- **requests.get()**

pd.json_normalize
- Convert the nested JSON file to a flat table using:
- pd.json_normalize()

Create DataFrame
- Store the result in a usable pandas dataframe

GitHub URL

https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api(3).ipynb

Data Collection - Scraping

Flowchart of web scraping process

Send HTTP Request
- Use request.get() to fetch HTML content required from the SpaceX Wikipedia page

Parse HTML
- Use BeautifulSoup to parse the HTML content

Locate Data Table
- Targetted the first HTML table to extract the launch records
- soup.find_all('table)[0]

Extract Data
- Iterated throught the rows of the table, pulling all relevant cells(e.g. launch site, payload, booster version)
- Stored extracted rows into a list of lists

Build DataFrame
- Converted list of rows into a pandas DataFrame with appropriate column headers

GitHub URL

https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping(1).ipynb

Data Wrangling

How the Data Was Processed

Clean Data

- The dataset was first inspected for missing values
- Then we reviewed datatypes of each column
- Next we filtered for single-core, single-payload missions

Calculate Launch Counts

- We used value_counts() to count launches per site and orbit

Feature Engineering

- We created a Landing Outcome Label
- Went on to define the landing_class column using a conditional:
- 0 if it's a bad_outcome
- 1 otherwise
- We then assigned it to a new column df['Class']

Calculate the average launch success rate

- To calculate the overall success rate we used:
- df["Class"].mean()

- https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling(1).ipynb

Summary of EDA with Data Visualization

We used a variety of visualisations to explore potential relationships between mission features and launch outcomes:

Scatter plots:

- *Payload vs Flight Number*: We used this scatter plot to explore how launch experience and payload size relate to success.
- *Flight Number vs Launch Site:* This scatter plot was created to check if success rates improved over time at different locations.
- *Payload Mass vs Launch Site:* Used to examine if certain sites handled heavier missions more reliably than others.
- *Flight Number vs Orbit Type:* We used this visualisation to assess whether experience influences success differently across orbit types.
- *Payload vs Orbit Type:* We explored this relationship to investigate whether payload size affected success in different orbits.

Bar Chart:

- *Success Rate by Orbit Type:* We used this bar chart to compare success rates across orbit types and identify reliable vs risky orbits

Line Chart:

- *Yearly Success Trend* – with this visualisation we wanted to see how the average lauch success changed over time

GitHub URL

https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/edadataviz(1).ipynb

EDA with SQL

Summary of SQL queries performed

- We first created a filtered table named 'SPACEXTABLE'. Rows with null 'Date' values were excluded.
- Next we identified the unique launch sites from SPACEXTABLE. This helped us discove where SpaceX has been launched from.
- The first 5 launches from sites starting with 'CCA" were then selected, allowing us to quickly preview some of their launch activity.
- Total payoad fot NASA(CRS) was then calculated in order to measure the the total mass deliverd by missions flown for NASA(CRS).
- We then calculated the average payload for launches using the 'F9 v1.1' booster version.
- Next, the date of the first successful ground pad landing was queries, giving us a milestone marker in SpaceX's reusability journey.
- We also queried for launches to find which booster versions landed on a drone ship while carrying payloads between 4001 – 5999 kg.
- To get a grasp of overall mission performance, we summarised all mission outcomes, grouping them by success and failure.
- We then identified the booster and payload mass for SpaceX's heaviest mission, highlighting its maximum launch capability.
- All failed drone ship landings in 2015 were queried, noting the launch month, booster version and launch site.
- Finally, we grouped outcomes over a specific date range allowing us to obserce how success rates varied over time.

GitHub URL: https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite(1).ipynb

Build an Interactive Map with Folium

Summary of map objects created and their purpose

- Circle markers were created to provide viewers with a quick visual reference for where each site is on the map
- Labelled text markers(DivIcons) were created to display the site names directly on map. Helping the map become more readable and allowing viewers to clearly identify each site location by name.
- Outcome markers we also added, which were colour-coded by success(green) and failure(red). By visualising individual mission outcomes, we helped reval spatial patterns in performance.
- We then used a marker clustering feature, which grouped nearby points when zoomed out and helped reduce clutter when displaying a large number of markers.
- A custom distance marker was added, to show us the distance between a launch site and its nearest coastline.
- The distance was then displayed by creating a labelled marker and a line(polyline) connecting the two points. This helped us visually representt how far inland each launch site was located.

- GitHub URL

- https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location(1).ipynb

Build a Dashboard with Plotly Dash

Summary of what plots/graphs and interactions were added to a dashboard

- We added a dropdown menu to allow users to filter results by launch site. This allowed for focused analysis of site-specific performance
- We created a pie chart to help show total successful launches across all sites. When a single site is selected, it switches to show success vs failure counts for the selected location. This interaction helps compare overall performance and reliability by site.
- A payload range slider was also included to filter the data based on payload mass. This gave the users flexibility to examine how payload weight effects launch outcomes.
- We also used a scatter plot to show the correlation between payload mass and mission success. The points were colour coded by booster version, helping reveal how different booster types performed across payload ranges.

GitHub URL

- https://github.com/anoddy/Coursera-Applied-Data-Science-Capstone/blob/main/Build%20an%20Interactive%20Dashboard%20with%20Plotly%20Dash.md

Predictive Analysis (Classification)

Summary: Building, Evaluating and Finding the Best Classifier

Building

- Data Preparation
  - Standardized features using:
  - Preprocessing.StandardScaler()
  - Data set was then split into training and testing sets
- Model Training
  - Classifiers chosen to be trained were:
  - LogisticRegression
  - Support Vector Machine(SVC)
  - DecistionTreeClassifier
  - KNeighboursClassifier

Evaluating

- Hyperparameter Tuning
  - Used GridSearchCV to tune the parameters for each model
- Evaluated models on test set using:
  - Model.score(X_test, Y_test) for accuracy scores
  - Produced a confusion_matrix to visualise prediction performance

Selecting

- Models were compared using their test set accuracy:
- .score(X_test, Y_test)
- The model with the highest test accuracy was then chosen.

