SPACEX FALCON 9 FIRST STAGE REUSABILITY PREDICTION FOR LAUNCH PRICING DECISION

FINAL CAPSTONE PROJECT REPORT

JOY U. OLAYIWOLA

25TH SEPTEMBER 2024

https://github.com/jaygirl/IBM-Data-Science-Certification-Capstone-Project/tree/main

IBM Developer

SKILLS NETWORK

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- **Overview**: The goal of this project is to predict the landing success of SpaceX's Falcon 9 first stage using machine learning models. Accurate prediction will help to determine the cost of launches and can provide a competitive edge for our company, Space Y.
- **Result**: The Decision Tree model achieved the highest accuracy in predicting first-stage landing success.
  - Four models were trained and used to test the data collected. These are Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbor.
  - The Decision Tree model came up to with an accuracy of 87.5%, about 3% higher than the others.
- **Impact**: Space Y can use this model to make informed decisions about launch pricing and planning.

INTRODUCTION

- **Background**: The commercial space industry is rapidly growing, with companies like SpaceX leading the way through reusable rocket technology.
- **Problem Statement**: Predicting the success of Falcon 9’s first-stage landing is crucial for reducing launch costs. Space Y aims to use predictive modeling to estimate the success of their rocket launches, and by extension estimate on a competitive launch cost with SpaceX.
- **Objective**: Develop a machine learning model to predict Falcon 9 first-stage landing success based on historical data.

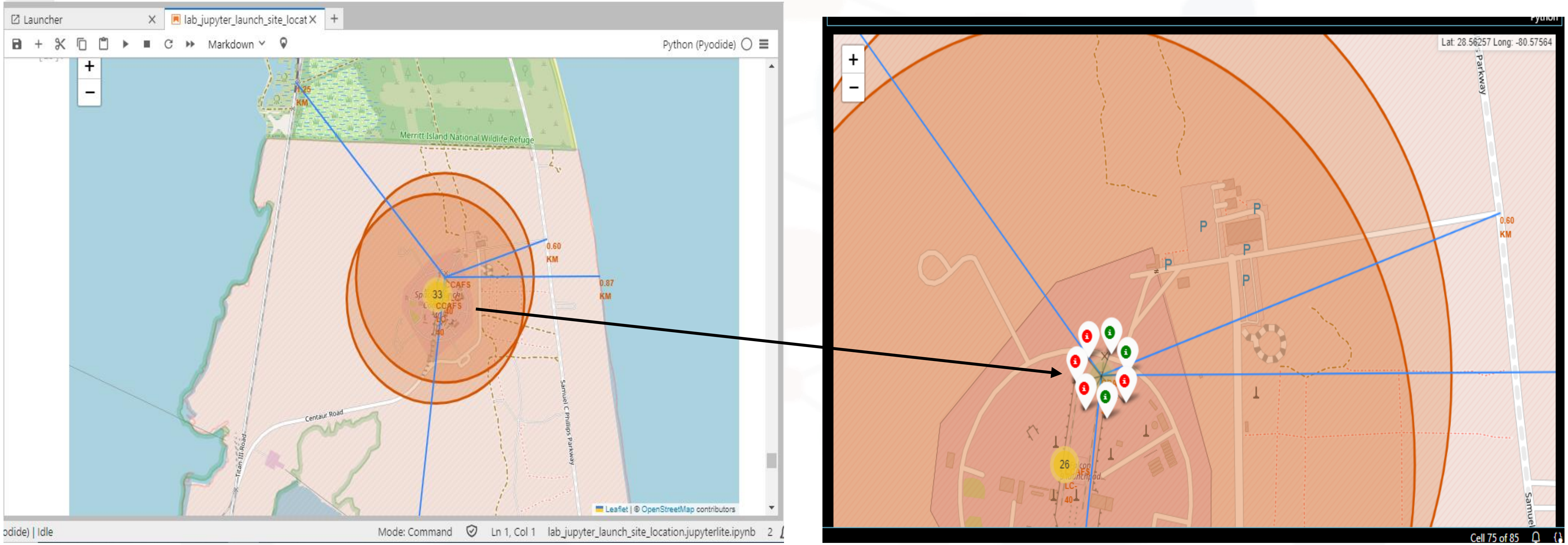
METHODOLOGY (DATA COLLECTION AND WRANGLING)

