IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rohan Pasricha
September 2025
rohanpasricha.work@gmail.com

Outline

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

Executive Summary

- Summary of methodologies:

  First, we collected data from the SpaceX API and by webscraping a Wikipedia page html table. Next, we did some data preprocessing/wrangling and performed EDA using SQL and visualizations. We also performed some feature engineering in this stage.

  Then we created an interactive dashboard using Plotly Dash to visualize relationships between features and discover interesting patterns. We also used Folium to analyze launch site locations on an interactive map.

  Finally, we prepared the data for modelling and used GridSearchCV with Cross Validation to compare the performance of 4 hyperparameter optimized classification models. We compared performance using accuracy and confusion matrices.

- Summary of all results:

  The best performing model was the DecisionTree classifier with an accuracy score of 0.944. All of the tested predictive models struggle with False Positives but seem to not have an issue with False Negatives.

  Some insights gained during EDA and interactive visual analytics: The Launch Sites are positioned as close to the equator line as possible while still remaining within US borders. The sites are well connected to transport lines, close to coastlines and far from cities. Launches carrying heavy payloads (over 10,000KG) seem to have a higher success rate than lower payload launches. The KSC LC-39A Launch Site has the highest total number of successful launches as well as the highest launch success rate ratio. The FT Booster Version has a high success rate relative to the performance of the other boosters. Orbit types chosen have an effect on mission success rate. The average success rate of launches per year has been generally trending upwards since 2013.

Introduction

- Project background and context:

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

  Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems we want to find answers to:

  - What insights can we gain from the Launch data? (for example, correlation between features like payload mass / chosen orbit / launch site / etc and outcome)
  - What insights can we gain about the geographical locations of the Launch Sites?
  - How accurately can we predict whether the first stage will land to be recovered and reused?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
- Data wrangling methodology
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

Data Collection

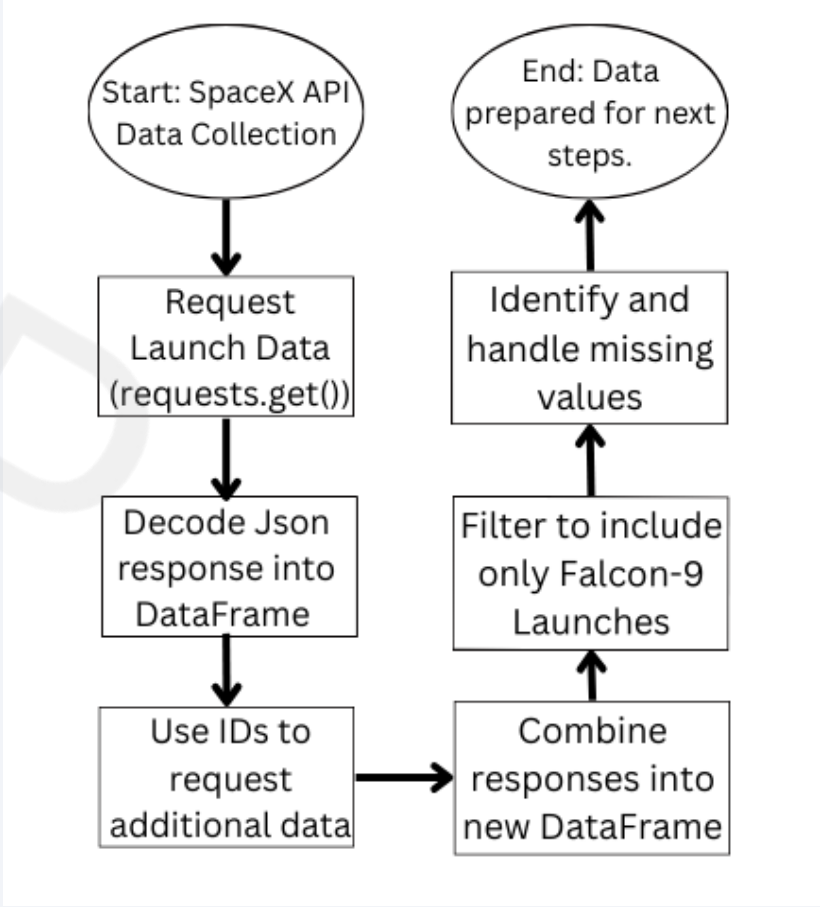
The datasets used for this analysis were collected using the *SpaceX API* (with the help of the requests library) as well as by *web scraping* an html table present on a *Wikipedia* page (using BeautifulSoup).

The data was formatted into pandas DataFrames for analysis after collection.

In the following slides, we will get into more detail about the data collection process and visualize the process using flowcharts.

Data Collection – SpaceX API

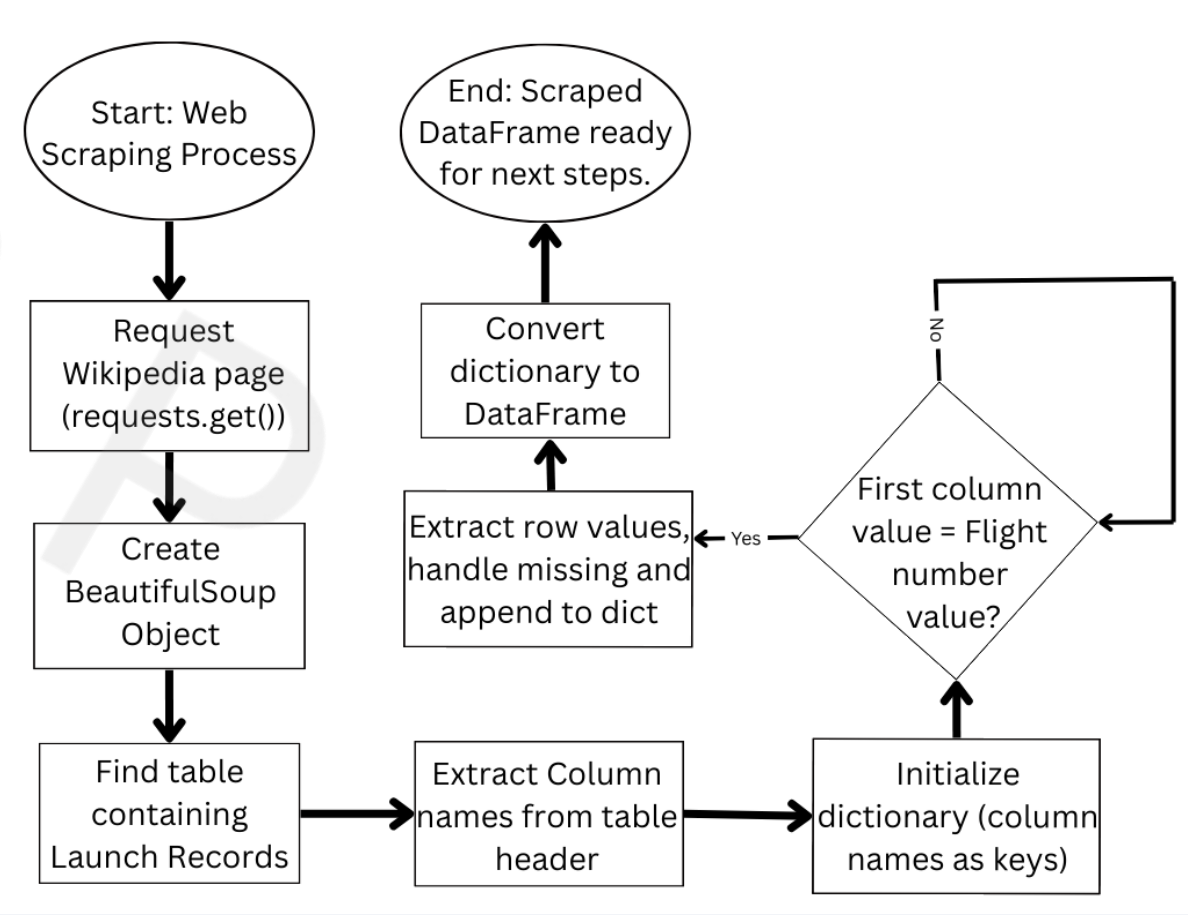
- First we use the requests library to request rocket launch data from the SpaceX API.
- Then, we decode the response content as a Json and turn it into a pandas DataFrame for further use.
- We notice that a lot of the columns contain IDs rather than useful information so we use the API again to gain information about the launches using the given IDs.
- The data from these responses will be stored in lists and combined to form a new DataFrame. Some columns we want more information on are rockets, payload, launchpad and core (replacing IDs with values).
- Next, we will filter the dataset for only Falcon9 Launches.
- Finally, we analyze and deal with missing values.