Results

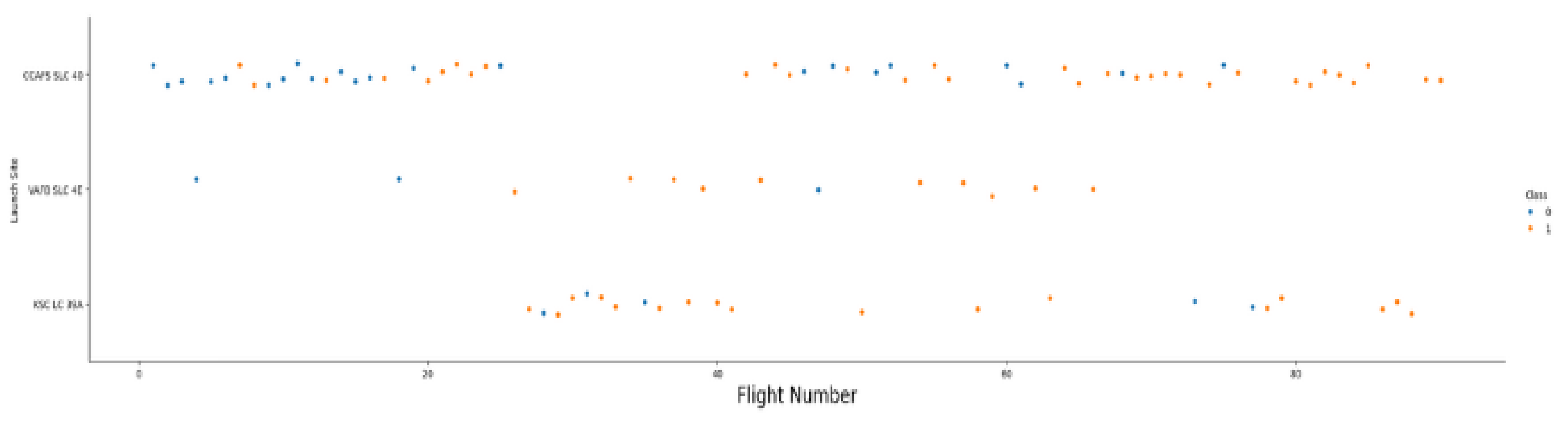
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

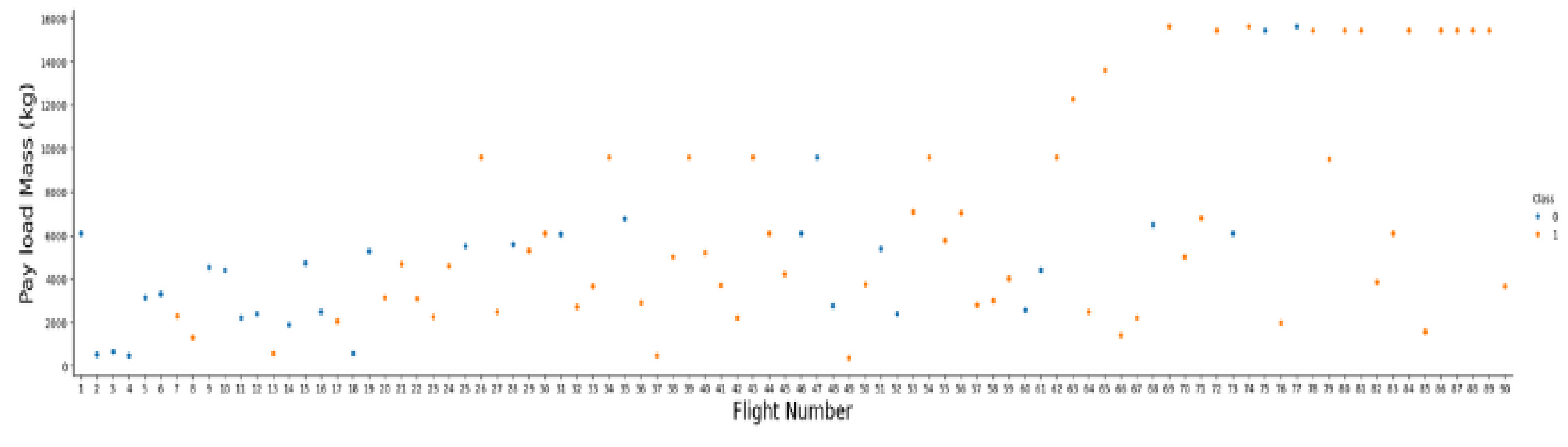
Flight Number vs. Launch Site Scatter Plot



With this scatter plot, we examined if there was any relationship between mission success and launch site. The plot shows us how frequently each site was used and if those launches were successful or not. The plot shows that launch site CCAFS SLC 40, not only had more launches but also showed a higher concentraion of successful missions.