- **Data Source**:
  - Collected data from SpaceX's launch records via SpaceX API using GET request
  - We also web scraped Wikipedia with Beautiful Soup to collect Falcon 9's historical launch records
  - Records collected include features like launch site, payload mass, orbit type, and booster version, etc.
- **Data Preprocessing**: We carried out data wrangling so as to
  - Handle missing values.
  - Turned the json data into dataframe using .json_normalize()
  - Encoded categorical variables and standardized numerical features.
  - Determine training labels
  - Performed EDA and Feature Engineering using Pandas and Matplotlib

METHODOLOGY (EDA AND INTERACTIVE VISUAL ANALYTICS)

- **Process**: We carried out launch sites location analysis with Folium (an interactive leaflet map). This visual was necessary to quickly answer questions or enable decision making on finding an optimal location for building a launch site. We were able to do the following on the map:
  - Map Launch Sites from the Space X records
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site and its proximities to the nearest city, nearest rail, nearest highway and nearest coastline.

METHODOLOGY (EDA AND INTERACTIVE VISUAL ANALYTICS CONT'D)



The image on the left zoomed out on the right to reveal the clusters that make up the total marker cluster on the left image. Also, the right image shows that the green markers represent successful launches on the selected launch pad, while the red depicts failed launches, totaling seven launches from the selected pad. The blue lines are proximity lines as mentioned in the previous slide.

METHODOLOGY (PREDICTIVE ANALYSIS METHODOLOGY)

- **Process**: Here we carried out machine learning prediction by creating an ML pipeline to predict if the Falcon 9 first stage will land given the available data we were working on

- **Method**: We performed EDA and determined the training labels. Then we,
  - Created a Class column (which was our target label) using NumPy method
  - Standardized our data using StandardScaler
  - Split our data into training and test data
  - Trained and evaluated the data with several models (Logistic Regression, SVM, Decision Tree, KNN) using cross-validation, to find the model that performs best (Decision Tree).

```
31]
...
Logistic Regression Best CV Accuracy:  0.8464285714285713
SVM Best CV Accuracy:  0.8482142857142856
Decision Tree Best CV Accuracy:  0.875
KNN Best CV Accuracy:  0.8482142857142858
The best model is: Decision Tree with a CV accuracy of 0.8750
+ Code    + Markdown
```

RESULTS (EDA WITH VIZUALIZATION)

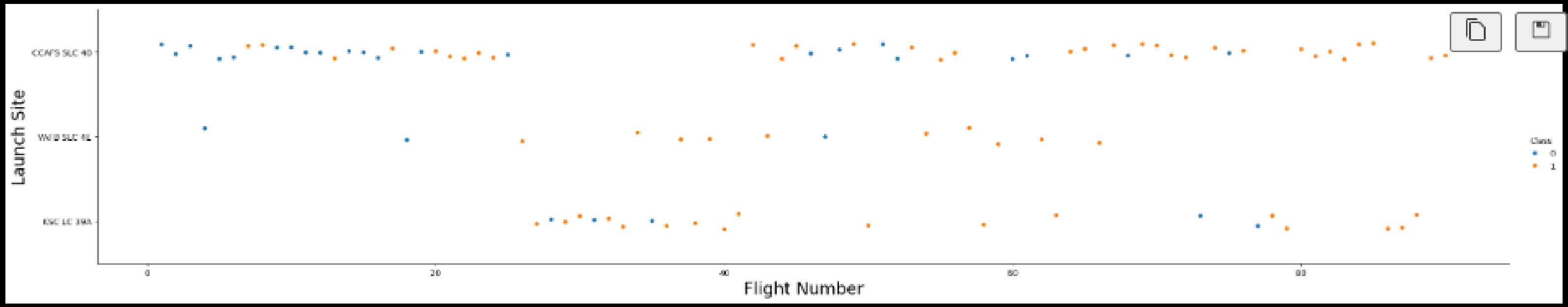
We performed EDA and Feature Engineering, using scatter plot (catplot), to visualize the relationships between the independent variables and how it affects the launch outcomes.

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```python
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

[7] Python



Launch Site CCAFS SLC-40 shows to have higher flight number (indicating more launch attempts), and the higher the flight number, the more likely for the first stage to land successfully from that site.

