IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gopi

February 28, 2026

Outline

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

Executive Summary

- **Project Overview:** https://github.com/gopikrishnan-t-s/Applied-Data-Science-Capstone
- **Analyzed historical SpaceX launch data to evaluate operational performance and mission success**
- **Methodology:**
  - Data cleaning, normalization, and validation
  - SQL-based exploratory data analysis (aggregations, success rates, temporal trends)
  - Visualization of payload, booster, and site performance
  - Predictive modeling using supervised classification
- **Key Results:**
  - Launch frequency increased significantly over time
  - Year-over-year reliability improvements observed
  - Booster version and payload mass influence mission success probability
  - Launch site performance differences identified

Introduction

Project Background & Context:

- Commercial space launch industry has experienced rapid growth and increased competition.
- SpaceX has significantly expanded launch cadence and reusable rocket operations.
- Reliable mission execution is critical for cost efficiency, customer trust, and operational scalability.

Business & Analytical Context:

- Large volumes of historical launch data provide opportunity for quantitative performance analysis.
- Structured data enables statistical evaluation of reliability, operational trends, and risk factors.

Key Questions / Problems to Address:

- What factors most influence mission success probability?
- How has launch reliability evolved over time?
- Do specific booster versions or launch sites perform better?
- Does payload mass correlate with mission outcome?
- Can historical data support predictive modeling of future launches?

Section 1

Methodology

Methodology

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

Data Collection- SPACEX API and Web Scrape

- **Objective:** Gather and organize launch data for SpaceX Falcon 9 and Falcon Heavy missions.
- **Sources:** Wikipedia launch tables, SpaceX API, and official press releases.
- **Data Retrieval**
    - Perform HTTP GET requests to the designated URLs or API endpoints.
    - Utilize BeautifulSoup to parse HTML tables.
    - Process JSON data returned from the API.
- **Header Extraction**
    - Retrieve column headers from <th> elements in HTML or keys within JSON objects.
    - Clean and standardize header names by removing excess whitespace.
- **Row Extraction**
    - Iterate through <tr> elements in HTML or arrays within JSON data.
    - Extract details such as payload, launch site, date, flight number, and mission outcome.
    - Manage any missing or incomplete values appropriately.

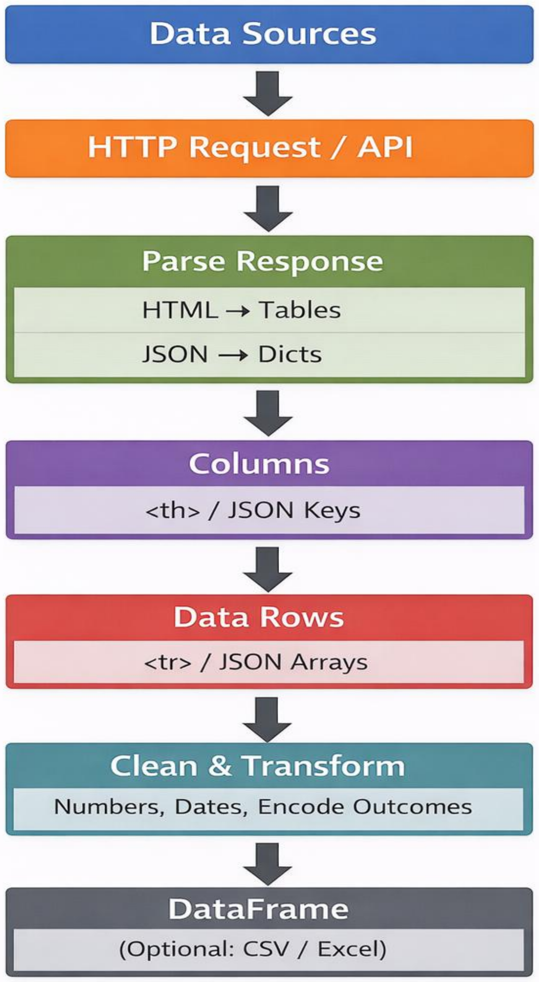
Data Collection continued

Data Preparation & Conversion

- Transform payload mass and flight number into numeric formats.
- Parse date strings into datetime objects.
- Encode categorical results:
    - Success as 1 and
    - Failure as 0.

- **Data Storage & Output**
- Save the data within a Pandas DataFrame.
- Optionally export to CSV, Excel, or a database.

Data Collection- Flowchart



Data Sources

HTTP Request / API

Parse Response

HTML → Tables

JSON → Dicts

Columns

<th> / JSON Keys

Data Rows

<tr> / JSON Arrays

Clean & Transform

Numbers, Dates, Encode Outcomes

DataFrame

(Optional: CSV / Excel)

https://github.com/gopikrishnan-t-s/Applied-Data-Science-Capstone/blob/main/SpaceX_Data_Collection_API.ipynb

Data Wrangling

Raw DataFrame

- Original collected data from web scraping or API

Handle Missing Values

- Drop rows with missing values or fill with placeholders

Convert Data Types

- Convert payload, flight number to numeric
- Convert date columns to datetime
- Ensure categorical columns are of type 'category'

Standardize Formats

- Normalize text fields (strip whitespace, fix casing)
- Standardize launch site names

Data Wrangling - continued

Encode Categorical Variables

- Encode Mission Outcome: Success = 1, Failure = 0
- Booster Version: one-hot or label encoding

Feature Engineering

- Extract month/year from date for analysis
- Create new features: payload ranges, success rates per booster

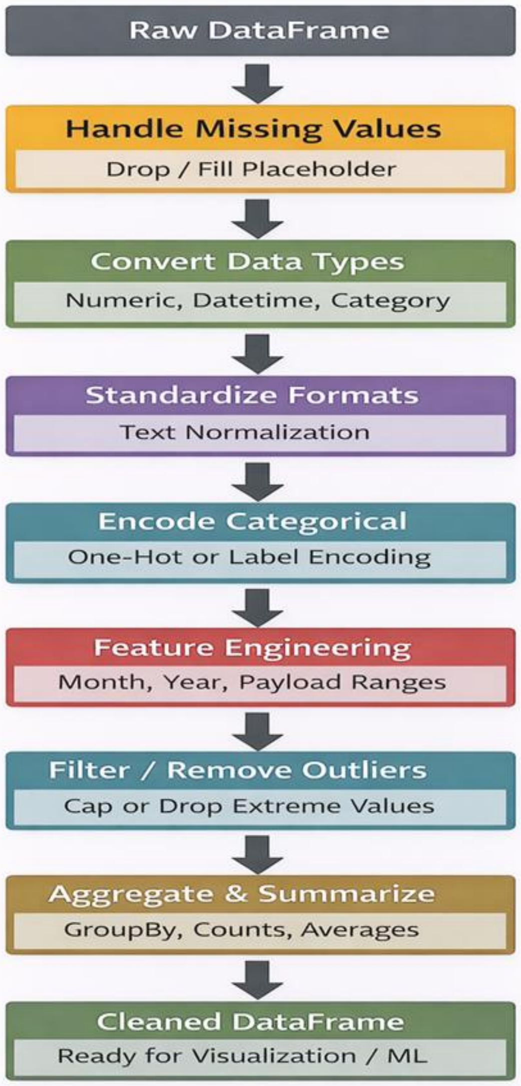
Filter / Remove Outliers

- Identify unrealistic payloads or flight numbers
- Remove or cap extreme values

Aggregate & Summarize

- Group by launch site, booster version, or year
- Compute counts, averages, success rates

Data Wrangling - Flowchart



https://github.com/gopikrishnan-t-s/Applied-Data-Science-Capstone/blob/main/SpaceX_Data_Wrangling.ipynb

EDA with Data Visualization

Explore and visualize SpaceX launch data to identify patterns, relationships, and trends

Dataset Summary

- Check data types and missing values
- Compute descriptive statistics (df.describe())
- Purpose: Ensure data is clean and ready for analysis

Univariate Analysis

- Histograms: Numeric distributions (Payload Mass, Flight Number) → detect skewness and outliers
- Boxplots: Identify spread and extreme values
- Bar / Pie Charts: Categorical frequencies (Mission Outcome, Launch Site) → understand proportion of categories

EDA with Data Visualization - continued

Bivariate / Multivariate Analysis

- Scatter plots: Flight Number vs Payload Mass → explore correlations and trends
- Boxplots: Payload Mass vs Mission Outcome → compare distributions across categories
- Correlation heatmap: Identify relationships between numeric features

EDA with SQL – Task 1

Objective of SQL-Based EDA

- Use structured queries to explore, summarize, and validate SpaceX launch data
- Extract statistical insights prior to visualization and modeling
- Identify trends, distributions, performance metrics, and data quality issues

Task 1: Display the names of the unique launch sites

- SQL query:

```
 """ SELECT DISTINCT Launch_siteFROM SPACEXTABLEORDER BY
Launch_site;"""cur.execute(query)unique_sites = cur.fetchall()unique_sites
```

EDA with SQL – Task 2 and Task 3

Task 2- Display 5 records where launch sites begin with the string 'CCA'

Query:

```
""" SELECT * FROM SPACEXTABLEWHERE Launch_Site LIKE 'CCA%'LIMIT 5;
"""cur.execute(query)rows = cur.fetchall()df_cca =
pd.DataFrame(rows,columns=df.columns)df_cca
```

Task 3 - Display the total payload mass carried by boosters launched by NASA (CRS)

Query:

```
""" SELECT SUM("PAYLOAD_MASS__KG_") AStotal_payload_massFROM
"SPACEXTABLE"WHERE Customer = 'NASA (CRS)'; """cur.execute(query)total_payload =
cur.fetchone()total_payload
```

EDA with SQL – Task 4 and Task 5

Task 4- Display average payload mass carried by booster version F9 v1.1

Query:

```
 """ SELECT AVG(PAYLOAD_MASS__KG_) ASavg_payload_massFROM SPACEXTABLEWHERE
Booster_Version LIKE 'F9 v1.1%'; """ cur.execute(query)avg_payload =
cur.fetchone()avg_payload
```

Task 5 - List the date when the first succesful landing outcome in ground pad was acheived.

