IBM Applied Data Science Capstone Project

IBM Developer SKILLS NETWORK

SpaceX Falcon 9 Landing Prediction

By Jung Hyun RYU

PRESENTATION OUTLINE

Executive Summary (1)

- This project aimed to predict the successful landing of SpaceX Falcon 9 first stages, a critical factor in SpaceX's cost advantage in the space launch market. Key points include:
- **Objective**: Develop a machine learning model to predict Falcon 9 first stage landing success.
- **Business Impact:** SpaceX's ability to reuse first stages allows them to offer launches at $62 million compared to competitors' $165+ million. Accurate predictions can help estimate launch costs and provide insights for competitive bidding.
- **Methodology:**
  - Data Collection: Gathered data from SpaceX API and web scraping.
  - Data Wrangling: Cleaned and preprocessed data, handling missing values and encoding categorical variables.
  - Exploratory Data Analysis (EDA): Used visualizations and SQL queries to uncover patterns and relationships.
  - Feature Engineering: Created new features and encoded categorical variables.
  - Machine Learning: Developed and compared multiple models (Logistic Regression, SVM, Decision Trees, K-Nearest Neighbors).

Executive Summary (2)

- Key Findings:
  - Launch success rates improved significantly over time.
  - Certain launch sites (e.g., KSC LC-39A) showed higher success rates.
  - The relationship between payload mass and launch success seems more complex than a simple positive correlation; Payload mass correlated positively with success rates for some orbits (LEO, ISS) but negatively for others (GTO).
  - The Decision Tree classifier demonstrated the highest accuracy (88%) among tested models.

- Conclusions: The project successfully developed a predictive model for Falcon 9 first stage landings and provided valuable insights into factors affecting launch success rates. These findings can significantly impact business decisions in the commercial space industry.

INTRODUCTION

Capstone Project: Predicting Falcon 9 First Stage Landing

- This project aims to predict whether the Falcon 9 first stage will land successfully—a critical factor in SpaceX's cost advantage.
- **How will this benefit SpaceX?** Accurate predictions help **estimate launch costs** and provide **insights for competitive bidding** against SpaceX.
- **How will we do it?** The project involves **collecting and formatting data via an API**, followed by **analysis and modeling**. Example outcomes include **identifying key factors behind successful launches**.

Section 1 : Methology

Methodology Overview

- **Step 1: Data Collection**
  - Use the **SpaceX API** and **web scraping** from Wikipedia to gather relevant data.
- **Step 2: Data Wrangling**
  - **Handle missing data** by replacing it with appropriate values (e.g., mean or other methods).
  - **Change data types** to match the analysis requirements.
  - **Encode categorical data** using techniques like **one-hot encoding** or **dummy variables**.
  - **Normalize and standardize data**, including handling missing values effectively.
- **Step 3: Exploratory Data Analysis (EDA)**
  - Perform EDA using **visualizations** and **SQL** to uncover patterns and relationships.
- **Step 4: Interactive Visual Analytics**
  - Create interactive dashboards using **Folium** and **Plotly Dash** for better insights.
- **Step 5: Predictive Analysis**
  - Develop and apply **classification models** to predict outcomes.
  - Focus on **building, tuning, and evaluating** models to ensure accuracy and reliability.

Data Collection Flowchart - (1) SPACEX API

- [Github URL : SPACEX API]

TASK 1: Request and parse the SpaceX launch data using the GET request

- Use the API again to get information about the launches using the IDs given for each launch.

TASK 2: Filter the dataframe to only include `Falcon 9` launches by data wrangling

- Filter the data dataframe using the `BoosterVersion` column to only k eep the Falcon 9 launches. Save the filtered data to a new dataframe call ed `data_falcon9`.

TASK 3: Dealing with Missing Values

- Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

Data Collection Flowchart - (2) WEB SCRAPPING

- [Github URL : Web Scrapping]

TASK 1: Request the Falcon9 Launch Wiki page from its URL

- Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

TASK 2: Extract all column/variable names from the HTML table header

- Collect all relevant column names from the HTML table header

TASK 3: Create a data frame by parsing the launch HTML tables

- Create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary is filled with launch records.

Data Wrangling Flowchart

- [Github URL : Data Wrangling]()

TASK 1: Calculate the number of launches on each site

- Use the method `value_counts()` on the column `LaunchSite` to determine the numb er of launches on each site

TASK 2: Calculate the number and occurrence of each orbit

- Use the method `.value_counts()` to determine the number and occurrence of each o rbit in the column `Orbit`

TASK 3: Calculate the number and occurence of mission outcome of the orbits

- Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes`.Then assign it to a variable landing_outcomes.

TASK 4: Create a landing outcome label from Outcome column

- Create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`

EDA with SQL - SQL Queries Summary

- **Table Creation:**
  - Created a new table `SPACEXTABLE` with data from `SPACEXTBL`, ensuring rows where the `Date` field is not null.
- **Distinct Launch Sites:**
  - Queried distinct `LAUNCH_SITE` values from the `SPACEXTBL` table.
- **Filtered Data Retrieval:**
  - Retrieved data from `SPACEXTBL` where `LAUNCH_SITE` starts with "CCA," limited to 5 rows.
- **Aggregations:**
  - Calculated the total payload mass (`SUM(PAYLOAD_MASS__KG_)`) for missions with `Customer = 'NASA (CRS)'`.
  - Calculated the average payload mass (`AVG(PAYLOAD_MASS__KG_)`) for missions where the `Booster_Version`matches "F9 v1.1."

[Github URL – EDA with SQL]

EDA with Visualization – Charts Summary

- **Scatter plots** are used to show: 1) The relationship between FlightNumber and PayloadMass, 2) The distribution of successful and failed landings based on various features
- **Categorical plots**, such as those created by seaborn's catplot function, are useful for showing 1) PayloadMass distribution across different Orbit types and 2) Success rates for different LaunchSites
- **Line plots** are employed to show 1) Success rate changes over time (by year or flight number), 2) Changes in payload mass over successive launches
- **Bar charts** are useful for showing 1) The count of launches for each orbit type, 2) Success rates for different booster versions

Github URL – EDA Visualization

Interactive Map with Folium (1) - Map Objects Created

Github URL – Map with Folium

- **Circles**
  - A blue circle at NASA Johnson Space Center's coordinates
  - Circles for each launch site location
- **Markers**
  - A marker at NASA Johnson Space Center
  - Markers for each launch site
  - Markers for the closest city, railway, and highway to a specific launch site (CCAFS SLC-40)
- **Popup Labels**
  - Added to circles and markers to display site names and distances
- **Lines**
  - Polylines connecting the CCAFS SLC-40 launch site to the closest city, railway, and highway
- **Marker Cluster**
  - Used to group nearby markers for better visualization when zoomed out

Interactive Map with Folium (2) - Purpose of Added Objects

- The purposes of the objects added include...
  - To visually represent the geographical locations of SpaceX launch sites
  - To highlight the NASA Johnson Space Center as a reference point
  - To show the proximity of important infrastructure (city, railway, highway) to a specific launch site
  - To **provide interactive information through popup labels**
  - To illustrate the distances between the launch site and nearby points of interest
  - To **improve map readability** by clustering markers when zoomed out

Dashboard with Plotly Dash (1) - Plots and Interactions

[Github URL – Plotly Dash]

- Dropdown Menu
  - To select a specific launch site or view data for all sites
- Pie Chart
  - To show the total successful launches count for all sites or success vs. failure counts for a specific site
  - To update dynamically based on the selected launch site from the dropdown
- Range Slider
  - To enable users to select a payload mass range
- Scatter Plot
  - To display the correlation between payload mass and launch success
  - To update based on the selected launch site and payload range

Dashboard with Plotly Dash (2) - Purpose of Added Elements

- <u>The purposes of elements added include...</u>
  - To provide an interactive and comprehensive view of SpaceX launch data
  - To allow users to explore the relationship between various factors and launch success rates
  - To visualize the impact of launch site selection on mission outcomes
  - To analyze how payload mass affects launch success across different sites
  - To enable data filtering and focused analysis on specific launch sites or payload ranges

