IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Andrew Clark

10th January 2024

Outline

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

Executive Summary – Data Science Methodologies

Executive Summary of Data Science Methodologies

- In this data science project, I have employed a comprehensive approach to data acquisition, processing, analysis, and visualization. My initial phase focused on data collection, utilizing advanced method such as data collection APIs and web scraping techniques.
- Once the data was gathered, I manipulated the data leveraging the power of Python and SQL. Python's versatility in data manipulation, combined with SQL's efficiency in handling database queries, allowed me to refine and structure the data effectively. This preparatory stage was crucial for ensuring data quality and integrity.
- Following data preparation, I conducted preliminary data analysis, primarily using SQL. This initial exploration provided valuable insights and guided subsequent analytical strategies. To complement my findings, I employed basic data visualization tools, notably Matplotlib, to depict my analyses clearly and concisely. These visual representations were instrumental in identifying patterns and trends within the dataset.
- Advancing my visualization techniques, I incorporated Folium and Plotly into the methodology. Folium, known for its interactive maps, enhanced the geographical data representation, offering a dynamic view of spatial patterns. Plotly, with its sophisticated graphical capabilities, allowed me to create more intricate and interactive visualizations. These tools brought a higher level of clarity and engagement to the presentation.
- Finally, the project culminated with predictive analysis. This phase was pivotal in forecasting trends and making data-driven predictions. By applying advanced statistical models and machine learning algorithms, I was able to provide meaningful insights and foresight into potential future scenarios.
- In summary, the data science project encompassed a thorough and strategic methodology, from initial data collection to advanced predictive analysis. Each step was carefully executed to ensure the most accurate, insightful, and valuable outcomes from our dataset.

Executive Summary – Summary of Results

Results

- **Results Summary ofMy Data Analysis and Forecasting Study**
- **Data Preparation:** Employed comprehensive methods for cleaning, analyzing, and visualizing data.
- **Machine Learning Application:** Tested various forecasting algorithms.
- **Key Finding:** Achieved highest accuracy (> 90%) with the Support Vector Machine (SVM) model.
- **Data Interpretation:**
  - Observed improvement in SpaceX's launch success rate over time.
  - Changes in the types of orbits requested have impacted launch outcomes, making this factor less predictive.
- **Model Effectiveness:**
  - SVM model provides accurate forecasts, suggesting some causal factors in launch success.
- **Broader Insights:**
  - Analysis has uncovered valuable insights into the nature of SpaceX's launch decision-making process.

Introduction

- **Introduction to Data Science Capstone Project**
- This project represents the culmination of my data science journey, showcasing a comprehensive analysis of SpaceX launches, leveraging datasets from SpaceX and Wikipedia. The core objective is to extract meaningful insights and develop predictive models concerning the landing success of SpaceX rockets.
- Key Focus Areas:

1. **Rocket Variants**: Analyzing different types of SpaceX rockets to understand how design and technology impact landing outcomes.
2. **Launch Sites**: Investigating various launch locations to reveal geographical influences on launch success rates.
3. **Payload Mass**: Examining the relationship between payload mass and the likelihood of successful landings.

- Methodologies:
- **Data Acquisition**: Sourcing data via APIs and web scraping, ensuring a rich and accurate dataset.
- **Data Wrangling and Analysis**: Utilizing Python and SQL for data cleaning, transformation, and preliminary analysis.
- **Visualization**: Employing tools like Matplotlib for initial visual representation, followed by advanced visualizations using Folium for geographical data and Plotly for interactive insights.
- **Predictive Modeling**: Applying statistical models and machine learning techniques to forecast launch outcomes.
- Outcome: This presentation will guide you through my key findings and demonstrate the data science techniques I've mastered. It aims to provide a deep understanding of factors influencing SpaceX rocket landings and showcase the potential of data science in aerospace analysis.

Section 1

Methodology

Methodology

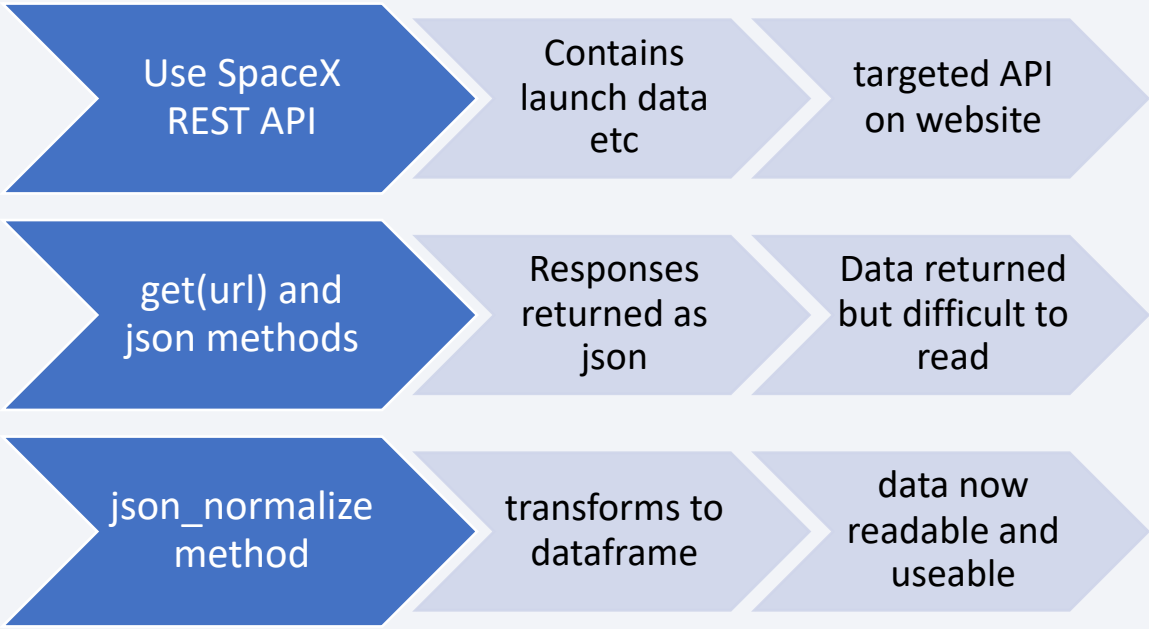
- **Executive Summary**
- **Data Collection Methodology:** Collected data from two distinct sources: a dynamic SpaceX API delivering data in JSON format and a static Wikipedia page. Techniques included API interfacing and web scraping using BeautifulSoup to extract and parse data from a complex HTML table.
- **Data Wrangling:** Processed the heterogeneous datasets to a uniform, clean, and analysis-ready state. Involved parsing JSON, transforming HTML table data, and normalizing various data formats using Python's data manipulation libraries.
- **Exploratory Data Analysis (EDA):** Utilized SQL for database-driven insights and Python's Matplotlib for preliminary visual trends. This stage laid the groundwork for identifying key factors affecting rocket landing success.
- **Interactive Visual Analytics:** Advanced our visual analytics with Folium for mapping launch site data and Plotly Dash for creating interactive, user-driven visualizations to explore complex relationships within the data.
- **Predictive Analysis Using Classification Models:** Developed classification models to predict landing outcomes. Involved splitting data into training and testing sets, feature selection, and model tuning to enhance predictive accuracy.
- **Model Building, Tuning, and Evaluation:** Iterative process of building and refining models with techniques like cross-validation and hyperparameter optimization. Evaluated models using metrics such as accuracy, precision, recall, and F1-score to ensure robust predictive performance.