Query:

```
""" SELECT MIN(Date) ASfirst_success_ground_padFROM SPACEXTABLEWHERE
"Landing_Outcome" = 'Success (ground pad)';"""cur.execute(query)first_success =
cur.fetchone()first_success
```

EDA with SQL – Tasks 6 and 7

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

Query:

```
"""SELECT DISTINCT "Booster_Version"FROM SPACEXTABLEWHERE "Landing_Outcome"
LIKE 'Success (drone ship)'  AND "PAYLOAD_MASS__KG_" > 4000  AND
"PAYLOAD_MASS__KG_" < 6000;"""cur.execute(query)boosters = cur.fetchall()#import
pandas as pdpd.DataFrame(boosters, columns=["Booster_Version"])
```

Task 7 - List the total number of successful and failure mission outcomes

Query:

```
"""SELECT "Mission_Outcome", COUNT(*) AS countFROM SPACEXTABLEGROUP BY
"Mission_Outcome";"""cur.execute(query)mission_counts = cur.fetchall()# import pandas as
pdpd.DataFrame(mission_counts, columns=["Mission_Outcome", "Count"])
```

EDA with SQL – Tasks 8 and 9

Task 8 - List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function

Query:

```
"""SELECT "Booster_Version", "PAYLOAD_MASS__KG_"FROM SPACEXTABLEWHERE
"PAYLOAD_MASS__KG_" = (    SELECT MAX("PAYLOAD_MASS__KG_")    FROM
SPACEXTABLE);"""cur.execute(query)max_payload_boosters = cur.fetchall()# pandas as
pdpd.DataFrame(max_payload_boosters, columns=["Booster_Version", "PAYLOAD__MASS_KG_"])
```

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.

Query:

```
"""SELECT    substr(Date, 6, 2) AS Month,    "Booster_Version",    "Landing_Outcome",
"Launch_Site"FROM SPACEXTABLEWHERE "Landing_Outcome" LIKE 'Failure (drone ship)'  AND
substr(Date, 1, 4) = '2015';"""cur.execute(query)failures_2015 =
cur.fetchall()pd.DataFrame(failures_2015, columns=["Month", "Booster_Version", "Landing_Outcome",
"Launch_Site"])
```

EDA with SQL – Task 10

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.

Query:

```
"""SELECT "Landing_Outcome", COUNT(*) AS Outcome_CountFROM SPACEXTABLEWHERE
Date BETWEEN '2010-06-04' AND '2017-03-20'GROUP BY "Landing_Outcome"ORDER BY
Outcome_Count DESC;"""cur.execute(query)landing_counts =
cur.fetchall()pd.DataFrame(landing_counts, columns=["Landing_Outcome", "Count"])
```

EDA with SQL- Unique launch sites

Unique launch sites in Space mission

- Query:

```
""" SELECT DISTINCT Launch_siteFROM SPACEXTABLEORDER BY
Launch_site;"""cur.execute(query)unique_sites = cur.fetchall()unique_sites
```

- Results:

```
[('CCAFS LC-40',), ('CCAFS SLC-40',), ('KSC LC-39A',), ('VAFB SLC-4E',)]
```

Build an Interactive Map with Folium

Interactive Map with Folium

- Provide spatial insight into launch performance and operational distribution.

Visualization Components

- Site Markers

Identify precise launch locations and key site metadata.

- Scaled Circle Markers

Size reflects launch volume (operational maturity).

Color indicates success rate (performance reliability).

- Flight Path Lines (PolyLines)
- Highlight geographic dispersion and operational corridors.

Build a Dashboard with Plotly Dash

EDA Dashboard – Plots, Visualizations & Interactions Summary

Dashboard Objective

- Create an interactive analytical environment to explore launch performance patterns, detect anomalies, identify relationships between variables, and inform feature engineering prior to predictive modeling.

Visualizations Included & Rationale

- Bar Chart – Launch Success by Site
- Shows total launches and success rate per site. Added to benchmark site-level performance and identify operational maturity differences.
- Success vs Failure Distribution (Pie or Stacked Bar)
- Scatter Plot – Payload Mass vs Launch Outcome
- Visualizes relationship between payload weight and mission success. Added to evaluate correlation, nonlinear behavior, and potential threshold effects.
- Scatter Plot – Flight Number vs Launch Outcome
- Examines success trends over time. Added to evaluate learning curve effects and operational improvement.

Predictive Analysis (Classification)

A structured machine learning workflow was implemented to build, evaluate, and optimize classification models predicting mission success. Multiple algorithms were tested, compared, and refined to identify the best-performing model.

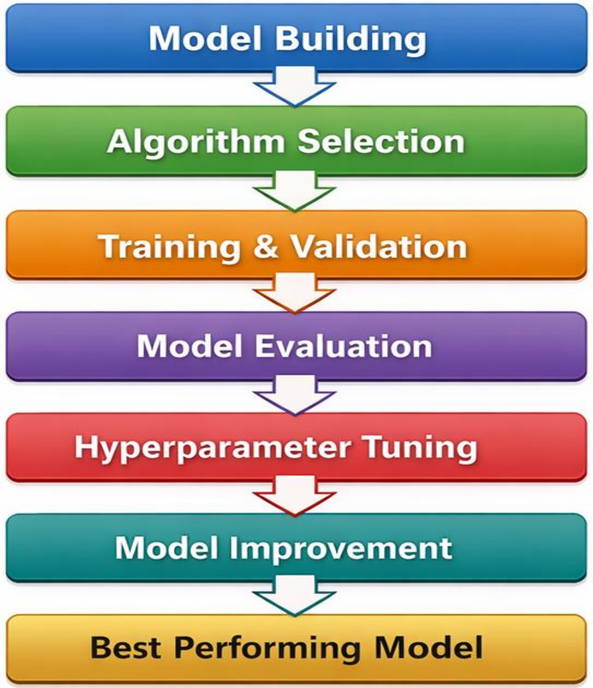
Model Development Process

- Data Preparation – Feature selection, encoding categorical variables, scaling numerical features.
- Train/Test Split – Partitioned dataset to prevent overfitting and ensure generalization.
- Baseline Model Training – Implemented Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM).
- Model Evaluation – Assessed accuracy, precision, recall, F1-score, and confusion matrix.
- Hyperparameter Tuning – Applied Grid Search to optimize model parameters.
- Model Comparison – Compared performance metrics across models.
- Model Selection – Selected best-performing classifier based on validation performance and generalization capability.

Evaluation & Improvement Summary

- • Logistic Regression – Provided strong interpretability and stable baseline performance.
- • Decision Tree – Captured nonlinear relationships but required pruning to prevent overfitting.
- • K-Nearest Neighbors – Performance sensitive to feature scaling and neighbor selection.
- • Support Vector Machine – Achieved highest validation performance after kernel tuning.
- After hyperparameter tuning and validation comparison, the best-performing model was selected based on highest balanced accuracy and stable confusion matrix results.

Predictive Analysis (Classification) - Flowchart



Model Building

Algorithm Selection

Training & Validation

Model Evaluation

Hyperparameter Tuning

Model Improvement

Best Performing Model

Results

1. **Data Analysis (Exploratory Data Analysis) Results**

- Exploratory analysis identified meaningful patterns in launch performance across launch sites, orbit types, and payload mass ranges. Visualizations including scatter plots, bar charts, and line charts revealed variability in success rates by operational location and mission configuration.
- Launch site comparisons demonstrated measurable differences in reliability, with certain sites achieving higher overall success ratios.
- Payload mass analysis suggested potential relationships between mission complexity and outcome probability, particularly within specific payload ranges.
- Yearly success trend analysis showed overall improvement in mission reliability over time, indicating operational learning and system refinement.
- 2. **Interactive Analytics Dashboard Results**
- The Plotly Dash dashboard enabled real-time exploration of launch data using interactive filters including launch site selection and payload range sliders.
- Dynamic pie charts displayed launch success distributions both across all sites and within individual sites, supporting rapid performance comparison.
- Interactive scatter plots visualized payload versus launch outcome relationships, allowing users to isolate payload intervals and observe performance patterns.
- The integrated visual analytics interface supports data-driven decision-making by combining filtering, visualization, and outcome comparison in a single environment.

Results

Predictive Analysis Results

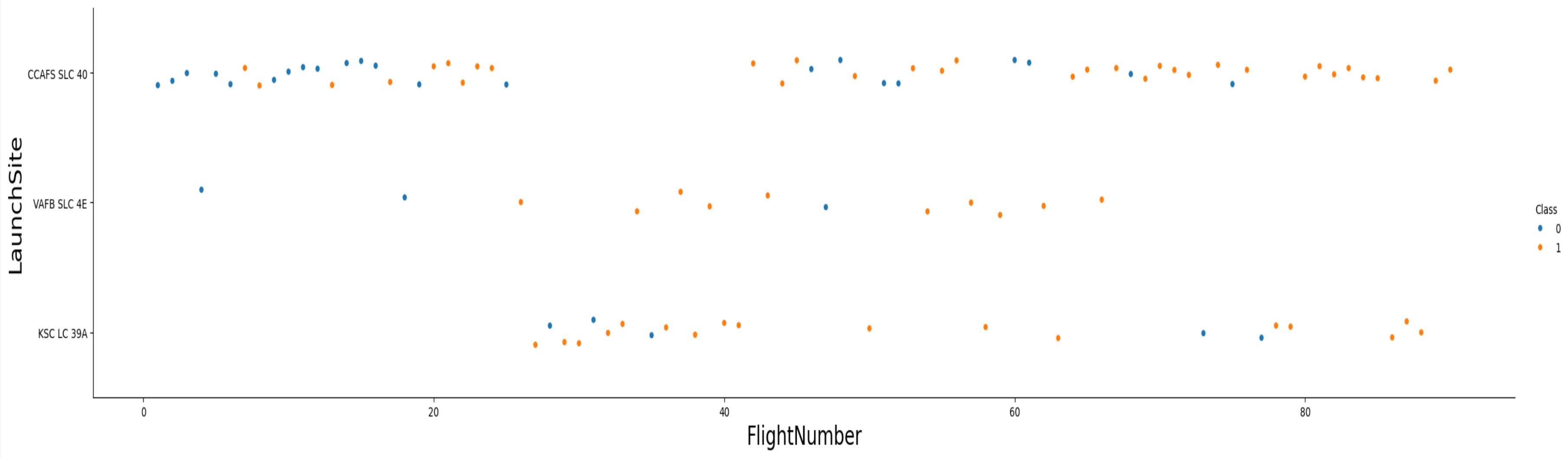
- Exploratory analysis identified meaningful patterns in launch performance across launch sites, orbit types, and payload mass ranges. Visualizations including scatter plots, bar charts, and line charts revealed variability in success rates by operational location and mission configuration.
- Launch site comparisons demonstrated measurable differences in reliability, with certain sites achieving higher overall success ratios.
- Payload mass analysis suggested potential relationships between mission complexity and outcome probability, particularly within specific payload ranges.
- Yearly success trend analysis showed overall improvement in mission reliability over time, indicating operational learning and system refinement.

