Winning Space Race with Data Science

Melusi Nyoni
2023/12/28

testrep/capstone-spacex-falcon-9-landing-prediction.ipynb at main · nyonimelusi/testrep (github.com)

Outline

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

Executive Summary

- I collected data from both the public Space X API and the Space X Wikepedia page. The dataset includes a labelled ‘class’ column, categorizing successful landings. To analyze the data comprehensively, I employed SQL queries, visualizations, Folium maps, and dashboards. Relevant columns were extracted s features, and categorical variables underwent one- hot encoding. After standardizing the data, I utilized GridSearch Cv to optimize parameters for machine learning models. The accuracy scores of four machine learning models-logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbor were visualized. Surprisingly , all models demonstrated a consistent accuracy rate of approximately 83.33%.Howevr , it was observed that they tended to overpredict successful landings. To enhance model determination and accuracy, it is suggested that additional data be acquired.
- Summary of all results. In summary, the process involved a thorough exploration of Space X landing data, feature engineering, and the application of machine learning models. Despite achieving consistent accuracy, the models exhibited a tendency to overpredict, emphasizing the importance of acquiring further data for further refinement

Introduction

- Project background and context
- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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. In this module, you will be provided with an overview of the problem and the tools you need to complete the course
- Problems you want to find answers: Space X has assigned us the task of training machine learning model to predict the successful recovery of stage 1.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Consolidated information from both Space X public API and the Space X Wikipedia page
- Perform data wrangling
  - Distinguishing as either successful or unsuccessful based on predefined criteria
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Optimized models through the use of Gridsearch

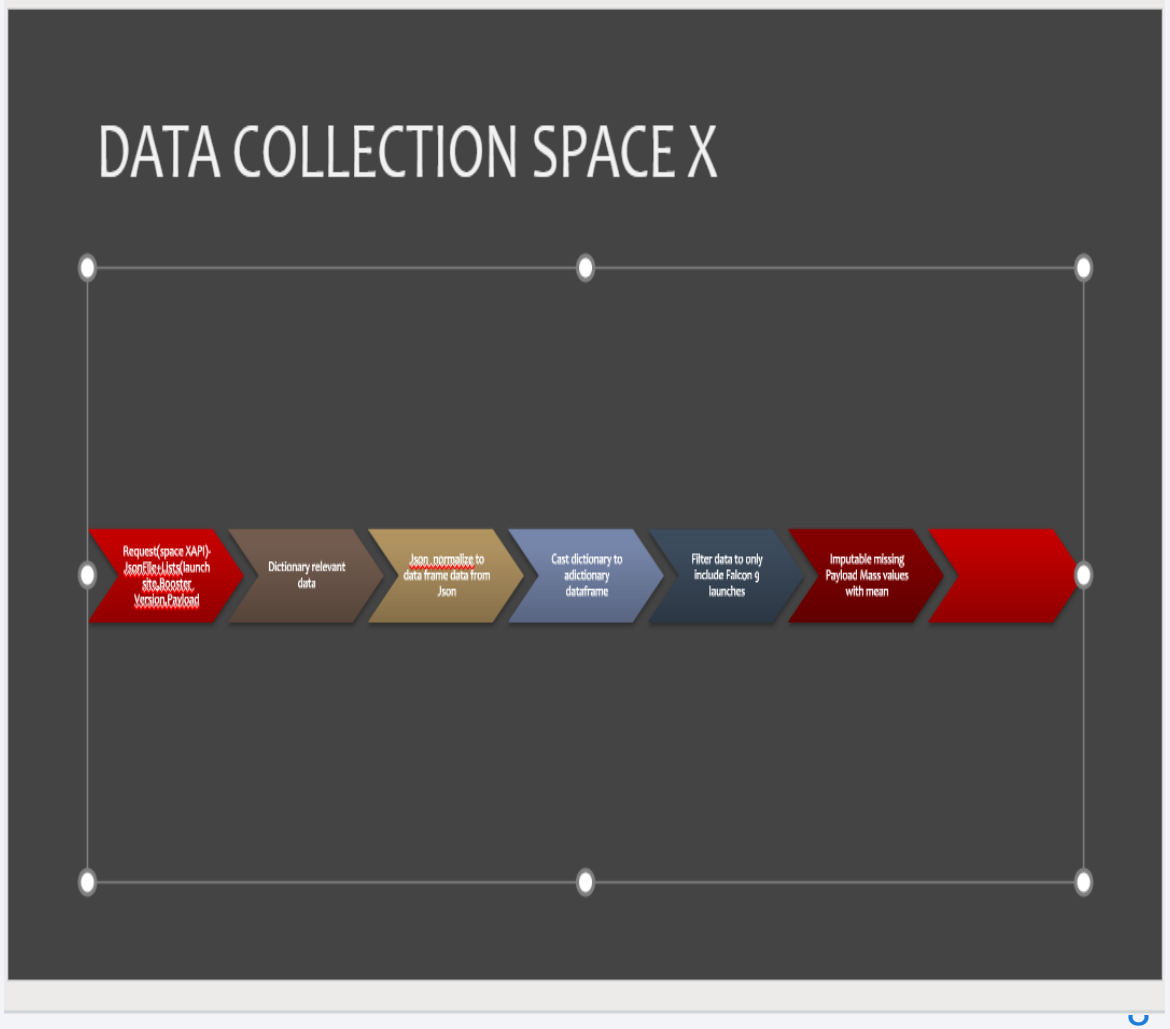
Data Collection

- Describe how data sets were collected.
- The process of collecting data involved a dual approach, combining API requests from Space X's public API and web scrapping data from a table within Space X's Wikipedia entry. The upcoming slide illustrates the flow chart detailing data collection from the API, while the subsequent one will depict the flow chart for data collection from web scrapping.
- Space X API Data Columns include:
- Flight number-Date-Booster version-Payload-Orbit-Luanchsite-Flights-Outcome-Reused-LandingPad-legs-Serial-Reusedcount-Longitute-latitude-Gridfins.
- Wikipedia Web scrape Data Columns Include: Flight no-Launch site-Payload-Payload Mass-Orbit-Customer-Launch Outcome-Version Booster-Booster landing -Date-Time