Data Collection

- **Data Collection Methodology**
- This project involved the meticulous gathering of data from two primary sources, each presenting unique challenges and requiring specialized methods for extraction and formatting.
- **1. SpaceX API:**
- **Source**: Accessed data directly from a web-based API provided by SpaceX.
- **Format**: Data was available in JSON format, which posed challenges in terms of extraction and conversion into a user-friendly structure.
- **Process**: Developed a robust methodology to parse JSON data, ensuring its transformation into a format suitable for detailed analysis.
- **2. Wikipedia - Public Data:**
- **Source**: Leveraged publicly available data from a static page on Wikipedia.
- **Challenge**: The task involved navigating a complex webpage to locate the specific dataset required for our analysis.
- **Method**: Employed web scraping techniques, utilizing BeautifulSoup to accurately extract information.
- **Focus**: Identified and extracted data from the third table on the page, which was then processed to convert it into a workable format.
- **Outcomes:**
- The SpaceX API provided real-time, dynamic data, offering insights into recent launches and technological advancements.
- The Wikipedia data, while static, contributed valuable historical context and supplementary information.
- Both data sources were integral in constructing a comprehensive dataset that underpins my analysis, offering a blend of current and historical perspectives on SpaceX's rocket launches.

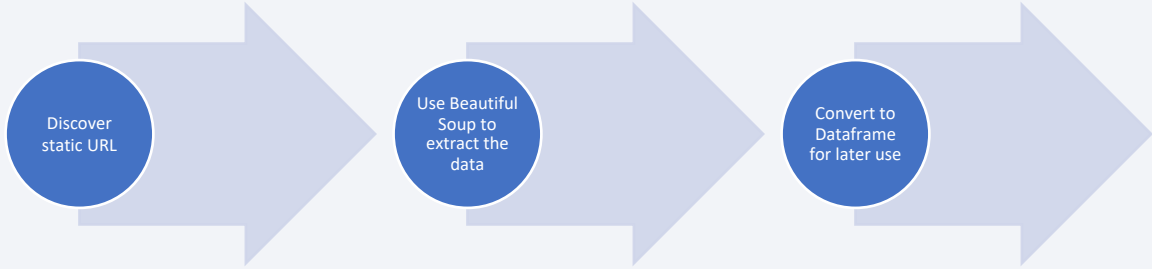
Data Collection – SpaceX API

- **Data Collection** was completed using the SpaceX REST API which covers launch, rocket, core, capsule, launchpad data etc from SpaceX Falcon Rocket launches.
- Our goal is to understand whether SpaceX will attempt to land a rocket, or not.
- The API can be found at this link.
- Requests were made using the get(url) and json() methods with responses being returned as json.
- I used the json_normalize method to turn the data into a dataframe for the purposes of this project.
- Later in the project I use web scraping with Beautiful Soup to retrieve the required additional information.
- The GitHub URL of the completed SpaceX API calls notebook is here.



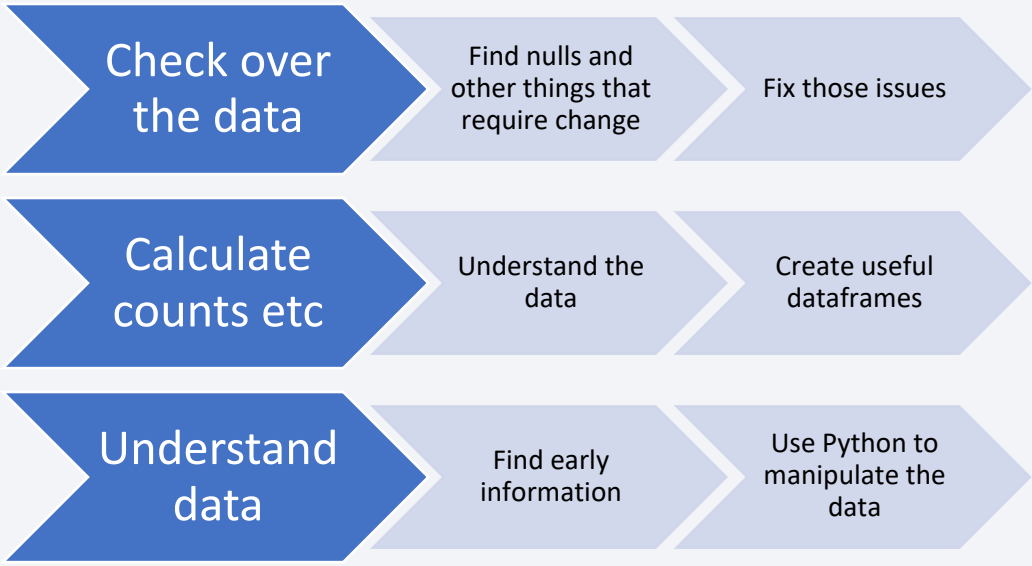
Data Collection – Web Scraping

- I performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page. A static URL was used with the link here.
- The launch records were stored in an HTML table. I used Beautiful Soup to extract first the column names and then the data. Which I then converted into a dataframe for later use.
- The GitHub URL of the completed web scraping notebook can be found here.



Data Wrangling

- A number of tasks needed to be completed before we could start using the data:
- I had a look around the data to see how many nulls, for example, were in each column and checked the types of the data to ensure they were in athe correct format
- I calculated the number of launches from each of the three launch sitesa
- I calculated the number of each type of the eleven orbit types
- I then calculated the number of each type of mission outcome
- I then created the set of bad outcomes to use in later queries.
- I tallied up the bad outcomes and good outcomes and came up with a success rate.



EDA with SQL

- As part of the work to understand the data further, I imported the data to a SQL database (sqllite) and made a number of queries:
- First I looked to understand the names of the Launch Sites, the total payload mass of the boosters launched by NASA and the average payload of the F9 rockets.
- To understand the data further I did a search to understand when the first successful landing outcome was, the number of successful outcomes and some more esoteric queries such as which booster was used for certain wrights of payloads.
- Finally I ranked the count of landing outcomes between two dates in descending order
- The GitHub URL of my completed EDA with SQL notebook can be found here.

Build an Interactive Map with Folium

- Folium allows us to build interactive visualizations. The task was to plot the locations of launch sites on a map for any insight that might give.
- I added the four launch sites to the map to see if there was an insight that could be gained. One thing of note is that all the launch sites are on the coast, presumably reducing the risk of danger to humans and allowing the rockets to be recovered at sea.
- Next I wrote a function that discriminated between red and green for success and failure and plotted those insights onto the map. Zooming in on the map allows interested parties to see the relative success of the different sites.
- Then I added a Mouse Position to get coordinates for mouse points. This is part of some work to analyse the proximities of the launch site. Choosing a location on the coast I was then able to put in a line that showed the distance from the shore.
- The GitHub URL of my completed interactive map with Folium map can be found here.

Build a Dashboard with Plotly Dash

- I used Plotly to create a dynamic visualization of launch success data for the various sites. The interactivity showcases the ability of Plotly to react to different real time searches.
- By creating a pie chart, a select box, a slider and a scatter plot I was able to show the data in a format that would be ideal for senior stakeholders etc and one that would keep them interested.
- A small amount of insight was gained through the use of Plotly but all could have been done elsewhere. This was really just used to showcase what it could do in this case.
- The GitHub URL of my completed Plotly Dash lab can be found here.