[GitHub URL of the completed SpaceX API calls notebook]

Data Collection - Scraping

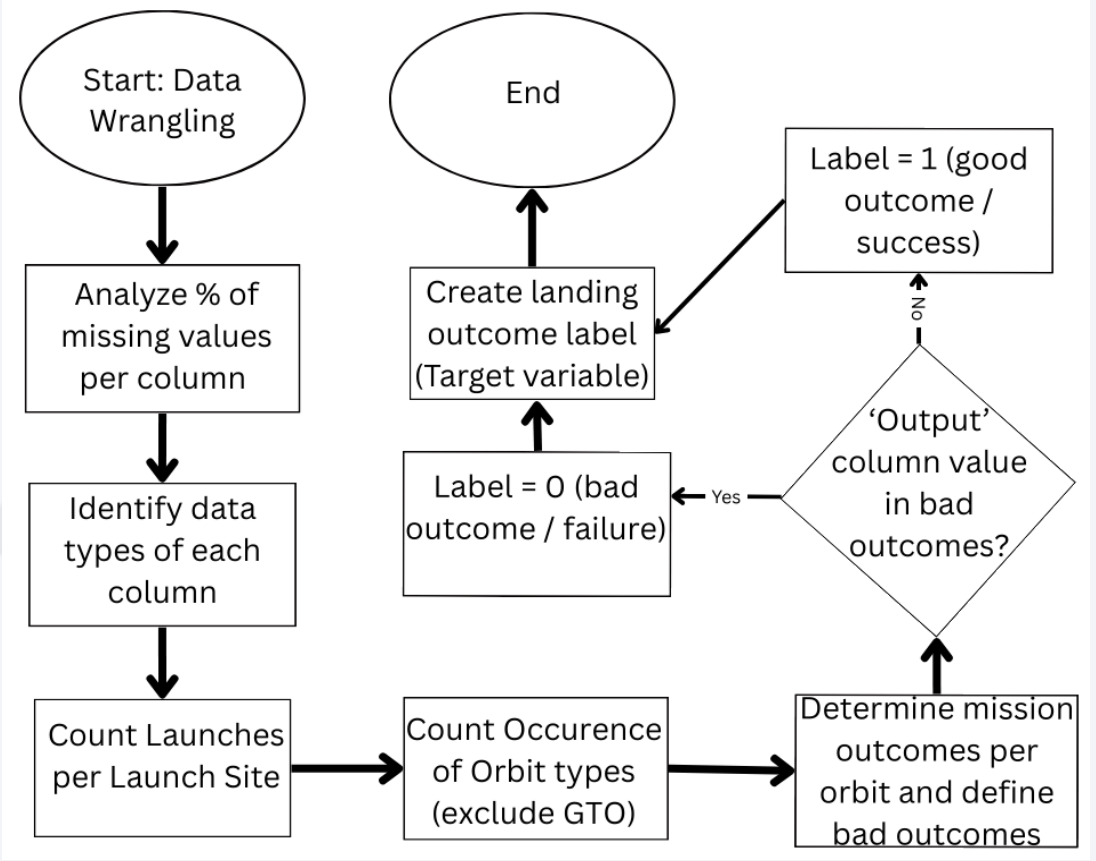
- To begin, we use the requests.get() method to request the Falcon9 Launch Wikipedia page as an HTTP response and create a BeautifulSoup object using this response.
- Next, we find our target table present on the page that contains the actual launch records and extract column/variable names one by one from the table header.
- Following this, we create a python dictionary using the extracted column names as keys to later convert to a DF.
- Now, we will check if the first column value corresponds to a Flight Number in order to simplify the parsing process.
- If True, we will extract relevant columns from the table row and append the values into the dictionary. This step also includes adding placeholder values if values are missing.
- After filling in the parsed launch record values into the dictionary, we create a DataFrame from the dictionary.



GitHub URL of the completed web scraping notebook

Data Wrangling

- First we performed some basic EDA, getting an initial look into the data to find some patterns.
- To do this we analysed the percentage of missing values per column and identified the data types of each column.
- Then, we calculated the number of launches from each launch site,
- We also calculated the number and occurrence of each orbit that launches were dedicated to (excluding GTO from our consideration since it is a transfer orbit and not itself geostationary).
- Next, we determined the number and occurrence of mission outcome per orbit type (and calculated a set of 'bad outcomes').
- Finally, we created a landing outcome label using the Outcome column and the set of bad outcomes calculated previously. The landing class variable represents the classification label that represents the outcome of each launch. This will be used as the target variable during modelling.



GitHub URL of the completed data wrangling notebook

EDA with Data Visualization

Created Visualizations:

- Flight Number Vs. Payload Mass scatterplot with outcome overlay to visualize how these variables affect launch outcome / success rate.
- Flight Number Vs. Launch Site scatterplot with outcome overlay to visualize their relationship and extract patterns. We can see how often each launch site is used, launch site preferences over time, and outcomes.
- Payload Mass Vs. Launch Site scatterplot with outcome overlay to observe if there is any relationship between the 2 variables and if specific combinations of payload + site are more successful than others.
- Bar chart for the success rate of each Orbit Type to visualize the most successful orbit types for launches.
- Flight Number Vs. Orbit Type scatterplot with outcome overlay to visualize relationships between the variables and check, for example, if some orbit types' success rates improve with more launches.
- Payload Mass Vs. Orbit Type scatterplot with outcome overlay to reveal the relationship between payload and orbit, such as checking if certain orbit types are more successful when the payload mass is higher.
- Year of launch Vs. Average Success Rate (over the year) line plot to visualize overall success rate of launches over time.

In this notebook, we also performed some feature engineering like one hot encoding and ensuring dtype consistency.

GitHub URL of the completed EDA with data visualization notebook

EDA with SQL

- Connected to DB and removed blank rows from table
- Displayed the names of the unique Launch Sites used in missions
- Displayed 5 records where Launch Site begins with the string "CCA"
- Displayed the total payload mass carried by boosters launched by customer 'NASA (CRS)'
- Calculated average payload mass carried by booster 'F9 v1.1'
- Checked when the first successful landing outcome in ground pad was achieved
- Listed boosters that had successful outcomes in drone ship with payload mass between 4000-6000
- Listed total number of successful and failure mission outcomes
- Displayed a list of all booster versions that have carried maximum payload mass
- Listed records displaying month names, failure landing outcomes in drone ship, booster version and launch site for the months in the year 2015
- Ranked the count of landing outcomes between 2010-06-04 and 2017-03-20 in Descending Order

GitHub URL of the completed EDA with SQL notebook

Build an Interactive Map with Folium

Map objects created and added to a folium map and their purpose:

- **Circles:** Added at NASA JSC and each launch site. **Purpose:** Highlight and visually emphasize the geographic position of launch sites.
- **Markers (& DivIcon)**: Placed at NASA JSC, each launch site, and proximity points (coastline, railway, highway, city). Custom labels display site names and distances. **Purpose:** Pinpointing exact locations and providing readable context directly on map.
- **MarkerCluster:** Contains individual launch markers color-coded by outcome where green represents success (class label=1) and red represents failure. **Purpose:** To show success/failure distribution while keeping map uncluttered.
- **Icons:** Used inside markers to indicate outcome with different colors. **Purpose:** Quick visual differentiation between successful and failed launches.
- **Polylines:** Drawn between launch sites and their nearest coastline, railway, highway, and city. **Purpose:** Visualize proximity relationships and measure logistical/safety distances.
- **Mouse Position:** Displays real-time coordinates when hovering over the map. **Purpose:** Helps extract coordinates for coastlines and infrastructure points needed in distance calculations.

GitHub URL of the completed interactive map with Folium map notebook

Build a Dashboard with Plotly Dash

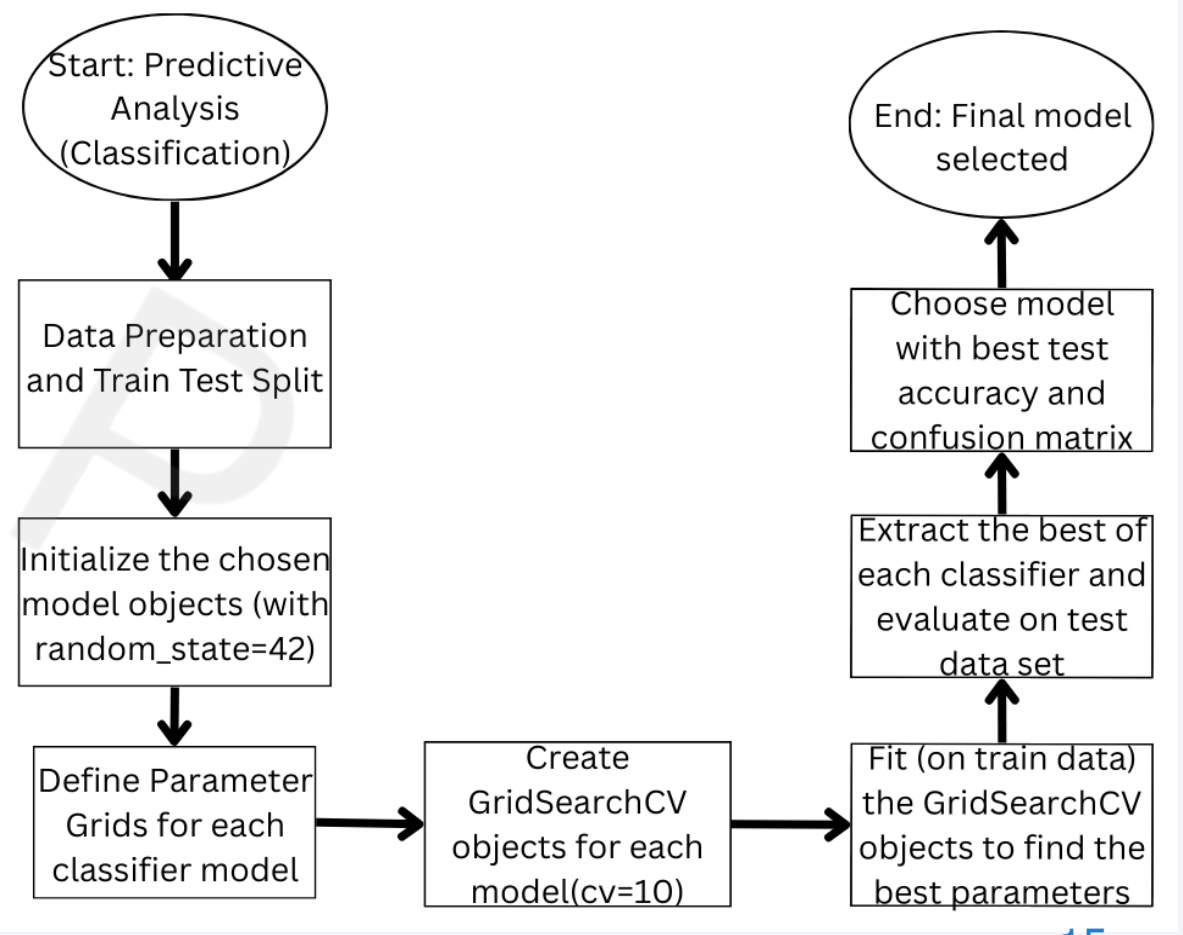
Dashboard components created and their purpose:

- **Dropdown Menu:** Options: All Launch sites / Any Individual Site **Purpose:** Let users filter the analysis by launch site, or view aggregated results across all sites. **Interaction:** Triggers updates in the pie chart and scatter plot.
- **Pie Chart**: Shows total successful launches by site or Success Vs. Failure rate of individual sites based on dropdown menu input. **Purpose:** Provides an overview of performance (success rate distribution) at different launch sites.
- **Range Slider:** Allows selection of a payload mass range (min to max payload). **Purpose:** Narrow down data to launches within a chosen payload ramge. **Interaction:** Updates scatter plot to only show points within the selected range.
- **Scatter Plot:** Plots Payload Mass (kg) Vs. Launch Outcome (class label representing success or failure) with data point color representing booster version. **Purpose:** Explore correlation between payload size and launch success, with insight into booster versions.