Data Collection - SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

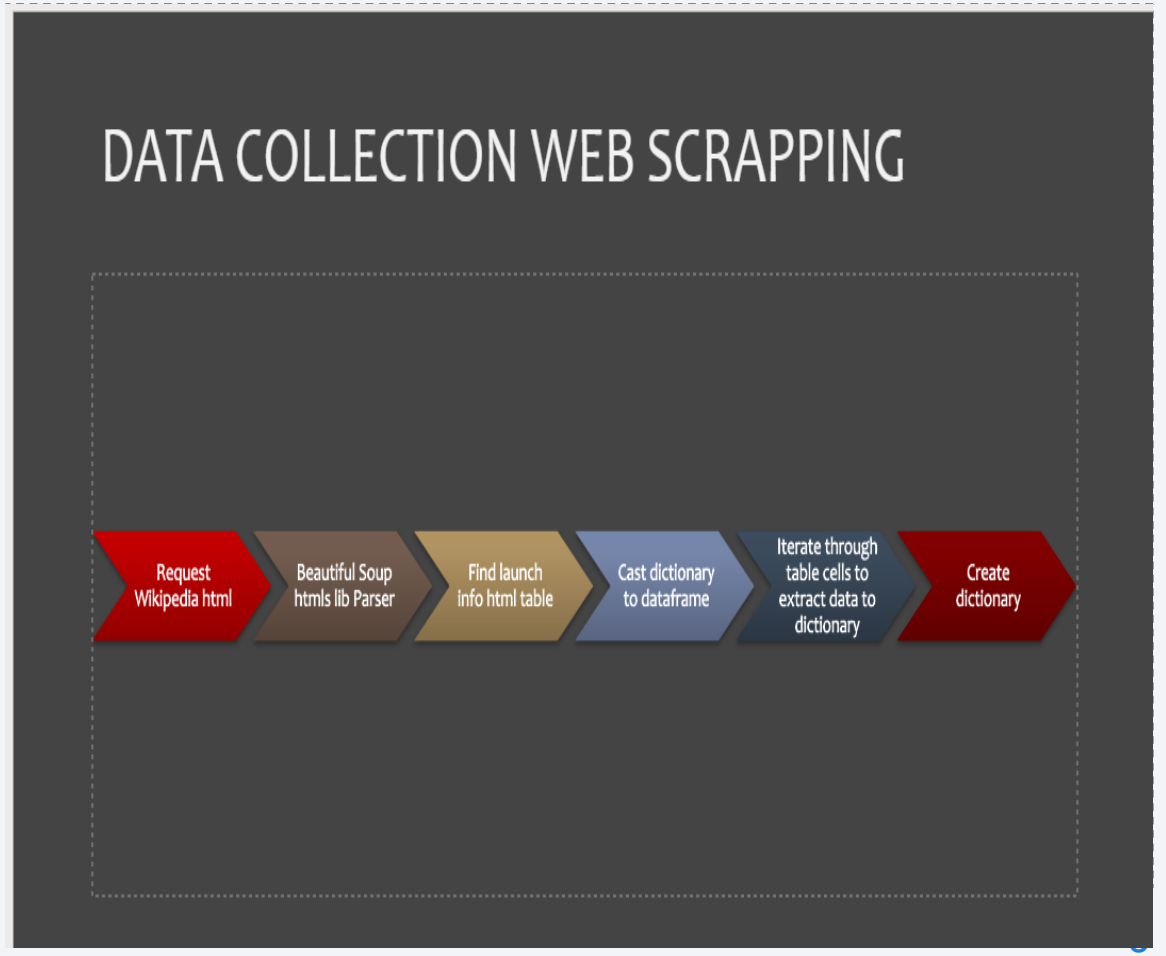
- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/nyonimelusi/testrep.git), as an external reference and peer-review purpose



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, [testrep/jupyter-labs-webscraping (1).ipynb at main · nyonimelusi/testrep (github.com)]



Data Wrangling

- Generate a training label for landing outcomes, assigning a value of 1 for success and 0 for failure. The ‘outcome’ column consists of two components: 'Mission Outcome 'and ‘Landing Location'. Introduce a new training label column named ‘class', assigning a value of 1 when ‘Mission outcome’ is True and 0 otherwise. Utilize the following value mapping:
- True ASDS,True RTLS, and True Ocean should be set to 1.
- None None, False ASDS,None ASDS,False Ocean, and False RTLS should be set to 0.
- testrep/Data Collection api lab.ipynb at main · nyonimelusi/testrep (github.com)

EDA with Data Visualization

- Conducted exploratory data analysis on Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year variables. Employed various plots, including Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs.Luanch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs. Orbit, and Success Yearly Trend.
- Utilized scatter plots, line charts, and bar plots to assess relationships between variables, determining their suitability for incorporation into the machine learning model during training.
- testrep/analysis with Folium.ipynb at main · nyonimelusi/testrep (github.com)

EDA with SQL

- Upload the dataset into an IBM DB2 Database and executed SQL queries using Python integration. Utilized queries to gain a deeper understanding of the dataset, extracting information on launch site names, mission outcomes, diverse payload sizes for customers, booster versions, and landing outcomes
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

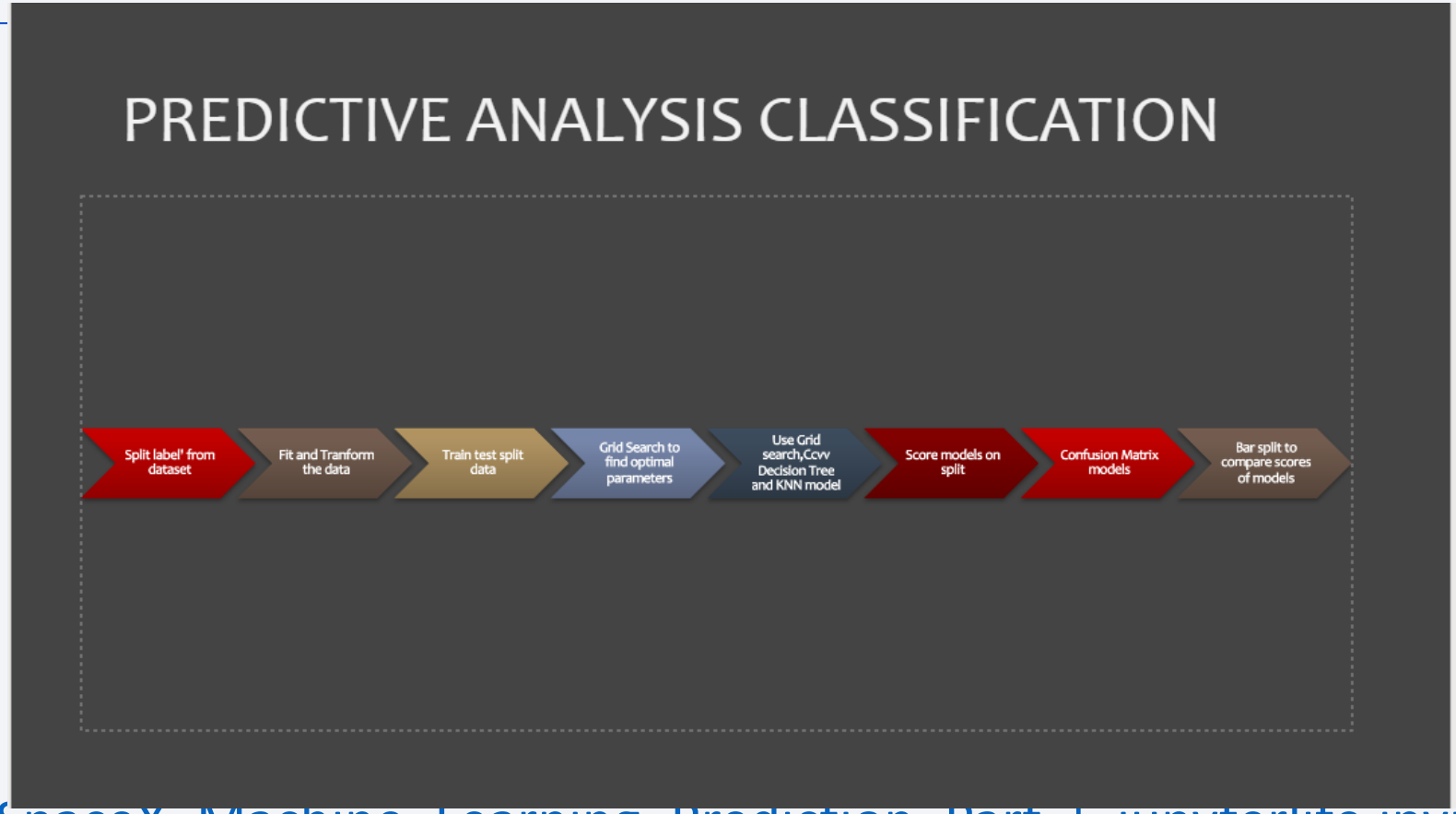
Build an Interactive Map with Folium

- Folium maps are employed to mark Launch Sites, successful and unsuccessful landings, along with proximity to key locations such as Railway, Highway, Coast, and City. This visualization aids in understanding the rationale behind the selection of launch site locations. Additionally ,the maps depict successful landings in relation to their specific geographic locations.
- testrep/analysis with Folium.ipynb at main · nyonimelusi/testrep (github.com)