Section 2

Insights drawn from EDA

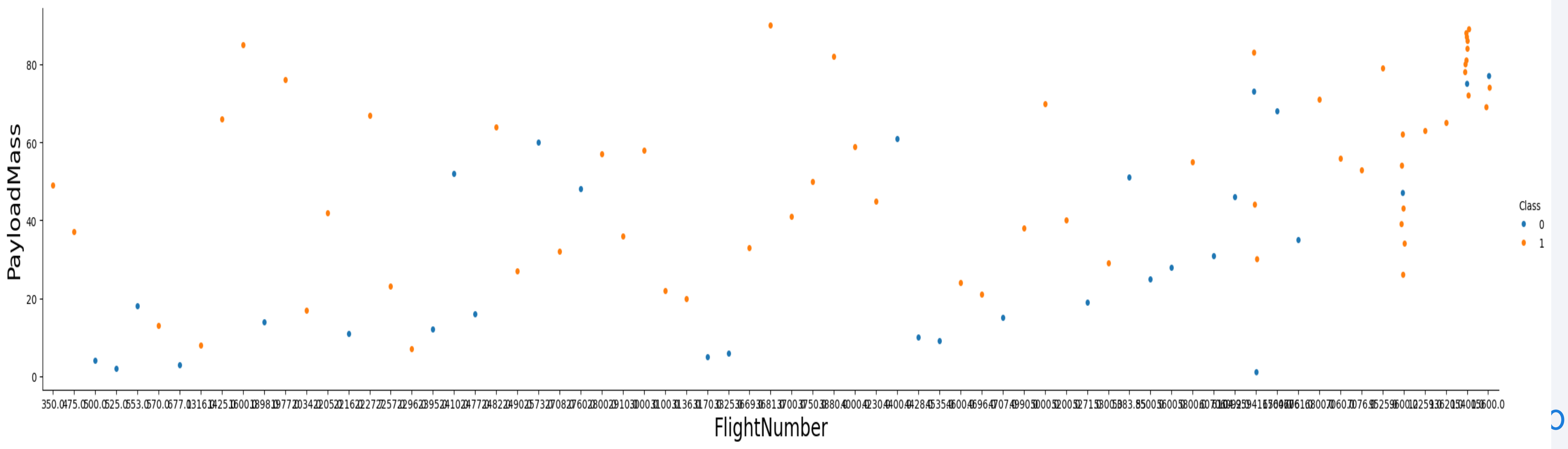
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- sns.catplot(y="LaunchSite", x = "FlightNumber", hue = "Class" , data=df, aspect = 5)plt.xlabel("FlightNumber", fontsize=20)plt.ylabel("LaunchSite", fontsize=20) plt.show()
- The scatter plot suggests a measurable learning curve, reinforcing the inclusion of launch experience as a predictive variable.



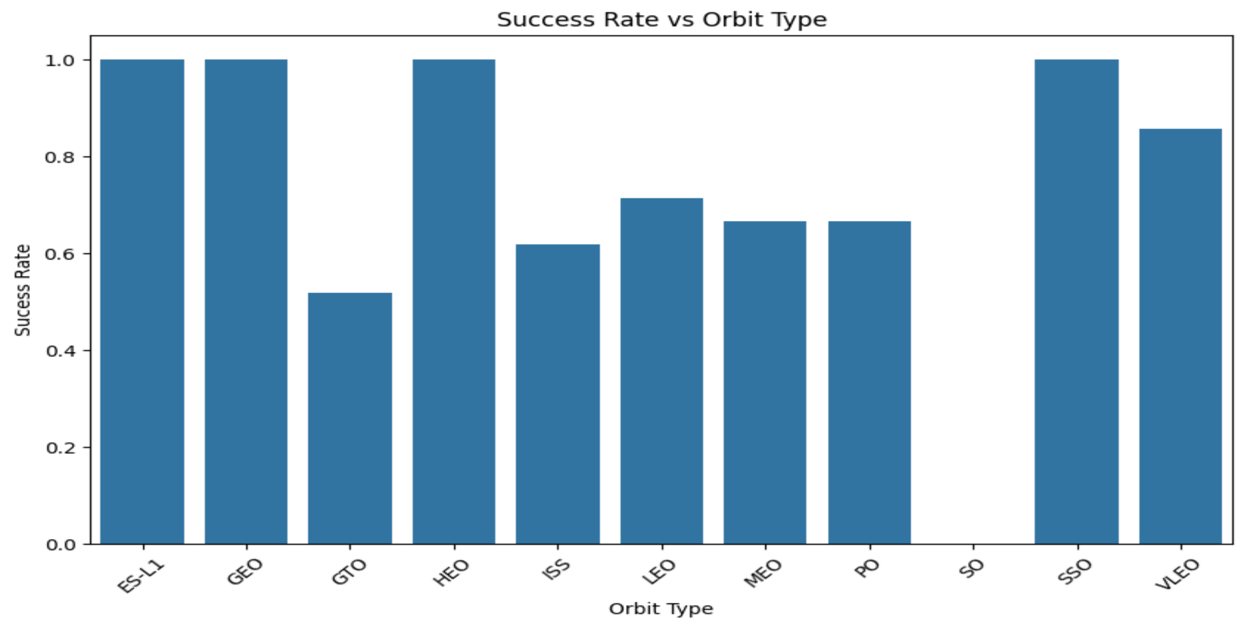
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- sns.catplot(y="FlightNumber", x = "PayloadMass", hue = "Class" , data=df, aspect = 5)plt.xlabel("FlightNumber", fontsize=20)plt.ylabel("PayloadMass", fontsize=20) plt.show()
- The scatter plot compares payload distribution across launch sites, identify site-specific payload specialization patterns and detect payload thresholds that may influence mission outcome.



Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- success_rate = df.groupby("Orbit")["Class"].mean().reset_index()plt.figure(figsize=(10,6))sns.barplot(data=success_rate, x="Orbit", y="Class")plt.title("Success Rate vs Orbit Type")plt.xlabel("Orbit Type")plt.ylabel("Sucess Rate")plt.xticks(rotation=45) plt.show()
- The bar chart shows the bar height which represents average success rate for each orbit.

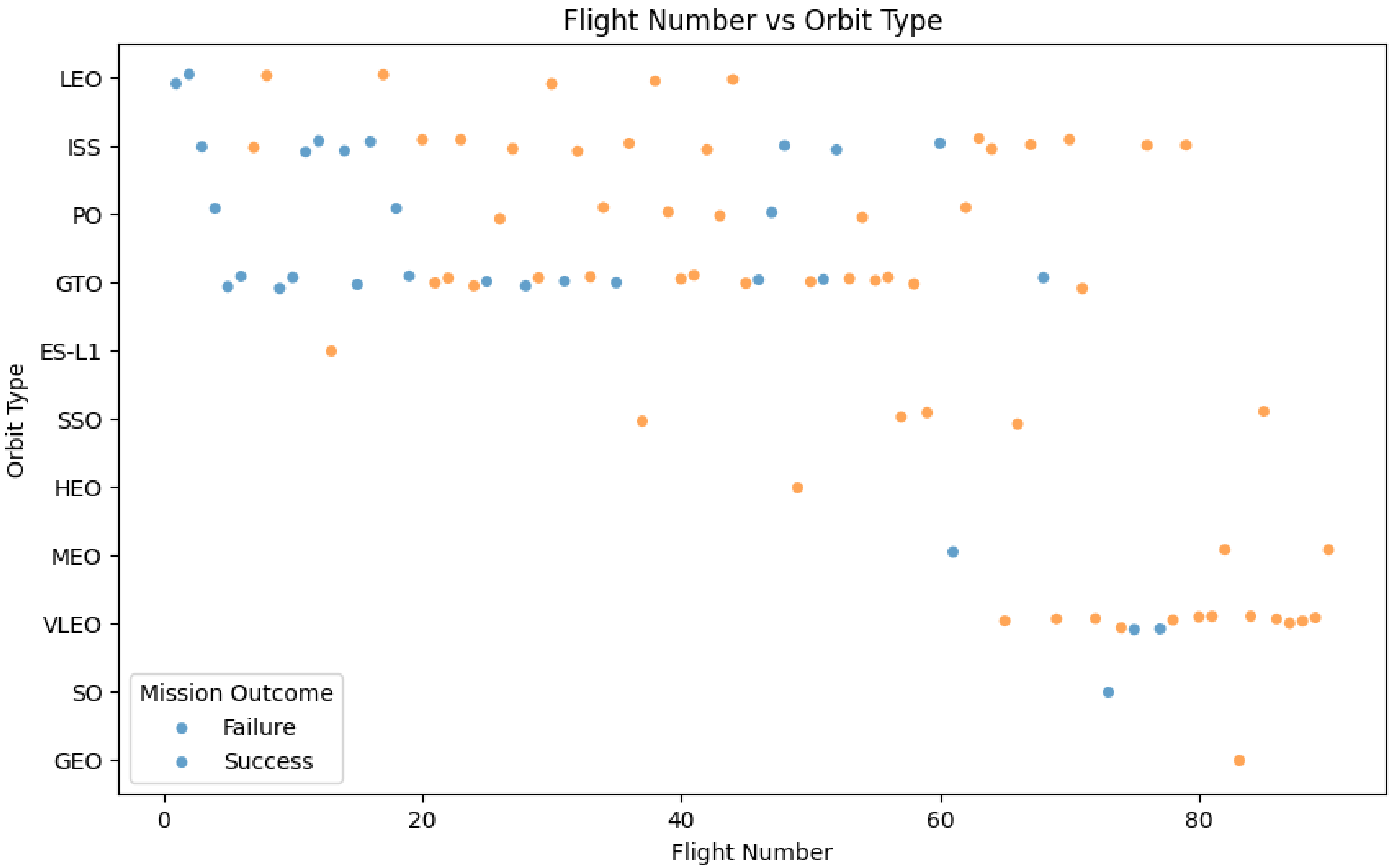


Flight Number vs. Orbit Type

Show a scatter point of Flight number vs. Orbit type

```
plt.figure(figsize=(10, 6))

# Orbit is categorical, so use a strip plot (scatter-style) rather than a numeric scatter
sns.stripplot(
    data=df,
    x="FlightNumber",
    y="Orbit",
    hue="Class",     # 1 = Success, 0 = Failure
    jitter=True,
    alpha=0.7
)

plt.title("Flight Number vs Orbit Type")
plt.xlabel("Flight Number")
plt.ylabel("Orbit Type")
plt.legend(title="Mission Outcome", labels=["Failure", "Success"])
plt.show()
```