Predictive Analysis (1) - Classification building process

Github URL - Classification

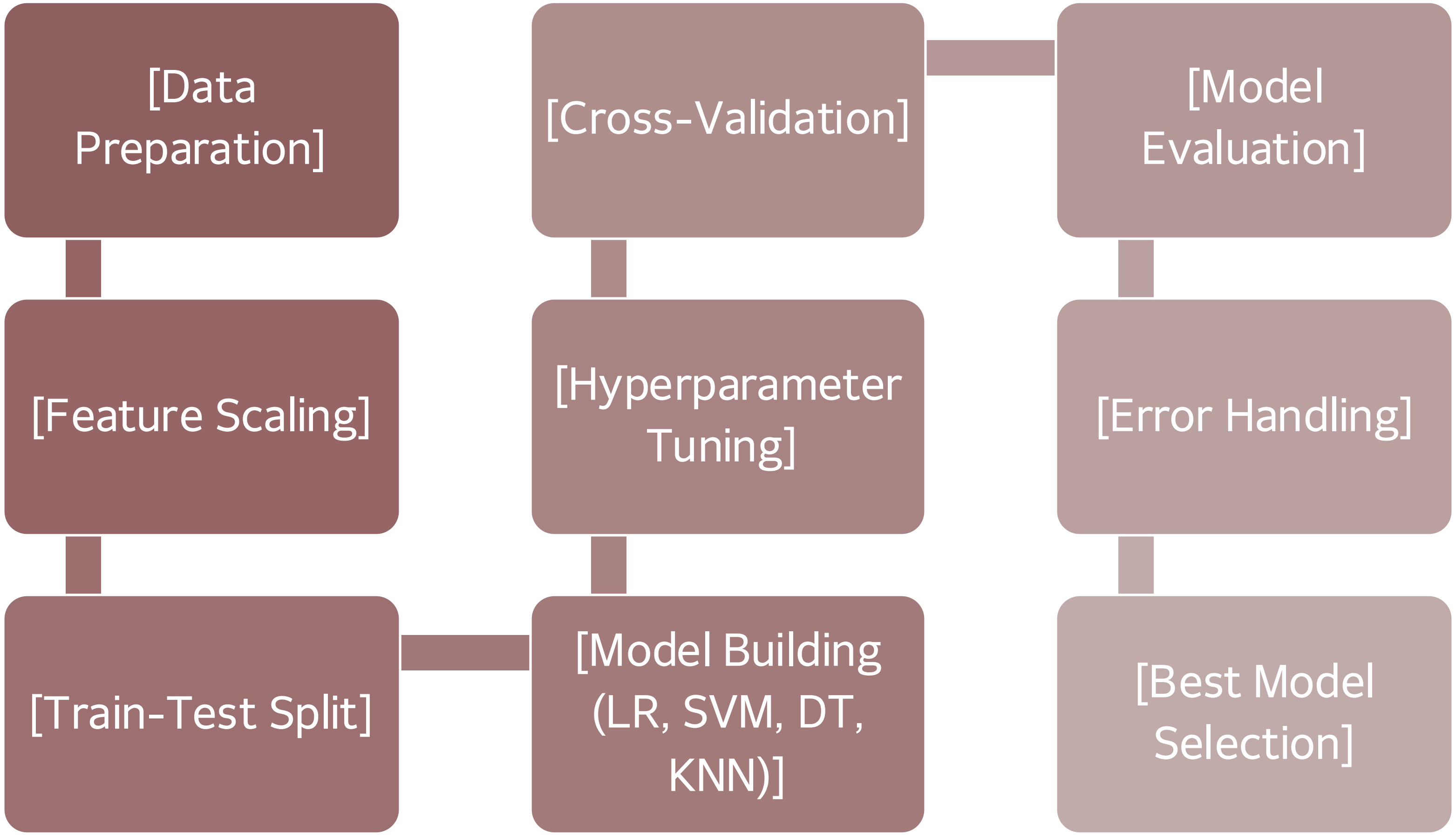
- Data Preparation - Created a 'Class' column for the target variable, Standardized the feature data using StandardScaler, Split data into training (80%) and test (20%) sets
- Model Building and Evaluation

- Logistic Regression
  - Created a GridSearchCV object with 10-fold cross-validation
  - Tuned hyperparameters: C, penalty, solver
  - Best parameters: C=0.01, penalty='l2', solver='lbfgs'
  - Validation accuracy: 84.64%
  - Test accuracy: 83.33%
- Support Vector Machine (SVM)
  - Used GridSearchCV for hyperparameter tuning
  - Explored different kernel types and C values
- Decision Tree
  - Utilized GridSearchCV for optimization
  - Tuned parameters: criterion, splitter, max_depth, max_features, min_samples_leaf, min_samples_split
  - Encountered issues with 'max_features' parameter
- K-Nearest Neighbors (KNN)
  - Applied GridSearchCV for finding optimal parameters

Predictive Analysis (2) - Classification building process

- Model Improvement
  - Iteratively refined models based on cross-validation results,
  - Addressed warnings and errors, particularly for Decision Tree's 'max_features' parameter
- Best Performing Model Selection
  - Compared test accuracies across all models,
  - Evaluated using confusion matrices and additional metrics

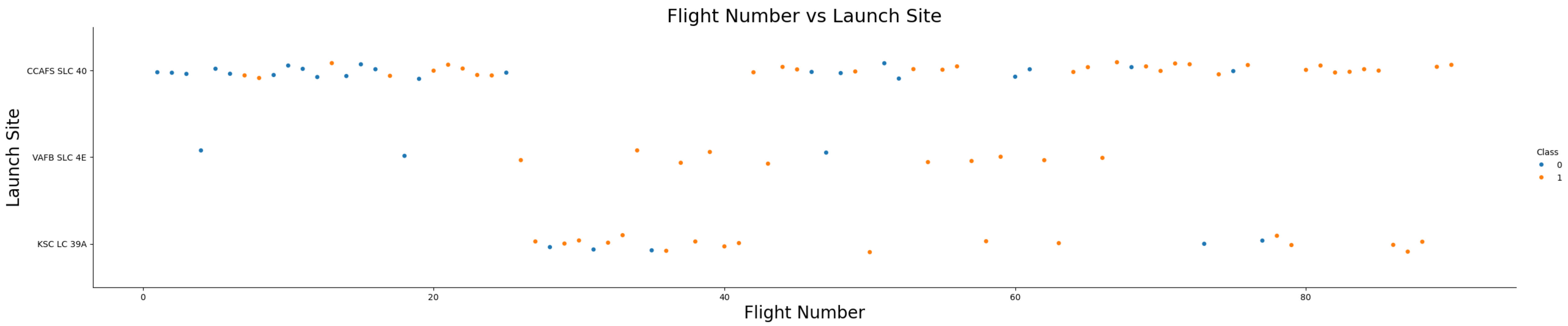
Predictive Analysis (3) – Flowcharts



SECTION 2.

Result & Insights drawn from EDA

(1) EDA with Visualization

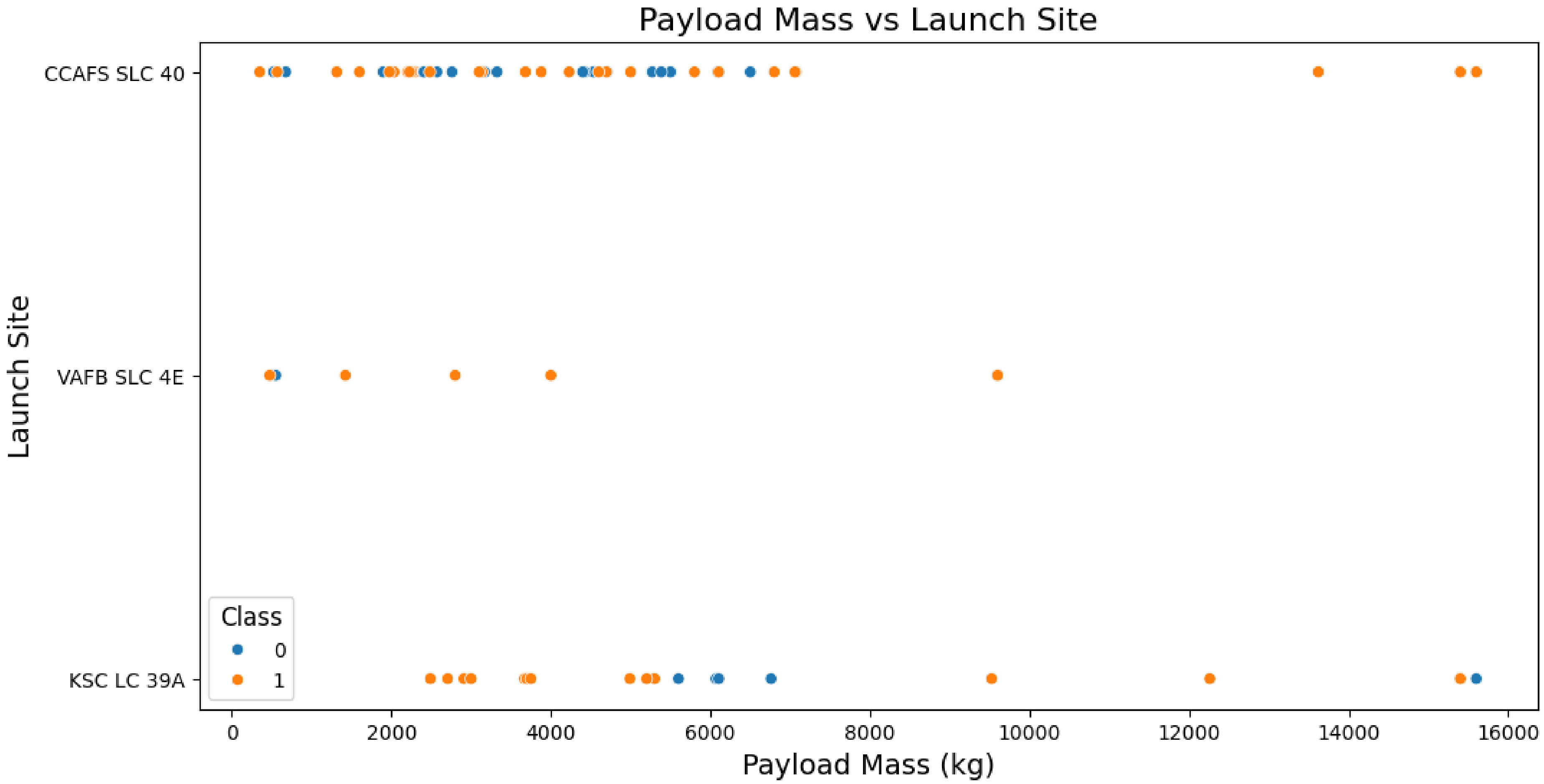


[Flight Number vs Launch Site]