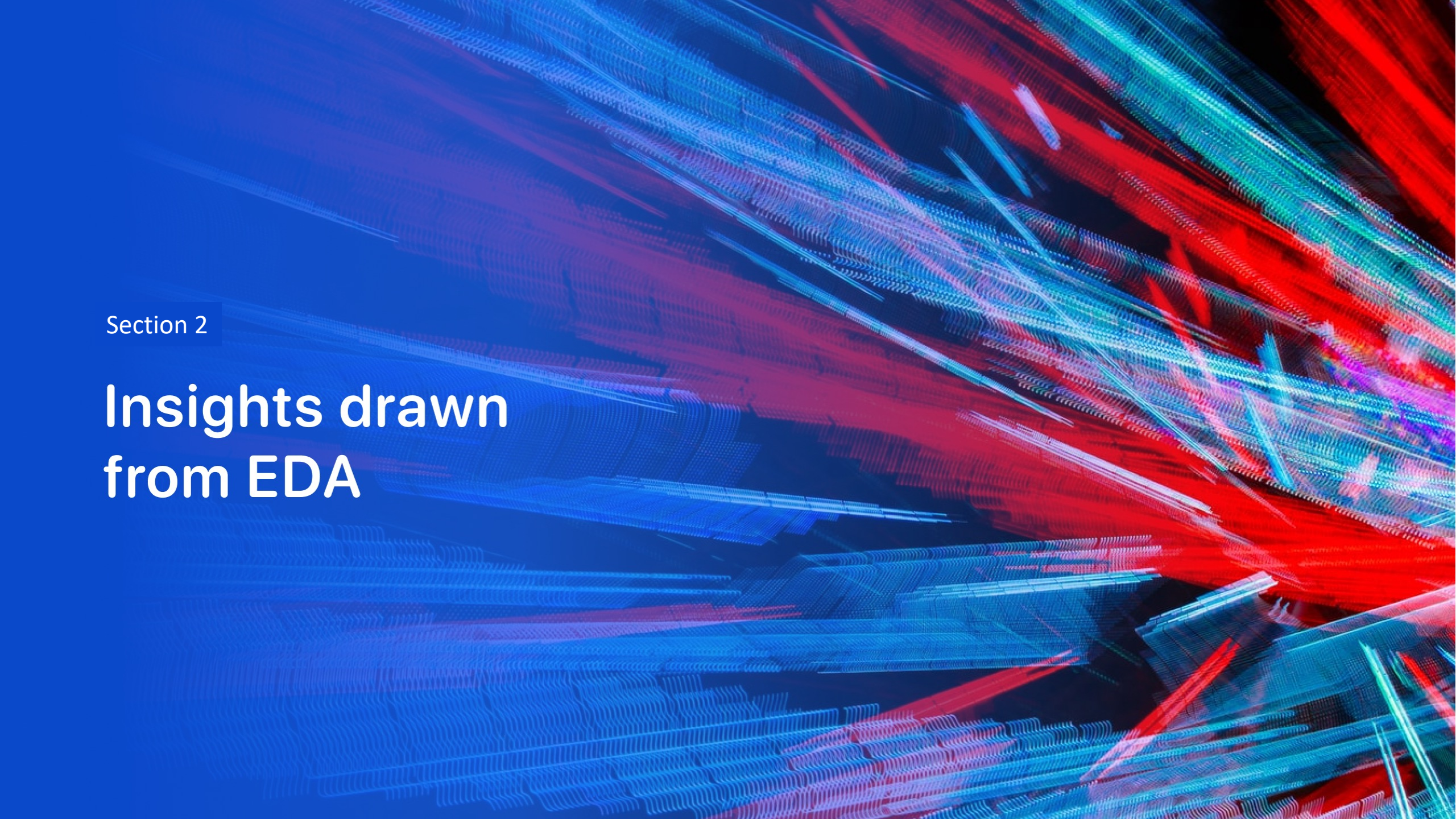
Predictive Analysis (Classification)

- The aim of this piece of work was to predict whether the first stage of Falcon 9 would land successfully. This was done by preprocessing data, using train test split to split our data into training data and testing data. I used features from the sklearn library for this analysis.
- I used Logistic Regression, SVM, Decision Tree Classifier, and K-nearest Neighbours in the analysys to output a Confusion Matrix
- Github URL is here.

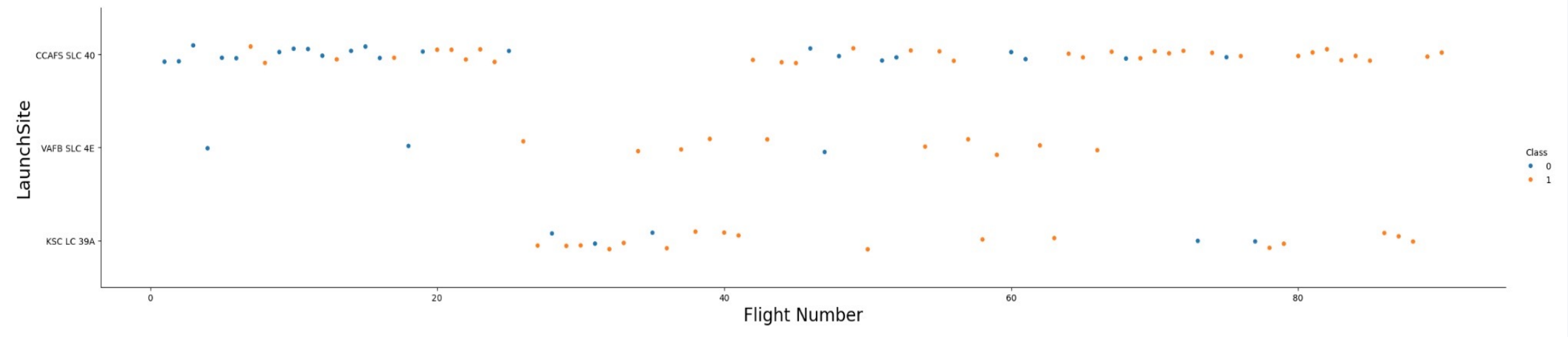
Results

Results

- **Results Summary ofMy Data Analysis and Forecasting Study**
- **Data Preparation:** Employed comprehensive methods for cleaning, analyzing, and visualizing data.
- **Machine Learning Application:** Tested various forecasting algorithms.
- **Key Finding:** Achieved highest accuracy (> 90%) with the Support Vector Machine (SVM) model.
- **Data Interpretation:**
  - Observed improvement in SpaceX's launch success rate over time.
  - Changes in the types of orbits requested have impacted launch outcomes, making this factor less predictive.
- **Model Effectiveness:**
  - SVM model provides accurate forecasts, suggesting some causal factors in launch success.
- **Broader Insights:**
  - Analysis has uncovered valuable insights into the nature of SpaceX's launch decision-making process.

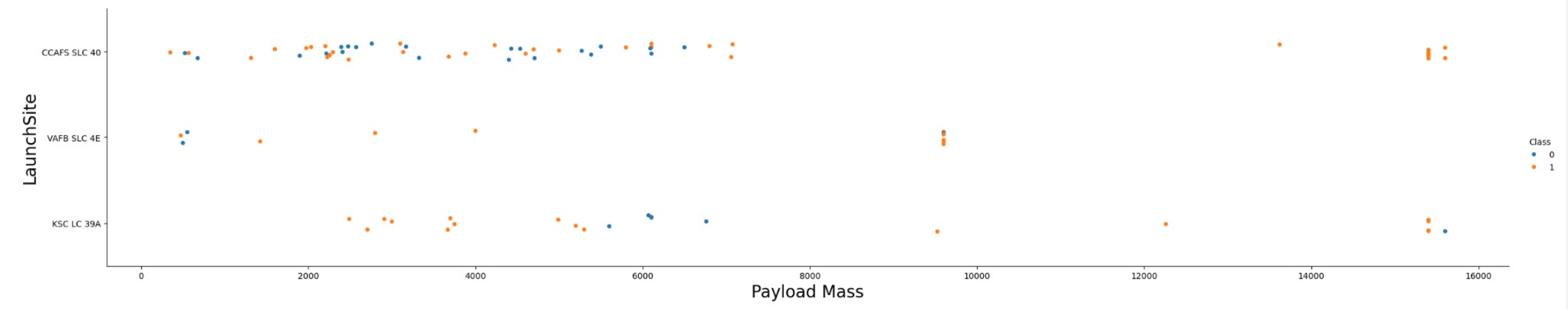

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- The scatter plot above shows Launch Site vs Flight Number and, via the key on the right, shows whether the rocket successfully or unsuccessfully landed.
- Earlier flight number are to the left which show a large number of failure to land. As flight numbers and time progress forward we can see that an increasing number of landings are successful. Presumably a function of SpaceX getting better with experience.