Flight Number vs. Orbit Type



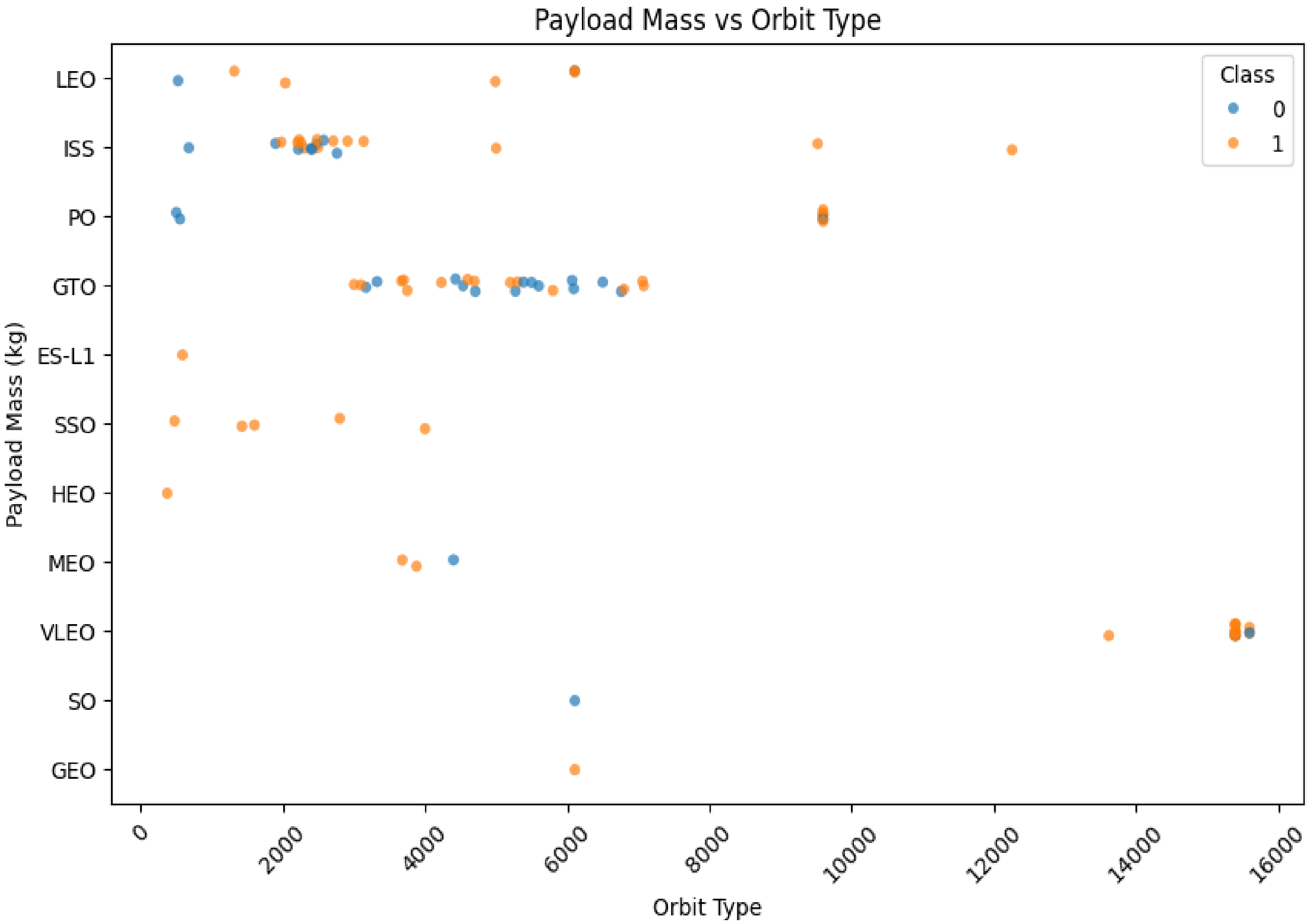
- Orbit Type is categorical; a strip plot preserves category labels on the y-axis.
- Jitter spreads points so overlapping launches remain visible.
- Hue by Class overlays success/failure patterns without requiring separate charts.
- Validate whether Orbit Type should be treated as an important categorical feature in predictive modeling.
- Support feature engineering decisions.
- Identify outliers or rare orbit categories that may require grouping or careful train/test splitting.

Payload vs. Orbit Type

Show a scatter point of payload vs. Orbit type

```
plt.figure(figsize=(10, 6))

# Orbit is categorical; a box plot summarizes payload distribution per orbit
sns.boxplot(
    data=df,
    x="Orbit",
    y="PAYLOAD_MASS__KG_"
)

plt.title("Payload Mass vs Orbit Type")
plt.xlabel("Orbit Type")
plt.ylabel("Payload Mass (kg)")
plt.xticks(rotation=45)

plt.show()
```

Payload vs. Orbit Type



- Assess whether Orbit Type is associated with payload mass differences (mission profile effects).
- Support feature engineering decisions (e.g., payload bins; orbit one-hot encoding; orbit–payload interaction).
- Identify outliers that may require cleaning or robust scaling before modeling.

Launch Success Yearly Trend

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

df['Year'] = df['Date'].dt.year
```

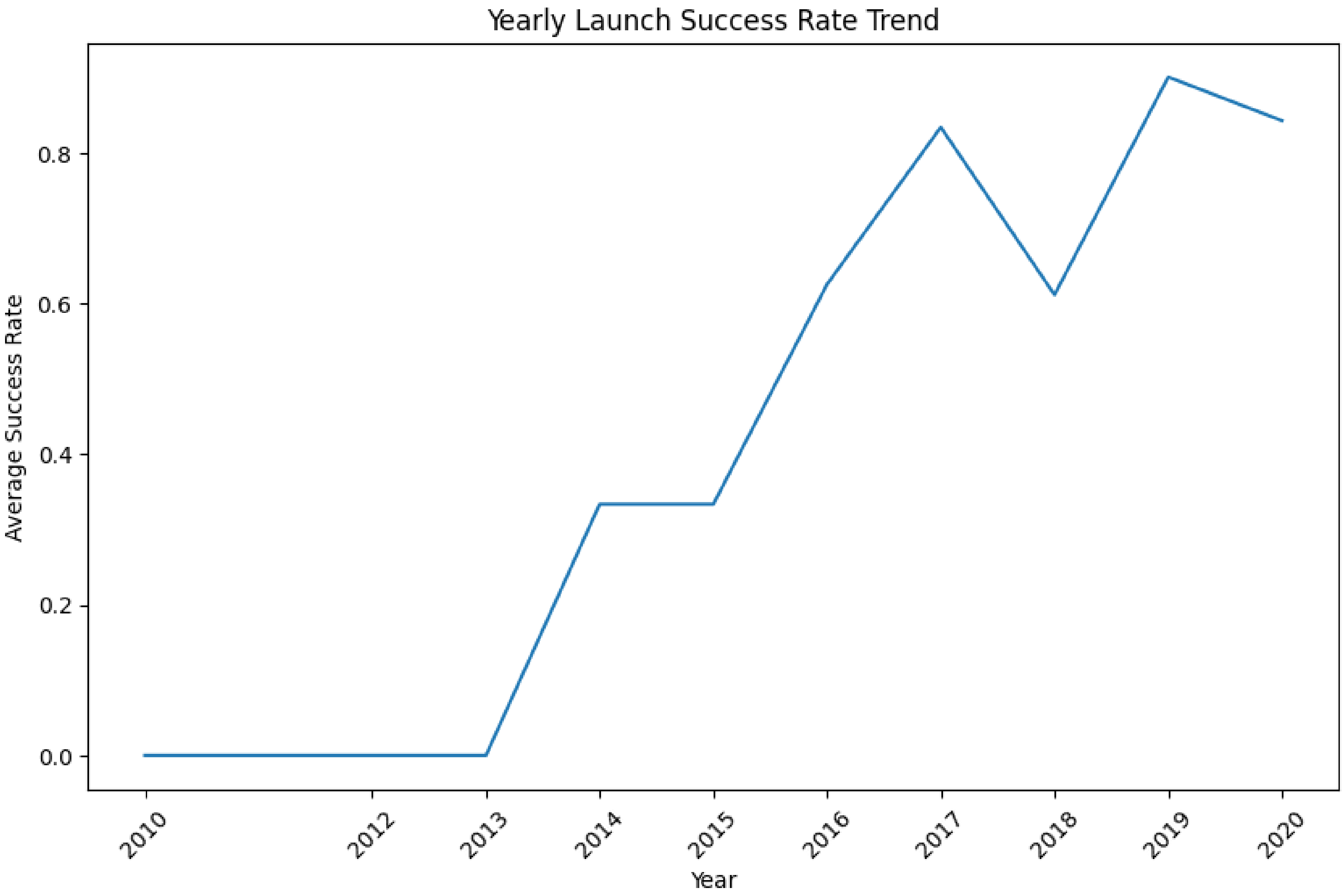
Step 2 – Compute Yearly Average Success Rate

```
yearly_success = (
    df.groupby('Year')['Class']
    .mean()
    .reset_index()
)

yearly_success.head()
```

Step 3 – Plot Line Chart

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))

plt.plot(
    yearly_success['Year'],
    yearly_success['Class']
)

plt.title("Yearly Launch Success Rate Trend")
plt.xlabel("Year")
plt.ylabel("Average Success Rate")
plt.xticks(yearly_success['Year'], rotation=45)

plt.show()
```

Launch Success Yearly Trend



- Shows reliability progression across years.
- Identifies learning curve effects and operational maturity.
- Detects volatility or stability phases in performance.
- Supports inclusion of temporal features in predictive modeling.