(Orange indicates success, while blue indicates fail)

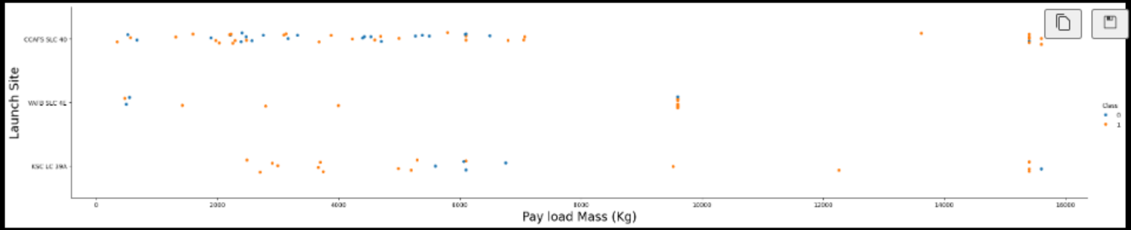
RESULTS (EDA WITH VIZUALIZATION)

TASK 2: Visualize the relationship between Payload Mass and Launch Site

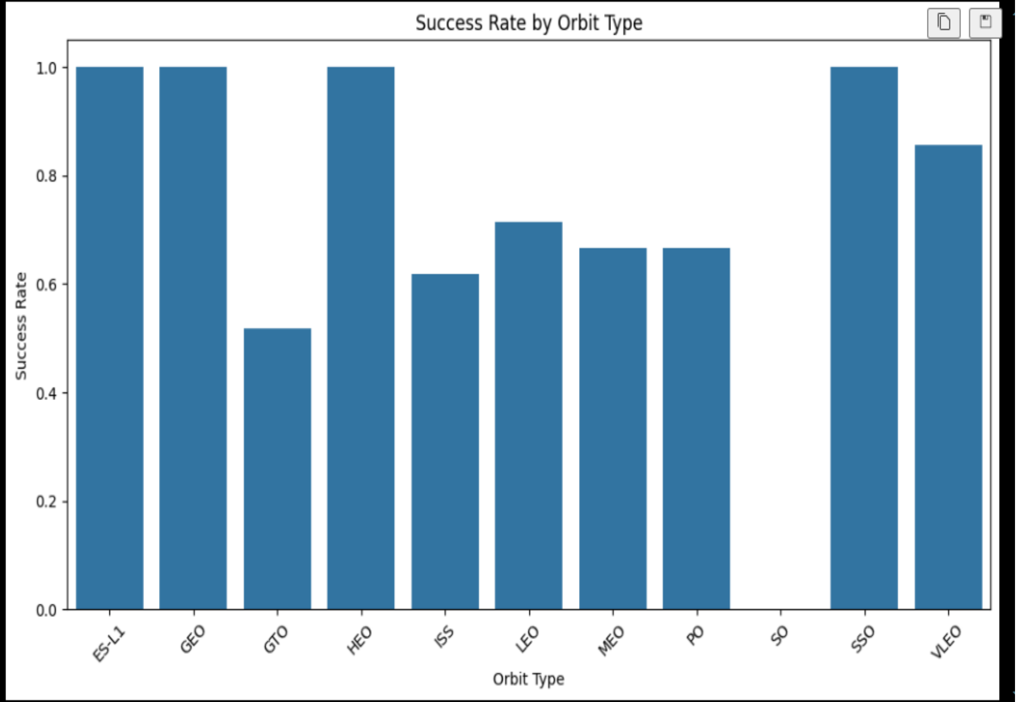
We also want to observe if there is any relationship between launch sites and their payload mass.

```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class va
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (Kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

[10] Python

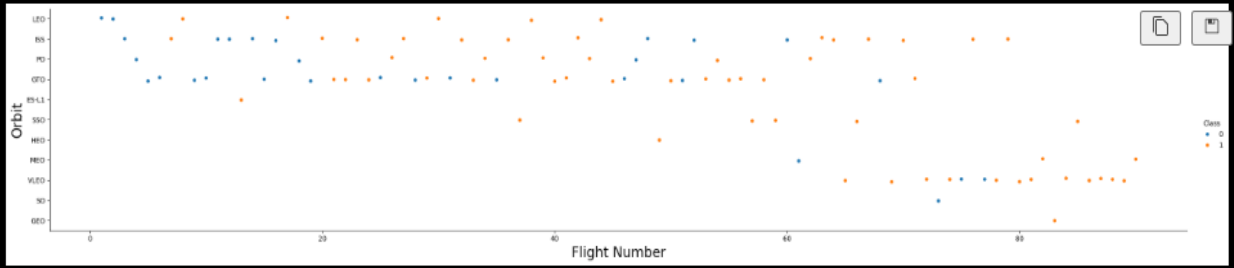


Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

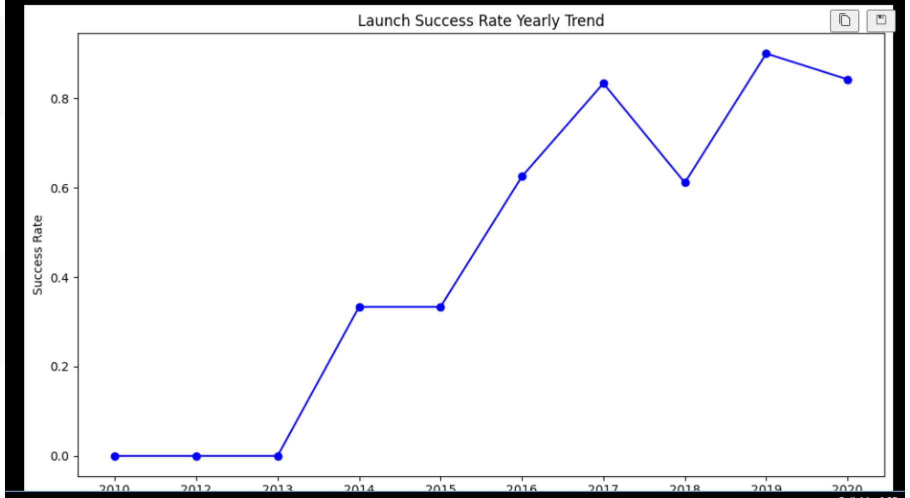


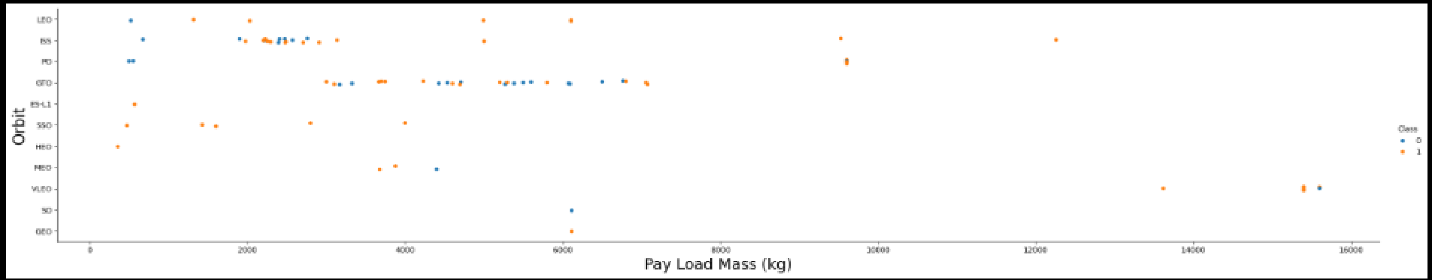
Cell 11 of 65

RESULTS (EDA WITH VIZUALIZATION)



You 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.





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

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

Success rate spiked from 2013

RESULTS (EDA WITH SQL)

We were able to load our dataset into a Db2 database, carried out several queries on it to answer specific questions

Task 1

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

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

[13]

* sqlite:///my_data1.db
Done.

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

Task 2

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

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5;
```

[14] Python

* 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

RESULTS (EDA WITH SQL)

Task 3

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

[23]
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS "TOTAL PAYLOAD BY NASA CRS" FROM SPACEXTABLE WHERE Customer = "NASA (CRS)";
```
Python

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

TOTAL PAYLOAD BY NASA CRS
45596

Task 4

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

[25]
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS "AVERAGE PAYLOAD BY F9 v1.1 Booster" FROM SPACEXTABLE WHERE "Booster_Version" = "F9 v1
```
Python

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

AVERAGE PAYLOAD BY F9 v1.1 Booster
2928.4

RESULTS (EDA WITH SQL)

Task 5

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

Hint:Use min function

```
%%sql
SELECT MIN(Date) AS "FirstSuccessfulLandingDate" FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

[28]

* sqlite:///my_data1.db
Done.

FirstSuccessfulLandingDate
2015-12-22

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 SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)' AND ("PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000);
```

[29] Python

* sqlite:///my_data1.db
Done.