GitHub URL of the completed Plotly Dash lab

Predictive Analysis (Classification)

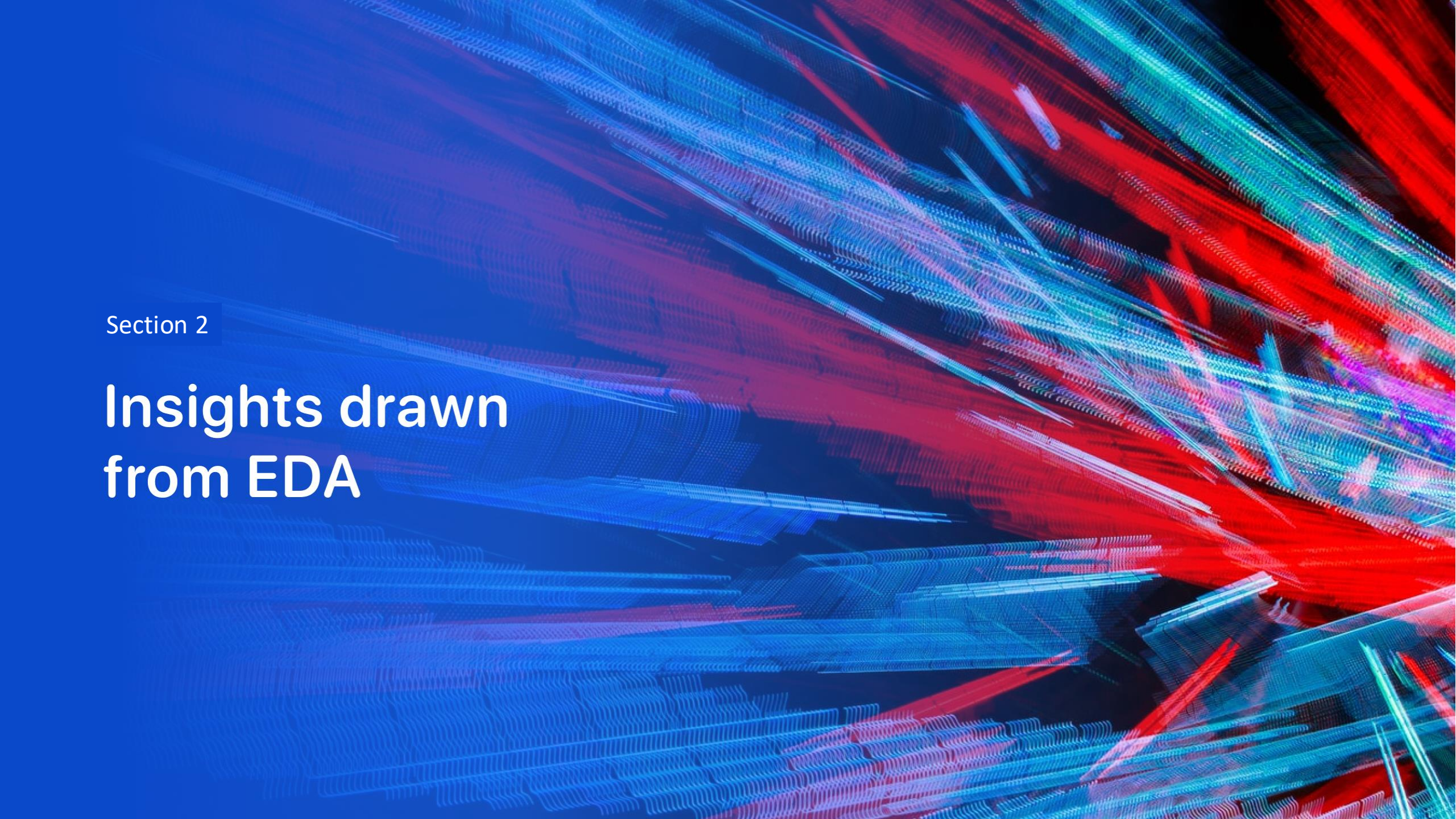
- **Data Preparation:** Extracted target variable Y from Class column, standardized features X using StandardScaler() and split into train/test sets (80/20) for evaluation.
- **Classification Models chosen for comparison:** Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier and K-Nearest Neighbors (KNN)
- **Model Training and Tuning:** Used GridSearchCV (cv=10, scoring=accuracy) for training and optimization of each chosen classification model.
- **Improvement Process / Hyperparameter Optimization:** Explored multiple hyperparameter combinations by defining parameter grids for each model and testing the combinations with the help of GridSearchCV. Set random_state=42 for model objects for reproducibility.
- **Model Evaluation:** Compared model performances using validation accuracy, test accuracy, and analyzing confusion matrices. In this stage, we also visualized test accuracy of each model side-by-side using a bar graph.



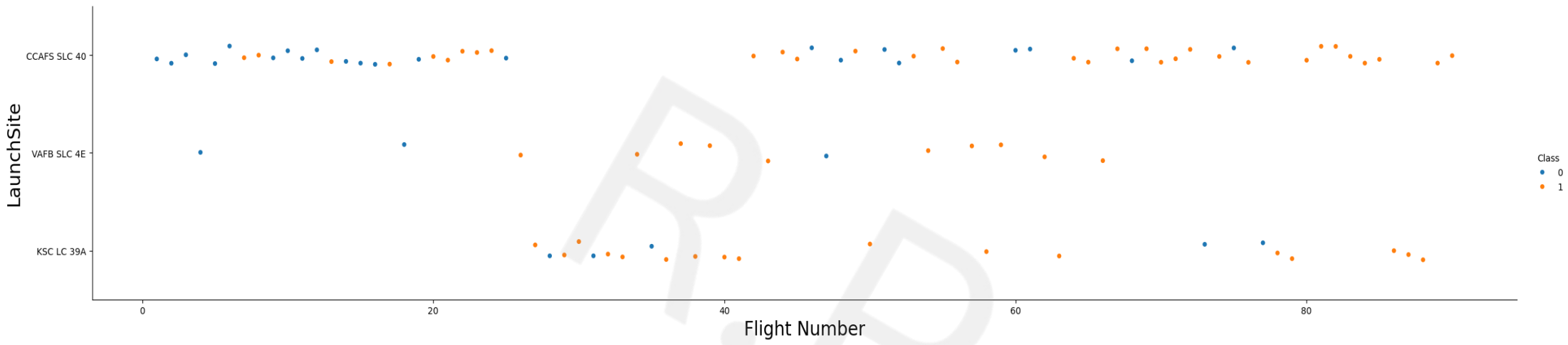
[The GitHub URL of the completed predictive analysis lab]

Results

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


Section 2
Insights drawn from EDA

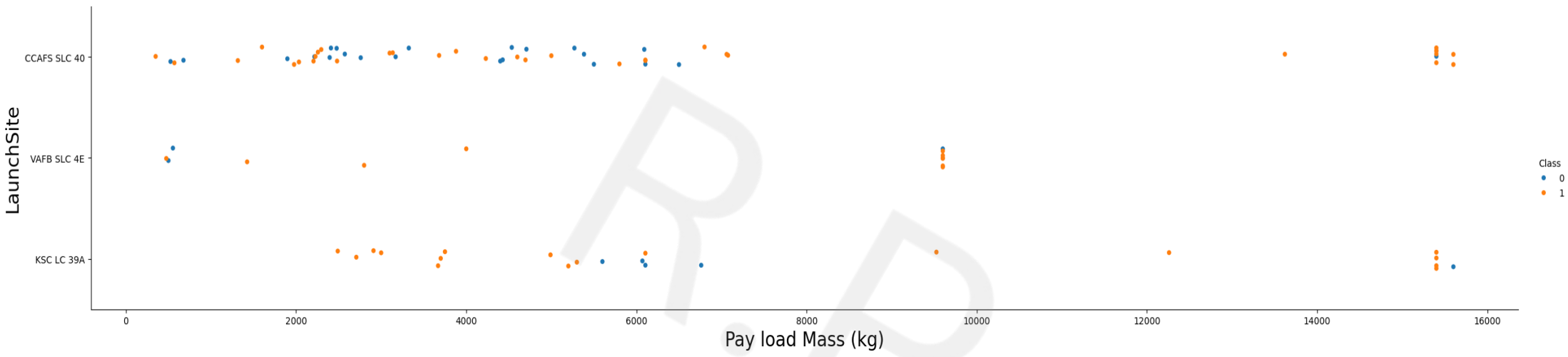
Flight Number vs. Launch Site



In the scatter plot we can observe that the CCAFS SLC 40 Launch Site is used the most, with newer launches(FlightNumber>70) from this site having a much higher success rate than before. The VAFB SLC 4E Launch Site has not been used for the last 20+ Launches even though 10 out of the last 11 launch attempts from this site have been successful(stage 1 recovered).

When considering all launches, KSC and VAFB launch sites seem to have comparable success rate with CCAFS trailing behind in success metrics but this can be misleading. We can clearly see that the success rate of launches from CCAFS in recent launches is very high and it is understandable that, since this site was used much more than the others early on, the overall success rate would be lower due to the novel approach.