All Launch Site Names

- Find the names of the unique launch sites
- Query =

```
""" SELECT DISTINCT Launch_site FROM SPACEXTABLEORDER BY
Launch_site;"""cur.execute(query)unique_sites = cur.fetchall()unique_sites
```

- **Launch_Site**

```
[('CCAFS LC-40',), ('CCAFS SLC-40',), ('KSC LC-39A',), ('VAFB SLC-4E',)]
```

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Query:

```
 """ SELECT * FROM SPACEXTABLEWHERE Launch_Site LIKE 'CCA%'LIMIT 5;
"""cur.execute(query)rows = cur.fetchall()df_cca =
pd.DataFrame(rows,columns=df.columns)df_cca
```

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	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Query:

```
 """ SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_massFROM
"SPACEXTABLE"WHERE Customer = 'NASA (CRS)'; """cur.execute(query)total_payload =
cur.fetchone()total_payload
```

Result:

(45596)

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

Query:

```
 """ SELECT AVG(PAYLOAD_MASS__KG_) ASavg_payload_mass

FROM SPACEXTABLEWHERE Booster_Version LIKE 'F9 v1.1%'; """
cur.execute(query)avg_payload = cur.fetchone()avg_payload
```

Result:

(2534.6666666666665)

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad

Query:

```
""" SELECT MIN(Date) ASfirst_success_ground_pad

FROM SPACEXTABLEWHERE "Landing_Outcome" = 'Success (ground
pad)';"""cur.execute(query)first_success = cur.fetchone()first_success
```

Result:

('2015-12-22',)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- **Booster Version**
- F9 FT B1022
- F9 FT B1026
- F9 FT B1021.2
- F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

Booster_Version	PAYLOAD_MASS_KG
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600

2015 Launch Records

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

Month	Booster_Version	Landing_Outcome	Launch_Site
01	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

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

- 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

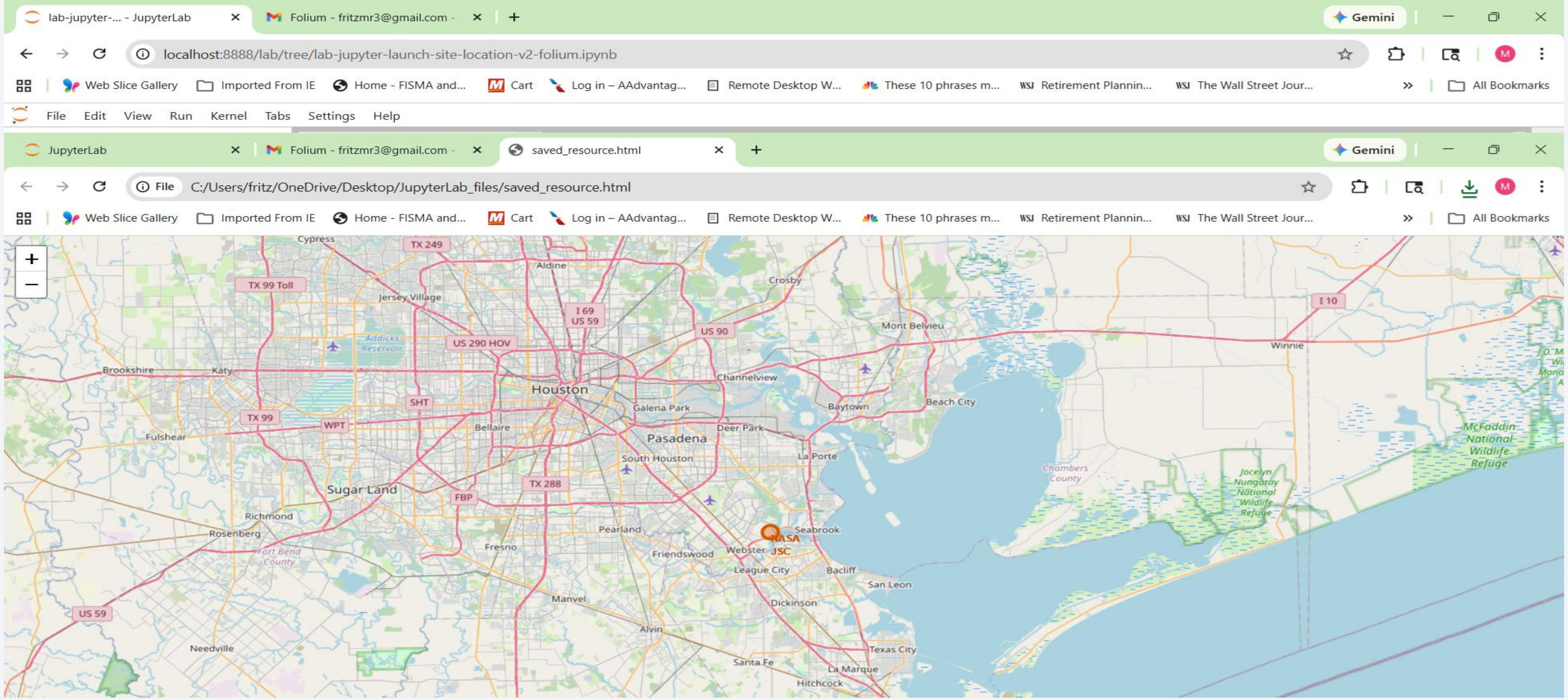
Landing_Outcome1	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