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

RESULTS (EDA WITH SQL)

Task 7

List the total number of successful and failure mission outcomes

```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS "Total_Count"
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

[31] Python

* sqlite:///my_data1.db
Done.

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

RESULTS (EDA WITH SQL)

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
SELECT "Booster_Version" FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
```

[32] Python

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

Booster_Version
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

Cell 31 of 42

RESULTS (EDA WITH SQL)

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql
SELECT
    CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS MonthName,
    "Landing_Outcome",
    "Booster_Version",
```

```
        END AS MonthName,
        "Landing_Outcome",
        "Booster_Version",
        "Launch_Site"
    FROM SPACEXTABLE
    WHERE "Landing_Outcome" = 'Failure (drone ship)'
    AND substr(Date, 0, 5) = '2015';
```

[33]

* sqlite:///my_data1.db
Done.

MonthName	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

RESULTS (EDA WITH SQL)

Task 10

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 "No of Outcomes"
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY "No of Outcomes" DESC;
```

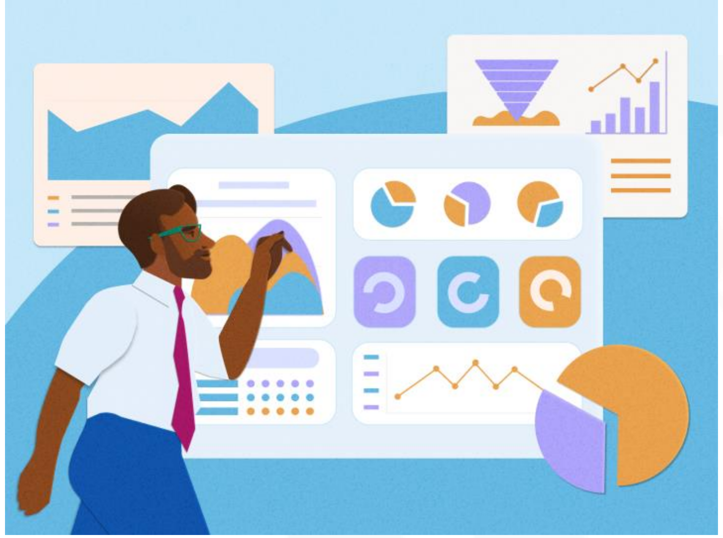
[34] Python

* sqlite:///my_data1.db

Done.

Landing_Outcome	No of Outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2

RESULTS - DASHBOARD



- We created an interactive dashboard using Plotly and Dash, allowing real-time predictions and analysis.
- **Key Features**:
  - Select launch site and payload mass to see predicted outcomes.
  - Visualize historical success rates.

•**Purpose**: The dashboard will allow the management team of Space Y to make real-time decisions based on input variables for each launch.