Payload vs. Launch Site

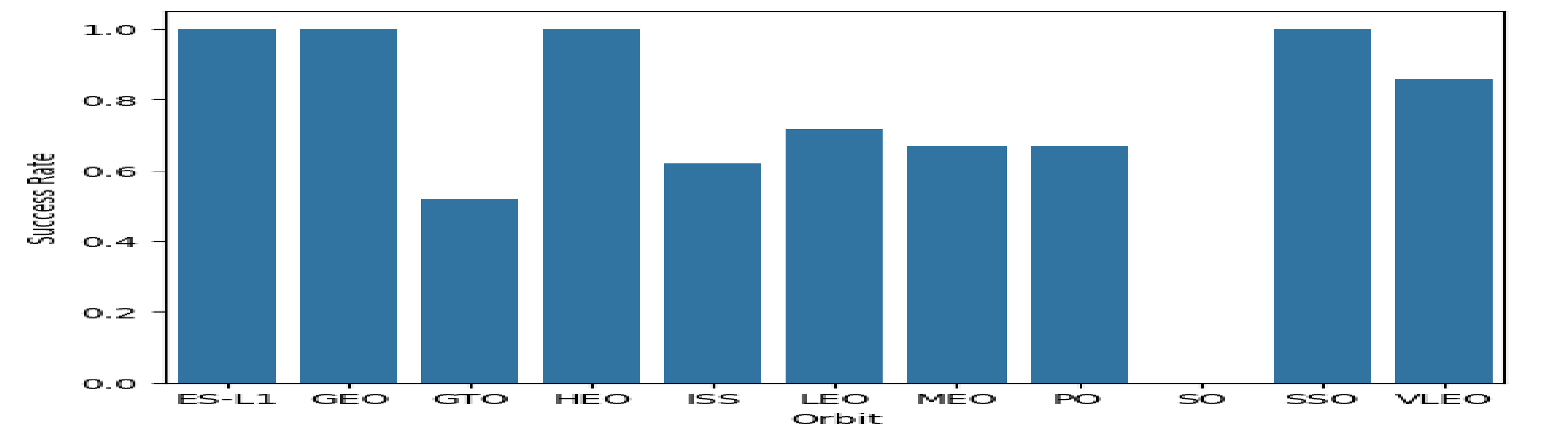
Payload vs. Launch Site Scatter Plot



We created this scatter plot to explore the relationsip between flight experience and mission success. It revealed that as flight numbers increased, the likelihood of a successful landing also improved. We also observed that higher payload masses did not appear to reduce success rates – suggesting that there is increased reliability over time, even for heavier missions.

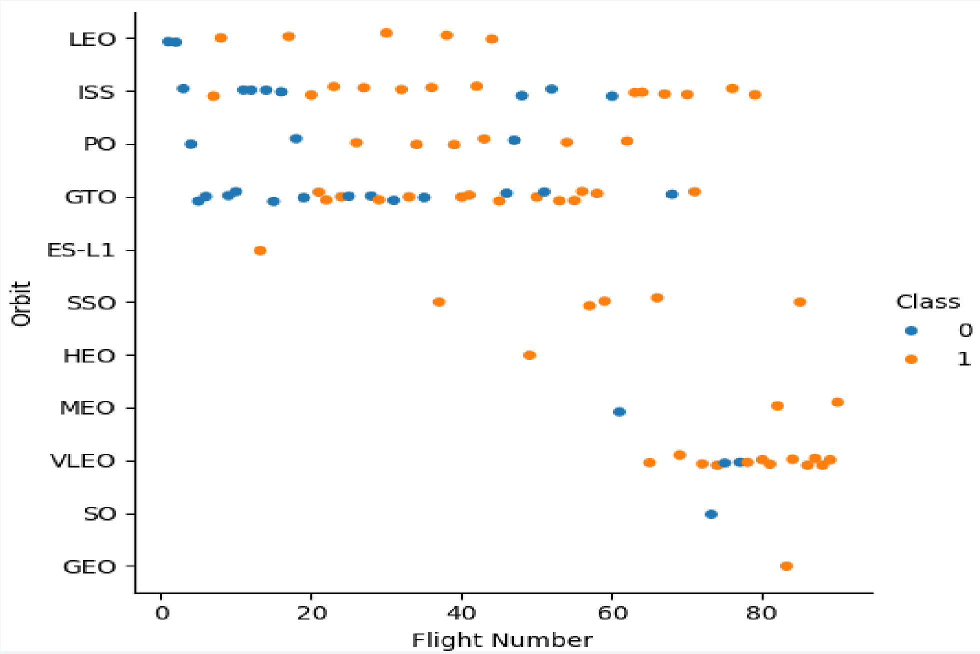
Success Rate vs. Orbit Type

Success rate of each orbit type



To examine the relationship between orbit types and mission success rates we used a bar chart. The chart reveals that ES-L1, GEO, HEO, and SSO achieved a 100% succes rate, making them the most reliable orbit types in the dataset. On the otherhand, orbit types GTO and ISS showed moderate success. Notably, orbit type SO had a 0% success rate, making it a key outlier and potentially highlighting higher mission risk for that orbit type