Build a Dashboard with Plotly Dash

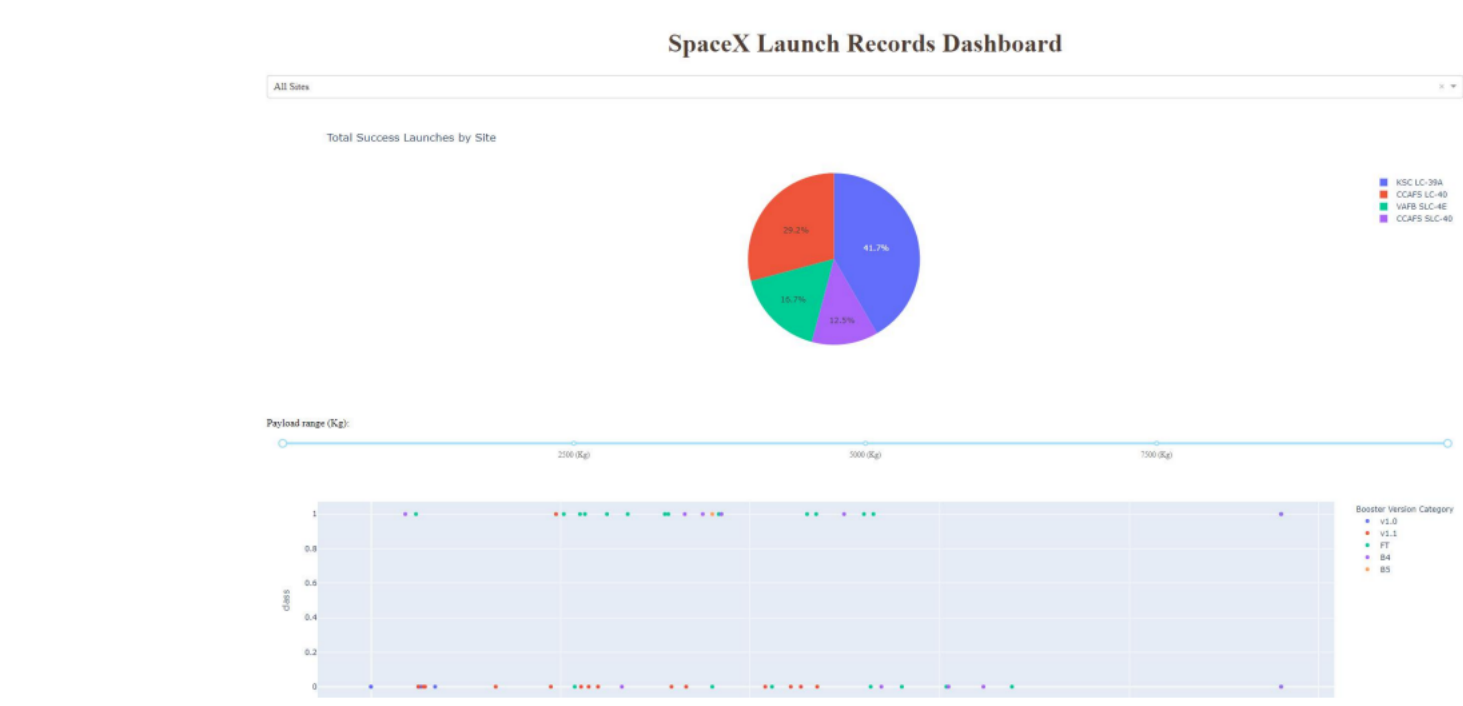
- The dashboard features a pie chart and a scatter plot. The pie chart provides options to display the distribution of successful landings across all launch sites or to focus on individual launch sites, showcasing their respective success rates. The scatter plot offers two inputs:either considering all launch sites or a specific individual site, with a slider for payload mass ranging from 0 to 1000kg.The pie chart facilitates the visualization of launch site success rates, while the scatter plot enables an exploration of how success varies across launch sites, payload masses,and booster version categories.
- Add the GitHub URL-testrep/spacex_dash_app.py at main · nyonimelusi/testrep (github.com)

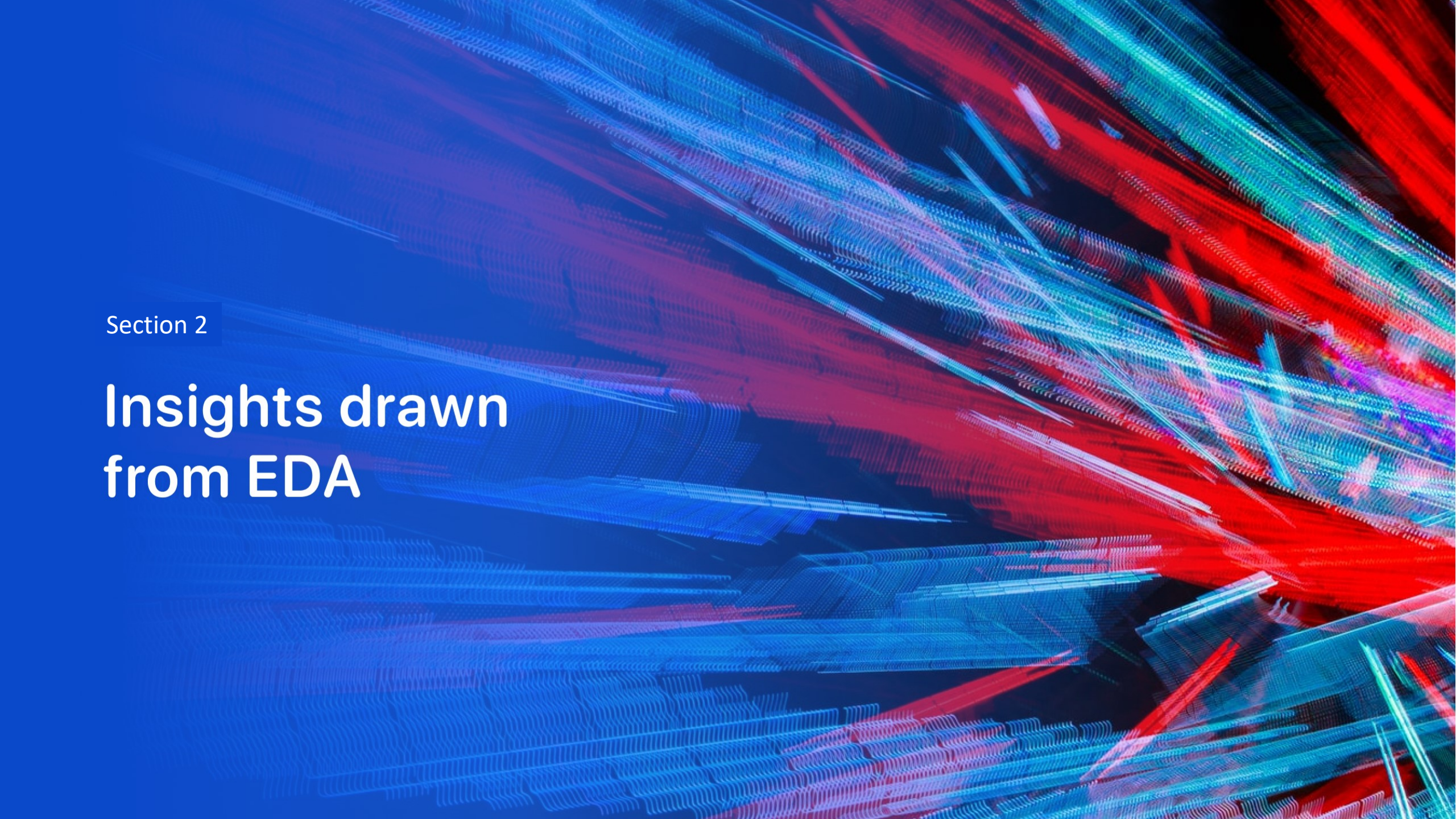
Predictive Analysis (Classification)