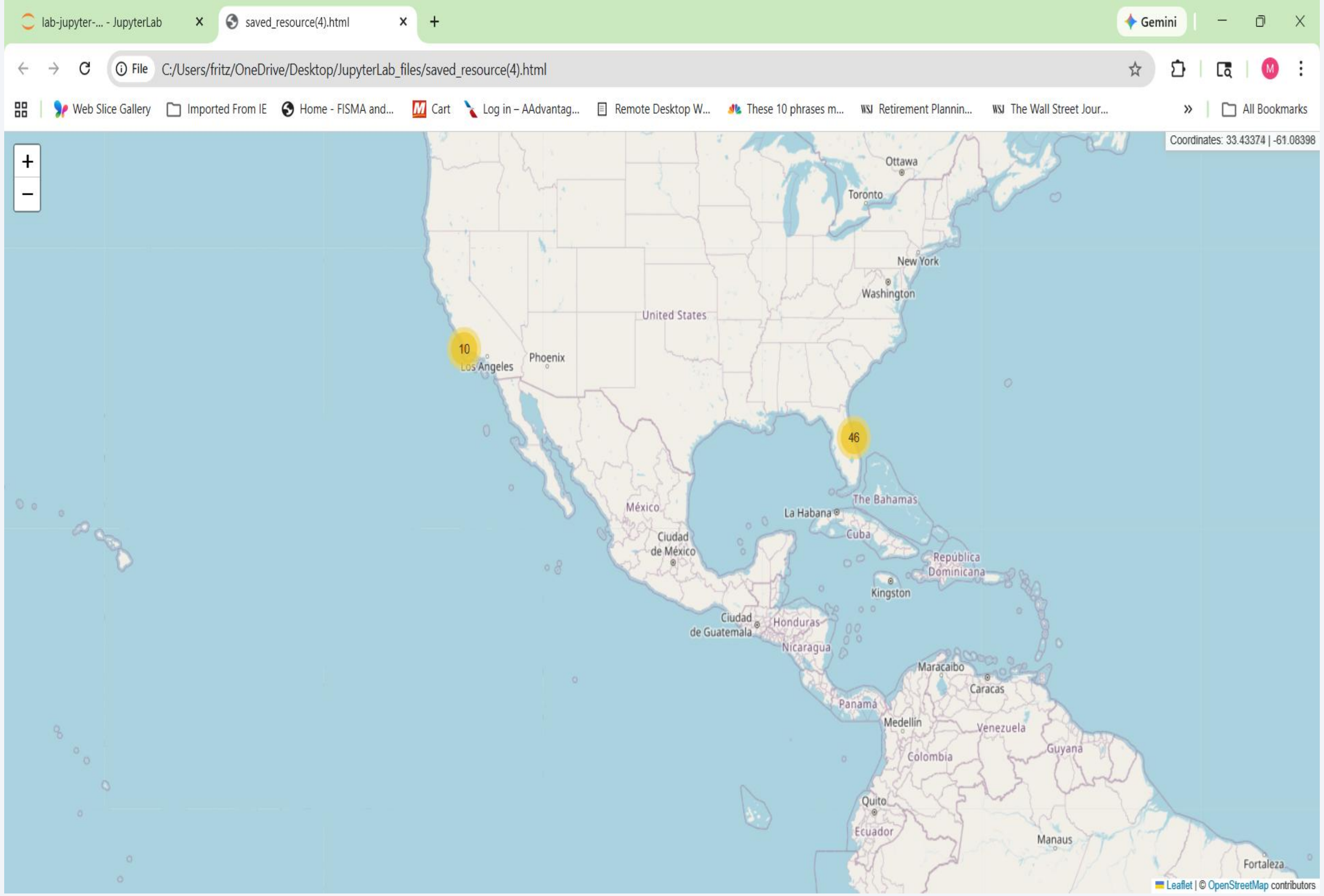
Launch Sites Proximities Analysis

All Launch Site on the Map

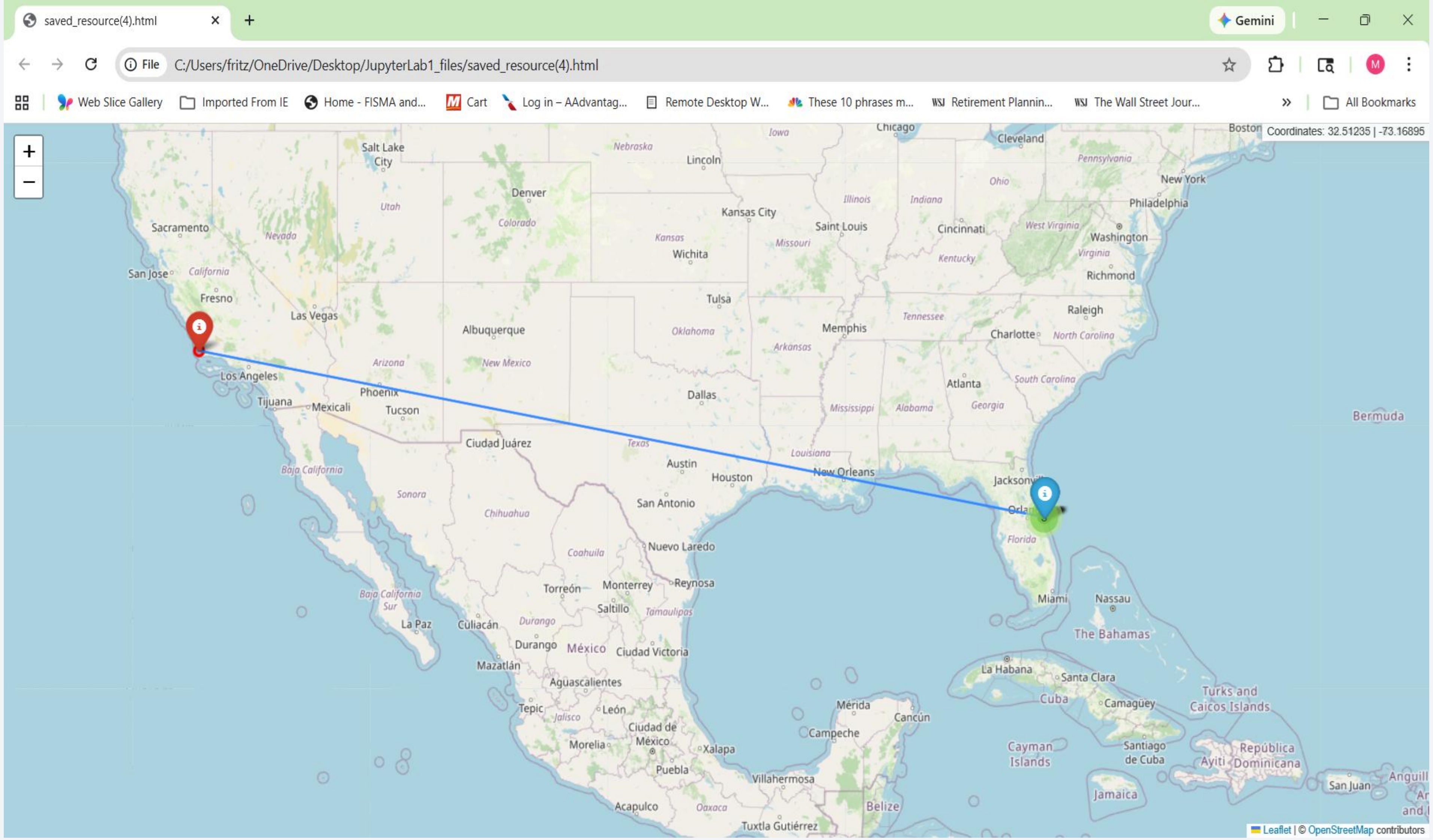
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