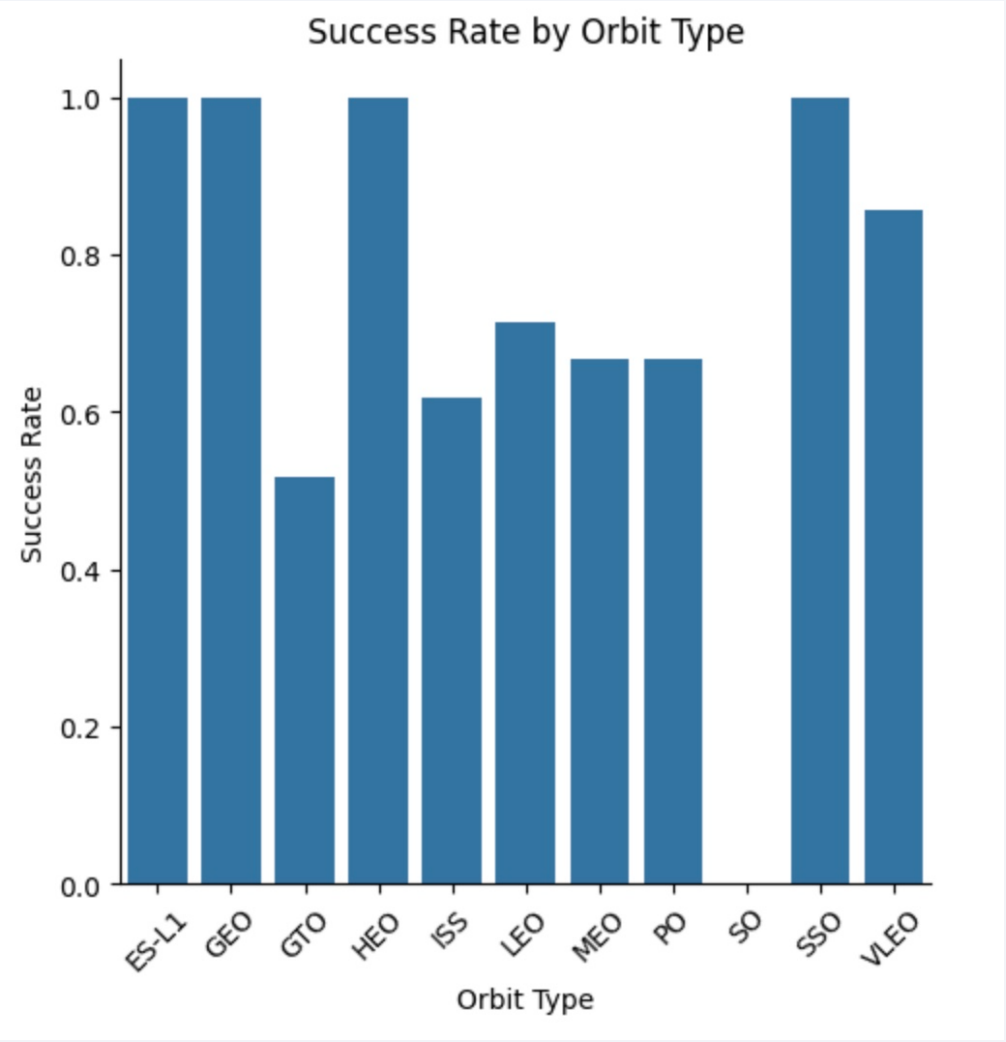
Payload vs. Launch Site



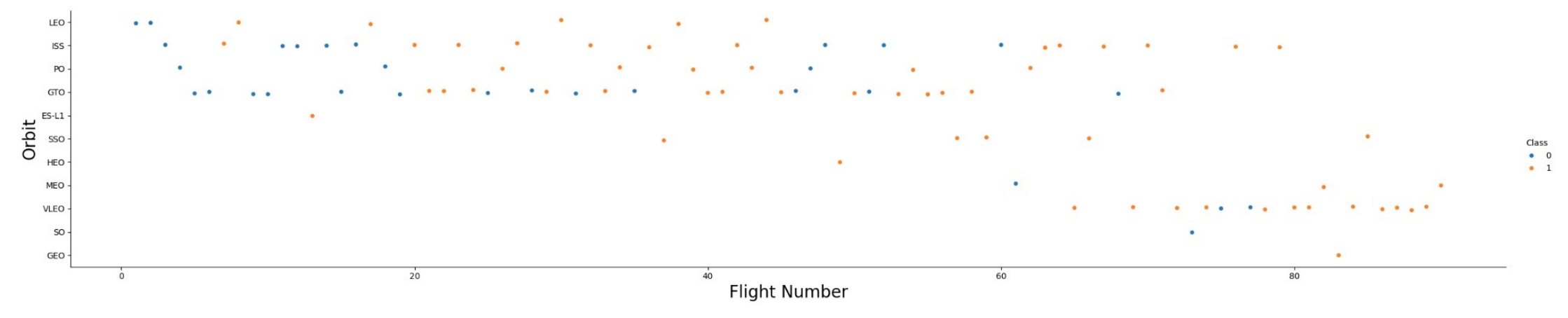
- This scatter plot shows payload versus launch site and identifies whether the recovery was successful or not.
- We can see a preponderance of unsuccessful launches with smaller payloads, and a clearly much more successful set of launches with larger payloads.
- We should check to see whether larger payloads just happened later on as the confidence levels increased rather than assuming that higher payloads are just more successful

Success Rate vs. Orbit Type

- The bar chart to the right shows the success rate by orbit type.
- While it is clear some orbits have a great success rate it is not clear that anything should be read into this. Some of these orbits, we have seen, were only used early on in the development of SpaceX rockets.

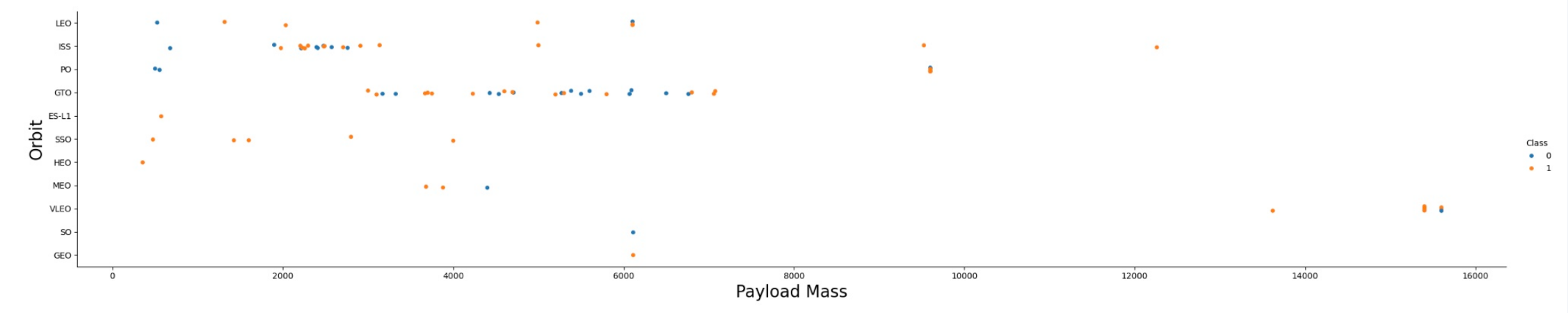


Flight Number vs. Orbit Type



- The scatter plot above shows the relationship between Orbit and Flight Number.
- The scatterplot clearly shows the different types of orbits that were being sought changing over time, presumably as SpaceX set their sights higher and higher.
- It is worth noting that some orbits have very high success rates while others are more hit and miss. Something worth investigating further.