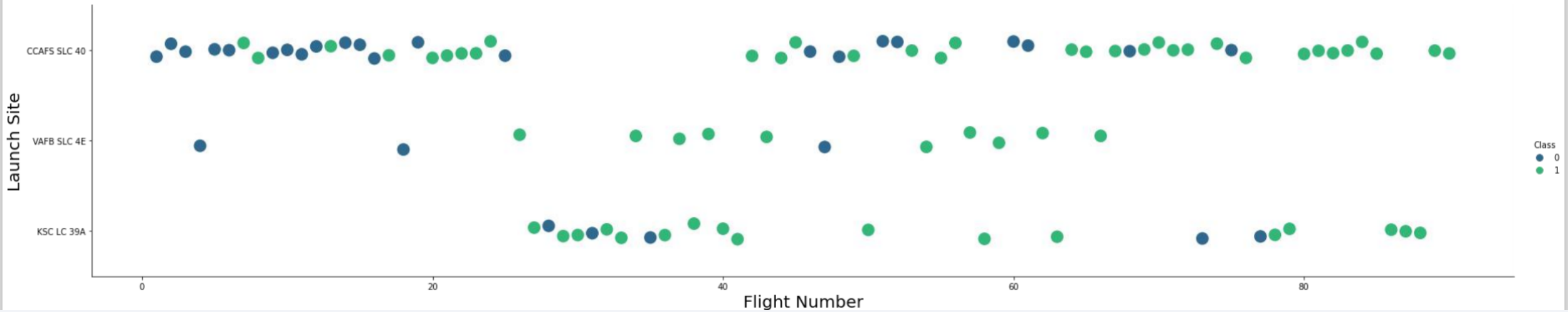
- testrep/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · nyonimelusi/testrep (github.com)

Results


SpaceX Launch Records Dashboard
All Sites
Total Success Launches by Site
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
2500 (Kg)
5000 (Kg)
7500 (Kg)
class
Booster Version Category
v1.0
v1.1
FT
B4
B5