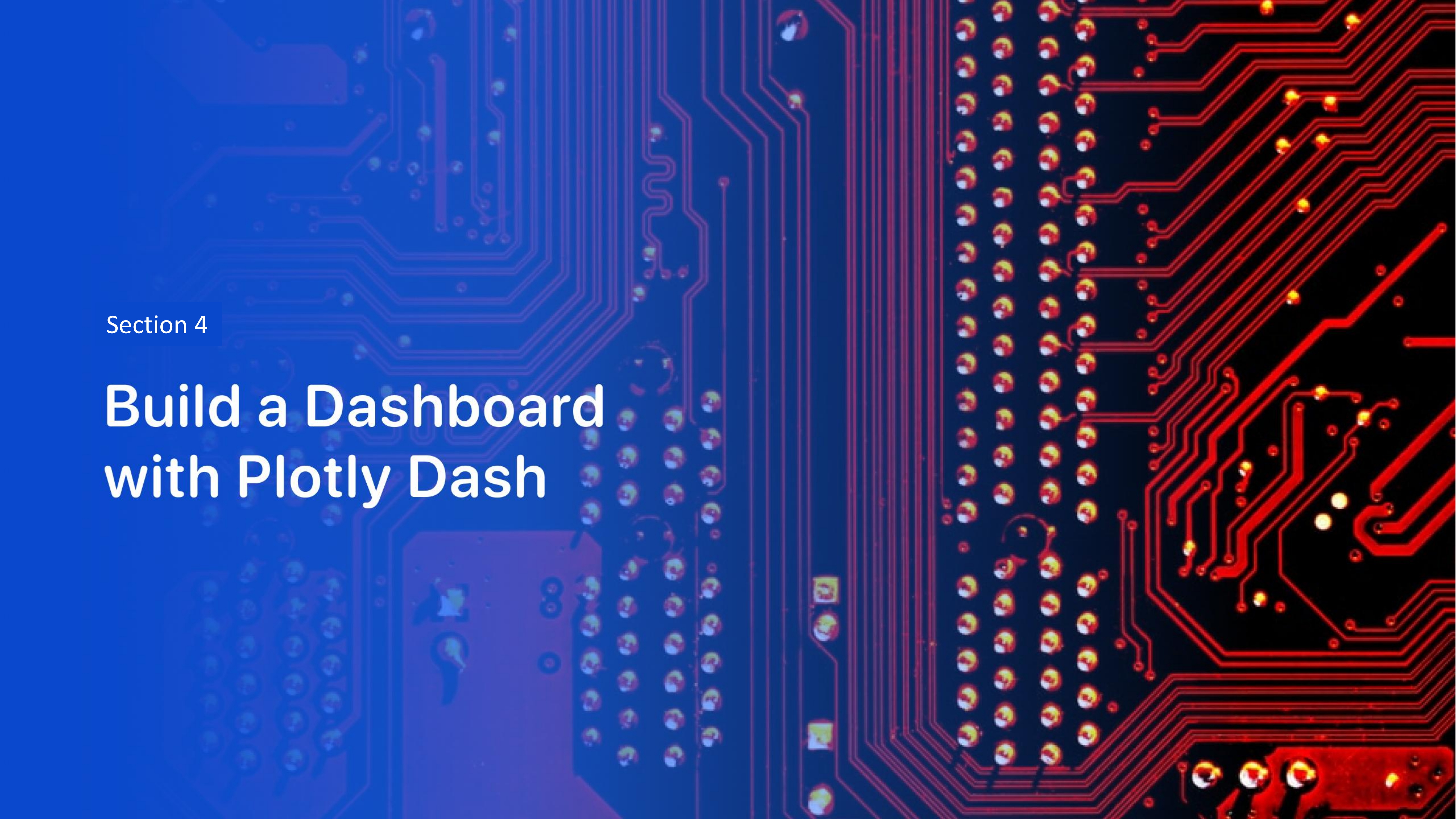


Mark the success/failed launches for each site on the map



Calculate the distances between a launch site - proximities

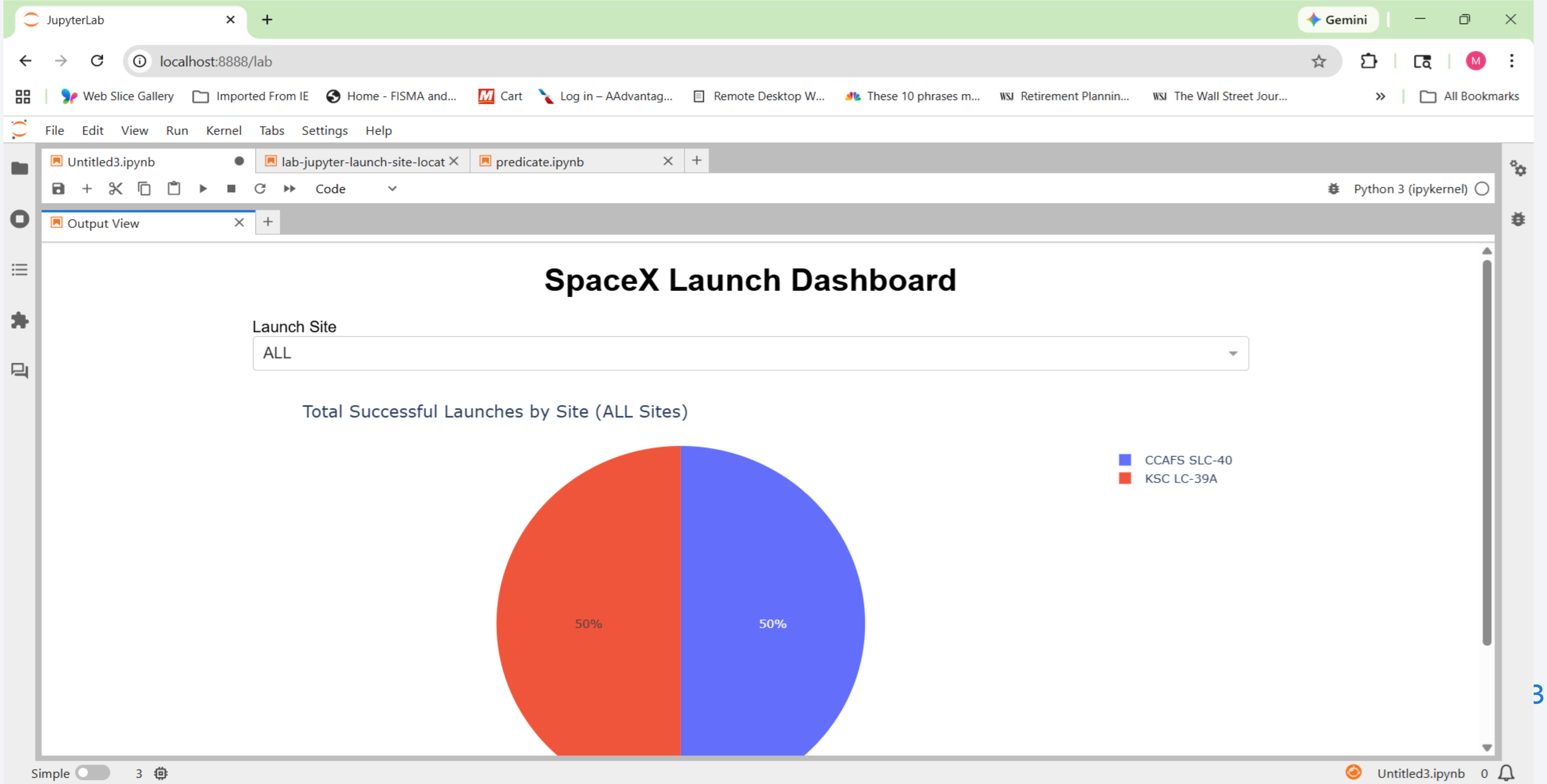




Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites, in a pie chart

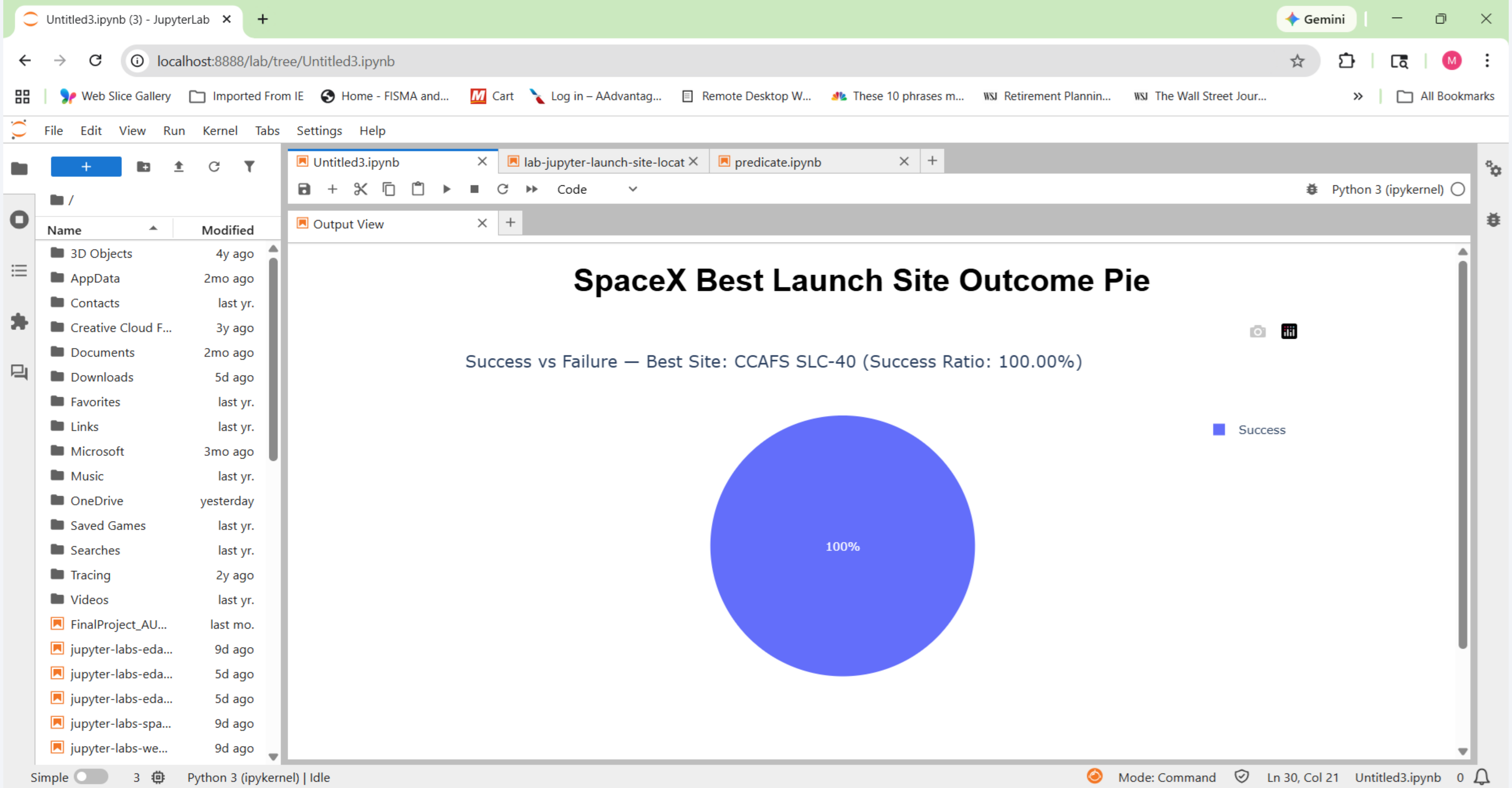


SpaceX Launch Dashboard

Launch Site: ALL

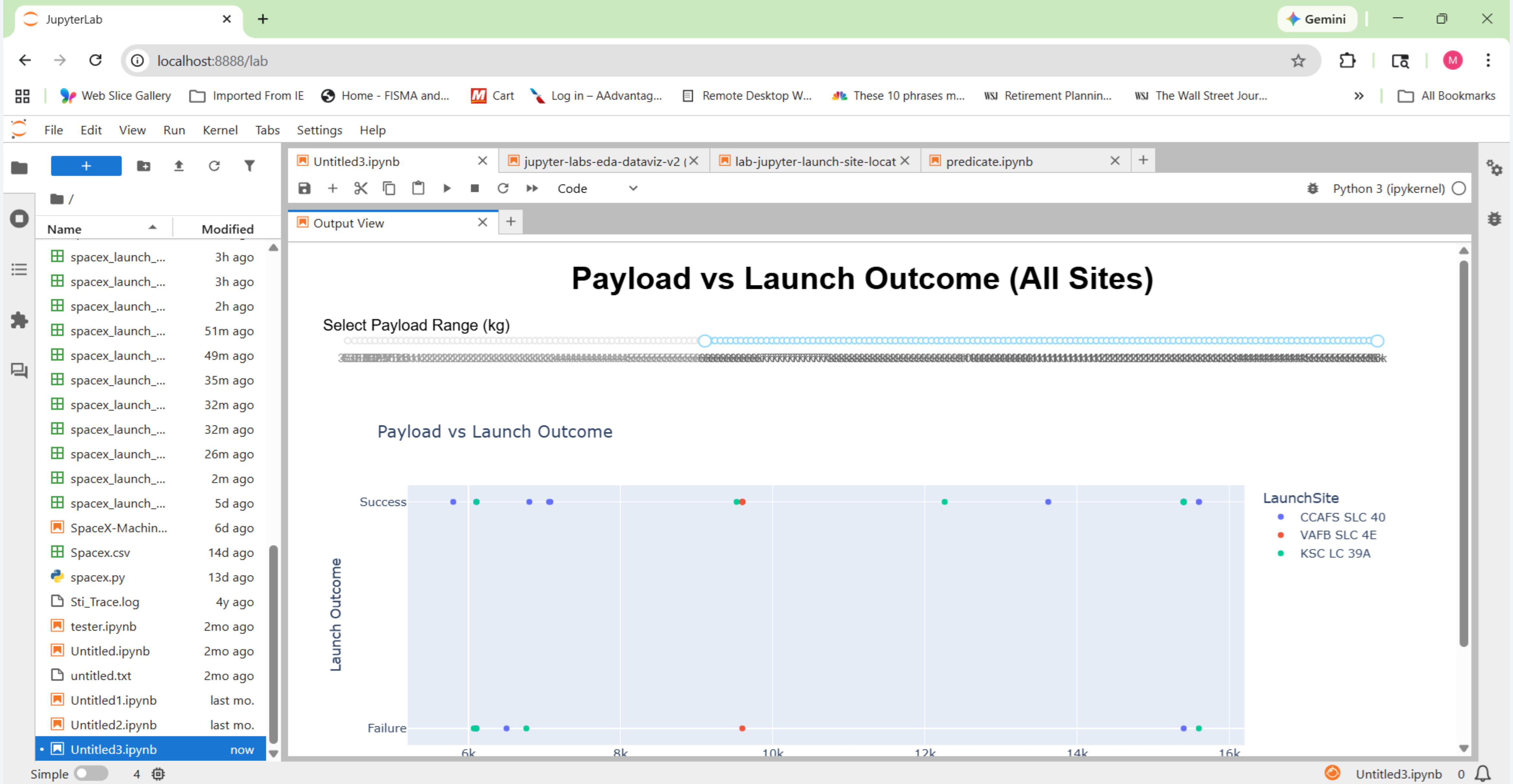
Total Successful Launches by Site (ALL Sites)

Pie chart for the launch site with highest launch success ratio



Untitled3.ipynb (3) - JupyterLab
localhost:8888/lab/tree/Untitled3.ipynb
File Edit View Run Kernel Tabs Settings Help
Untitled3.ipynb
lab-jupyter-launch-site-locat
predicate.ipynb
Output View
Python 3 (ipykernel)
SpaceX Best Launch Site Outcome Pie
Success vs Failure — Best Site: CCAFS SLC-40 (Success Ratio: 100.00%)
100%
Success
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 30, Col 21
Untitled3.ipynb