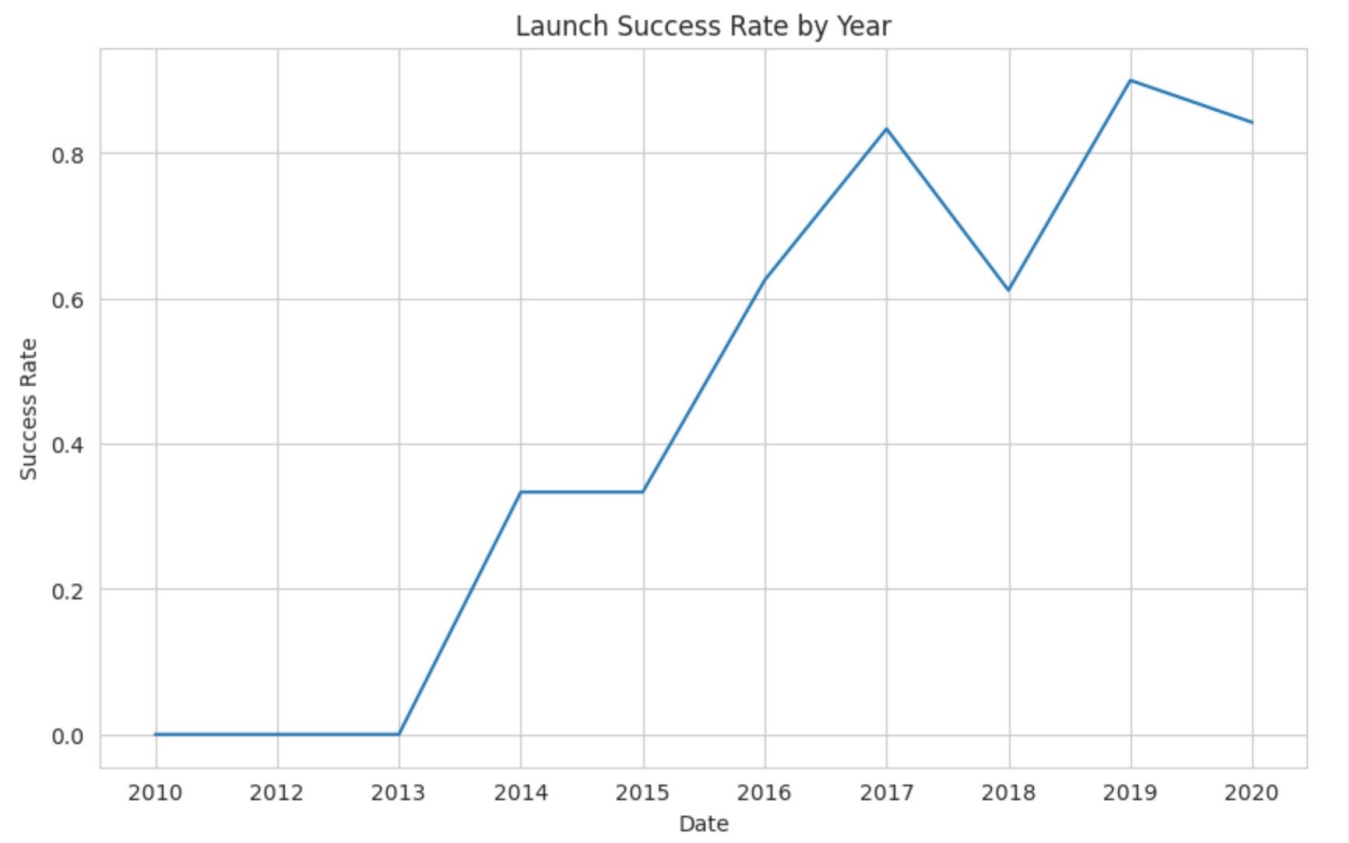
Payload vs. Orbit Type



- Payload vs Orbit type offers another view of the data.
- It is clear that the payload mass is a feature of different orbit types. The heaviest loads by some margin are within one orbit type, and we can also see how the mass for certain types clusters around low or medium payloads. Something to understand.

Launch Success Yearly Trend

- On the right hand side, I've plotted the success rate by year.
- This plot is very insightful as it shows how SpaceX have improved their success rate.
- There is an interesting trend in 2018, I would be interested to know what caused this, a change of launch site perhaps or a change into different orbits.



All Launch Site Names

- To find the names of all the Launch sites I did a simple select DISTINCT query from the main table called SPACEXTBL. The code and results and posted in the image to the right.

Task 1

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

```
%sql select DISTINCT "Launch_Site" from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- To display all records containing CCA is performed a simple SELECT statement but used the WHERE clause and pattern matched CCA%. I then LIMITed my query to 5 records as can be seen in the screen grab on the right.

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%'
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

Total Payload Mass

- To calculate the total payload mass carried by the boosters I did a standard SELECT query but used the SUM function against the column “Payload_Mass_KG_” and limited the results to records which pattern-matched “%NASA%”.

Task 3

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

```
[10]: %sql SELECT SUM("PAYLOAD_MASS_KG_") AS "PAYLOAD" FROM SPACEXTBL WHERE "Customer" like '%NASA%'
 * sqlite:///my_data1.db
Done
```

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by F9 boosters my query used the AVG function against the Total Payload KG column in the databases. This result was filtered for where the Booster version was pattern-matched to “F9%”

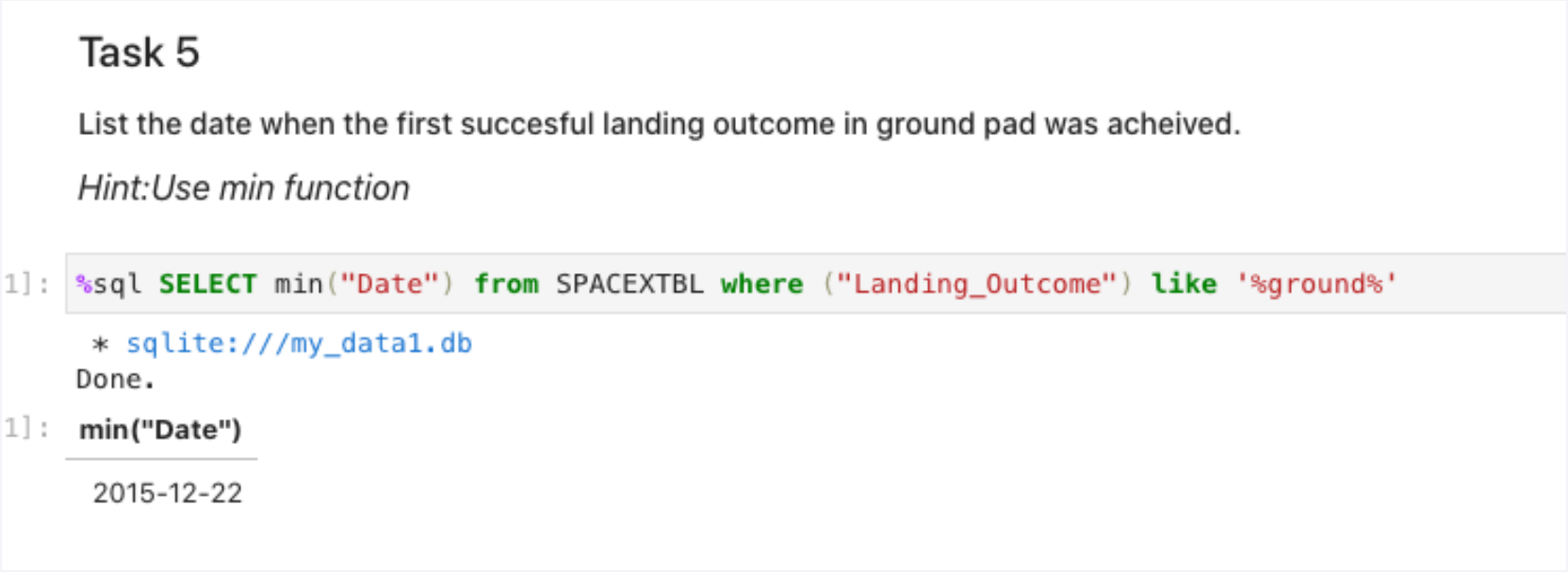
Task 4

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

```
49]: %sql SELECT AVG( ) AS "Average_Payload" FROM SPACEXTBL WHERE "Booster_Version" like "F9%"
 * sqlite:///my_data1.db
Done.
```

First Successful Ground Landing Date

- To find the first successful Ground Landing Date I used the min function against the Data column in the database and filtered for where Landing_Outcome was pattern-matched to "%ground%".
- The earliest successful ground landing date was the 22nd of December 2015



Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
1]: %sql SELECT min("Date") from SPACEXTBL where ("Landing_Outcome") like '%ground%'
```

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

1]:

min("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To discover the list of boosters that had a successful landing with a payload greater than 4000 and 6000 I used the BETWEEN function against the PAYLOAD_MASS_KG_ column in the database.