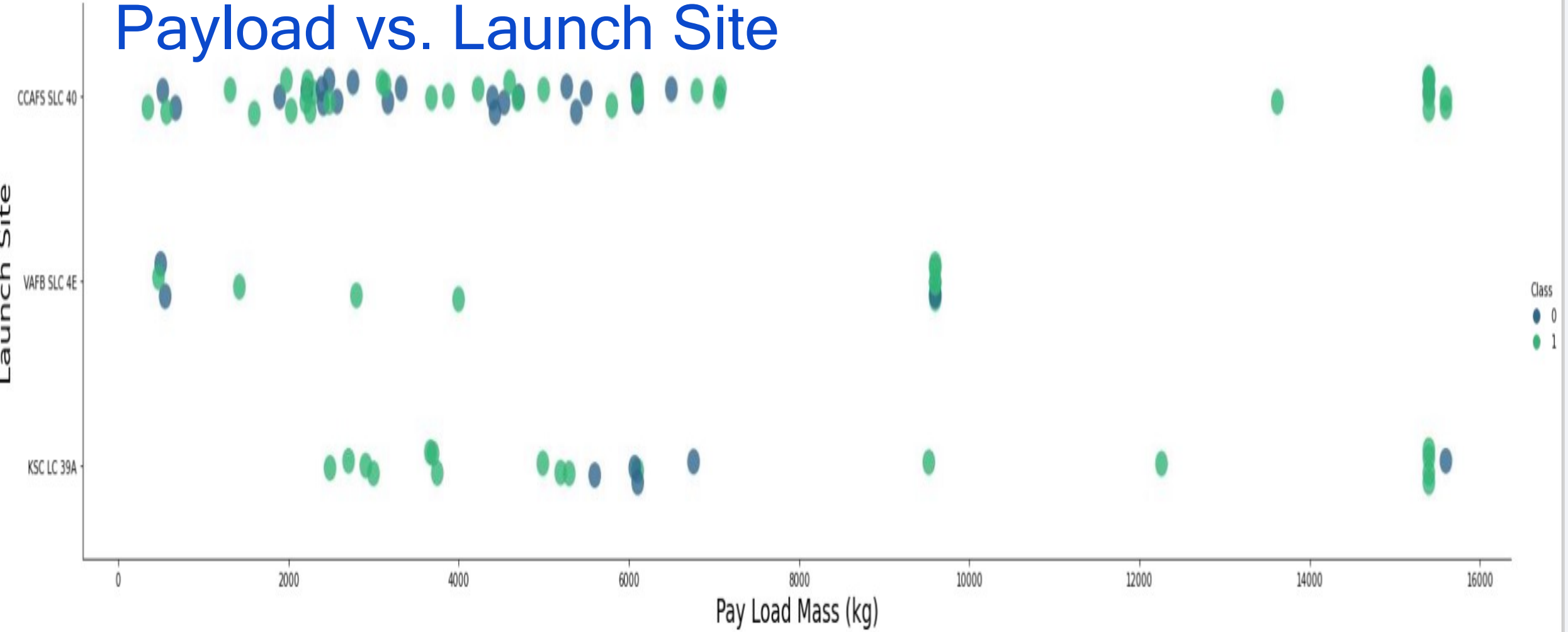

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

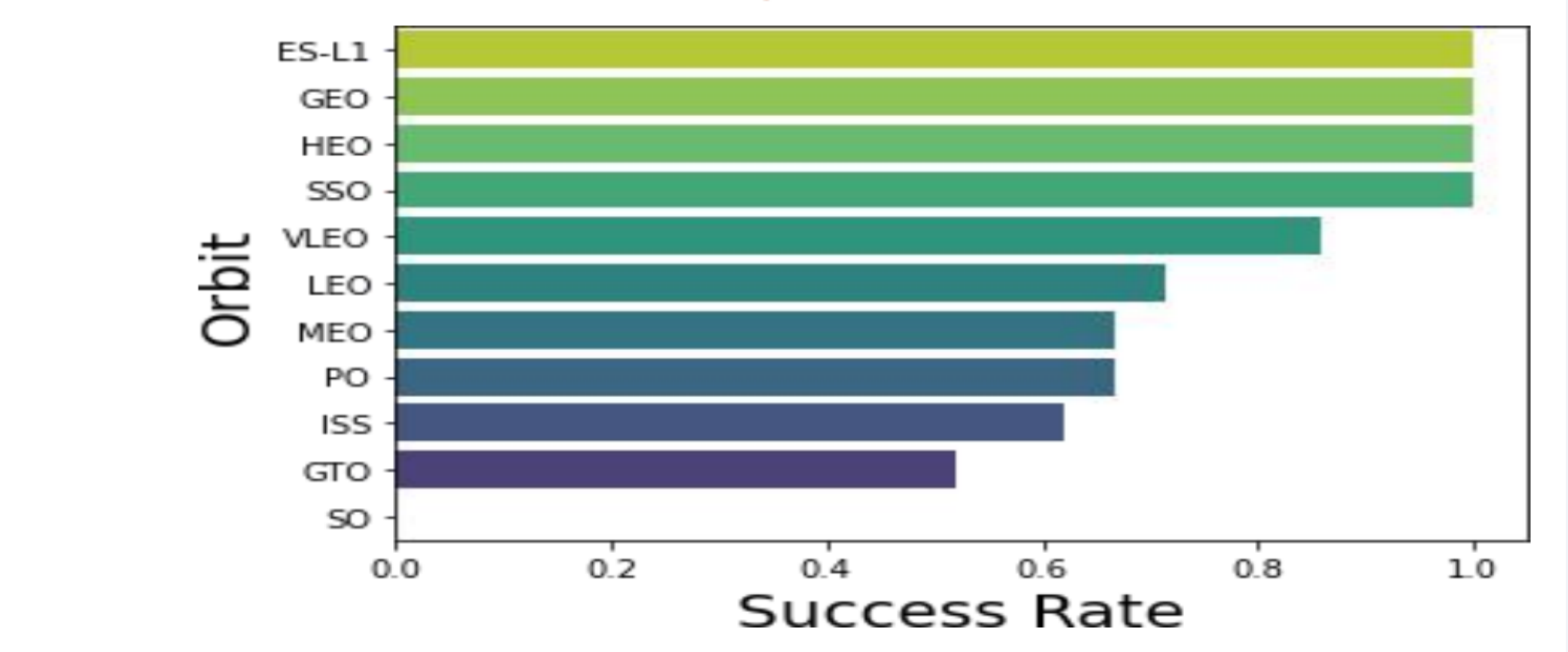


- Green indicates successful launch and purple indicates unsuccessful launch

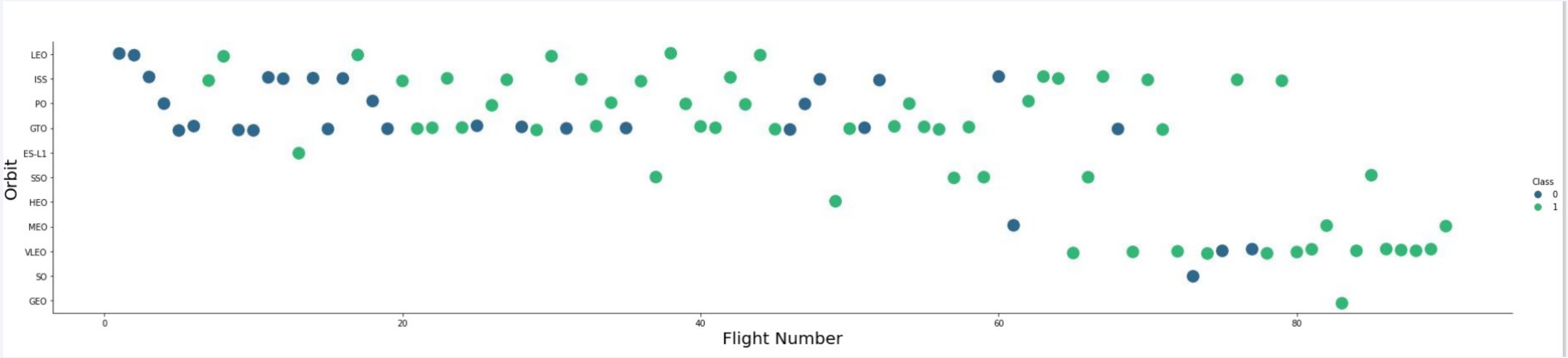
Payload vs. Launch Site



Success Rate vs. Orbit Type



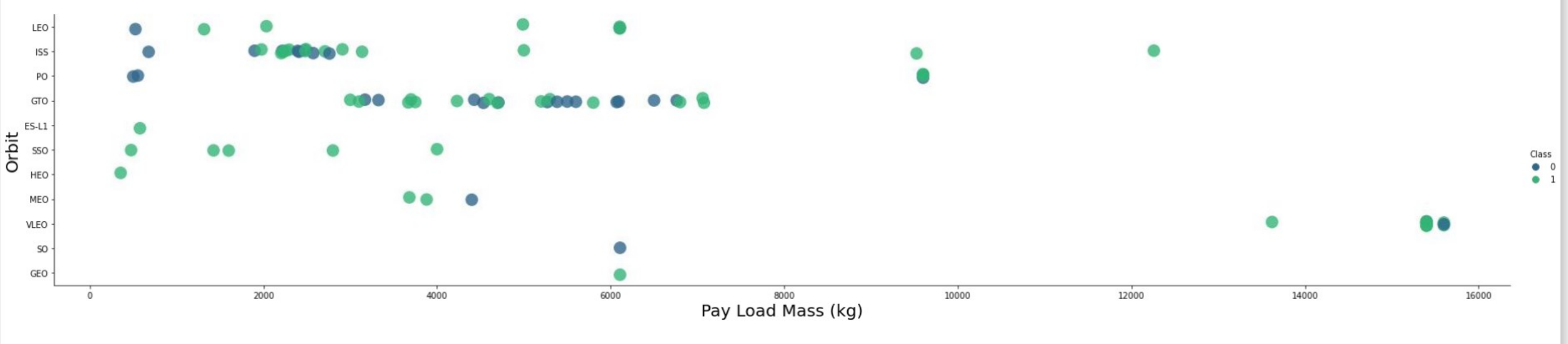
Flight Number vs. Orbit Type



- Green indicates successful launch & purple unsuccessful launch

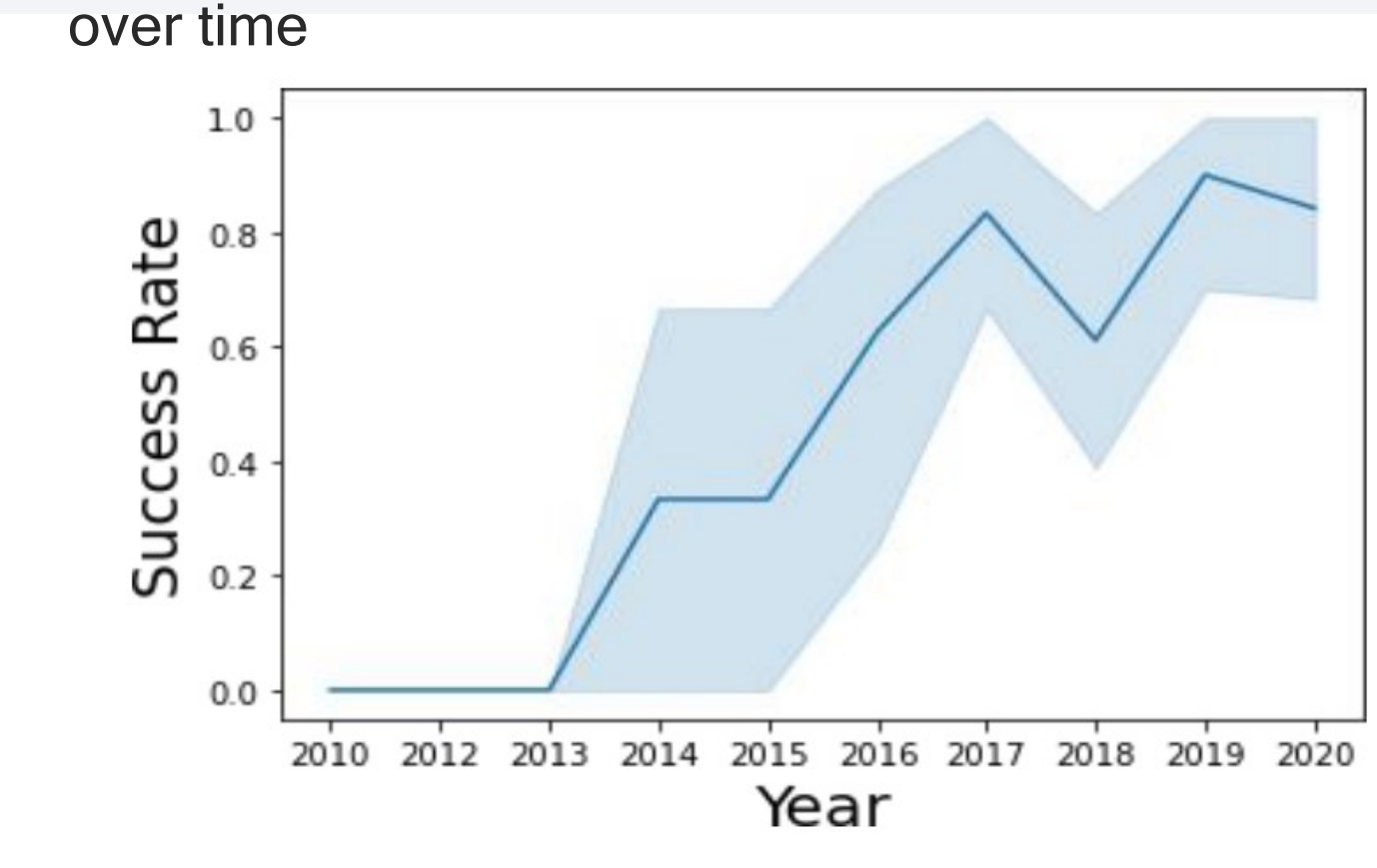
Payload vs. Orbit Type

Payload seems to be in correlation with orbit



Launch Success Yearly Trend

- Success general increases over time



```
q = pd.read_sql('select distinct Launch_Site from spacexdata', conn) q
```

All Launch Site Names

Out[42]:

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Out[43]:

	index	Date	Time_(UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
0	0	2010-06-04 00:00:00	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
1	1	2010-12-08 00:00:00	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
2	2	2012-05-22 00:00:00	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
3	3	2012-10-08 00:00:00	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
4	4	2013-03-01 00:00:00	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

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

```
]:
q = pd.read_sql("select sum(PAYLOAD_MASS__KG_) from spacexdata where Customer='NASA (CRS)'", conn)
q
```

]:

	sum(PAYLOAD_MASS__KG_)
0	45596

Average Payload Mass by F9 v1.1

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

[45]:
```
q = pd.read_sql("select avg(PAYLOAD_MASS__KG_) from spacexdata where Booster_Version='F9 v1.1'", conn)
q
```

[45]:

	avg(PAYLOAD_MASS__KG_)
0	2928.4

First Successful Ground Landing Date

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

```
q = pd.read_sql("select min(Date) from spacexdata where Landing__Outcome='Success (ground pad)'", conn)
q
```

	min(Date)
0	2015-12-22 00:00:00

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
q = pd.read_sql("select distinct Booster_Version from spacexdata where Landing__Outcome='Success (drone s
q
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
q = pd.read_sql("select substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from spacexdata  group b
q
```

	Mission_Outcome	count(*)
0	Failure	1
1	Success	100

Boosters Carried Maximum Payload

```
q = pd.read_sql("select distinct Booster_Version from spacexdata where PAYLOAD_MASS__KG_ = (select max(P
q
```

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 Launch Records

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

```
q = pd.read_sql("select distinct Landing__Outcome, Booster_Version, Launch_Site from spacexdata where Lar
q
```

	Landing__Outcome	Booster_Version	Launch_Site
0	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
2	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
3	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
4	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

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