- **Launch success improved over time**: The earliest flights mostly failed, while recent flights have had a much higher success rate.
- **CCAFS SLC-40 is the most frequently used site**: This launch site accounts for approximately half of all SpaceX launches.
- **Higher success rates at certain sites**: VAFB SLC-4E and KSC LC-39A have demonstrated better success rates compared to other launch sites.
- **Increasing success trend**: There appears to be a trend of increasing success rates for newer launches, likely due to improvements in technology and procedures over time.

[Payload Mass vs Launch Site]

- **Payload mass correlation**: For every launch site, there is a general trend that higher payload mass correlates with higher success rates. • High payload mass success: The majority of launches with payload mass over 7000 kg were successful.
- KSC LC-39A performance: **KSC LC-39A demonstrates the highest overall success rate** among launch sites.
  - It has a **100% success rate** for payload mass **under 5500 kg.**
- **Success rate variations:** Successful launches (likely represented by green dots) appear to be distributed across various payload masses; there are likely other factors influencing launch outcomes, such as improvements in technology and procedures over time.



SpaceX has optimized its Falcon 9 rocket for heavier payloads, particularly at the KSC LC-39A launch site.

However, it's important to note that correlation does not imply causation, and other variables like **technological advancements and operational experience likely contribute to the improved success rates** over time.

Success Rate by Orbit Type

100% Success Rate:

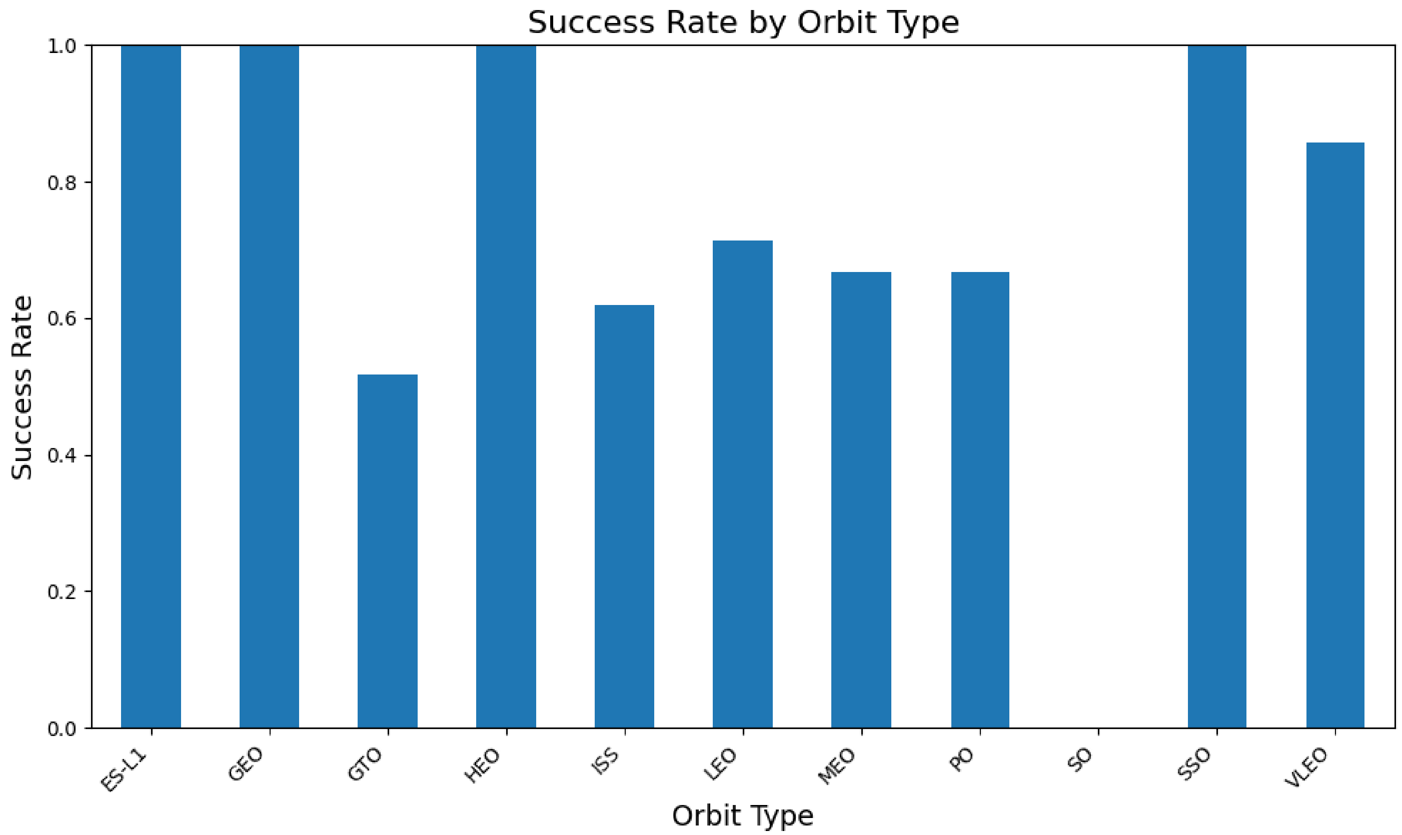
- ES-L1 (Earth-Sun Lagrange Point 1)
- GEO (Geostationary Earth Orbit)
- HEO (Highly Elliptical Orbit)
- SSO (Sun-Synchronous Orbit)

0% Success Rate:

- SO (Sub-Orbital)

50-85% Success Rate:

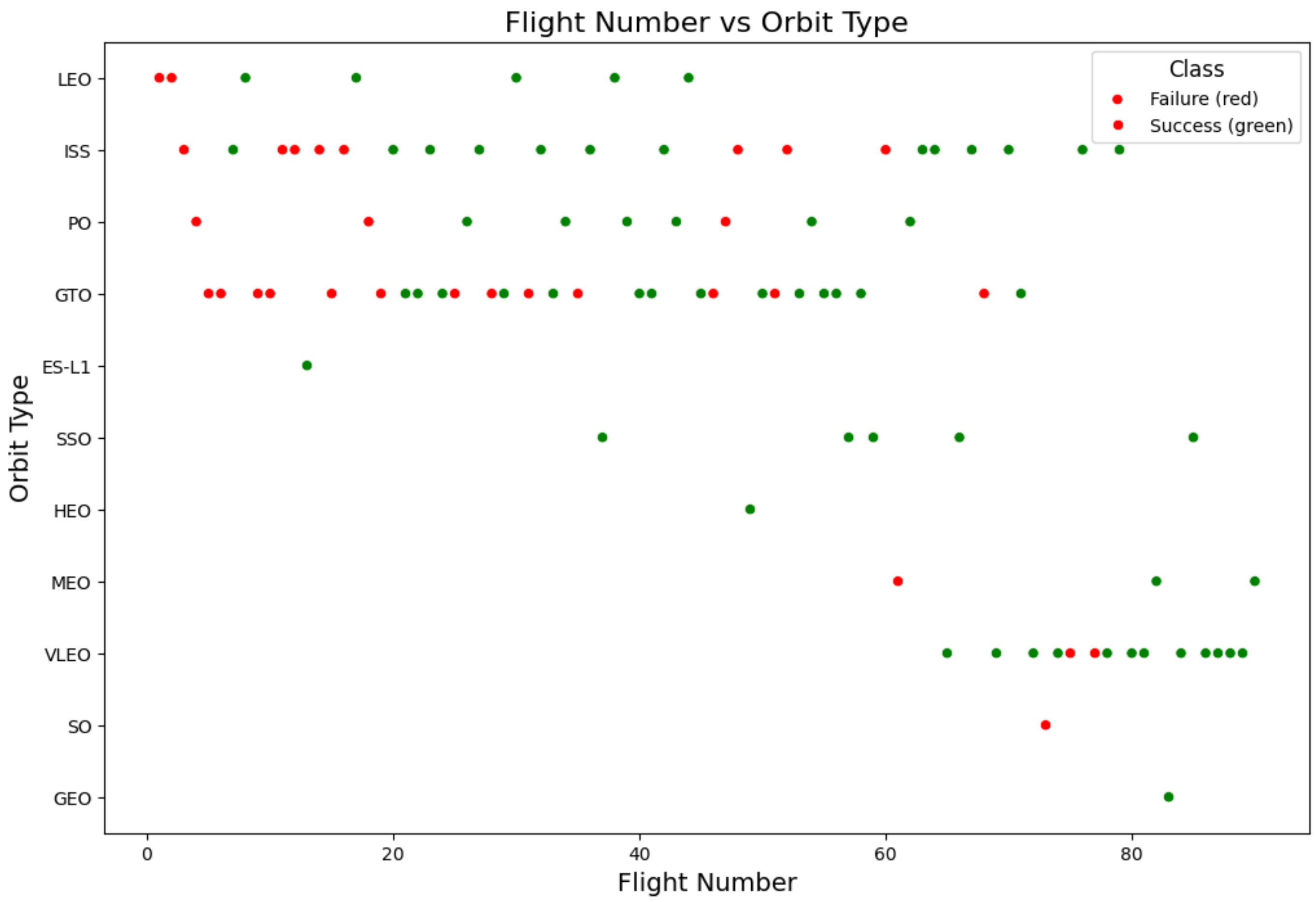
- GTO (Geosynchronous Transfer Orbit)
- ISS (International Space Station orbit)
- LEO (Low Earth Orbit)
- MEO (Medium Earth Orbit)
- PO (Polar Orbit)