Payload vs. Launch Site



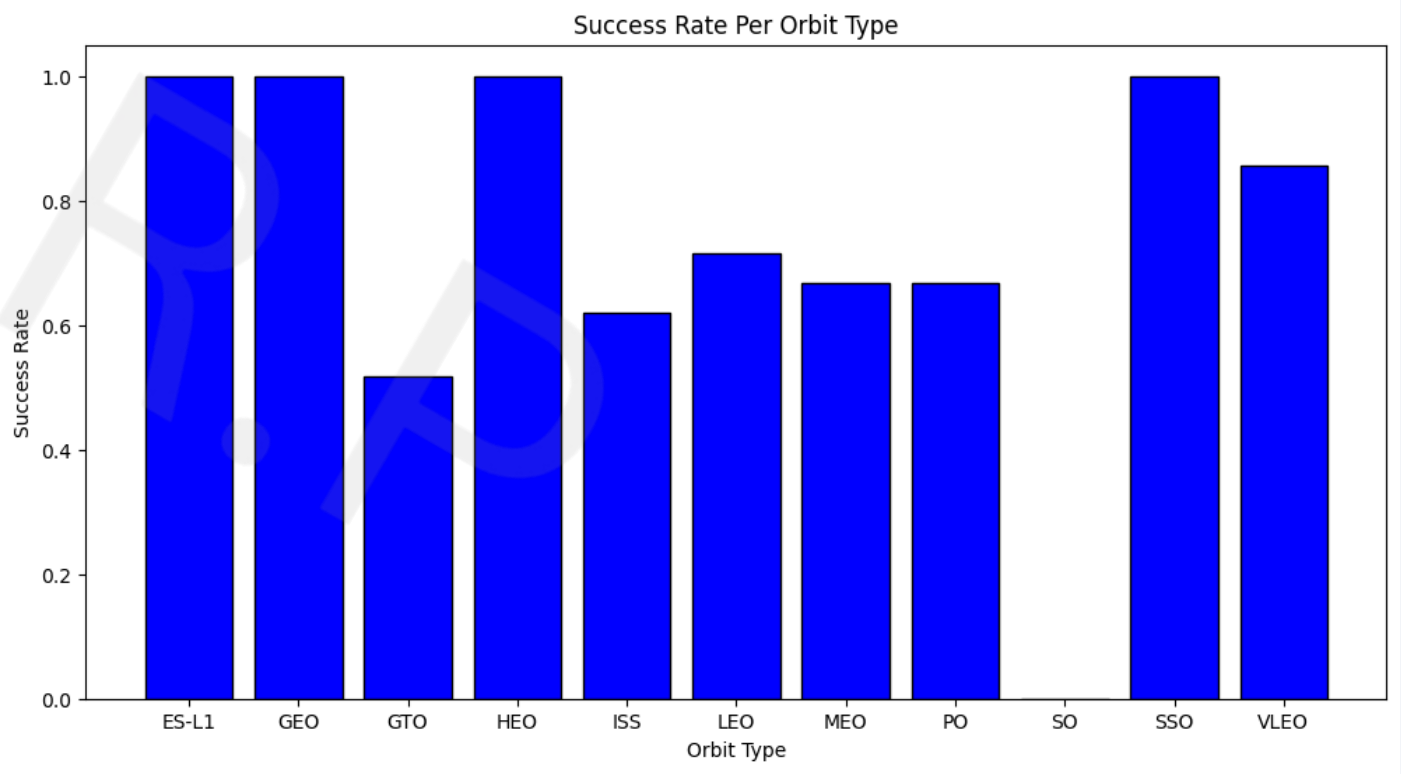
If we observe the Payload Mass Vs. Launch Site scatter point chart we find that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10,000kg)

Additionally, we can also observe that the success rate of the Launches with a heavy payload mass (more than 10,000kg) seem to have a higher success rate than lower Payload launches. Even launches with a payload mass between 8,000 and 10,000kg seem to perform quite successfully.

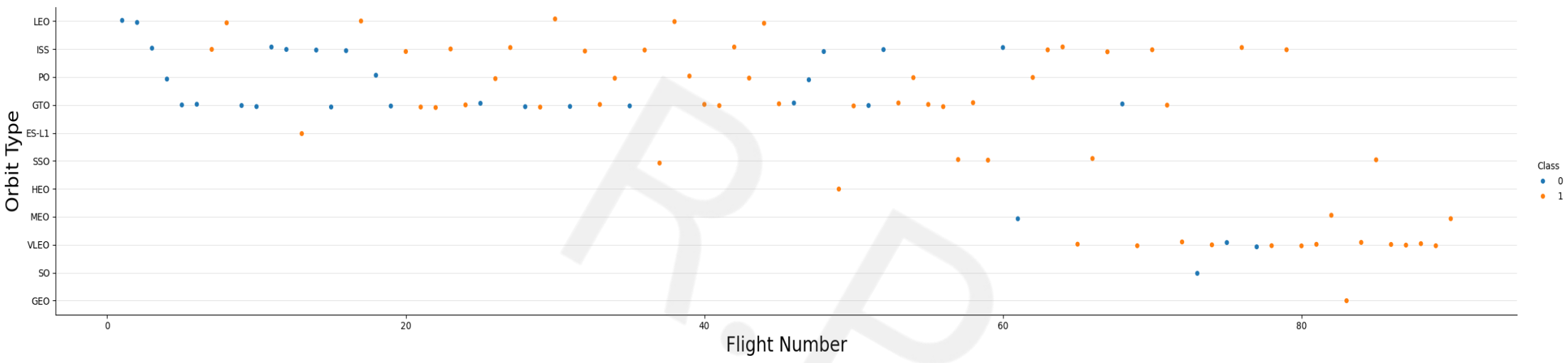
Success Rate vs. Orbit Type

In this bar chart, we can clearly see that the ES-L1, GEO, HEO and SSO orbit types have the highest success rate (1 or 100%) with VLEO a close 2nd place at around 0.85.

Although this graph seems conclusive, we must also consider the number of launches per orbit type for a more complete picture because, currently, an orbit with 10 out of 10 successful launches would be valued the same as an orbit with 1 out of 1 successful launches.



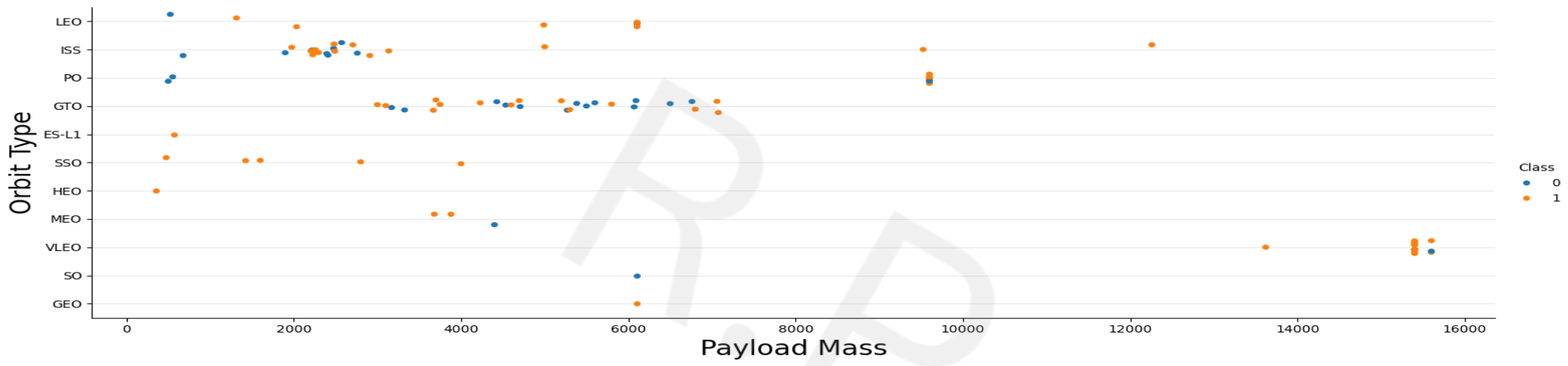
Flight Number vs. Orbit Type



We can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Additionally, success rates in the previous plot can be somewhat misleading if interpreted without context. For example, HEO and GEO have only 1 launch, both successful, and so we see a success rate of 1 but the sample size is also only 1. On the other hand, SSO orbit type launches have maintained their success rate of 1 over 5 launches and even VLEO orbit type launches seem to have an impressive success rate over a larger number of launches (12 out of 14 successful launches)

Payload vs. Orbit Type

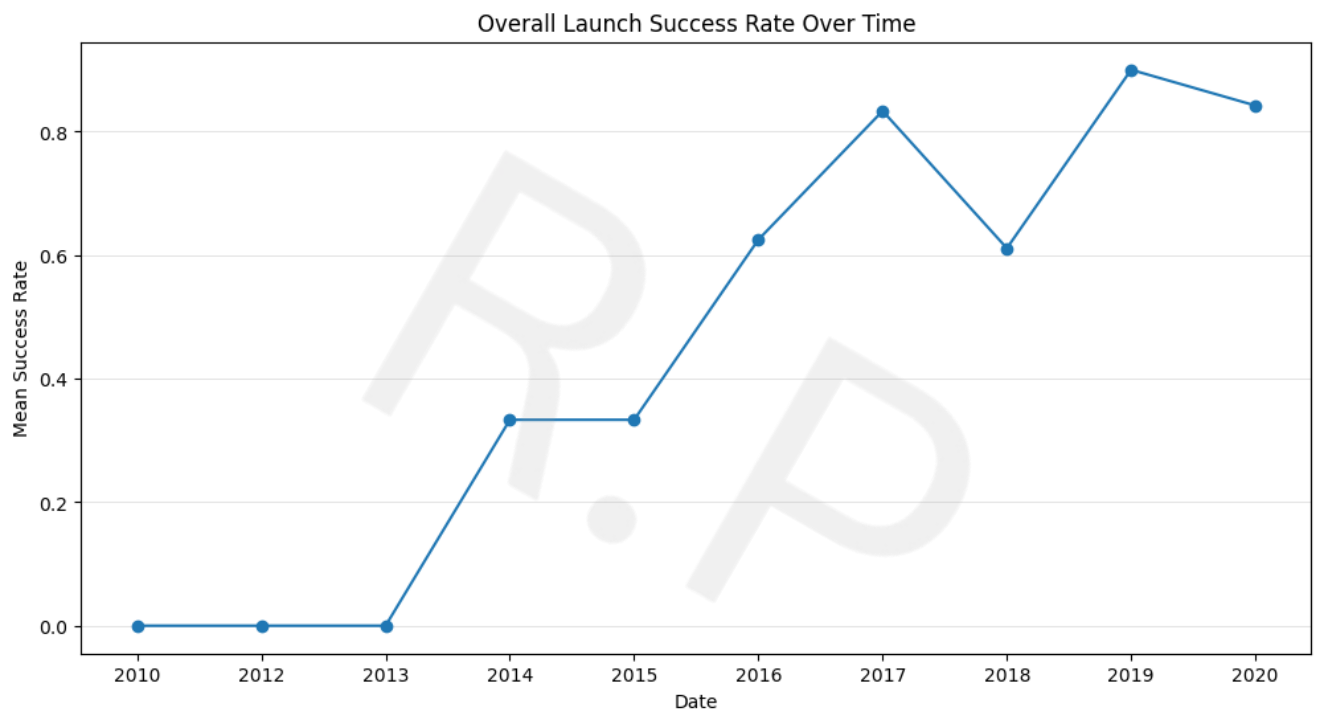


With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS orbit types.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

For MEO, we could argue that the success rate decreases with higher payload mass but since there are such few data points and also the difference in payload seems to be less than 1000kg: this seems inconclusive. Even for VLEO, since all of the launches have had a high payload mass, there is not enough of a payload mass range to determine correlation between payload and success rate for that orbit at this time.