Task 6

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

```
]: %sql SELECT "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS_KG_" BETWEEN 4000 and 6000
 * sqlite:///my_data1.db
Done.
```

Total Number of Successful and Failure Mission Outcomes

- As my query didn't fit on the page I have pasted it here instead with a picture of the results:
- Query was: SELECT SUM(CASE WHEN Mission_Outcome LIKE '%success%' THEN 1 ELSE 0 END) AS Successful_Launches, SUM(CASE WHEN Mission_Outcome LIKE '%failure%' THEN 1 ELSE 0 END) AS Unsuccessful_Launches FROM your_table_name;

Boosters Carried Maximum Payload

My query was as follows:

```
SELECT "Booster_Version"

FROM SPACEXTBL

WHERE "PAYLOAD_MASS__KG_" = (

    SELECT MAX("PAYLOAD_MASS__KG_")

    FROM SPACEXTBL

);c
```

2015 Launch Records

Did not complete

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

My query for this one looked like this:
%sql SELECT Landing_Outcome, COUNT(*) as Outcome_Count FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC

This did a count of outcome between the two dates specified which I then ordered. The DESC function then lists the results from the most to the least.

Task 10

Rank the count of landing outcomes (such as Failure (dror
2017-03-20, in descending order.

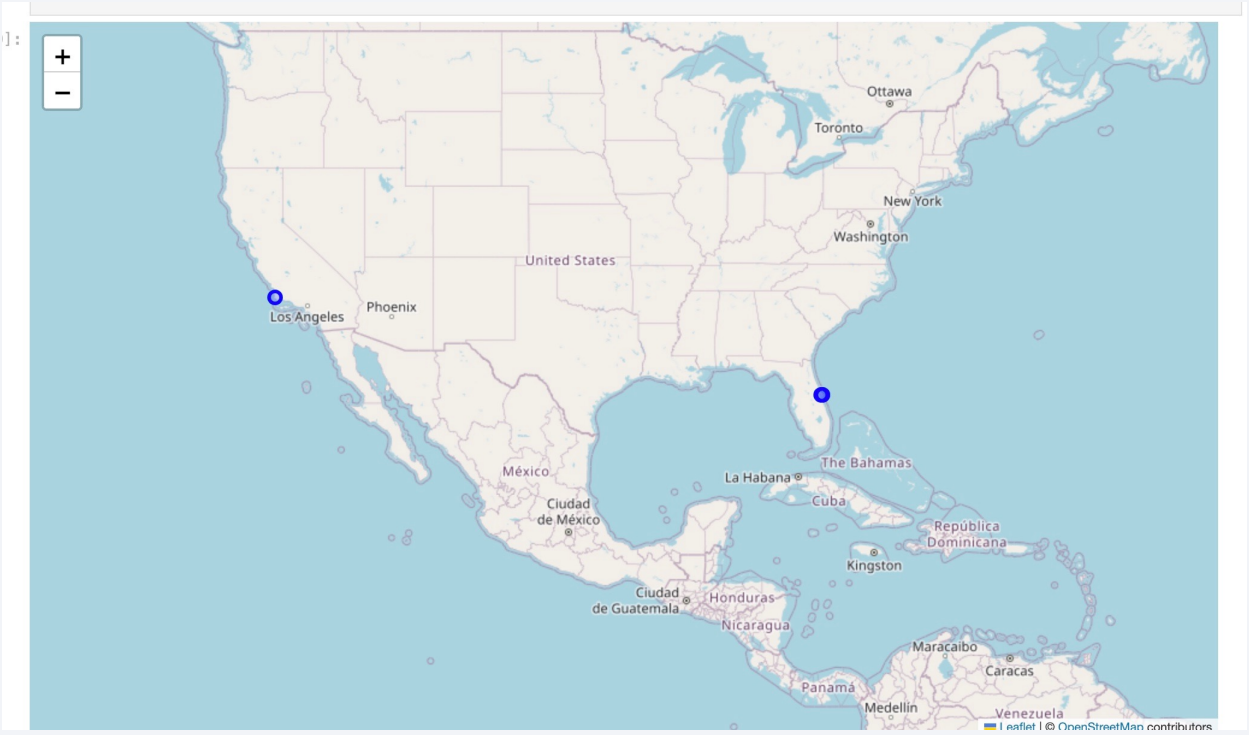
```
%sql SELECT Landing_Outcome, COUNT(*) as Outcome_
 * sqlite:///my_data1.db
Done.
```