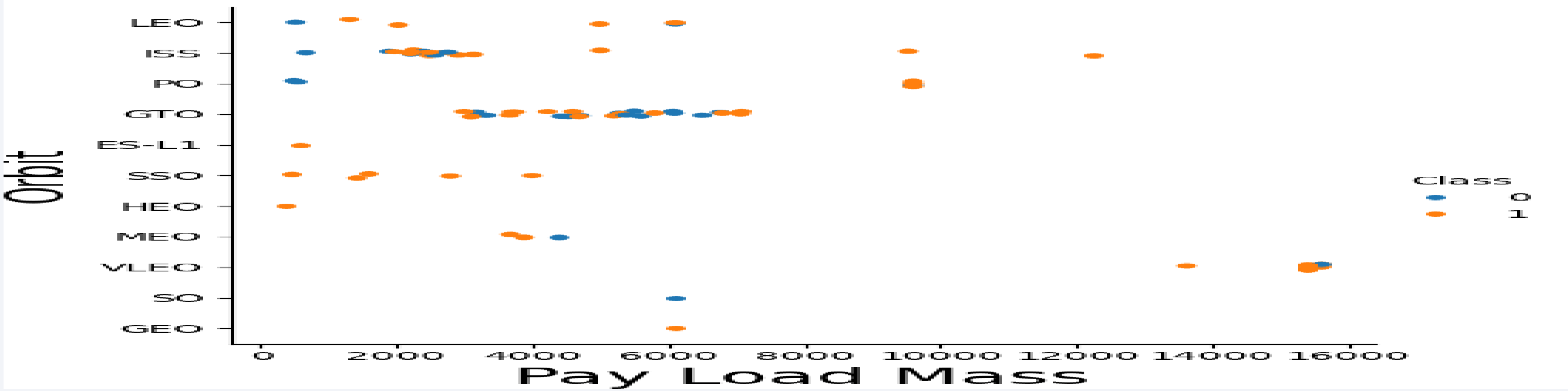
Flight Number vs. Orbit Type



We then used a scatter plot again to explore whether there was a relationship between FlightNumber, Orbit type and success rate. We observed that for most orbit types, like LEO, PO, SSO, VLEO , they appear to have a positive correlation between higher flight numbers and success rate, suggesting that reliability improves over time. Although with orbit types like, ISS and GTO, success appears more evenly distributed across flight numbers, indicating little relationship between flight number and experience

Payload vs. Orbit Type

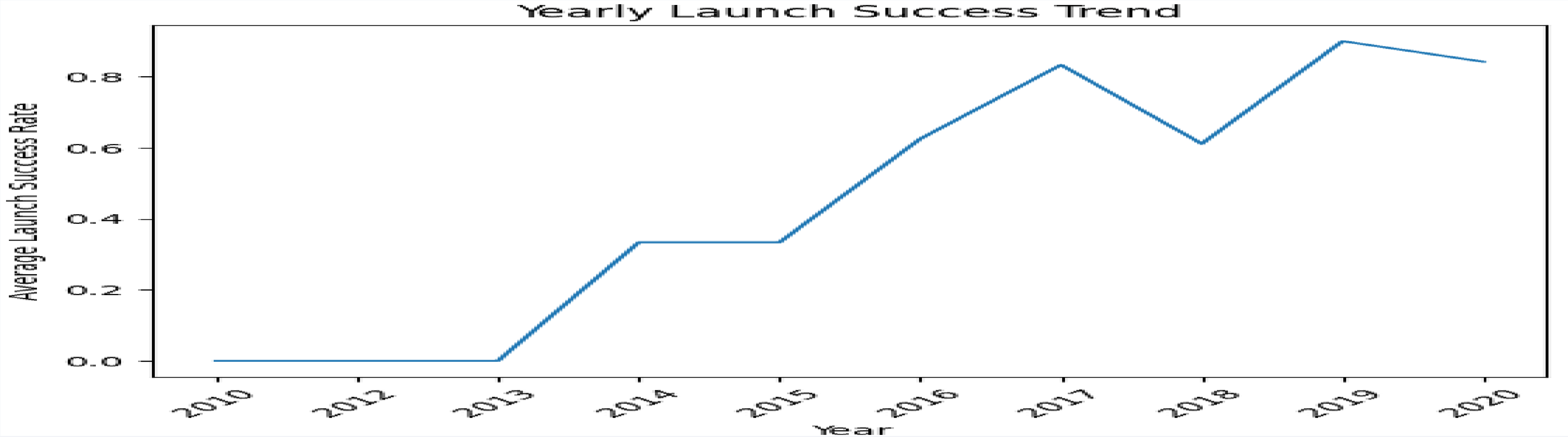
Payload vs. orbit type



We also explored how launch success varies with payload mass across different orbit types. Orbits types like PO, LEO, and ISS showed us more successful outcomes with heavier payloads. However, for orbits like GTO, the distribution of success and failure appears to be mixed, telling us there is no clear correltation between payload size and outcome for this orbit type.

Launch Success Yearly Trend

Line Chart of Yearly Average Success Rate



We used a line chart to map out the changes in average launch success rate over time. The chart shows us q clear upward trend beginning in 2013, with success rates steadily improving and peaking around 2019. This suggests to us that there is a strong correlation between increased launch experience and better landing outcomes over the years.

All Launch Site Names

Find the names of the unique launch sites

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

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

We used the query SELECT DISTINCT to retrieve the unique launch sites names from the dataset. The query returns all distinct entries in the Launch_Site column. Knowing where the SpaceX missions were launched from gave us the foundation for site-specific analysis later.

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

```
#
%sql SELECT * FROM SPACEXTABLE 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

To help us get a quick overview of the mission details for the sites that begin with 'CCA', we used the query:
SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
to filter and preview for the first 5 records from such launch sites.

The LIKE 'CCA%' condition matches all sites starting with that prefix, and LIMIT 5 restricts the output to the first 5 results.

Total Payload Mass

Calculated total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';
 * sqlite:///my_data1.db
Done.
```

Total_Payload
48213

To calculate the total payload mass carried by SpaceX boosters boosters for NASA CRS missions, we used the query:

SELECT SUM(PAYLOAD_MASS_KG_) AS Total_Payload FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';.

The SUM() function adds upall payloadss where the customer field contains 'NASA (CRS)', and the LIKE condition ensures we capture any partial matches.

The total payload result was 48,213 kg, giving us a little insight into NASA's cargo volume on SpaceX missions.