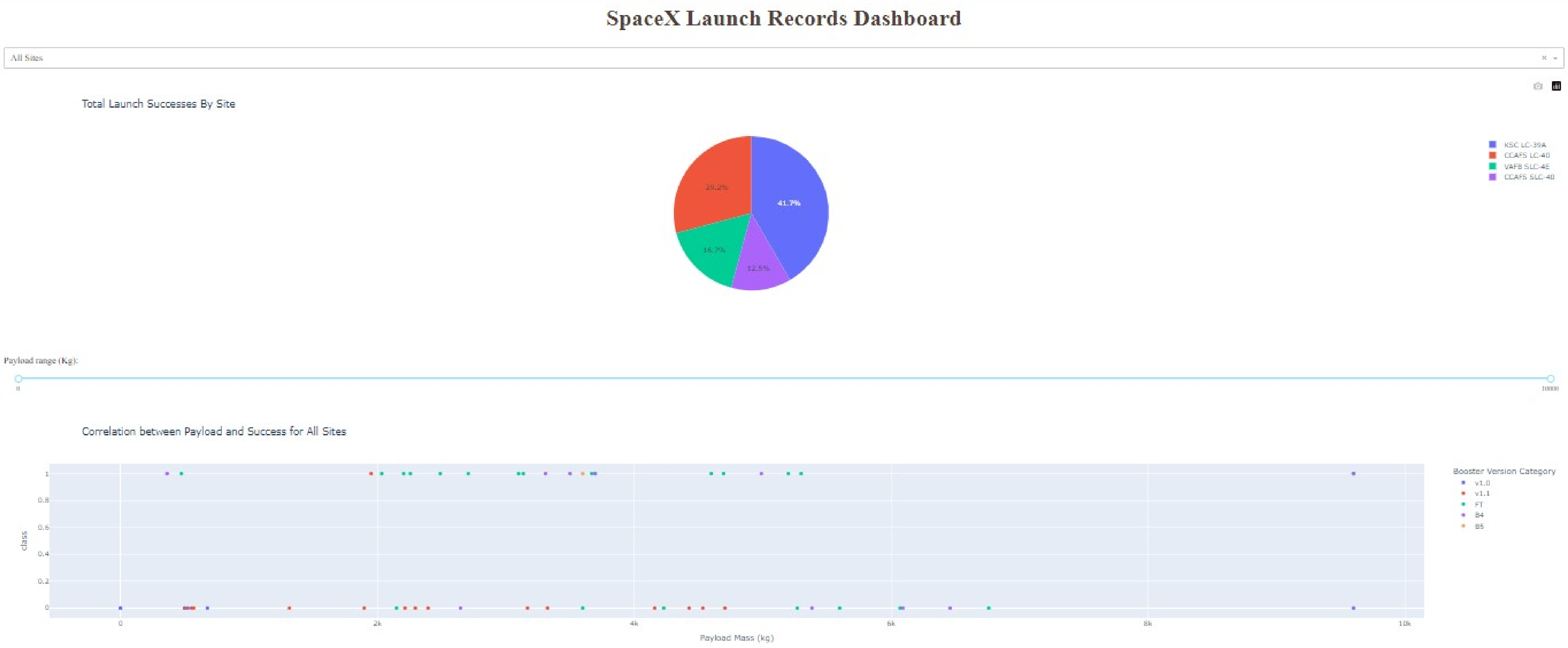
•Below is the link to the dashboard on github

https://github.com/jaygirl/IBM-Data-Science-Certification-Capstone-Project/tree/main

RESULTS - DASHBOARD CODE SNIPPET

```python
# Import required libraries
import pandas as pd
import dash
import dash_html_components as html
import dash_core_components as dcc
from dash.dependencies import Input, Output
import plotly.express as px

# Read the airline data into pandas dataframe
spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()

# Create a dash application
app = dash.Dash(__name__)

# Create an app layout
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503D36',
                                               'font-size': 40}),
                                # TASK 1: Add a dropdown list to enable Launch Site selection
                                # The default select value is for ALL sites
                                dcc.Dropdown(id='site-dropdown',
                                             options=[{'label': 'All Sites', 'value': 'ALL'},
                                                      {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
                                                      {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'},
                                                      {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
                                                      {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'}
                                             ],
                                             value = 'ALL',
                                             placeholder = "Select a Launch Site here",
                                             searchable = True
                                             ),
                                html.Br(),

                                # TASK 2: Add a pie chart to show the total successful launches count for all sites
```

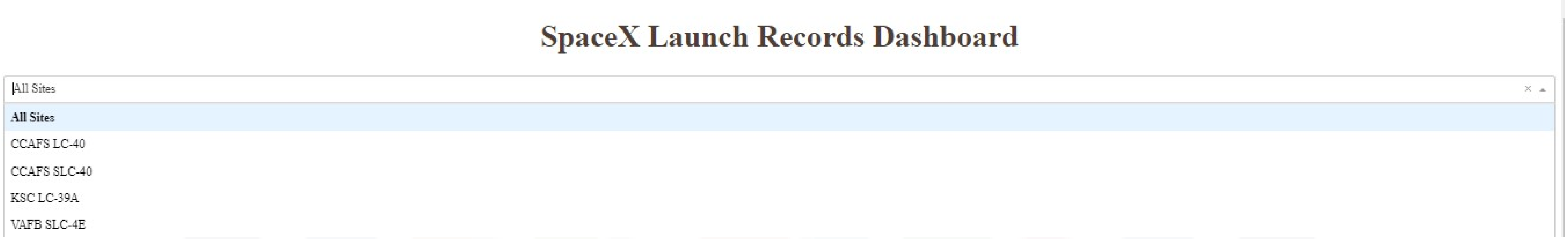
RESULTS - DASHBOARD (FULL VIEW)