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


Section 3
Launch Sites
Proximities Analysis

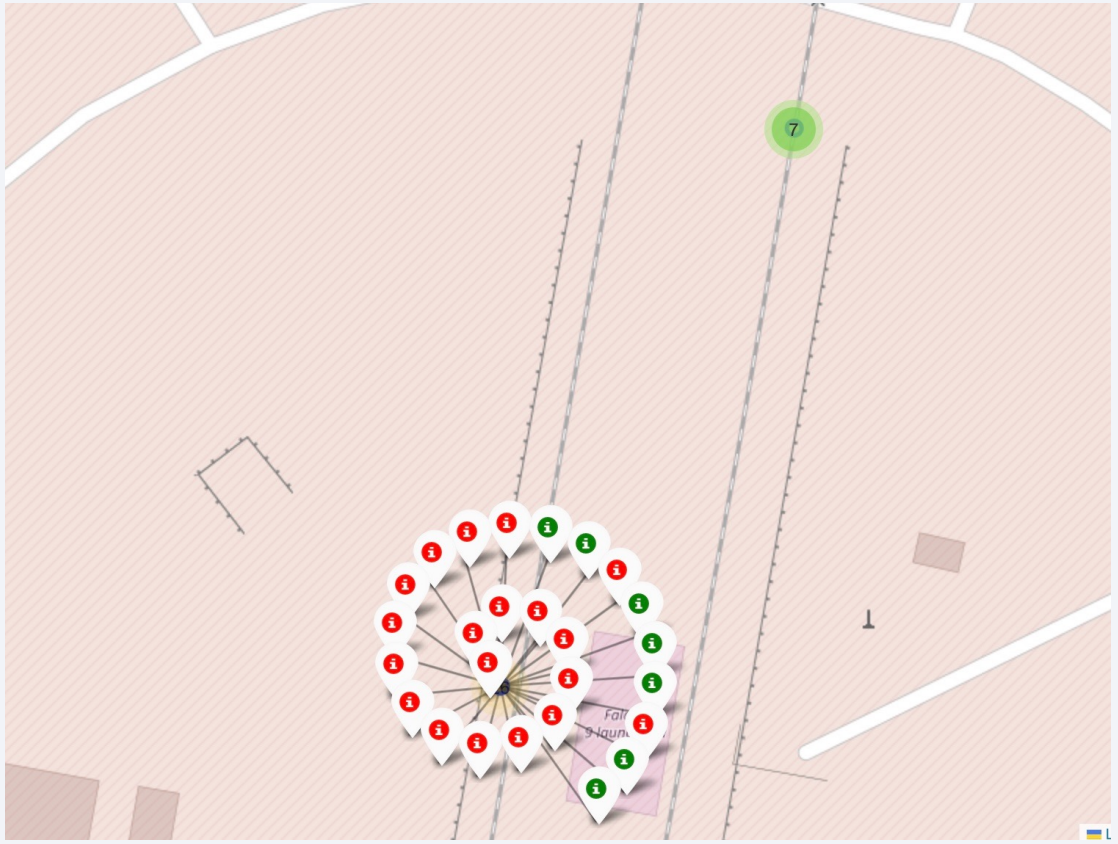
Folium – Plotting Launch Location

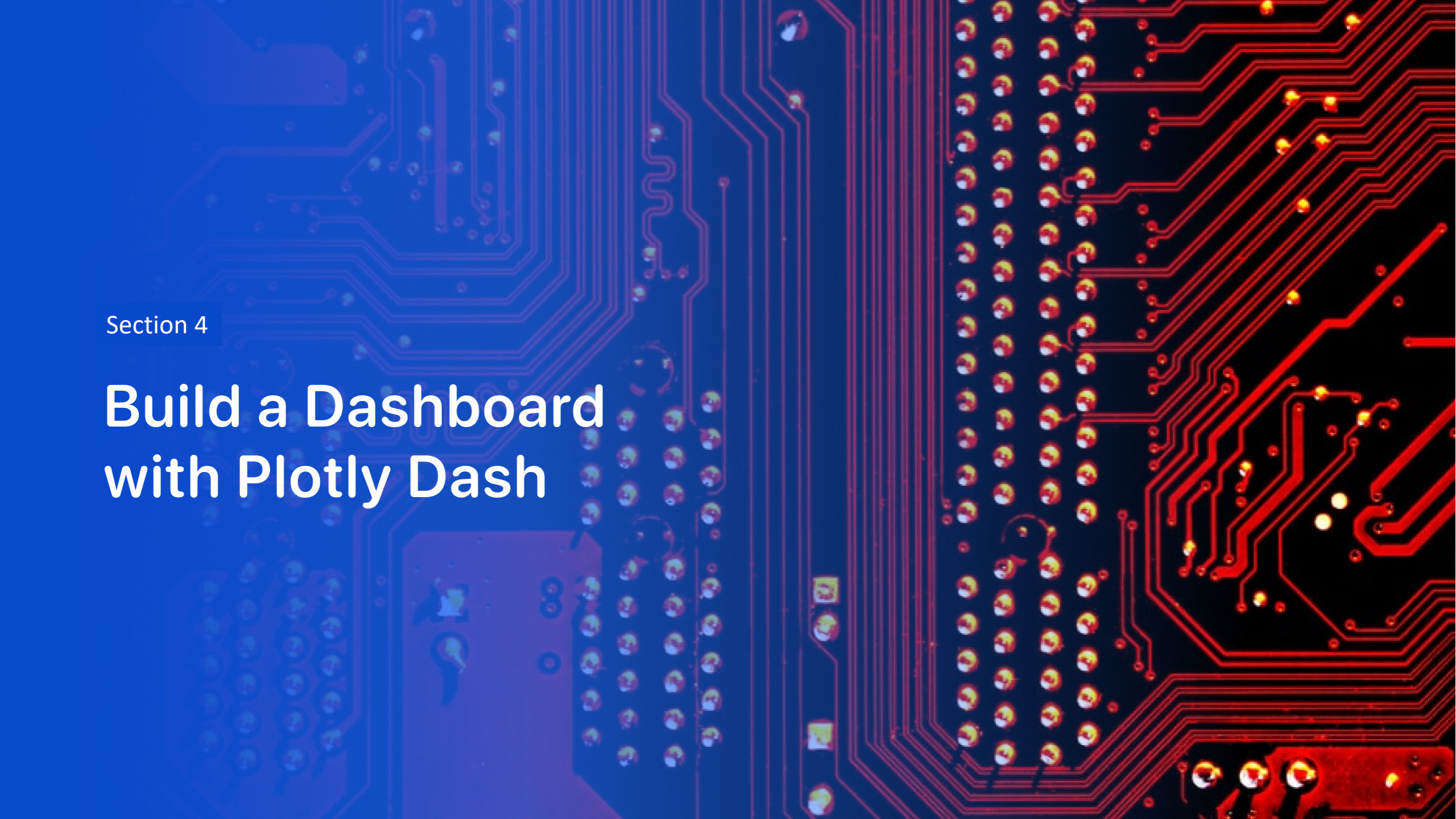
- I was able to plot the launch locations on the embedded map. By zooming in you can see that the launch locations are both close to each other (in one case two different landing pads) and also close to the see.
- Clicking on the circles with provide a popup with the name of the launch site.
- The coastal locations and distance from habitation were the key findings of this piece of work.



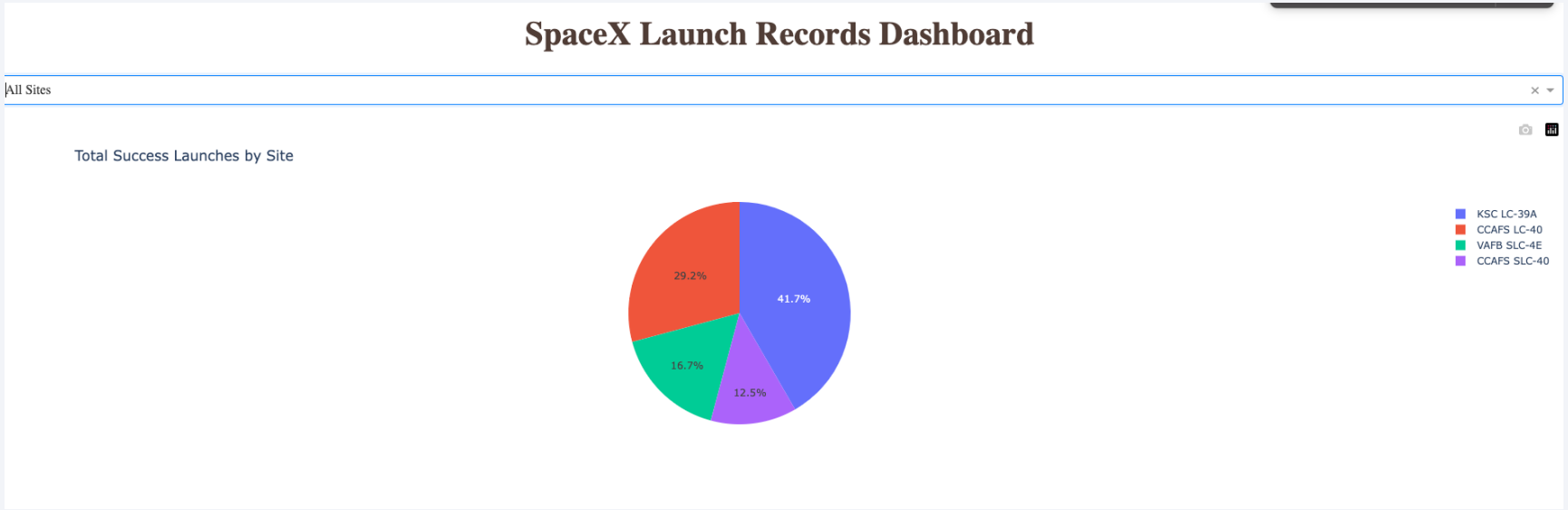
Folium 2 – Zooming in

- I was able to plot a label for each of the launches using Folium and assigning red or green to success or failure to land.
- In the top right you can see the second local launch site. Clicking on that also shows the success/failure of each launch.