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload FROM SPACEXTABLE WHERE Booster_Version LIKE '%F9 v1.1%'
```

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

Average_Payload
2534.6666666666665

We then caclculated the average payload mass carried by Falcon 9 v1.1 boosters, using the query:

SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload FROM SPACEXTABLE WHERE Booster_Version LIKE '%F9 v1.1%';

The AVG() function computes the mean of all payload masses where the Booster_Version field contains 'F9 v1.1'. The LIKE condition ensures we include any records that partially match the version name.

The average payload result was approximately 2,534.67 kg, giving us insight into the typical cargo carried by this specific booster model.

First Successful Ground Landing Date

Querying for the dates of the first successful landing outcome on ground pad

```
%sql SELECT MIN(DATE) AS First_Succes_Date FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
```

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

First_Succes_Date
2015-12-22

To find the first successful ground landing by SpaceX, we used the query:

SELECT MIN(DATE) AS First_Succes_Date FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)';

The MIN() function returns the earliest date in the dataset where the landing outcome was recorded as a successful ground pad landing. The LIKE clause ensures that we only include landings with that exact outcome description.

The result shows that the first successful ground landing occurred on 2015-12-22.

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

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4001 AND 5999;
```

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

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

To identify the booster versions that successfully landed on a drone ship while carrying payloads greater than 4000 but less than 6000, we used the query:

SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4001 AND 5999;

The DISTINCT keyword ensures only unique booster versions are returned, while the BETWEEN clause filters for missions with payloads in the specified range.

The result shows four booster versions — F9 FT B1022, B1026, B1021.2, and B1031.2 , that met the conditions in question.

Total Number of Successful and Failure Mission Outcomes

Calculated the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success%' OR Mission_Outcome LIKE 'Failure%' GROUP BY Mission_Outcome;
```

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

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

To count the number of successful and failed missions, we used the query:
SELECT Mission_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success%' OR Mission_Outcome LIKE 'Failure%' GROUP BY Mission_Outcome;

The COUNT(*) function tallies how many times each type of mission outcome appears, while the LIKE conditions filter for outcomes that begin with either 'Success' or 'Failure'.The GROUP BY clause groups the results by outcome type for a clear count of each category.

The result shows 98 successful missions, with a few other outcomes like ‘Success (payload status unclear)’ and ‘Failure (in flight)’ also recorded.

Boosters Carried Maximum Payload

Subquerying names of the booster which carried the maximum payload mass

```
%sql SELECT Booster_Version, Payload_Mass__KG_ FROM SPACEXTABLE WHERE Payload_Mass__KG_ = (SELECT MAX(Payload_Mass__KG_) FROM SPACEXTABLE)

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

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

We also found the booster versions which were able to carry the maximum payload mass by following a query with a subquery:

SELECT Booster_Version, Payload_Mass__KG_ FROM SPACEXTABLE WHERE Payload_Mass__KG_ = (SELECT MAX(Payload_Mass__KG_) FROM SPACEXTABLE);

The subquery (SELECT MAX(...)) identifies the maximum payload mass in the dataset. The outer query then retrieves all booster versions that match this maximum value.

The result shows that several F9 B5 boosters carried the maximum recorded payload of 15,600 kg.

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 1, 4) = '2015';
```

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

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

Next we used the following query, to list the failed drone ship landings along with their booster versions and launch sites for the year 2015, the following query:

SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 1, 4) = '2015';

The substr() function extracts the year and month from the Date column. We filtered the results to include only entries from 2015 where the Landing_Outcome was a failure on a drone ship.

The result shows us two failed landings in January and April 2015, both launched from CCAFS LC-40 using Falcon 9 v1.1 booster versions.

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

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.

```
%sql 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;

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

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

To find out which landing outcomes occurred most frequently between June 2010 and March 2017, we used the 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;

The COUNT(*) function tallies the number of missions for each Landing_Outcome, and the ORDER BY clause ranks them from most to least common. The BETWEEN condition filters the results to only include landings within the specified date range.

The result shows that 'No attempt' was the most common outcome during this period, followed by both 'Success (drone ship)' and 'Failure (drone ship)', with several other outcomes occurring less frequently.

Section 3

Launch Sites Proximities Analysis

Overview of Launch Sites Globally

Generated folium map of all launch sites' location markers on a global map

Important Elements and Findings

- We marked each launch site with a circle marker, allowing for easier visibility on a global map
- We added DivIcon text labels to display the names of each site directly on the map, enhancing readability
- By adding these elements, we have a quick overview of SpaceX's global launch site locations, which tells us that they are all based in North America.

Visualising Success/Failed Launches

Folium Map with Coloured Labelled Launch Outcomes

We created a folium map to help visualise the individual mission outcomes using colour-coded markers:

- Green markers to indicate successful launches.
- Red markers to indicate failed launches.
- And marker clustering to reduce visual clustering to reduce visual clutter and help us better assess outcome distribution when zooming in and out.

Through the visualisations we were able to find out and interpret that:

- The launch activity was heavily concentrated around Florida, particularly KSC LC-391 and CCAFS SLC-40.
- Both successful and failed launches can occur at the same sites, telling us that location alone doesn't determine outcome.