IBM Developer

SKILLS NETWORK

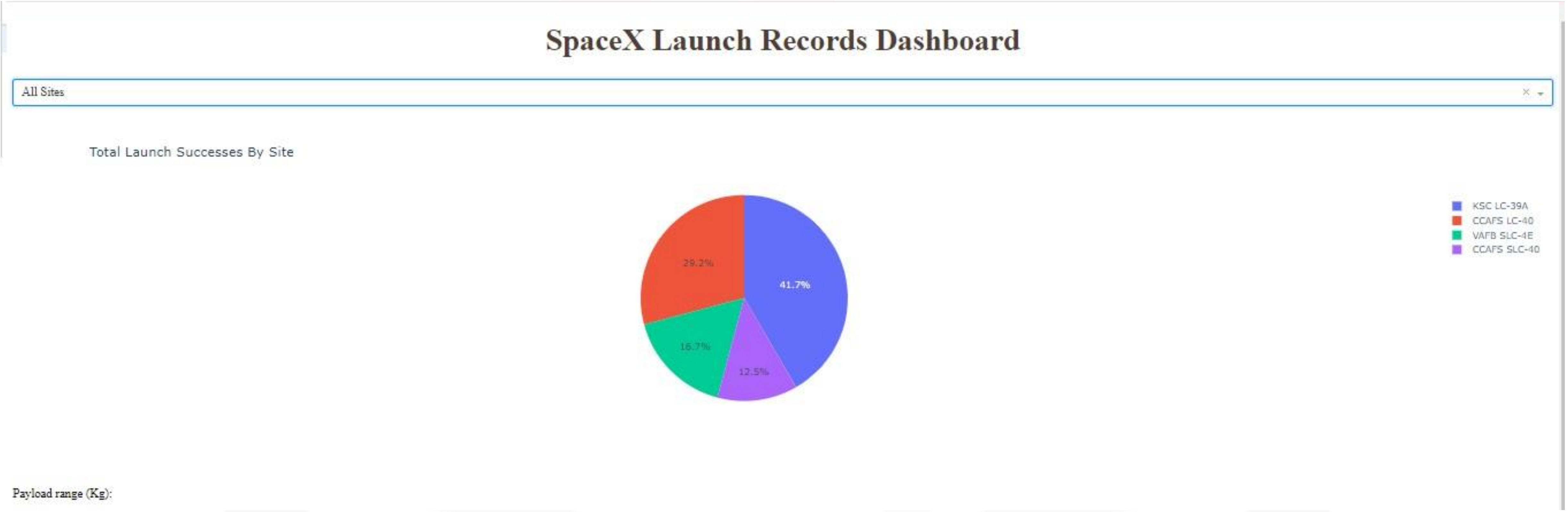
RESULTS - DASHBOARD



Launch Site Drop-down Input Component

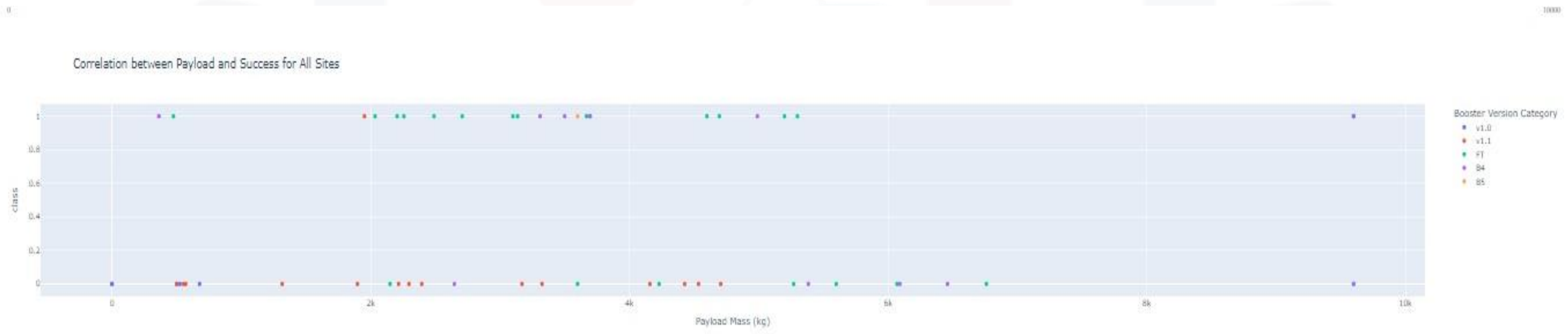
Range Slider to Select Payload

RESULTS - DASHBOARD



Callback function rendering success-pie-chart based on selected site dropdown

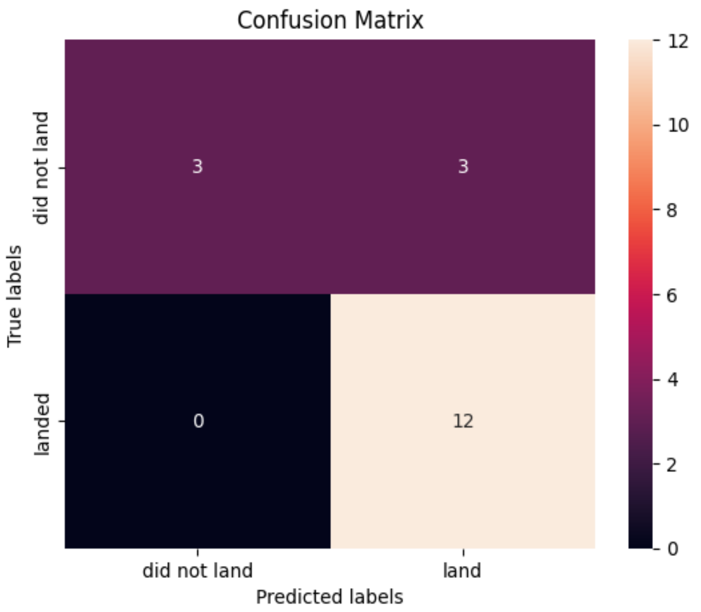
RESULTS - DASHBOARD



Callback function to render the success-payload-scatter-chart scatter plot

RESULTS – PREDICTIVE ANALYSIS (CLASSIFICATION)

- **Model Performance**: The Decision Tree model outperformed other algorithms in predicting the success of Falcon 9 first-stage landings. It achieved an accuracy of 87.5%**,** the highest among the models tested. The other models performed well too.
- **Confusion Matrix**: The Decision Tree model's confusion matrix, just like the other models, highlights that it accurately classified the majority of the landings, with minimal misclassifications. It effectively predicted both successful and failed landings, showing a balanced performance.
- **Model Interpretability**: One key advantage of the Decision Tree model is its interpretability. It provides clear insights into which features are most important in predicting landing success. In this case, features like **launch site**, **orbit type**, and **payload mass** were critical in determining whether the Falcon 9's first stage would successfully land or not.




```
Logistic Regression Best CV Accuracy:  0.8464285714285713
SVM Best CV Accuracy:  0.8482142857142856