Launch Success Yearly Trend



In the above line chart of yearly average launch success rate, we observe that the average success rate of launches per year has been following a general trend of increasing success rate since 2013 with a dip of 0.2 in 2018(as compared to 2017) and a small dip of 0.1 or lesser from 2019 to 2020. In general, the success rate does seem to be improving over time and it would be reasonable to assume that, at the very least, the success rate would not tend to reduce moving forward (and might continue following the trend of improvement).

All Launch Site Names

Query used:

select DISTINCT "Launch_Site" from SPACEXTABLE

Result:

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

Explanation:

By making use of the DISTINCT keyword, we extract unique values from the "Launch_Site" column of the DB table "SPACEXTABLE"

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

Launch Site Names Begin with 'CCA'

Query used:

select * from SPACEXTABLE WHERE "Launch_Site" like "CCA%" LIMIT 5

Result:

Extracted all columns for first 5 rows where "Launch_Site" column value started with CCA.

Explanation:

The like operator is used in a where clause to search for a specific pattern in a column. The % sign represents 0,1 or multiple characters. The limit clause is used to specify the number of records to return.

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 used:

select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer" = "NASA (CRS)"

Result:

45,596 kg

SUM("PAYLOAD_MASS__KG_")
45596

Explanation:

By making use of the SUM() function and where clause, we calculated the total payload mass carried by boosters launched by NASA(CRS).

Average Payload Mass by F9 v1.1

Query used:

select avg("PAYLOAD_MASS__KG_") as "Average Payload Mass in Kg" from SPACEXTABLE WHERE "Booster_Version" = "F9 v1.1"

Result:

2928.4 kg

Average Payload Mass in Kg
2928.4

Explanation:

By using the AVG() function and where clause, we calculated the average payload mass carried by booster version F9 v1.1

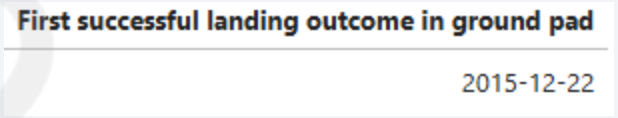
First Successful Ground Landing Date

Query used:

select min(Date) as "First successful landing outcome in ground pad" from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"

Result:

2015-12-22



First successful landing outcome in ground pad
2015-12-22

Explanation:

By making use of the MIN() function and where clause, we extracted the first successful ground landing date.

Successful Drone Ship Landing with Payload between 4000 and 6000

Query used:

```
select DISTINCT "Booster_Version" from SPACEXTABLE where
"Landing_Outcome"="Success (drone ship)" and
"PAYLOAD_MASS__KG_">4000 AND
"PAYLOAD_MASS__KG_"<6000
```

Result:

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

Explanation:

By using the DISTINCT keyword, WHERE clause, and the AND operator we are able to extract the Booster Versions which have success in drone ship with payload mass between 4000-6000 kg.

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

Total Number of Successful and Failure Mission Outcomes

Query used:

```
SELECT "Landing_Outcome" AS "Landing Outcome",
COUNT("Landing_Outcome") AS "Count" FROM SPACEXTABLE WHERE
"Landing_Outcome" LIKE '%Success%' OR "Landing_Outcome" LIKE
'%Failure%' GROUP BY "Landing_Outcome"
```

Result:

Total Successes:61 Total Failures:10

Explanation:

Using the wildcard % and like operator, we are able to include all landing outcomes with 'Success' or 'Failure' as part of their name and see their counts. I was not sure whether to interpret the question as displaying total count of each success and failure mission outcome or combined so I displayed each and calculated the combined total number of successes and failures.

Landing Outcome	Count
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

Query used:

```
SELECT DISTINCT "Booster_Version" AS "Booster versions that carried max payload" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

Result:

A list containing each of the booster version names that carried max payload mass in a launch. (result screenshot displayed)

Explanation:

By using the DISTINCT keyword, WHERE clause, and a subquery with a suitable MAX() function we are able to extract all the (unique) booster versions that carried maximum payload in a Launch.

Booster versions that carried max payload
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

Query used:

SELECT SUBSTR(Date,6,2) as month, "Landing_outcome", "Booster_version", "Launch_site" from SPACEXTABLE where substr(Date,0,5)="2015" AND "Landing_Outcome" = 'Failure (drone ship)'

Result:

A list containing only 2 records of drone ship landing attempt failures in 2015, 1 in Jan and the 2nd in April. Both from CCAFS LC-40 Launch Site using different boosters.

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

Explanation:

By using the SUBSTR function and WHERE clause, we are able to get the month from the Date column, check the year from the date column and extract launch records which were drone ship failures in that year.

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

Query used:

```
select  "Date", "Landing_outcome" as "Landing Outcome",
COUNT("Landing_outcome") as "Landing Outcome Counts" FROM
SPACEXTABLE WHERE "Date"<'2017-03-20' and "Date">'2010-06-
04'\