<Folium Map Screenshot 3>

Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

Fig 1. Coastline

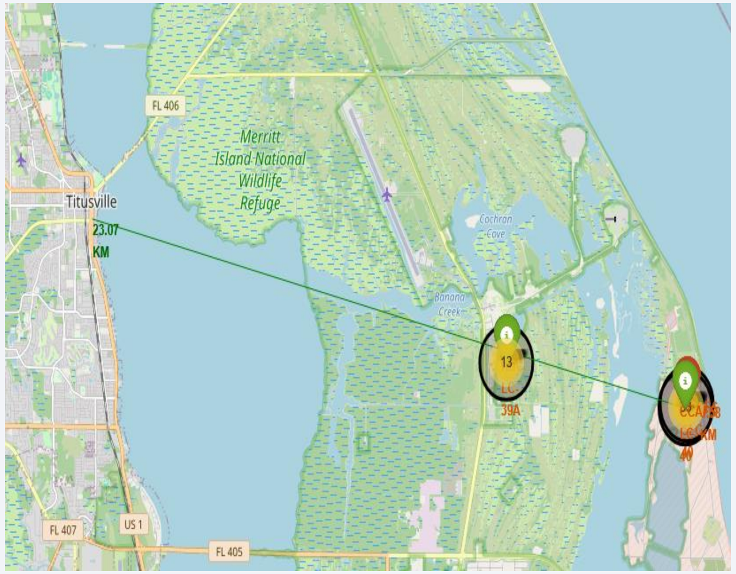


Fig 2. Closest City

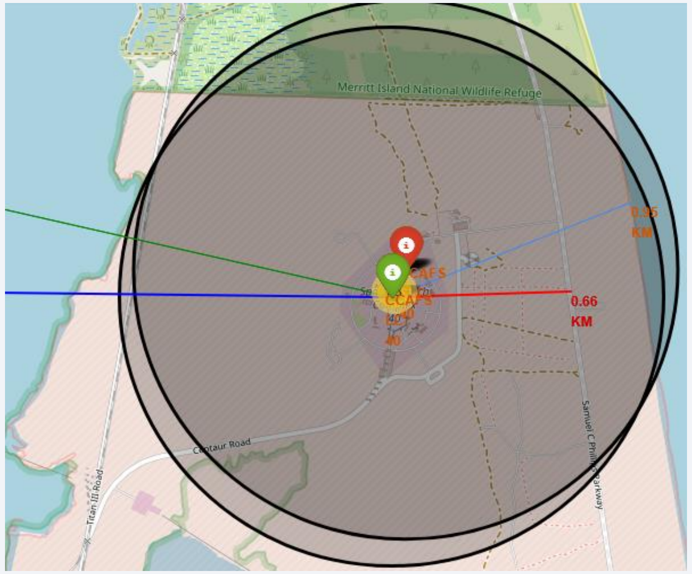


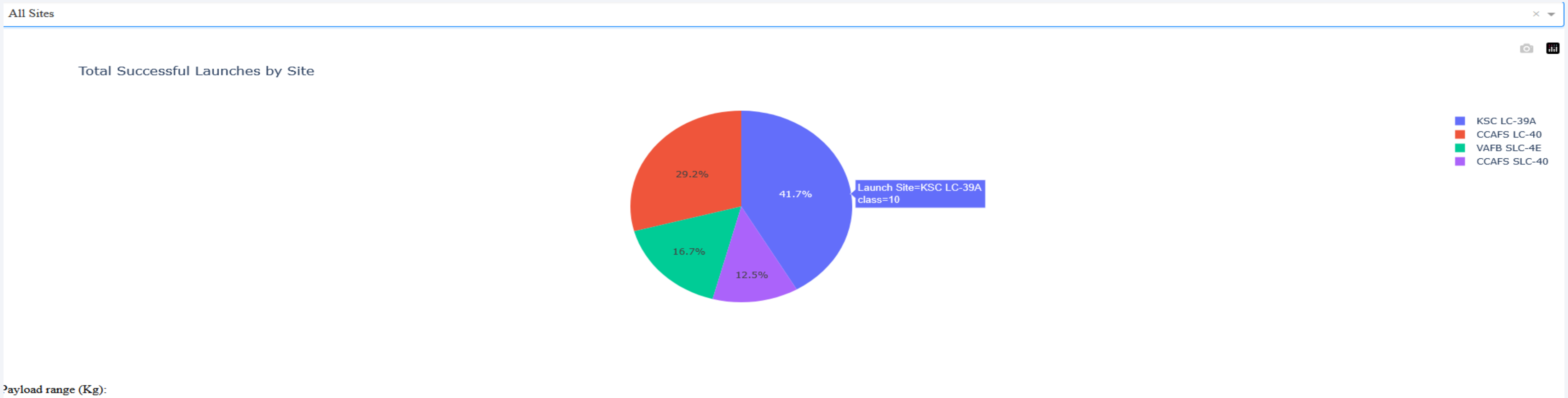
Fig 3. Highway & Railway

Section 4

Build a Dashboard with Plotly Dash

Launch Successes per Site

Show the screenshot of launch success count for all sites, in a piechart



Import Elements and Findings

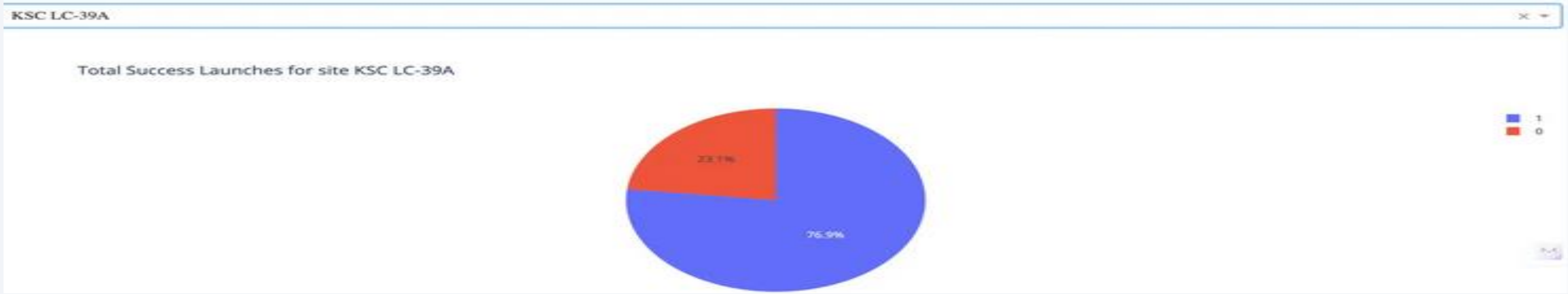
- Above we have a pie chart to help us quickly identify the proportions of launch successes per site. The segments within the pie chart represent the different launch sites, with each slice's size showing users the proportions of successful launches from that site.