Decision Tree Best CV Accuracy:  0.875
KNN Best CV Accuracy:  0.8482142857142858
The best model is: Decision Tree with a CV accuracy of 0.8750
```

DISCUSSION

- **Model Selection Rationale**: Decision Tree was chosen due to its superior performance in accuracy.
- **Challenges**:
  - Data Imbalances: Although our data set was not large, some launch sites had significantly more data
- **Generalization**: The model performed well on test data and is expected to generalize to future launches.

OVERALL FINDINGS & IMPLICATIONS

Findings

- The Decision Tree model accurately predicts the success of Falcon 9 landings based on the features provided.
- Launch site, orbit type, and payload mass are key determinants of landing success.

Implications

- **For Space Y**: This predictive model provides critical insights for planning and pricing launches, potentially lowering costs and improving competitiveness.
- **For the Industry**: Accurate predictions of reusable rocket landings can lead to significant cost reductions and more frequent commercial launches.

CONCLUSION

- **Summary**:
  - Successfully built a machine learning model (Decision Tree) to predict Falcon 9 first-stage landing success.
  - Achieved high accuracy (87.5%), which can help Space Y optimize its operations and compete with SpaceX.
- **Future Work**:
  - Incorporate more features, such as weather conditions or sea state, to further improve accuracy.
  - Deploy the model in a production environment for real-time launch predictions.