- SpaceX has optimized its Falcon 9 rocket for certain specialized orbits (ES-L1, GEO, HEO, SSO), achieving perfect success rates.
- The most challenging orbit appears to be Sub-Orbital (SO), with no successful launches recorded.
- The majority of SpaceX launches target orbits with moderate success rates(GTO, ISS, LEO, MEO, PO), indicating these are common but more challenging destinations.

Flights Number vs Orbit Type

- LEO missions benefit from SpaceX's accumulated experience
- GTO missions face consistent challenges regardless of flight number
- Factors other than flight experience may play a more significant role in GTO mission success

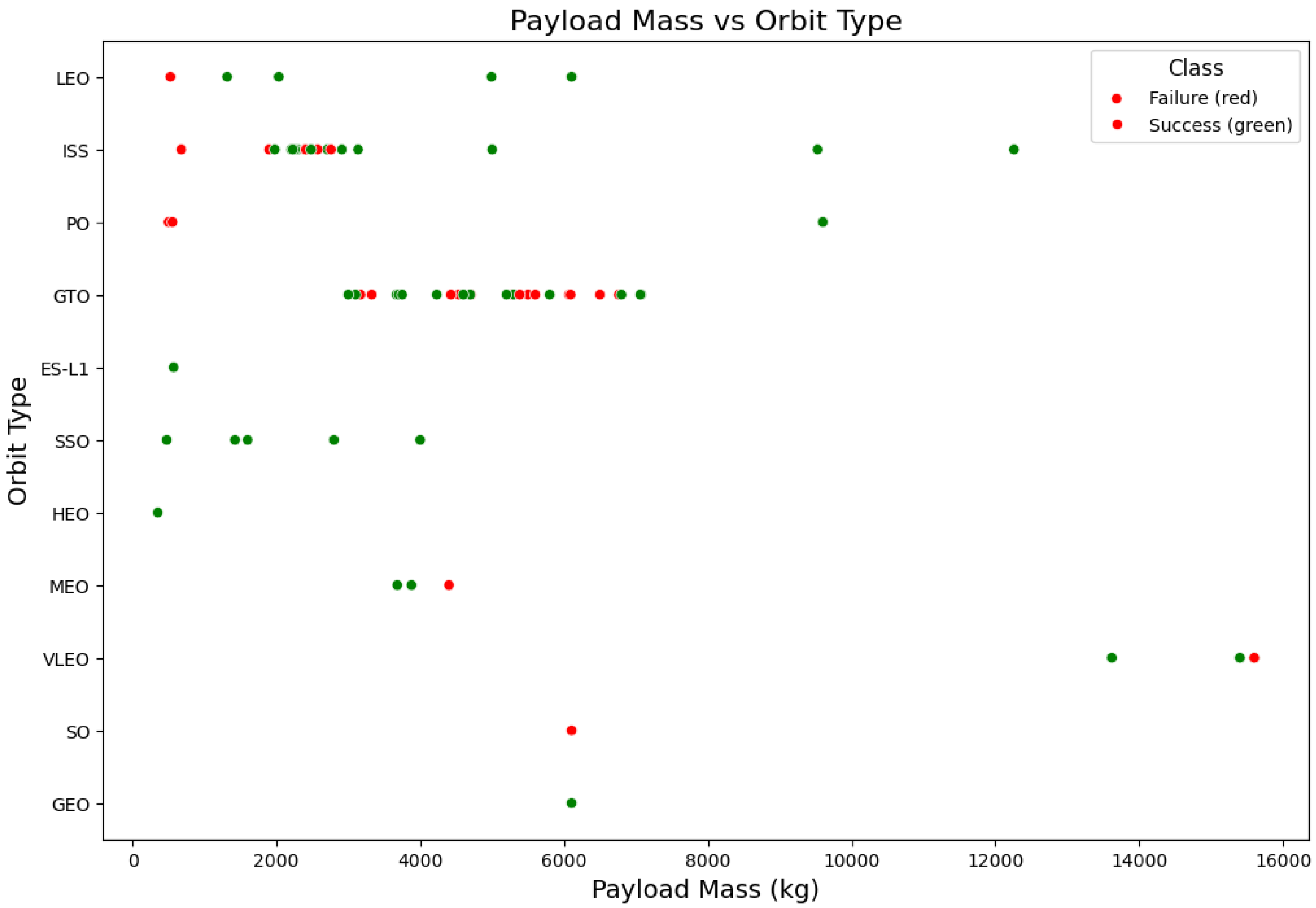


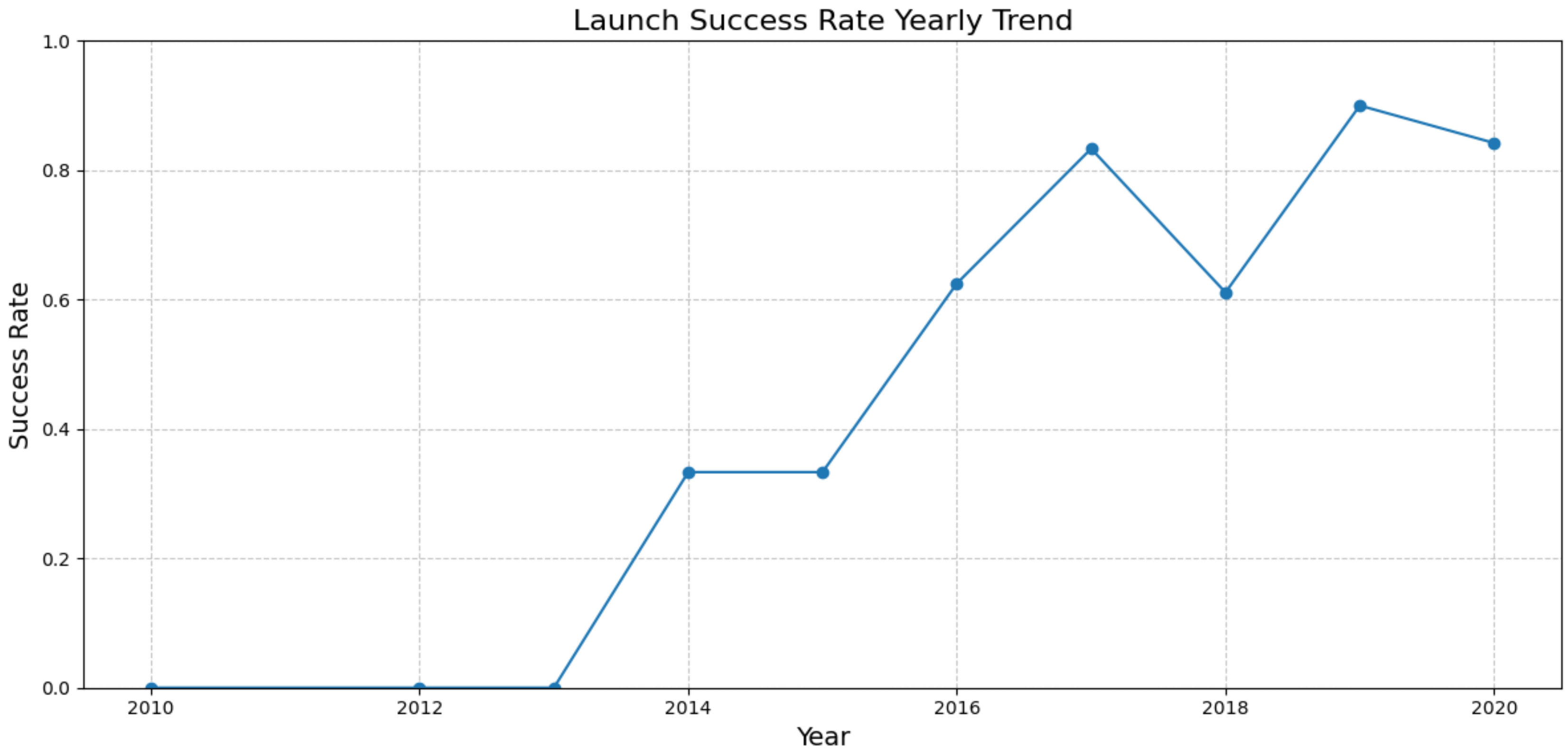
Payload Mass vs Orbit Type

- The challenges of reaching GTO with heavier payloads may reduce the likelihood of successful launches

- Missions to the ISS seem better optimized for handling larger payloads

- The impact of payload mass on launch success varies significantly depending on the target orbit





Launch Success Rate Yearly Trend

- The success rate since 2013 kept increasing till 2020.
- More is better :)

SECTION 2.

Result & Insights drawn from EDA

(2) EDA with SQL

All Launch Sites Name

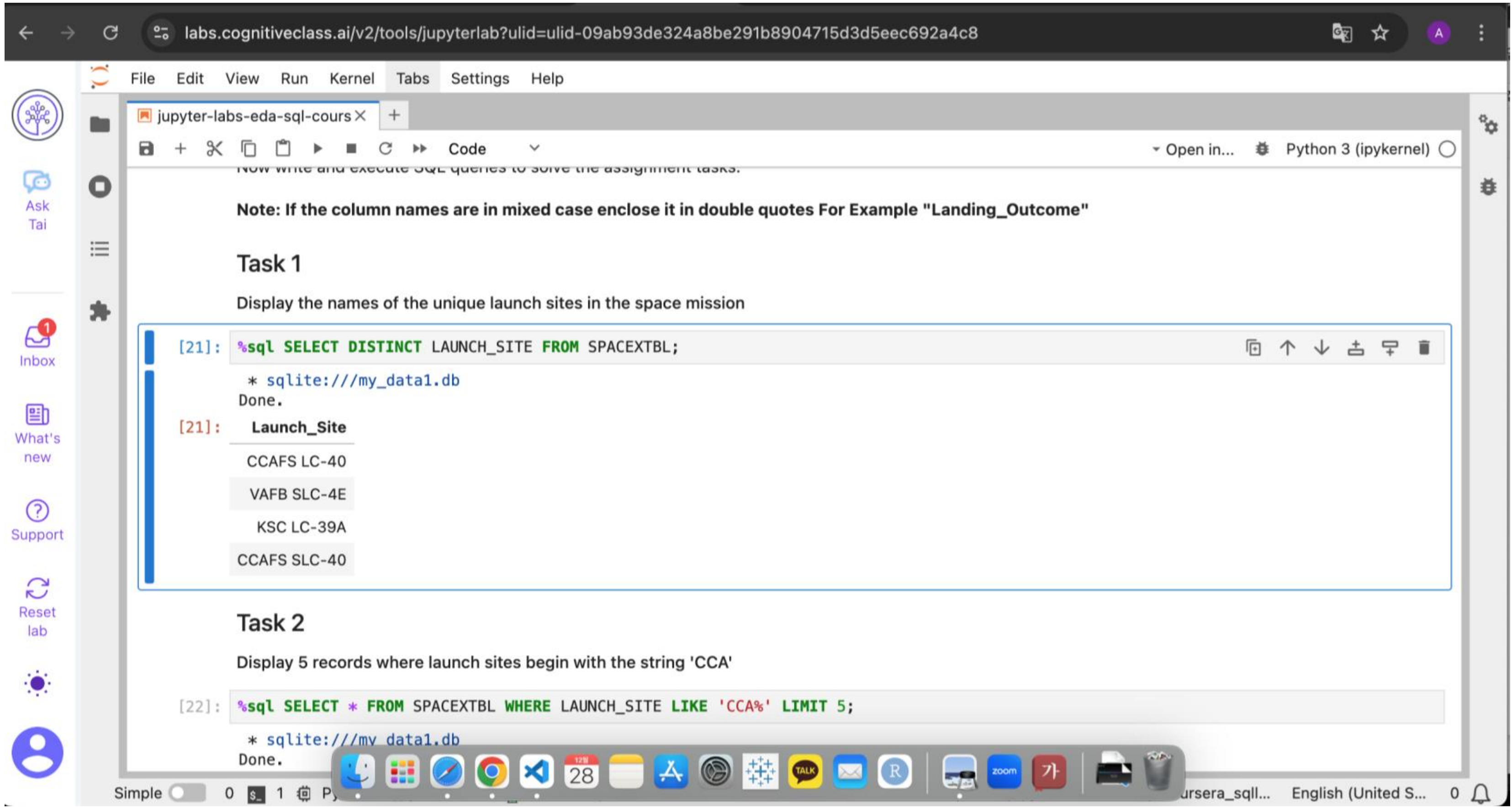
- The launch sites include

1)CCAFS LC-40

2)VAFB SLC-4E

3)KSC LC-39A

4)CCAFS SLC-40



Launch sites beginning with the string 'CCA'

Task 2

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

[22]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

[22]:

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

Task 3

Total payload mass by boosters launched by NASA (CRS)

Task 3

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

```
[23]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

[23]:

Total_Payload_Mass
45596

Average payload mass by booster ver. F9 V1.1

Task 4

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

```
[24]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1';
```

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

[24]:

Average_Payload_Mass
2928.4