Payload vs. Launch Outcome scatter plot for all sites



Name	Modified
spacex_launch_...	3h ago
spacex_launch_...	3h ago
spacex_launch_...	2h ago
spacex_launch_...	51m ago
spacex_launch_...	49m ago
spacex_launch_...	35m ago
spacex_launch_...	32m ago
spacex_launch_...	32m ago
spacex_launch_...	26m ago
spacex_launch_...	2m ago
spacex_launch_...	5d ago
SpaceX-Machin...	6d ago
Spacex.csv	14d ago
spacex.py	13d ago
Sti_Trace.log	4y ago
tester.ipynb	2mo ago
Untitled.ipynb	2mo ago
untitled.txt	2mo ago
Untitled1.ipynb	last mo.
Untitled2.ipynb	last mo.
Untitled3.ipynb	now

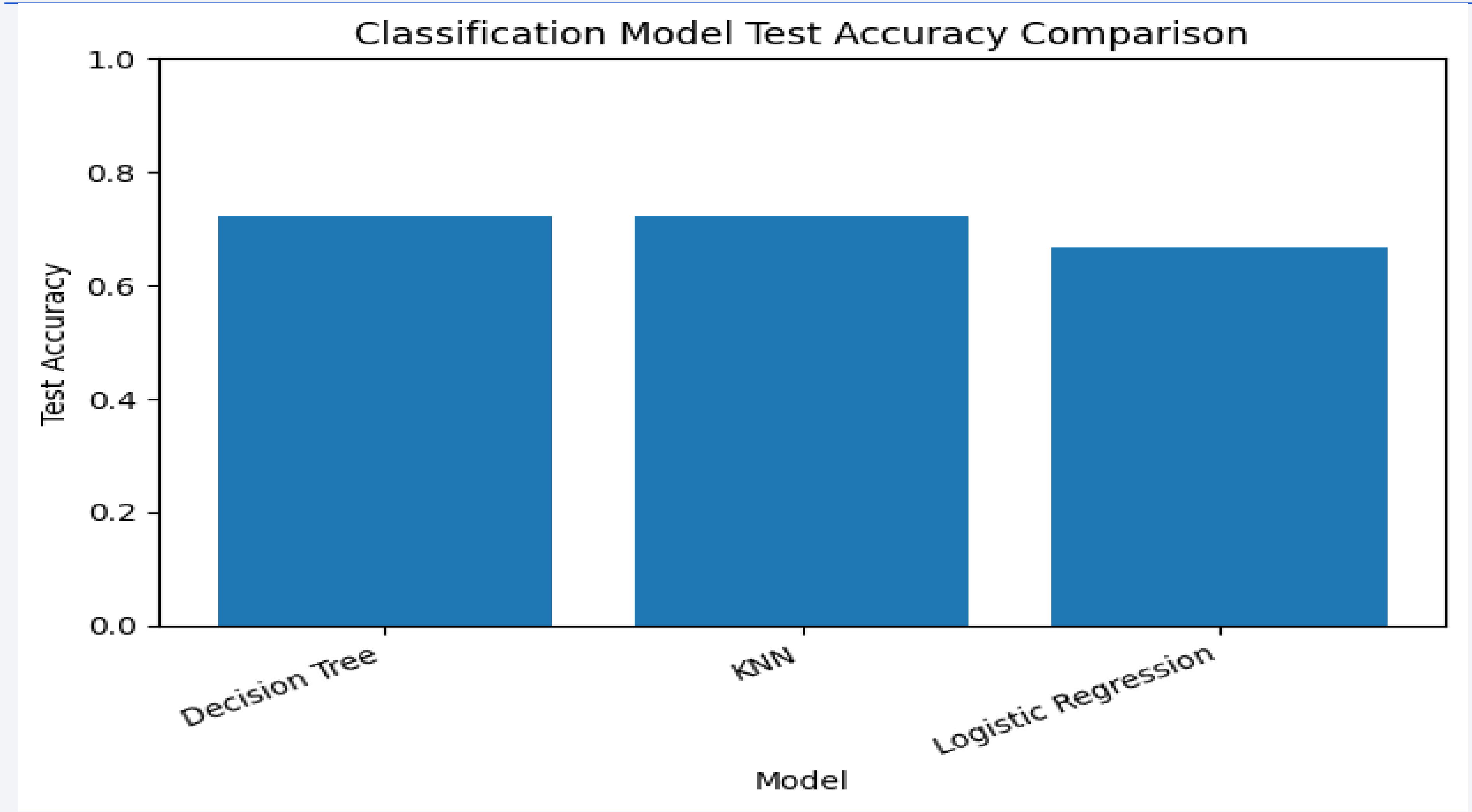
Payload vs Launch Outcome (All Sites)

Select Payload Range (kg)

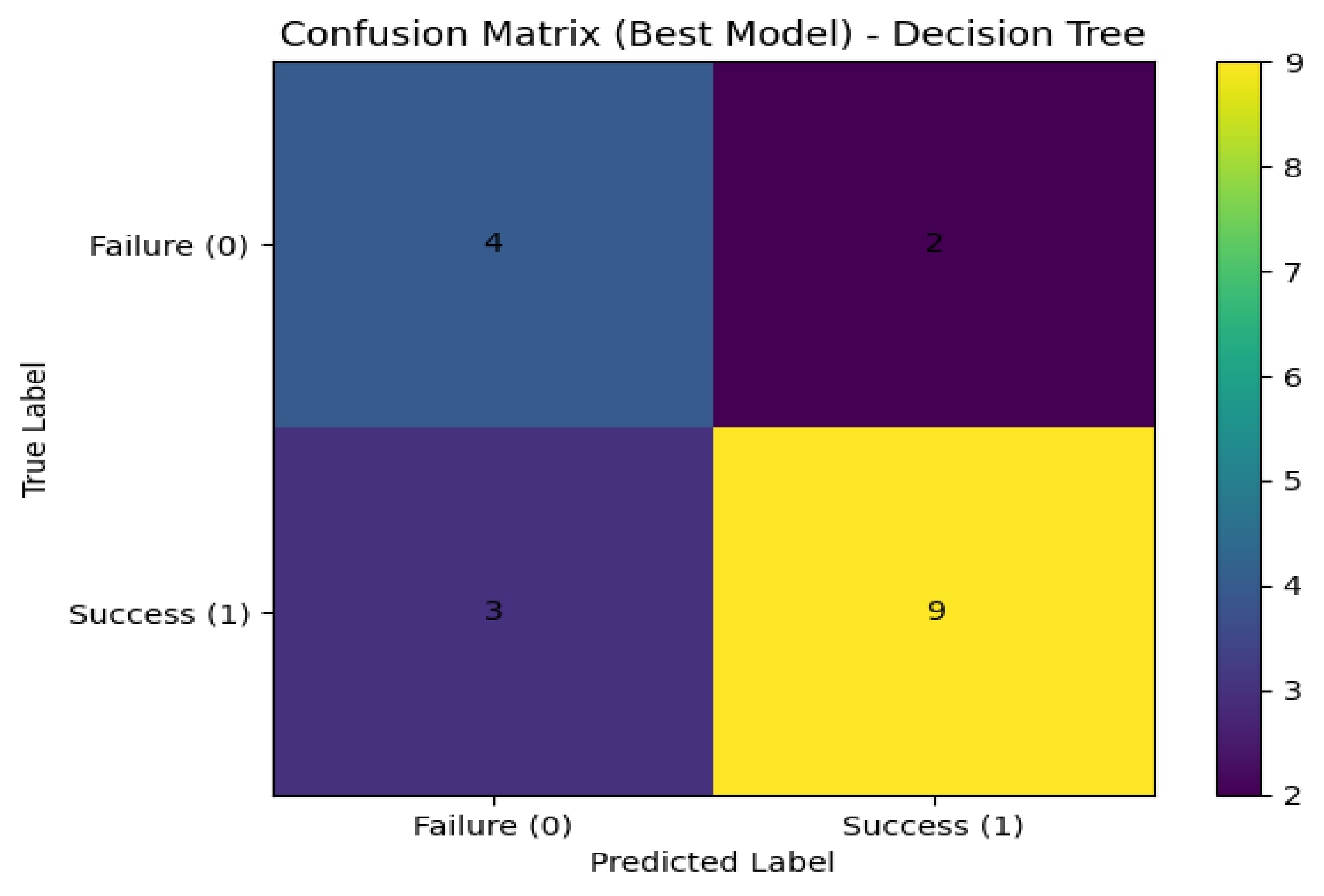
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

1. Launch Site Performance Varies Significantly

Analysis revealed measurable differences in launch success ratios across launch sites. Certain sites consistently demonstrated higher reliability, suggesting operational, environmental, or infrastructure factors influence mission outcomes.

2. Payload Mass Influences Launch Outcomes

Exploratory analysis identified observable patterns between payload mass and mission success. Different payload ranges exhibited varying performance distributions, indicating mission complexity and vehicle load contribute to risk variability.

3. Predictive Models Demonstrate Strong Performance

Multiple classification algorithms (Logistic Regression, Decision Tree, KNN) were trained and optimized using GridSearch. The best-performing model achieved strong test accuracy and demonstrated reliable predictive capability using payload, orbit, and launch site features.

4. Interactive Dashboard Enables Data-Driven Insights

The integrated Plotly Dash application combines filtering, visualization, and predictive modeling. This provides a practical decision-support tool for evaluating launch configurations, assessing mission risk, and identifying high-performing operational sites.

- https://github.com/gopikrishnan-t-s/Applied-Data-Science-Capstone

Innovative Insights

Operational Learning Curve Impact

- Analysis of multi-year launch data indicates a clear upward trend in mission success rates, reflecting measurable operational maturation. The results suggest that sustained launch cadence, process refinement, and accumulated engineering experience significantly enhance reliability over time.
- This finding reinforces the strategic value of continuous execution, data-driven performance monitoring, and iterative improvement as core drivers of long-term mission success.
- In practical terms, reliability growth appears to be organizationally cultivated rather than purely hardware-dependent, demonstrating the competitive advantage of disciplined operational learning.

Thank you!