   GROUP BY "Landing_outcome" ORDER BY
COUNT("Landing_outcome") DESC
```

Result:

A list containing each of the landing outcome names, the date of their first occurrence and their counts between the specified date range. (result screenshot displayed)

Date	Landing Outcome	Landing Outcome Counts
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2015-06-28	Precluded (drone ship)	1
2010-12-08	Failure (parachute)	1

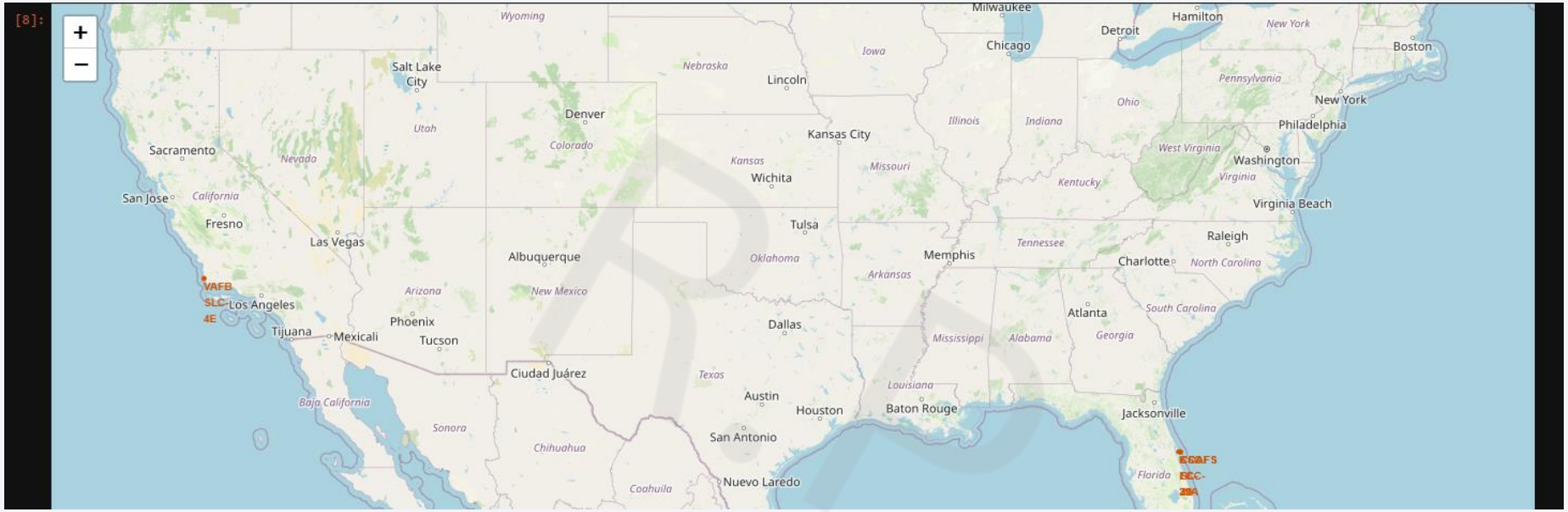
Explanation:

By using the COUNT() function, WHERE clause, GROUP BY clause, ORDER BY clause and DESC keyword we can rank landing outcome counts between the specified date range and arrange it in descending order.

Section 3

Launch Sites Proximities Analysis

Launch Sites Marked on Folium Map



It seems that SpaceX attempted to keep launch sites as close to the equator as possible while still remaining within US borders. The reasoning for this would probably be that the surface speed of the Earth is highest at the equator line and rockets can essentially use this surface speed as a running start, thus maximizing fuel efficiency and reducing costs and weight by requiring lesser fuel.

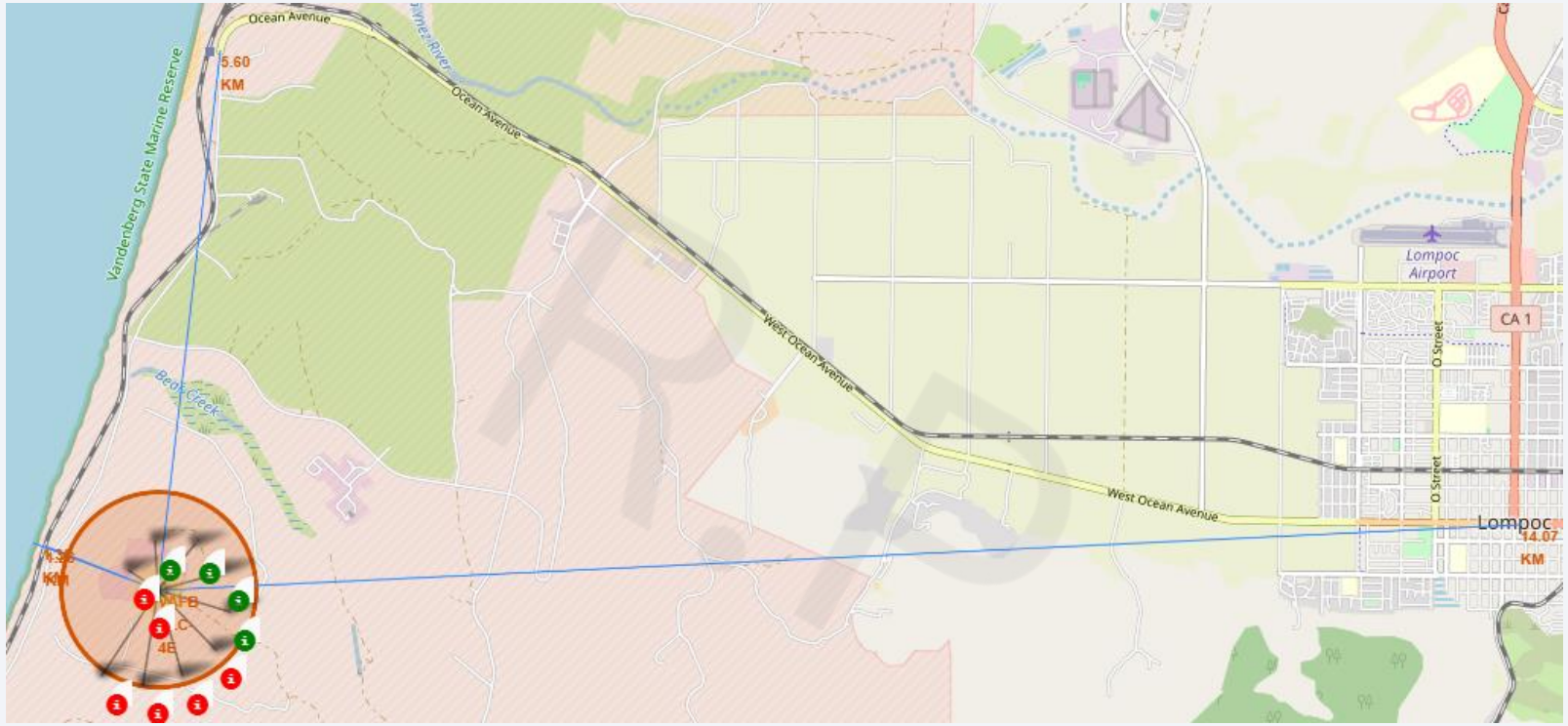
Also, the launch sites are in very close proximity to the coast. I believe the logic behind this decision is simply to minimize danger to human life in case a launch or a booster recovery goes wrong. Rockets, especially older ones, often drop spent boosters or stages during launch and, as bad as polluting the ocean is, dropping it into the ocean when it can't be recovered is a better decision than dropping it inland where it may threaten human lives. Fortunately, companies like SpaceX try to recover boosters for reuse (the motivation for this is surely saving massive costs but it works out well for dealing with the environmental concerns too).

Successful/Failed Launches for Each Launch Site

Marking the successful and failed launches for each site allows us to easily visualize the launch sites which have a relatively high success rate.

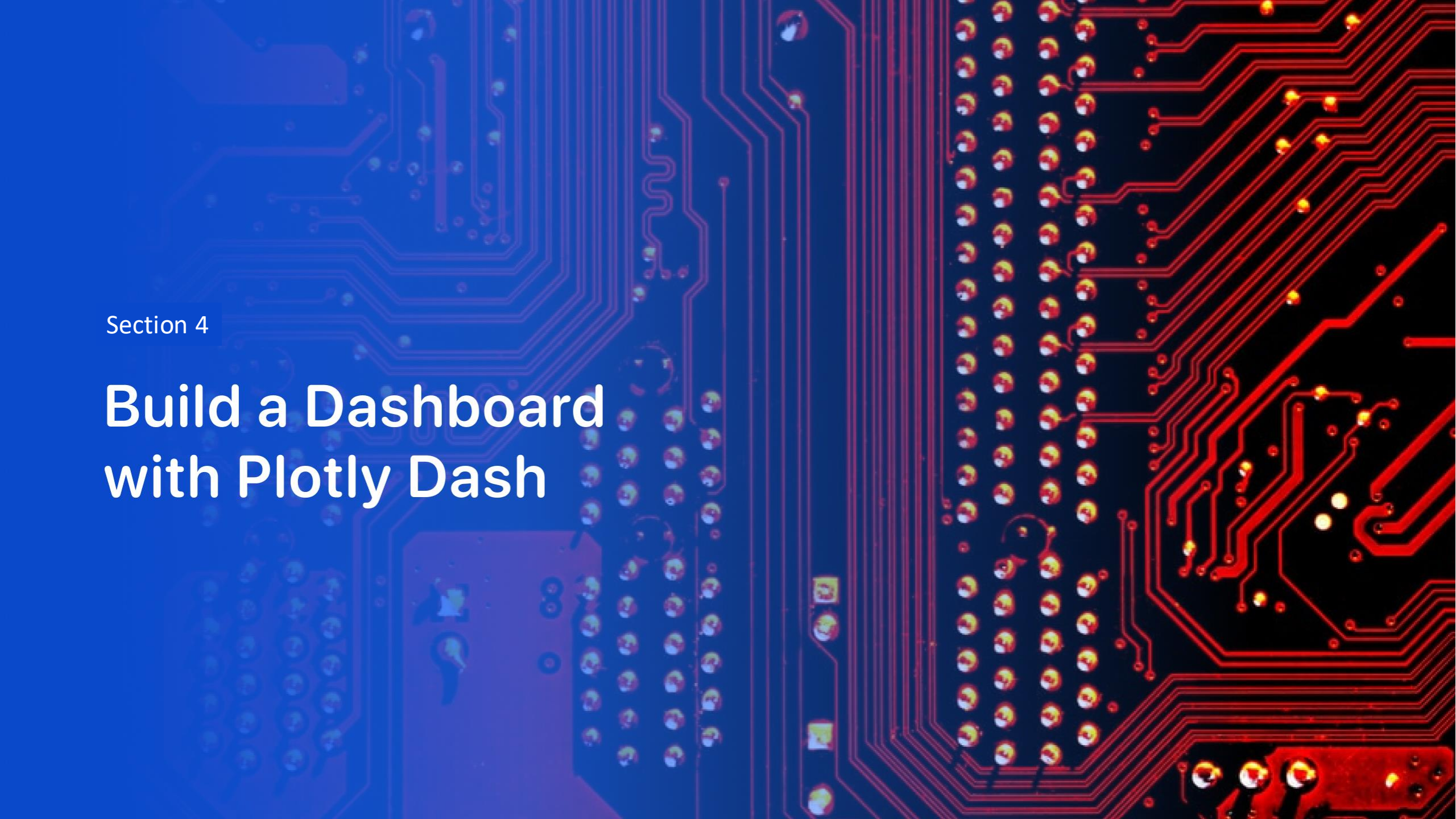
The smaller screenshot provided on the left is for the VAFB SLC-4E Launch Site and we can see that it has 4 successful launches out of 10 so a 40% success rate. In contrast to this, the KSC LC-39A Launch Site has a success rate of ~77%, CCAFS-LC40 has a success rate of approximately 27% and CCAFS-SLC40 of about 43%. We can clearly see that the KSC LC-39A Launch Site is the best performing Launch Site by a significant margin.

VAFB SLC-4E Launch Site's Distance to its Proximities

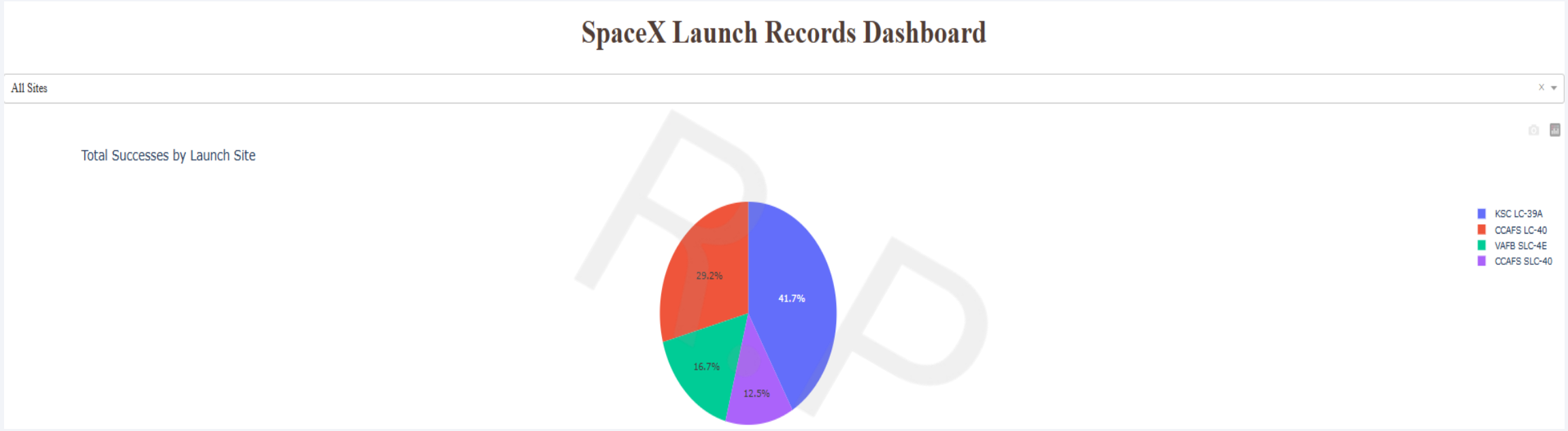


We can observe that the launch site is in close proximity to the coastline and railway (1.38km and 1.26km respectively). It is also relatively close to the nearest highway at around 5.6km. This facilitates well connected transportation and smooth movement of important parts/components.

The site also maintains a safe distance away from the nearest city (Lompoc) at about 14.07km. This positioning decision, along with the close proximity to the coastline, is a safety precaution in order to minimize threat to human life.


Section 4
Build a Dashboard with Plotly Dash

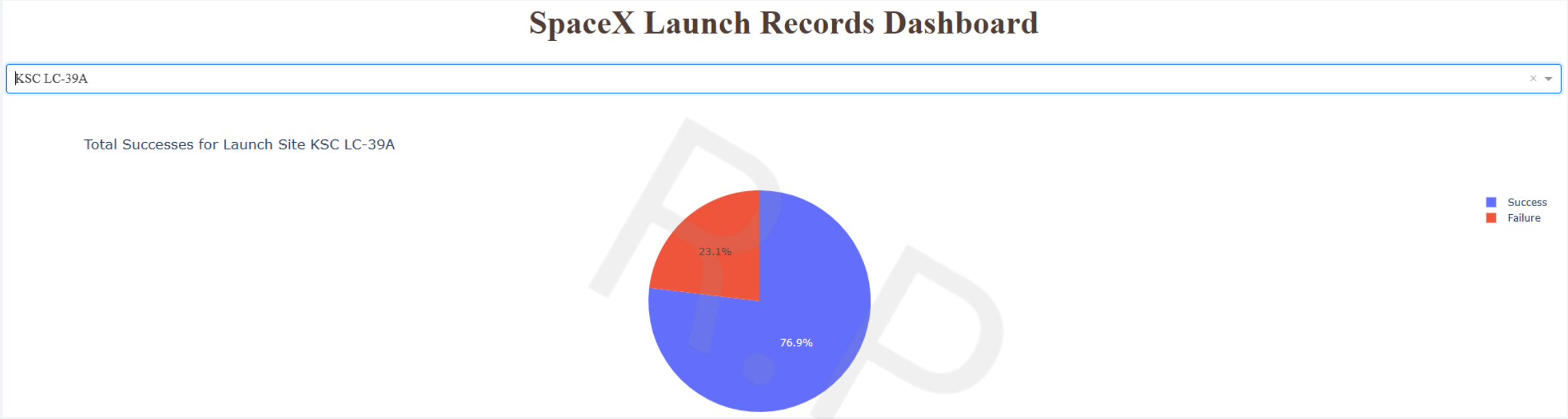
Launch Success Count for All Launch Sites