Date when the first successful landing outcome in ground pad

Task 5

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

Hint:Use min function

[25]:
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

[25]:

MIN(Date)
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

[26]:
```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND
```

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

[26]:

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

Successful Drone Ship Landing with Payload between 4000 and 6000

: %sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

Total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

[28]:
```
%sql SELECT count(Mission_Outcome) AS Total_Mission_Outcome FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

[28]:

Total_Mission_Outcome
1
98
1
1

Boosters Carried Maximum Payload

Task 8

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

[27]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

[27]:

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

Launch records in 2015

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
```

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

]:

Month	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

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

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 Count, RANK() OVER (ORDER BY COUNT() DESC) AS Rank FROM SPACEXTBL WHERE Date BETWEEN '2010-
```

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

]:

Landing_Outcome	Count	Rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8

SECTION 3.
Launch Sites Proximities Analysis

Interactive map with Folium

Task 1: All launch sites on a map

```
            icon_anchor=(0,0),
            html=f'<div style="font-size: 12; color:#d35400;"><b>{site["Launch Site"]}</b></div>'
        )
    )

    # Add circle and marker to the map
    site_map.add_child(circle)
    site_map.add_child(marker)

# Display the map
site_map
```

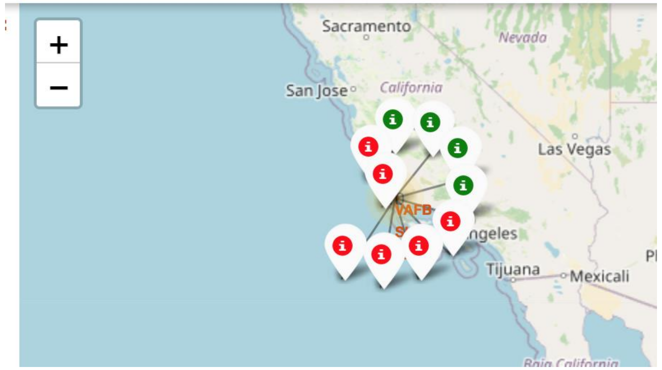
The generated map with marked launch sites should look similar to the following:

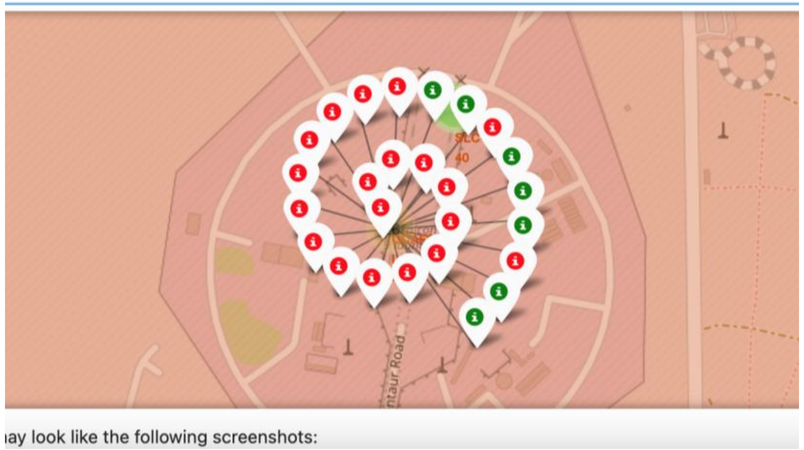
0 1 Python (Pyodide) | Idle   Mode: Command   Ln 10, Col 11   lab_jupyter_launch_site_location.ipynb

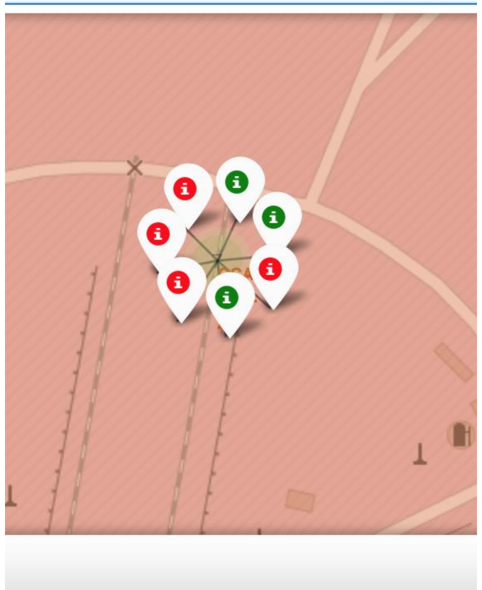
Task 2: Success/failed launches for each site on the map

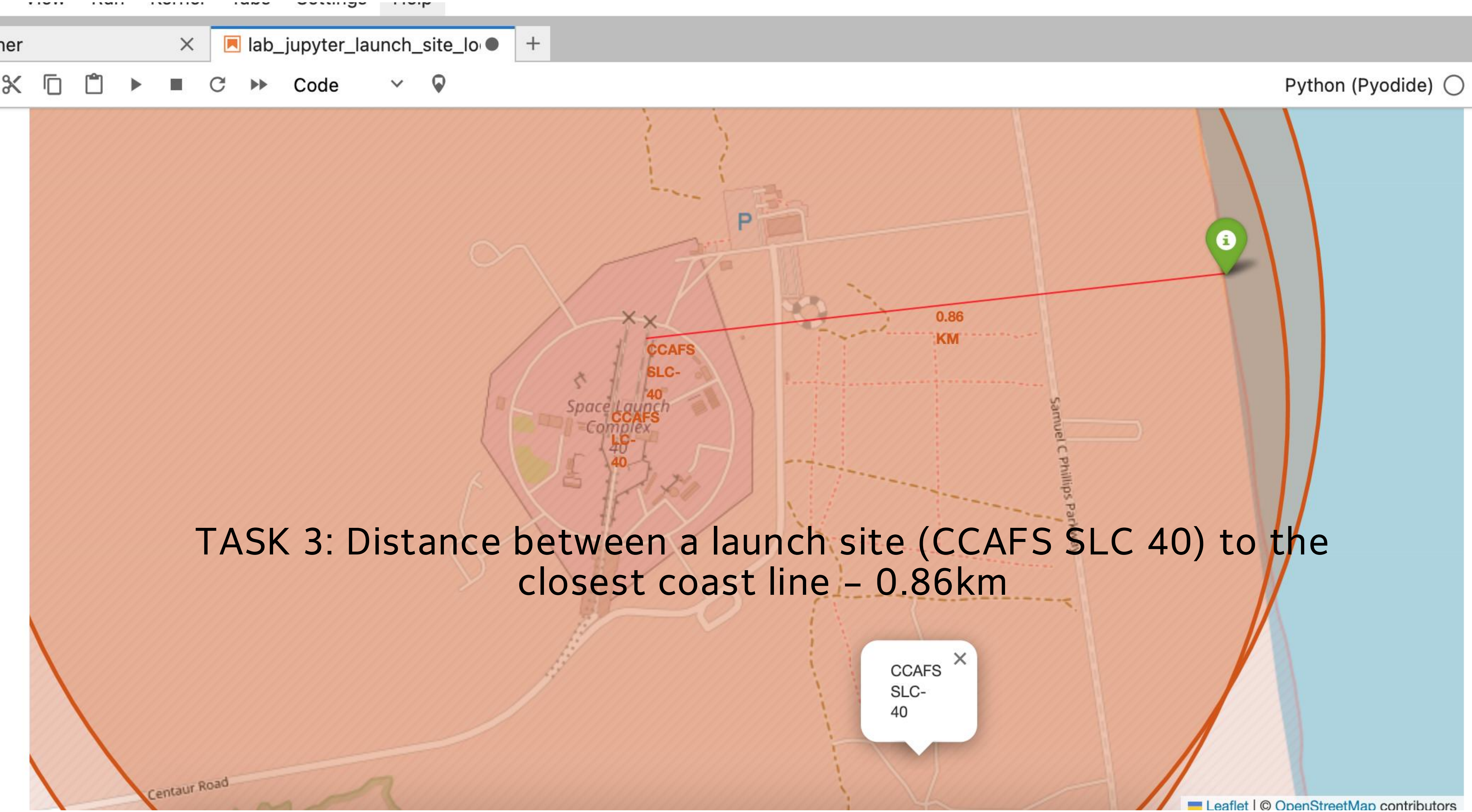
- Green = Successful
- Red = Failed
- <u>Launch Site **KSC LC-39A (fourth on the right)**</u> has a very high Success Rate.

```python
    # Create and add a Marker cluster to the site map
    marker = folium.Marker([record['Lat'], record['Long'
              icon=folium.Icon(color='white', icon_c
    marker_cluster.add_child(marker)
site_map
```









TASK 3: Distance between a launch site (CCAFS SLC 40) to the closest coast line – 0.86km

TASK (additional):
A marker and lines with distance to a closest city, railway, highway, etc.

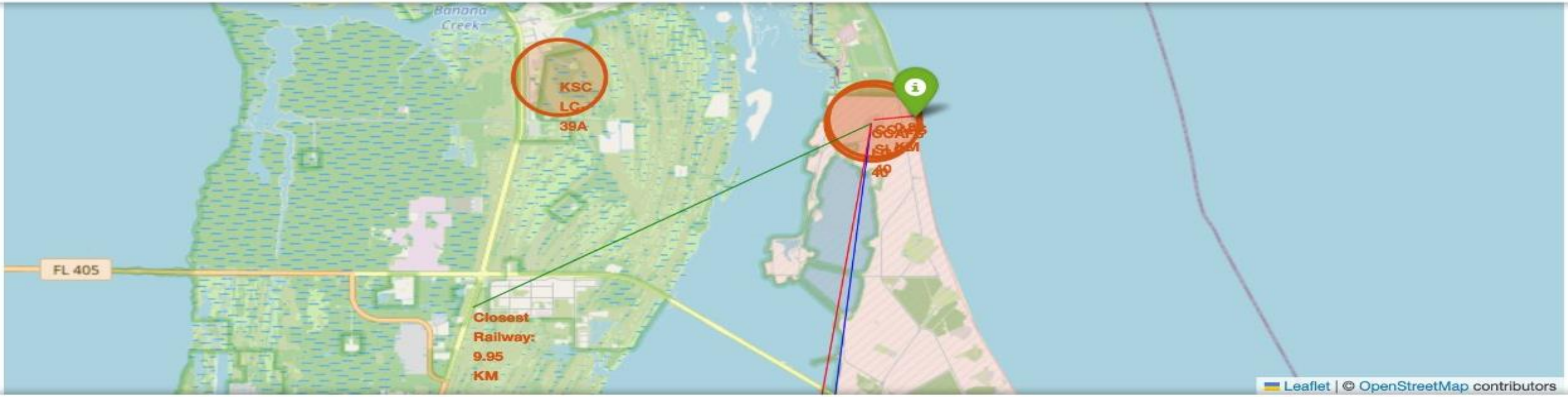
Edit View Run Kernel Tabs Settings Help

Launcher | lab_jupyter_launch_site_lo

Code

Python (Pyodide)

```
        html=f'<div style="font-size: 12; color:#d35400;"><b>Closest Highway: {highway_distance:.2f} KM</b></div>'
    )
)
site_map.add_child(highway_marker)
```