- The visualisation also includes a color coded legend on the right. The different colors identify the different launch sites.

- The pie chart above shpws is that while the successful launches are spread across the different sites, launch site KSC LC-39A has the largest share of the pie, meaning that it is the dominant site for successsful missions.

KSC LC-39A Launch Success Breakdown

Breakdown of Launch Success at site KSC LC-39A



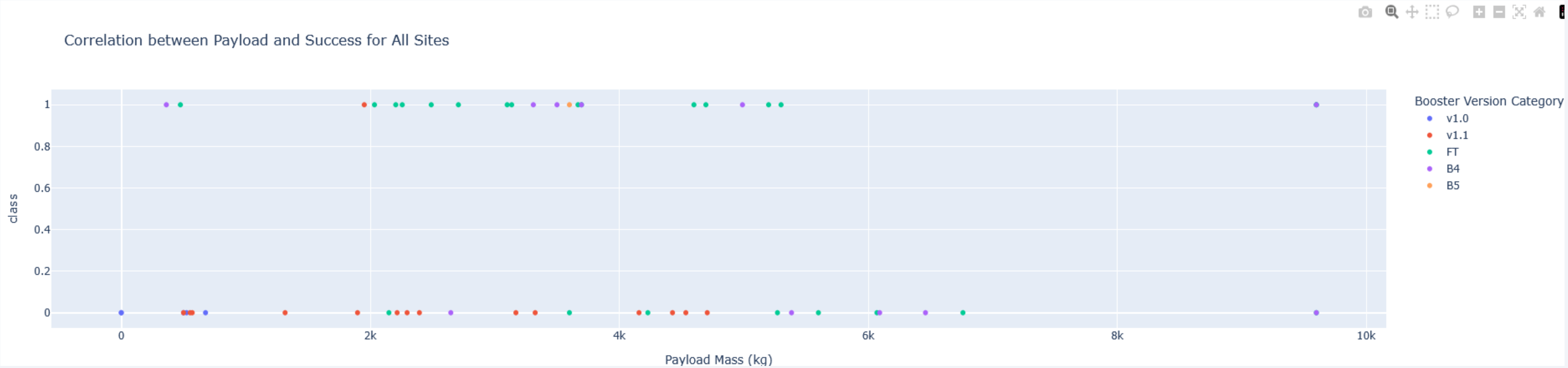
Important Elements and Findings

- The pie chart above is broken into 2 segments, showing us the two different classes. On the right of the dashboard we have a legend, showing users that the red portion is failure(0) and the lilac portion is for success(1)

- We can see from the pie chart that launch site KSC LC-39A has a high success rate of 76.9% and a relatively low failure rate of 23%.

Correlation between Payload and Success of All Sites

Payload vs. Launch Outcome scatter plot for all sites



Important Elements and Findings

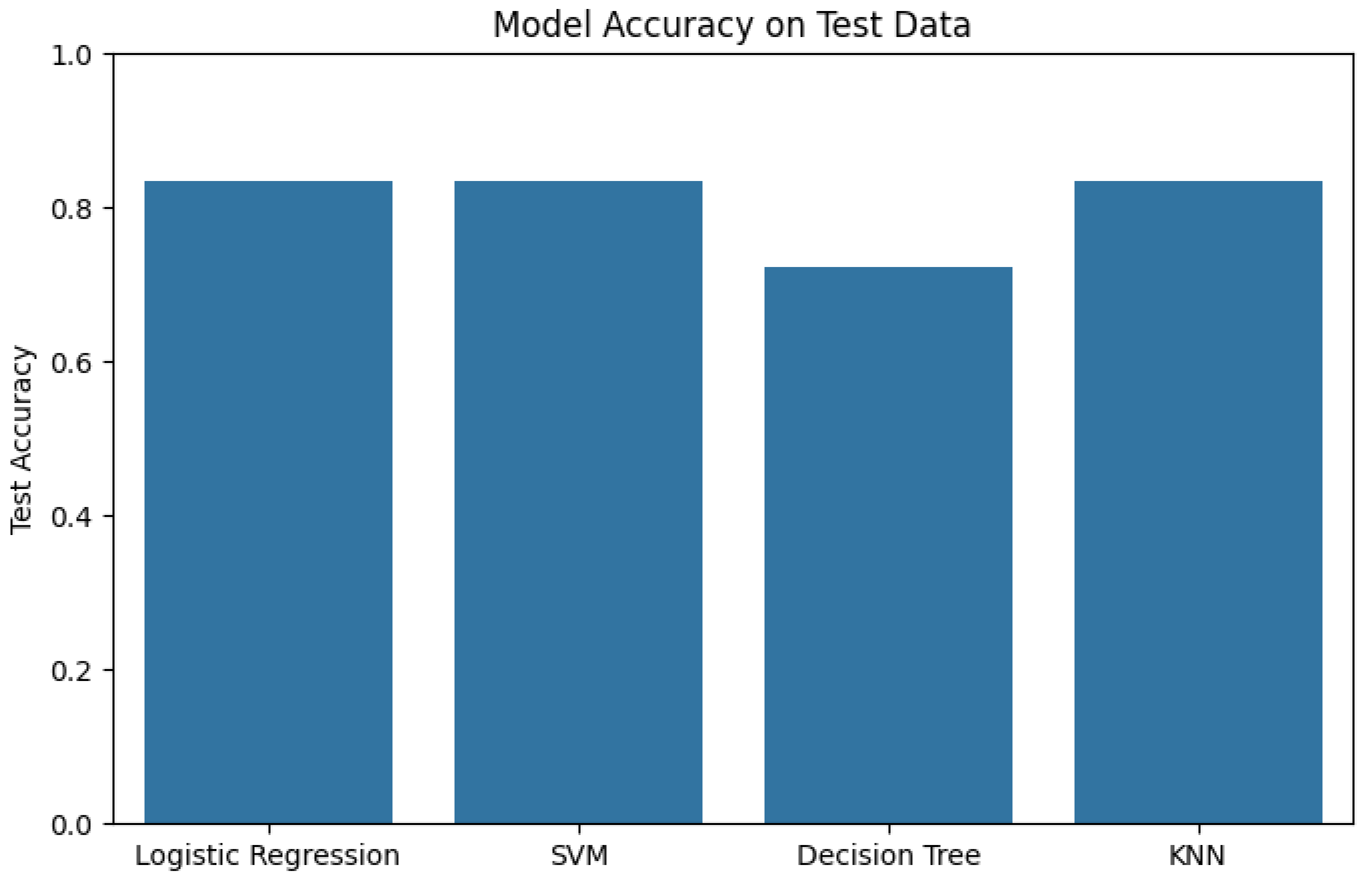
- Above we have an interactive slider embedded with the functionality to allow users to filter by payload range, which helps users explore how success varies at different launch weights.
- The X-axis represents the different payload masses(in kilograms) and the Y-axis represents the class, telling us whthere the mission was a success(1) ot a failure(0). On the right of the scatter plot there is a legend, with color coding to show users which booster version was used for each launch.
- From a quick glance, we can see that there are slightly more failed launches than successful launches. We can also see that the payload range between 2000 and 4000 has more success launches than the other ranges, and also more successes than failure within this range.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