In this screenshot, we selected All Sites in the Dropdown menu in order to see the total successes by launch site. As shown by the legend on the right of the pie chart, the color corresponding to the maximum count of successes is the KSC LC-39A Launch Site with 41.7% of total successful launches over all sites. Lowest count of successful launches is the CCAFS SLC-40 with 12.5% of total successful launches.

However, this pie chart only takes into account total successful launches rather than % of successful launches so we must be aware of this when interpreting the chart.

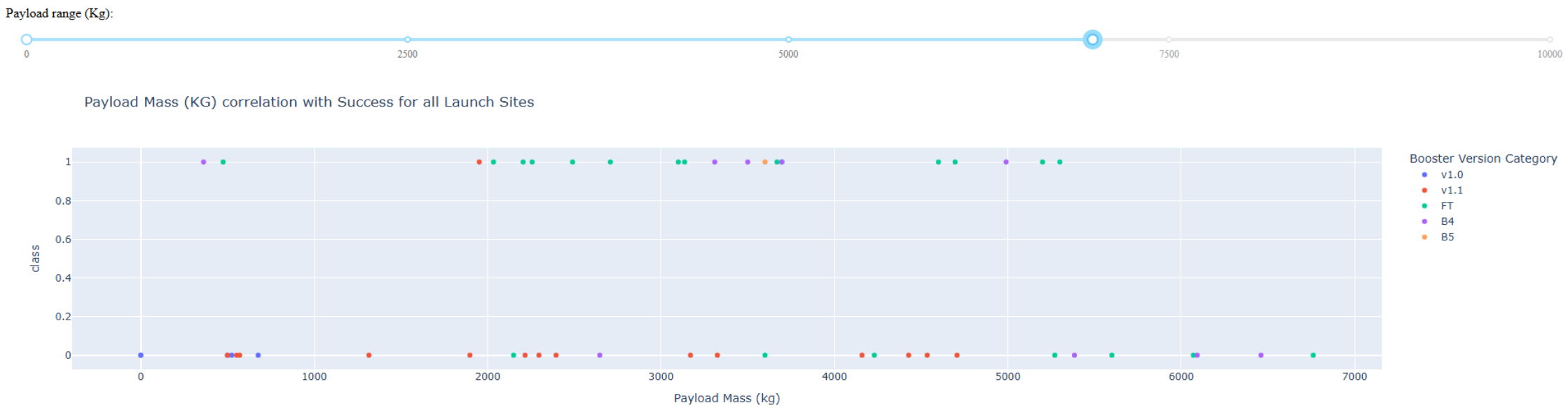
Launch Site with Highest Launch Success Ratio



The KSC LC-39A Launch Site has the highest launch success ratio at 76.9%. This is significantly higher than all of the other Launch Sites. If we also consider the previous pie chart from the last slide, we can see that this Launch Site has the highest count of successful launches as well as the highest success ratio. This is valuable information as launches could be scheduled here more often and also we could try to extract some features of this launch site in order to determine why it has a much higher success rate than others and use that to improve other sites.

Also, if we do not specify the color ordering in plotly, plotly will assign the first outcome it comes across as blue and the other outcome as red. If we want the pie charts to have consistent color for class labels for the different Launch Sites, we must explicitly mention this otherwise the class labels may have different colors for different launch sites based on whatever outcome was encountered first in the dataset.

Payload Mass Vs. Outcome in Payload range 0-7000KG



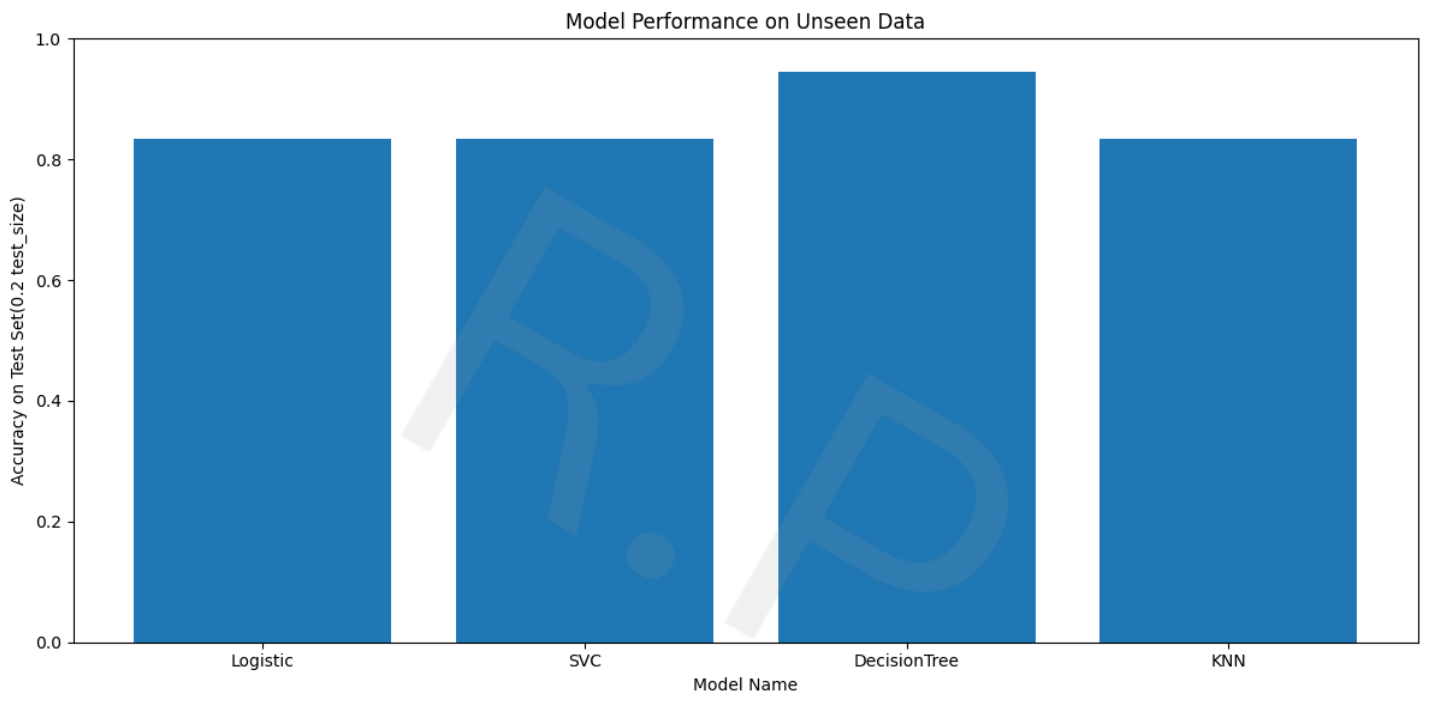
As we can see in the plot, launches with payload mass 0-1000kg and 6000-7000kg have a very low success rate and launches with payload mass 2000-4000kg have a relatively high success rate (in this payload range of 0-7000KG).

We can also observe that the Booster Version 'FT' seems to have a very high success rate relative to the other booster versions (ignoring B5 as it only has 1 data point so it has a perfect success rate but this is misleading with such little data)

Section 5

Predictive Analysis (Classification)

Classification Accuracy



When comparing the models' predictions on the test set data to the test set ground truth labels, the Decision Tree Classifier model achieved an accuracy score of 0.944 or 94.4%, beating the other models which all had a similar accuracy score of 83.3%. I used random_state=42 for reproducibility.

Disclaimer: When I first ran the notebook, all the models had the same accuracy score of 83.3%. When I ran it again, the DecisionTree classifier performed worse than the others. However, when setting the random state I happened to stumble upon one where it is the best performer. The Logistic Regression model on the other hand, seemed to have much more consistent performance over multiple runs. Therefore, in practice we might choose to go with another model since the DecisionTree classifier seems somewhat unstable.

Confusion Matrix

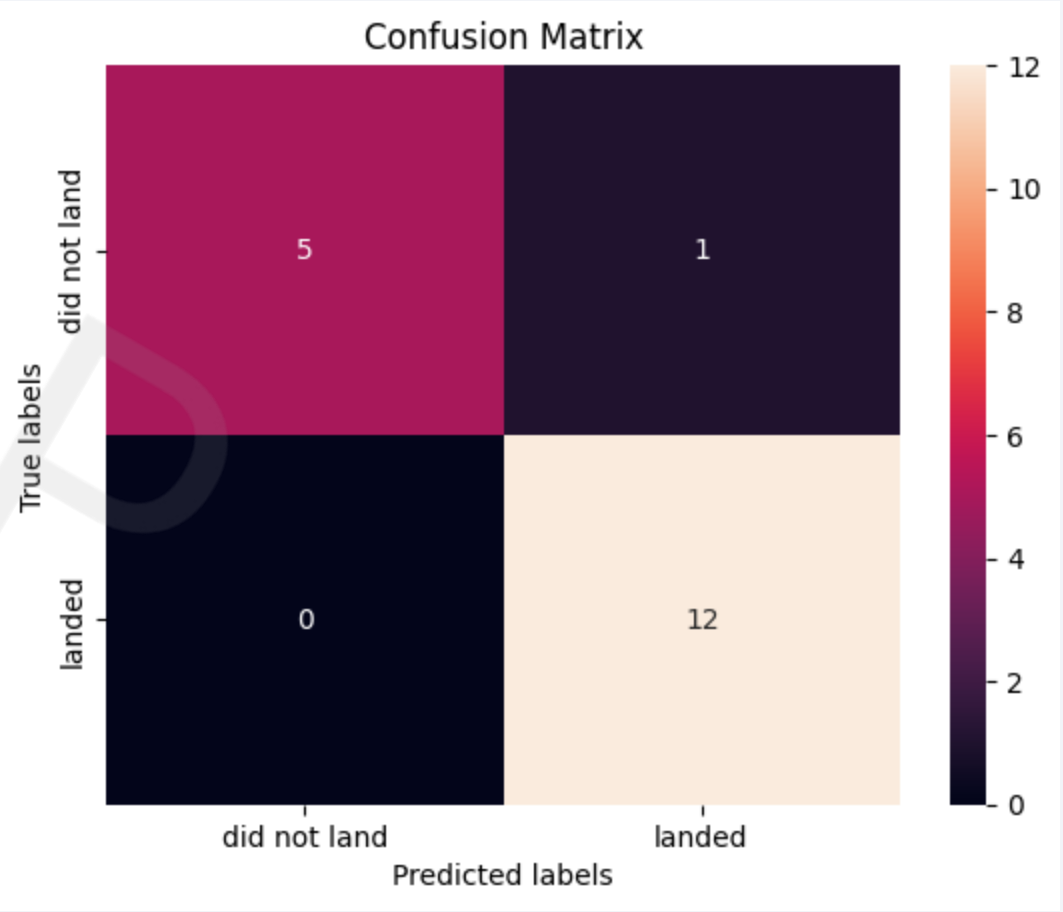
The best performing model in terms of accuracy score on test (unseen) data and confusion matrix analysis was the Decision Tree Classifier.

On the right we can see the Confusion Matrix for the Decision Tree Classifier Model (random_state=42 for reproducibility).

The model predictions are on the X axis with the Ground Truth Labels on the Y axis. The top left block represents True Negatives, top right is False Positives, bottom left is False Negatives and bottom right is True Positives.