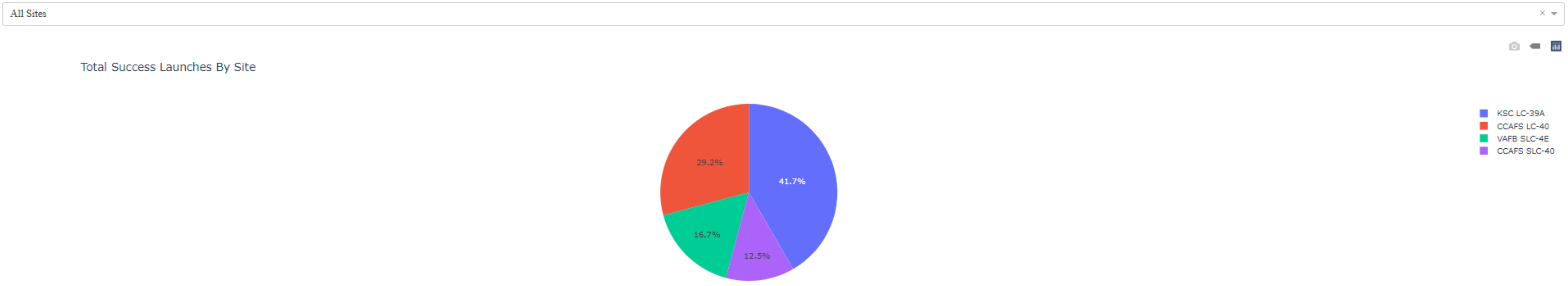


[19]:

```
# Draw lines between markers and launch site
folium.PolyLine (locations=[[launch_site_lat, launch_site_lon], [city_lat, city_lon]], weight=1, color='blue').add_to(site_map)
folium.PolyLine(locations=[[launch_site_lat, launch_site_lon], [railway_lat, railway_lon]], weight=1, color='green').add_to(site_ma
```

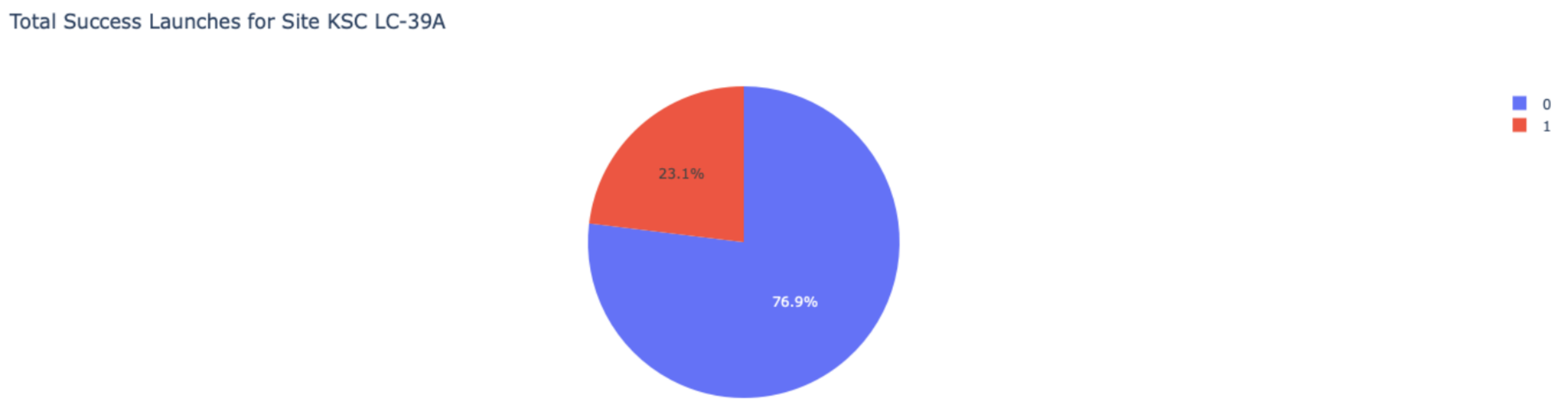
SECTION 4.