Section 4
Build a Dashboard with Plotly Dash

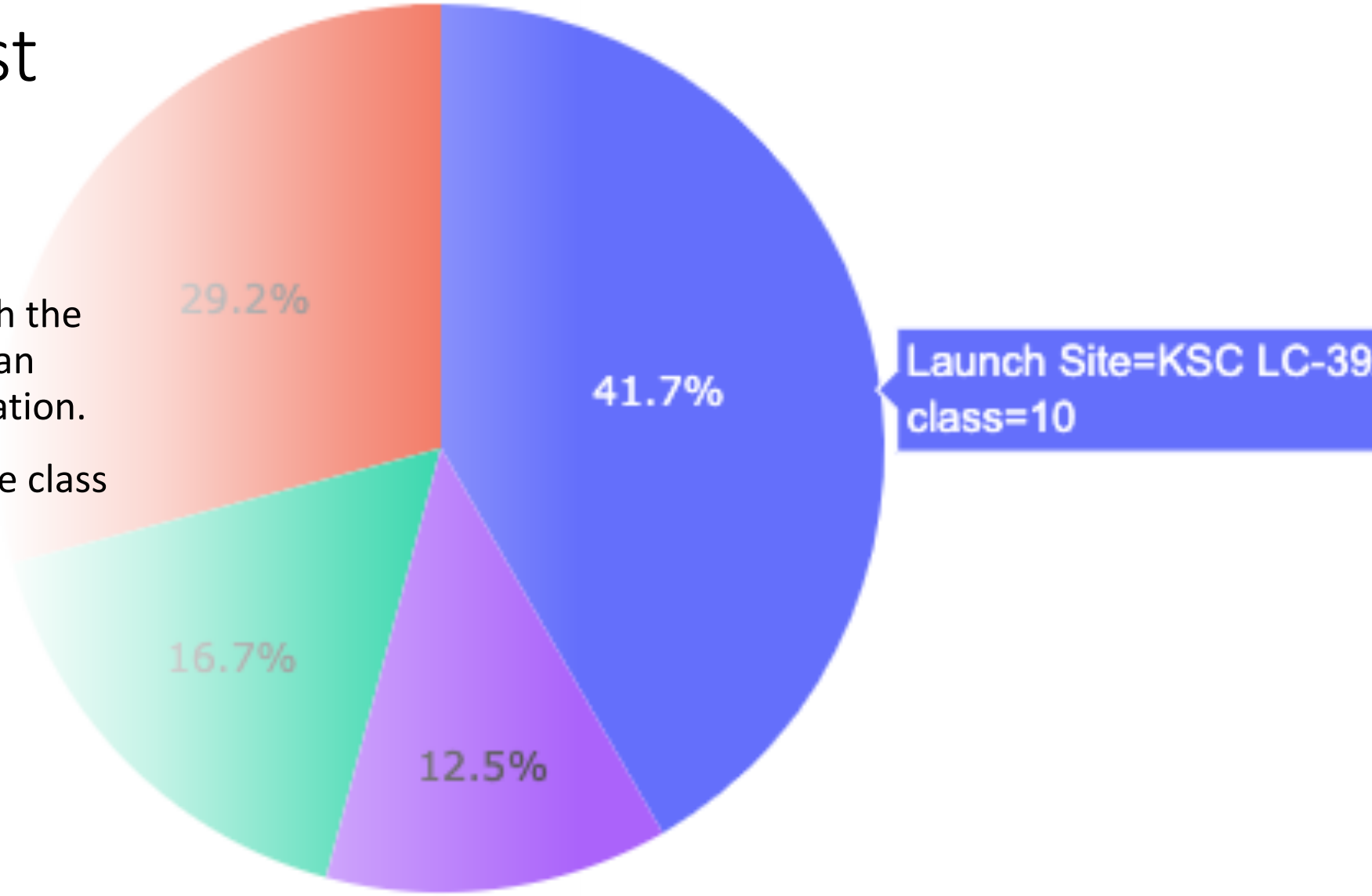
Plotly – Launch Success Count Pie Chart



- This pie chart shows the launch success rate in the four different sites used for launches. You can drill down further by clicking on the selector above the pie chart.
- Zooming in and hovering the mouse shows additional data about the selected piece of pie.

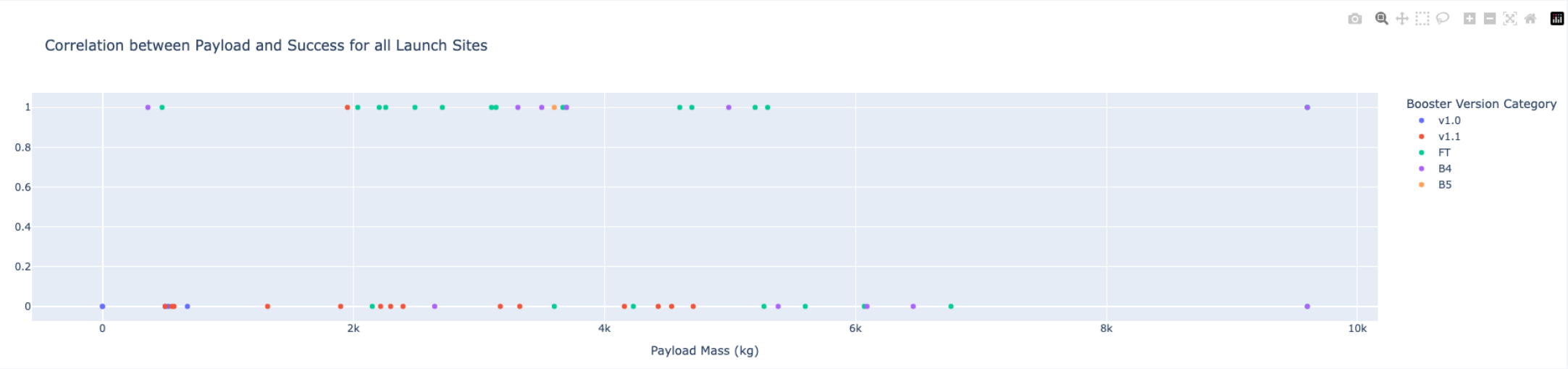
Plotly – Highest Success Ratio

- Hovering over the site with the highest success ratio we can provide additional information.
- In this case we see that the class = 10



Plotly – Payload vs Launch Outcome for each site

- Finally we are interested in showing the relationship between payload and launch outcome for each site.
- The Scatter point below shows the relationships.
- It is worth noting that the FT type booster version category has a very high success rate which is easily visible on this diagram. As various payload amounts.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

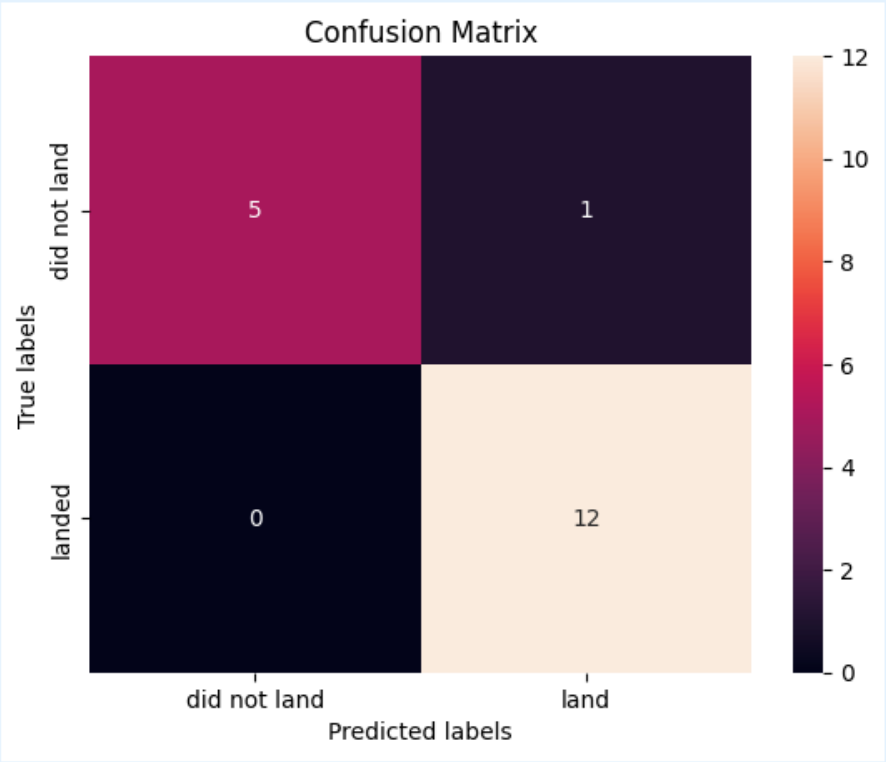
- Did not complete

Confusion Matrix

- The confusion matrix is a tool often used in machine learning to visualize the performance of a classification algorithm. It is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. This allows you to see how well your classifier is doing and, importantly, in what ways it might be failing.
- Here's a breakdown of each term:

1. **True Positives (TP):** These are cases in which the model correctly predicted the positive class.
2. **True Negatives (TN):** These are cases in which the model correctly predicted the negative class.
3. **False Positives (FP), also known as Type I Error:** These are cases in which the model incorrectly predicted the positive class.
4. **False Negatives (FN), also known as Type II Error:** These are cases in which the model incorrectly predicted the negative class.

```
[18]: yhat=lr_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Various models were tried using the dataset that we were given.
- The predicting models had different results and their levels of accuracy varied. It is worth noting that all models predicted with an accuracy of over 80%.
- The confusion matrices did not vary significantly which is something I expected to see.
- The SVM model was the best performing model in my experiments with an accuracy well over 90%
- It was interesting to see the algorithms running in realtime as it was possible to see how long some of these predictions take even with a small data set.

Thank you!