We can see that the model correctly classified 17 out of 18 test data records and the incorrect classification was a False Positive meaning that it predicted success (landed) but in reality it was a failure (did not land).

This is consistent with the confusion matrices of the other models where they also struggled with False Positives rather than false negatives (they had more false positives but did not have any false negatives either, this is probably related to the train test split).

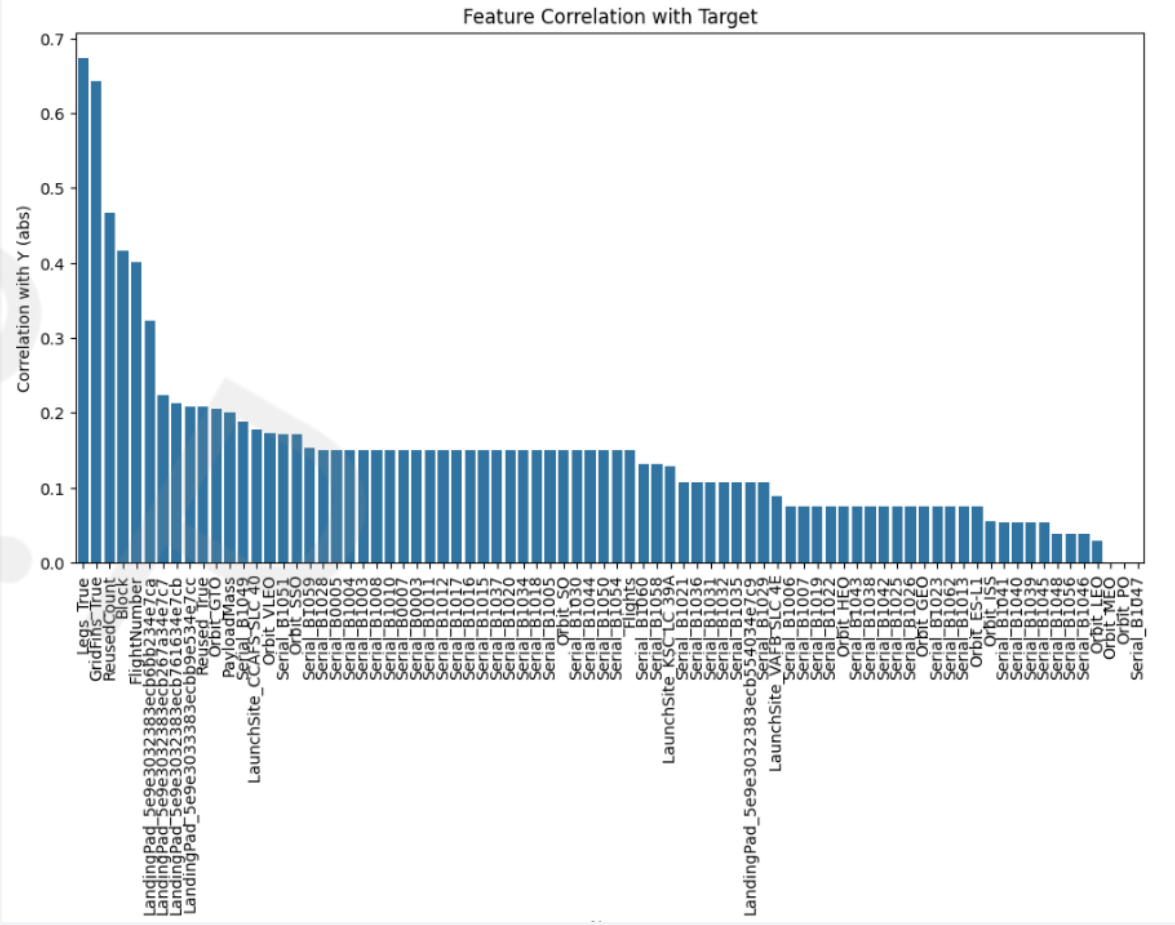


Conclusions

- We are able to, in theory, predict whether the Falcon9 first stage will land with a 94.4% accuracy rate using the Decision Tree Classifier. Practically, we might want to go with another more stable model that does not depend on random state as much (like Logistic Regression with an accuracy of 83.3%)
- While analyzing the Launch Data, we saw that the success rate of higher payload launches (over 10,000KG) is much higher than the success rate of lower payload launches.
- We also observed that the success rate is influenced by selection of orbit type (with SSO being the most successful) and booster version(with FT booster being the most successful). Additionally, we identified that the KSC LC-39A Launch Site has the highest total number of successful launches as well as the highest launch success rate ratio.
- Average Launch Success rate has steadily followed an increasing trend over time since 2013. The LEO orbit especially, has had an improved success rate after more launches.
- Launch Sites are strategically positioned as close to the equator as possible, close to coastlines and transportation lines like railways and highways and they are far from cities.
- Each of the trained classifier models seems to struggle with False Positives more than with False Negatives based on the existing train test split.
- For **future work** we may want to take a deeper look into some of the relationships we discovered and try to determine if we are confusing correlation with causality or if some seemingly interesting connections are actually coincidental. For example, we saw that the FT booster version seems to perform very well, we also saw that the KSC LC-39A Launch Site has a much higher success rate than the others. Is part of the reason for this because that is where the majority of FT boosters are launched from? Or maybe it's the other way around? Or maybe it's neither and the inferences we gained are completely independent from each other?

Appendix

- I performed a feature correlation analysis with the target variable and plotted the absolute values in the provided graph.
- Also, I implemented L1 regularization with Logistic Regression to see if it assigns a coefficient of 0 to any features and then tried re-training the models after removing the 0 coef features.
- Additionally, I performed more extensive hyperparameter testing and used stratified kfold rather than just regular cv=10. I also tried different sized test sets. This is all available under the "Extra" section in my final ML notebook.



- GitHub Main URL
- The screenshots on slide 30 and 32 look slightly different because I discovered some improvements I could make to the queries so I updated my sql eda file, took the new screenshots in the editor whereas the other screenshots were from the uploaded files on GitHub.

Thank you!