Interactive Dashboard with Plotly dash

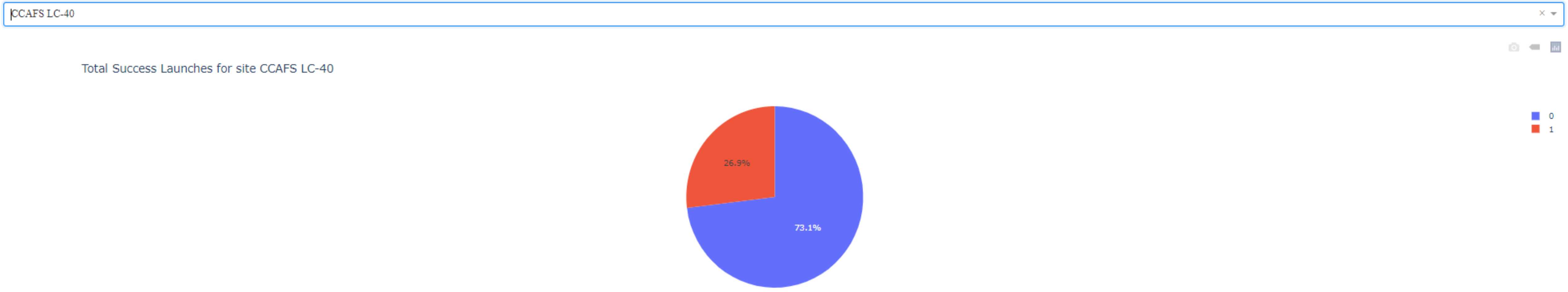


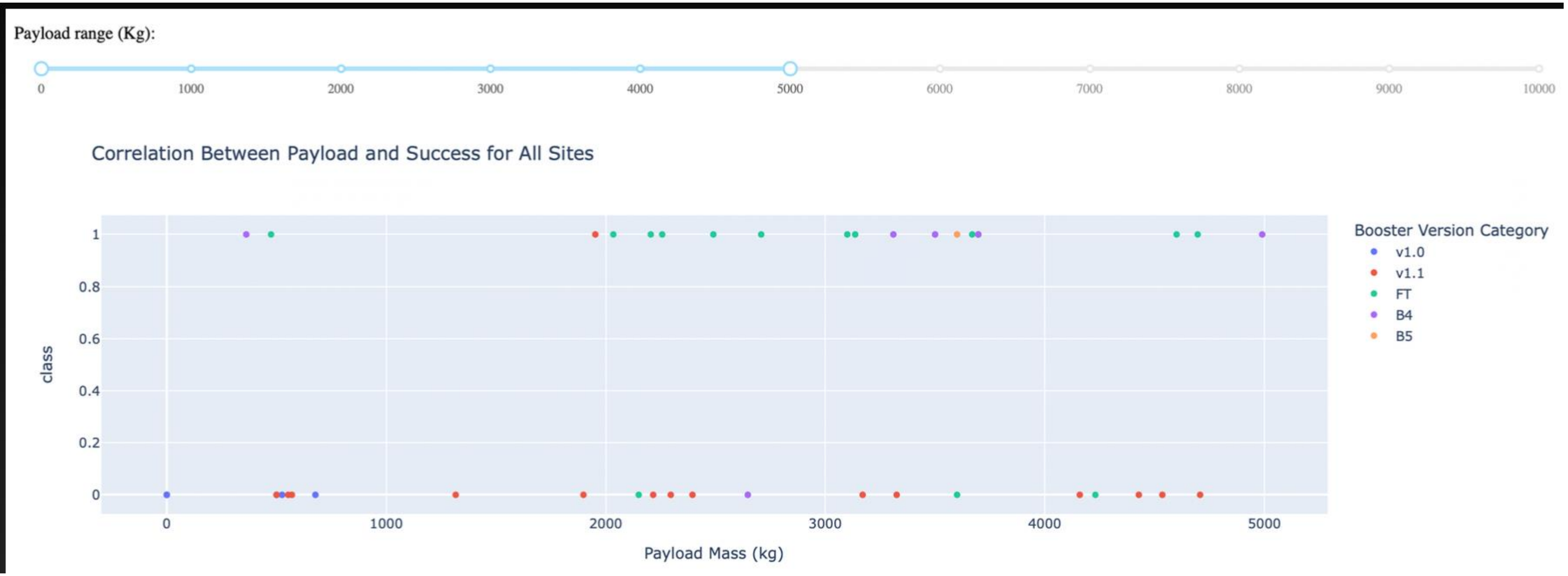
Total Launch Success at All Sites

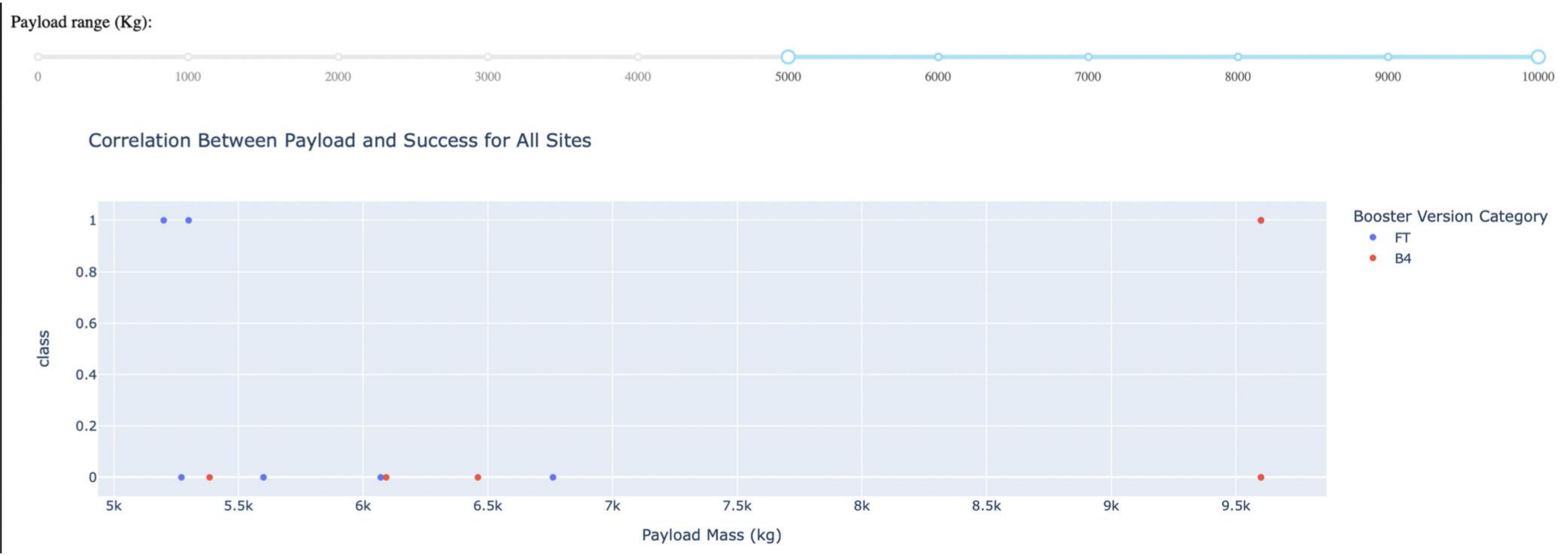
- The chart shows that from all the sites, KSC LC-39A has the most successful launches.



KSC LC-39A has the highest launch success rate (76.9%), while other sites have lower rates (e.g., CCAFS LC-40 with 73.1%)







The charts show that payloads between 2000 and 5500 kg have the highest success rate.

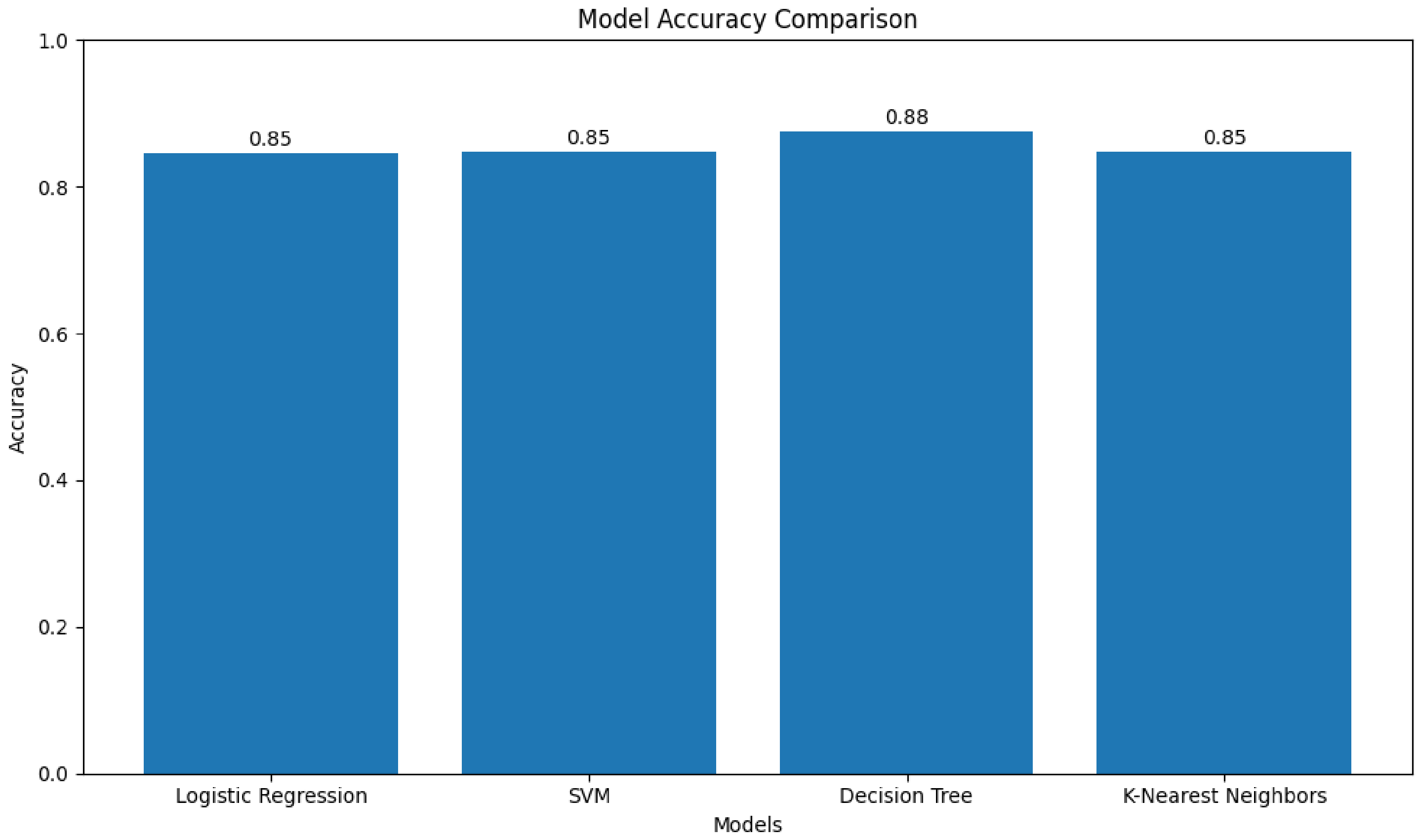
SECTION 5.

Predictive Analysis

Classification

CLASSIFICATION ACCURACY
(BAR CHART)

- Based on the model comparison results, **the Decision Tree classifier demonstrates the highest accuracy** among the four models tested, achieving a score of 0.88. This performance appears to improve with increased testing, suggesting robust generalization capabilities.
- Key points:
  - Decision Tree outperforms other models
  - Accuracy score: 0.88
  - Performance tends to improve with more testing
  - Indicates good generalization ability



CONFUSION MATRIX

<u>The Decision Tree classifier demonstrates the highest accuracy</u> among the four models tested, achieving a score of 0.88. at the first trial.

The accuracy above the matrix of 0.903 is the second trial. Therefore, we can assume that the performance of the model would improve with increased testing, suggesting robust generalization capabilities.

CONCLUSION (1)

- **Key Insights:**
  - Launch success rates improved significantly over time.
  - The relationship between payload mass and launch success seems more complex than a simple positive correlation; Heavier payloads correlated with higher success rates for certain orbits (LEO, ISS) but showed challenges for GTO.
  - KSC LC-39A has the highest success rate of the launches from all the sites.
  - Specific orbits (ES-L1, GEO, HEO, SSO) showed 100% success rates, while Sub-Orbital launches were the most challenging.
  - Launch site and payload mass were significant factors in predicting landing success.

- **Business Impact:**
  - The predictive model can help estimate launch costs more accurately.
  - Insights can be used for competitive bidding against SpaceX and for optimizing launch strategies.

CONCLUSION (2)

- **Data Collection and Wrangling:**
  - Successfully gathered data from SpaceX API and web scraping.
  - Cleaned and preprocessed data for analysis, handling missing values and encoding categorical variables.

- **Exploratory Data Analysis:**
  - CCAFS SLC-40 was the most frequently used launch site.
  - KSC LC-39A showed the highest overall success rate.

- **Interactive Visual Analytics:**
  - Created interactive maps and dashboards to visualize launch site locations and success rates.
  - Analyzed proximity of launch sites to infrastructure and its potential impact on success rates.

- **Predictive Analysis:**
  - Developed and compared multiple machine learning models (Logistic Regression, SVM, Decision Trees, K-Nearest Neighbors).
  - The Decision Tree classifier demonstrated the highest accuracy (88%) among the models tested.
  - Model performance improved with increased testing, suggesting good generalization capabilities.

THANK YOU!

[IBM Data Science Professional Certificate](#)

[Coursera](#)