```
q = pd.read_sql("select Landing__Outcome, count(*) from spacexdata where Date between '2011-06-04' and '2
q
```

	Landing__Outcome	count(*)
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

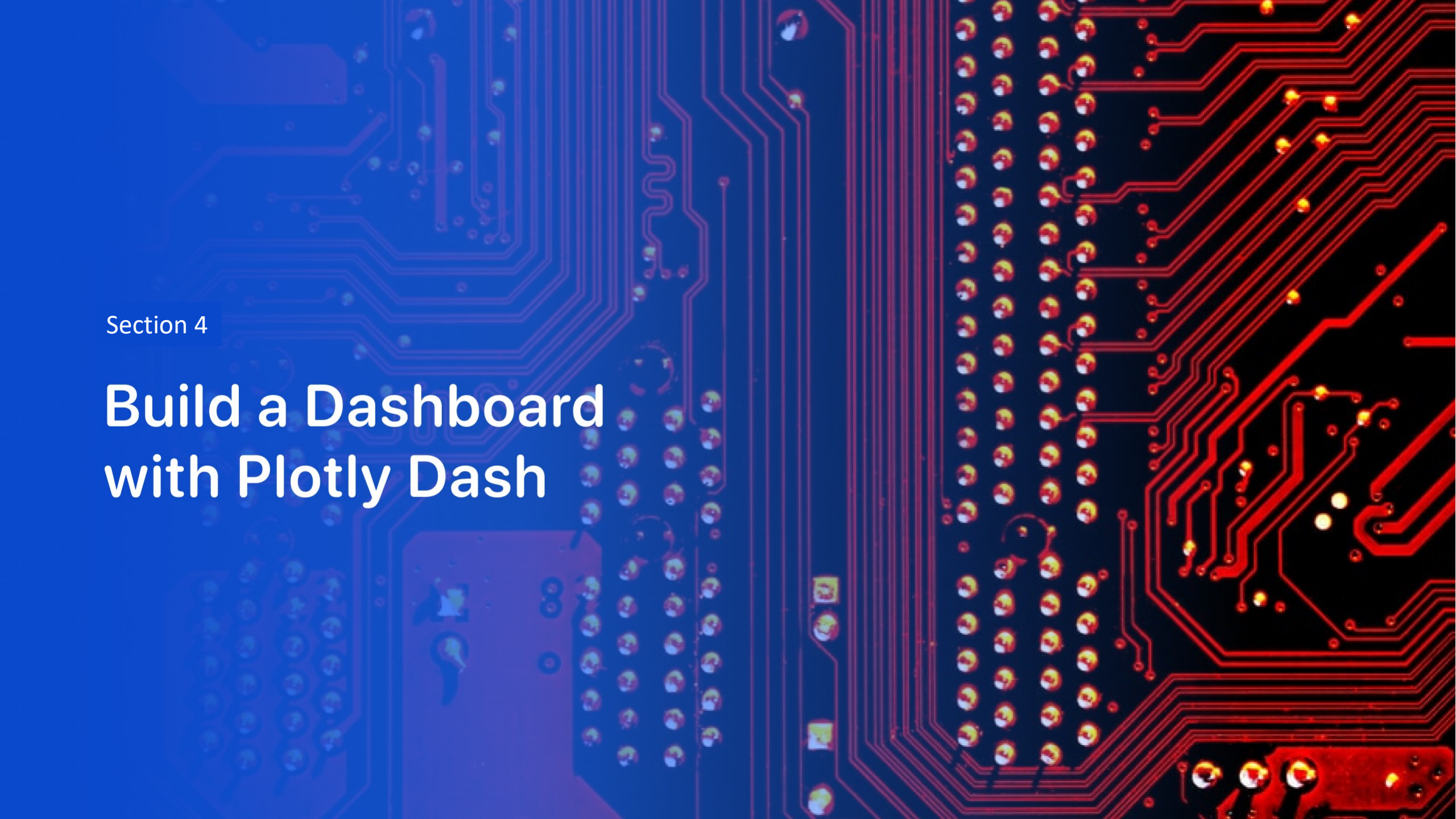
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

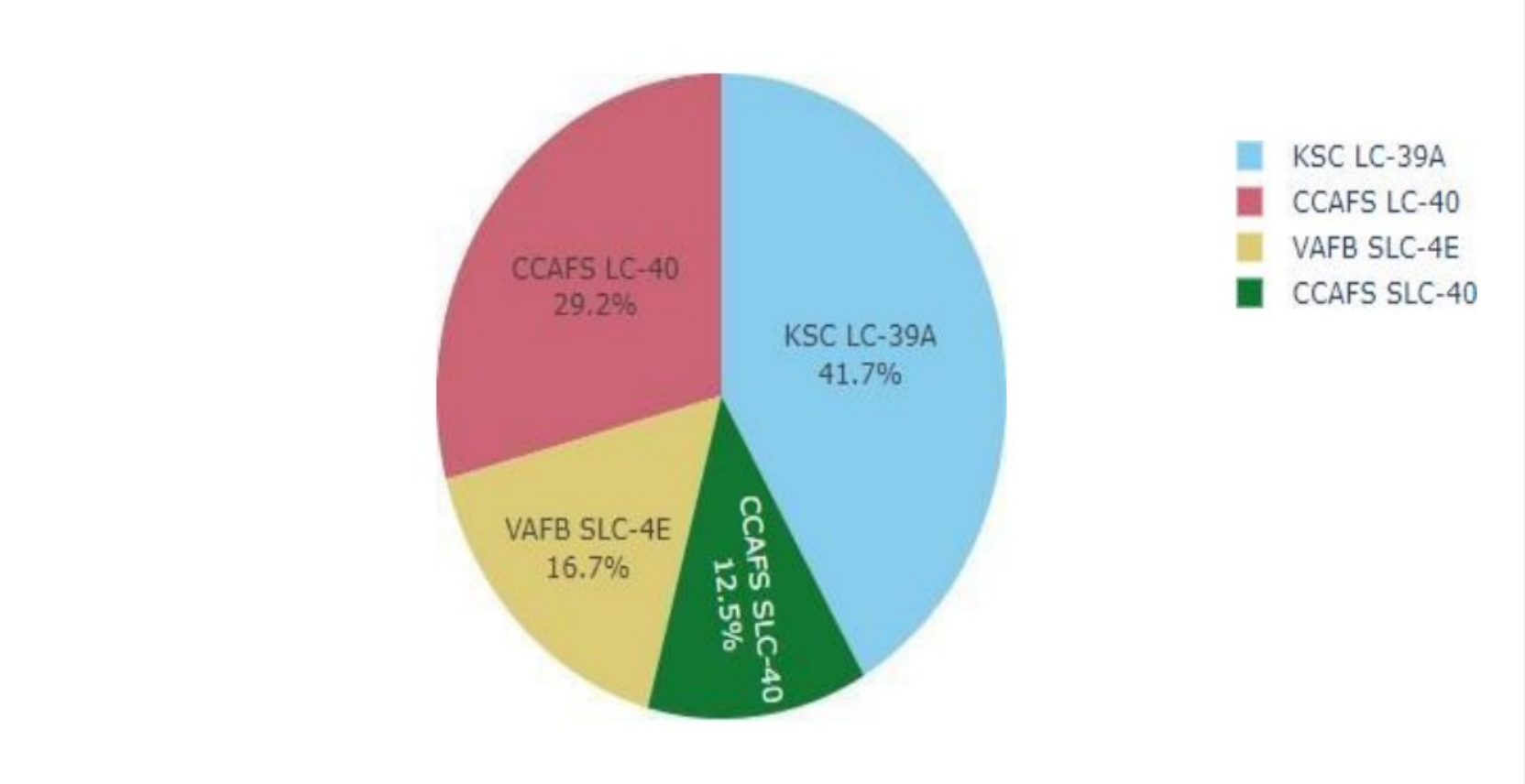
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

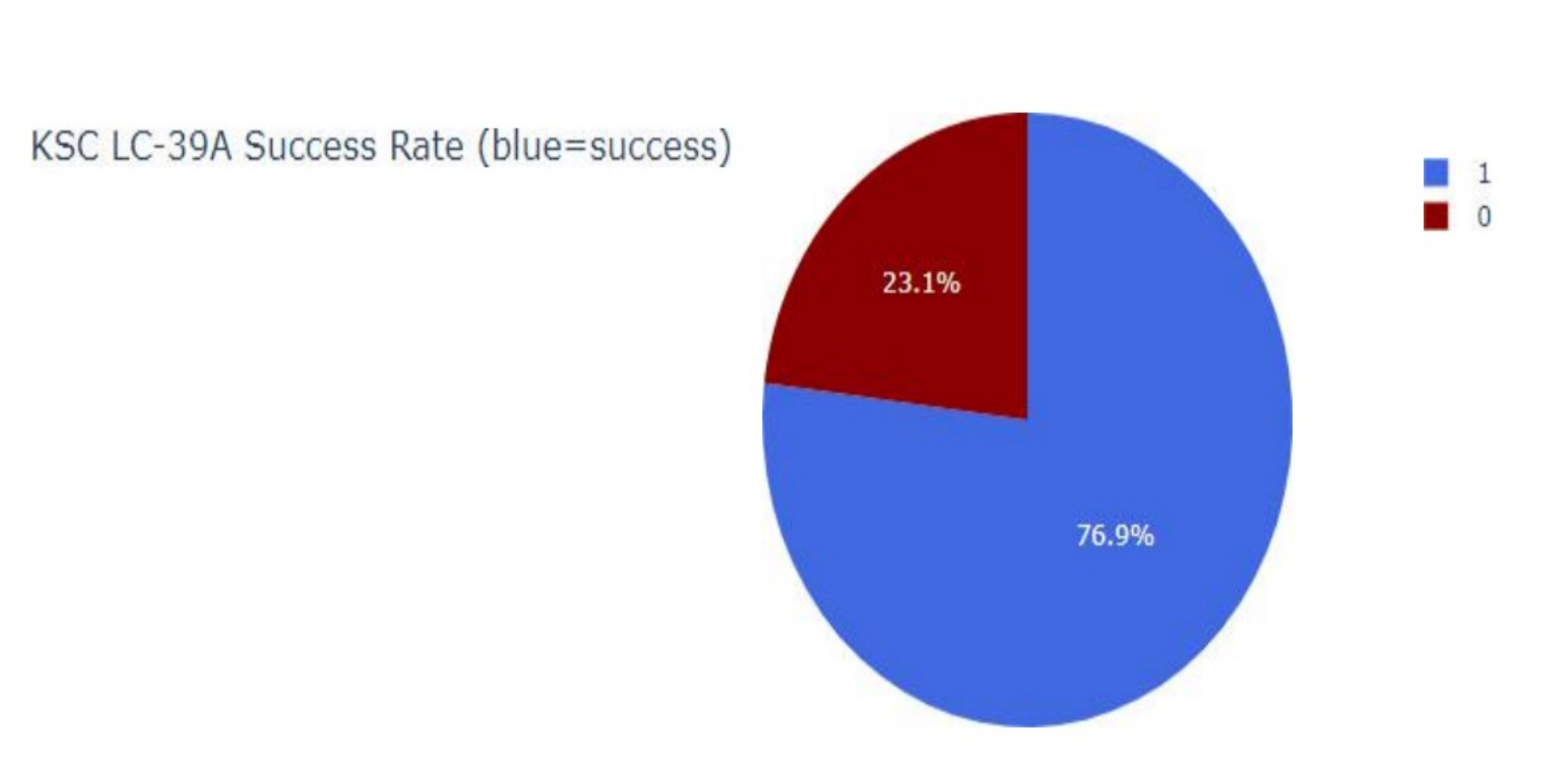
- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot


Section 4
Build a Dashboard with Plotly Dash

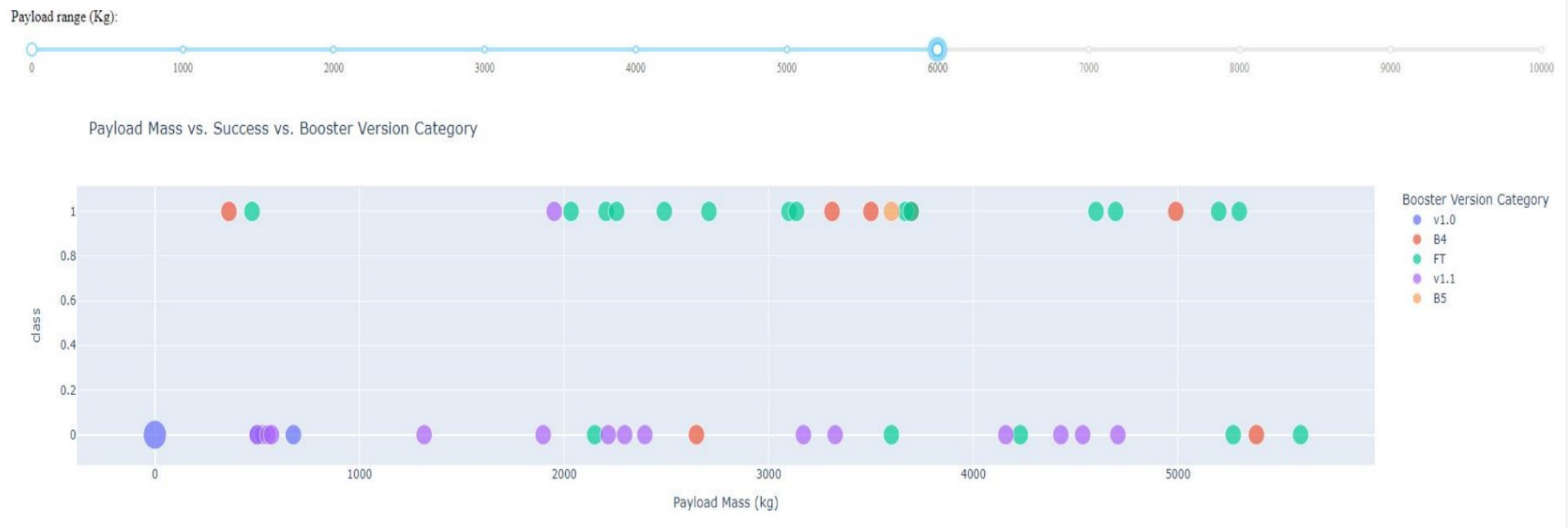
<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



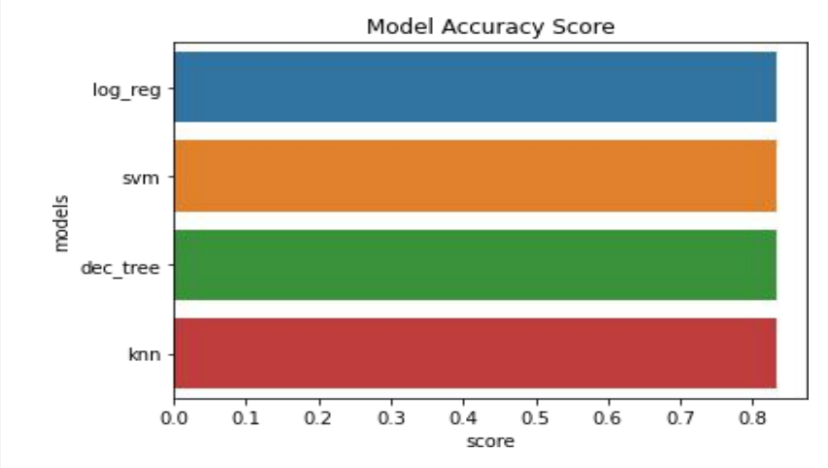
<Dashboard Screenshot 3>



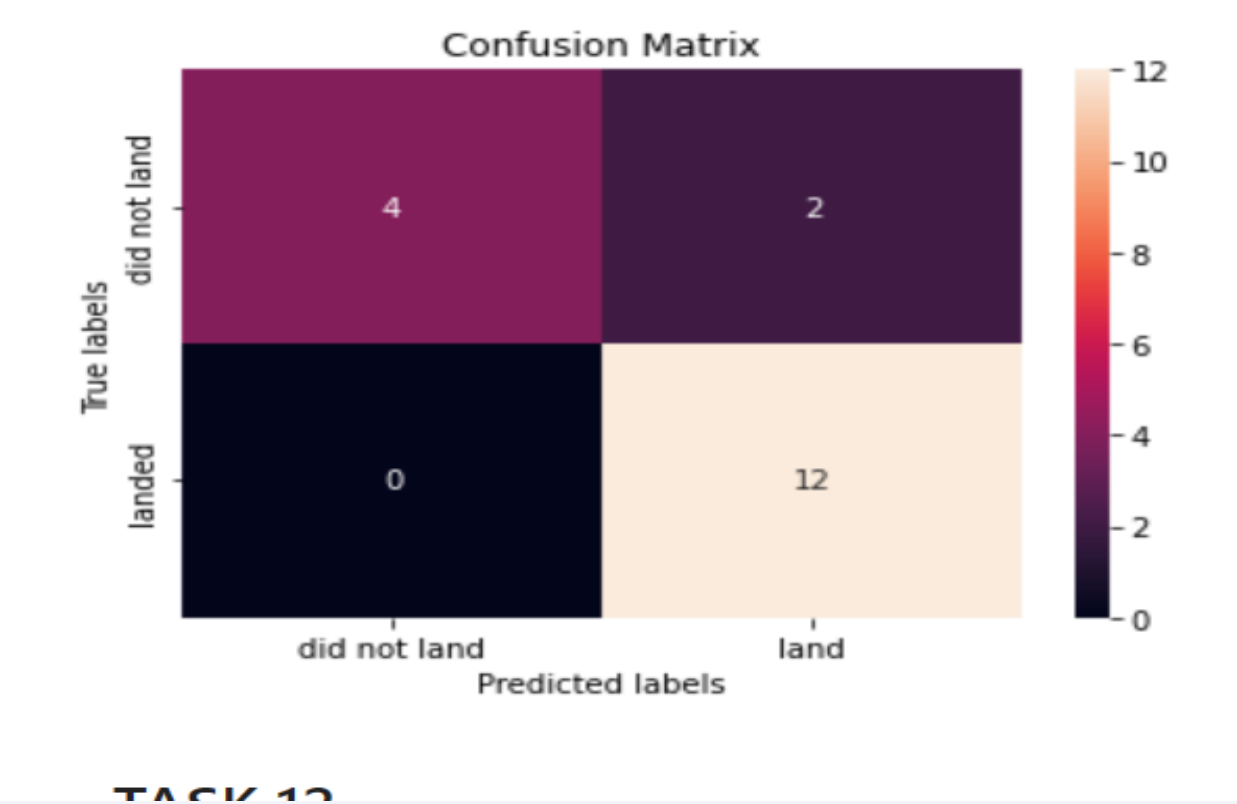
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- Our objective is to develop a machine learning model for Space Y, enabling them to competitively bid against SpaceX . The primary aim of the model is to predict the successful landing of Stage 1,potentially saving around$100 million. We utilized data sourced from a public SpaceX API and performed web scrapping on the Space X Wikipedia page. After creating data labels, the information was stored in SQL database
- To enhance visualization and insights, we designed a dashboard. The machine learning model we constructed achieved an accuracy rate of 83%.This model provides Elon Musk and the Space Y team with capability to predict, with considerable accuracy, whether a launch will witness a successful Stage 1 landing before it occurs. This predictive capability is instrumental in making informed decisions about whether to proceed with a launch.
- For further refinement and accuracy improvement, it is recommended to gather more data.
- This additional data will contribute to amore robust determination of the optimal machine learning model for the task at hand.
- …

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!