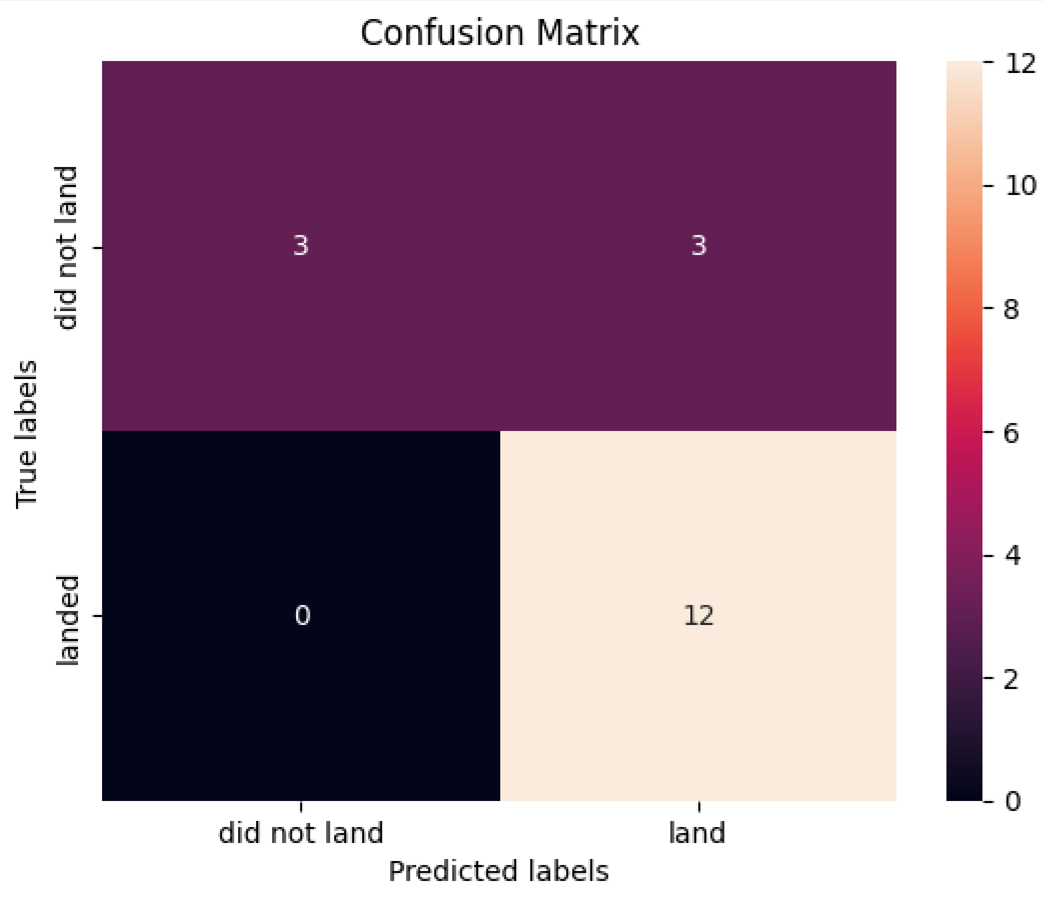
Bar Chart Visualisation of All Model Accuracies




```
The best model based on test accuracy is: Logistic Regression with accuracy: 0.8333
```

Confusion Matrix

Logistic Regression Confusion Matrix



Findings:

- The logistic regression model is quite good at detecting actual landing, having recorded 12 True Positives, meaning that the model correctly predicted 12 successful landings.

- However, it's not as reliable when predicting failed landings. With 3 False Positives and 3 True Negatives recorded. Meaning that the model misclassified 50% of failures(3 from 6) as landings when they were in fact failures.

Conclusions

- Our analysis showed us that launch site, payload mass, and booster version all influence Falcon 9 landing success.
- There is a upward trend of better performances and more successes as time passes and as the launch attempts increase.
- Launch Site KSC LC-39A had the highest success rate, while payloads between 2000 – 4000 kg were consistently successful.
- Booster versions F9 & B5 performed the best, carrying heavy payloads with frequent successful landings.
- We found that our Logistic Regression Model performed the best when predicting outcomes, achieving a test accuracy score of 83.33%. It successfully identified all the landings.
- It did, however misclassify 50% of failed landings, showing high recall but lower precision.

Thank you!