IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Saurav Gupta
17-Mar-2025

Outline

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

Executive Summary

- Summary of methodologies
  - For this analysis, we used the SpaceX launch data to investigate the factors influencing mission outcomes. We primarily focused on finding correlations, particularly between payload mass and launch success. To do this, we used Python with Pandas for data handling, Plotly for creating visual representations, and Folium for geospatial analysis. These tools enabled us to explore the data thoroughly and illustrate our key findings
- Summary of all results
  - We found that payload mass and launch success are definitely connected—heavier payloads tended to have more issues. Also, the 'CCAFS LC-40' launch site stood out with a really good success record. On the other hand, things like how close a launch site was to the coast or a city didn't seem to make much of a difference. These findings point to payload weight as a key consideration for future launches, and suggest that it would be worthwhile to further explore the success of 'CCAFS LC-40' site

Introduction

- Project background and context
- The commercial space industry has witnessed significant growth in recent years, with companies like SpaceX playing a pivotal role in advancing space exploration and satellite deployment. Understanding the factors that contribute to successful space missions is crucial for optimizing launch operations and ensuring mission reliability. This project aims to analyze SpaceX launch data to gain insights into the key determinants of launch outcomes. The analysis will focus on the relationships between payload mass, launch site characteristics, and mission success, providing valuable context for future launch planning
- Problems you want to find answers
- This analysis seeks to address the following key questions:
    - What is the correlation between payload mass and launch success rates?
    - How do different launch sites compare in terms of mission success
    - Does the proximity of launch sites to geographical features significantly influence launch outcomes
- By answering above questions, we aim to provide actionable insights that can contribute to improved launch planning and mission success for SpaceX

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using
  - SpaceX Rest API. The endpoint used was the data for past launches using the url api.spacexdata.com/v4/launches/past
  - Using (Wikipedia) Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records
- Perform data wrangling
  - The steps conduced to processes data are listed below,
  - Data Cleaning – Handling missing values, dplicate records,
  - Feature Engineering – Encoding categorical value Normalising numerical features
  - Data organization – Move data into Tabular format
  - Data splitting - Data splitting into test and train to test the various Models fit

Data Collection

- Describe how data sets were collected.
- The Data was collected using below 2 methods
  - SpaceX Rest API to retrieve past launch records
  - Web scraping Falcon 9 launch records
- You need to present your data collection process use key phrases and flowcharts
- The Primary data collection was from the SpaceX rest API and the secondary collection was from web scraping
- The data collection method was automated as they were read from the API or the Web directly. There were no manual data entry with the source providers.
- The Data cleaning and pre-processing were done after the data was loaded into a readable structure like table and columns

[Start] --> [Identify Data Sources] --> [Select Collection Methods] --> [Develop Data Collection Tools/Protocols (Python script and libraries)] --> [Data Collection Begins] --> [Data Validation/Cleaning (Pandas Data frame)] --> [Data Storage (Cloud Storage)] --> [Data Analysis Preparation to check various Models] --> [End]

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
  - *API Endpoint Selection*
  - *HTTP GET Requests*
  - *Pagination Handling*
  - *JSON Response Parsing*
  - *Data Extraction*
  - *Data Storage*
  - *Automated Data Retrieval*

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

A[Start: Define Data Collection Goals] -->

B{Identify Key Data Sources}; B --> C[Select Appropriate Collection Methods]

C --> D[Develop Data Collection Tools/Protocols]

D --> E{Pilot Test (If Needed)?}

E -- Yes --> F[Refine Tools/Protocols]

F --> G[Begin Data Collection]

E -- No --> G

G --> H[Data Validation & Cleaning]

H --> I[Data Storage & Organization]

I --> J[Data Analysis/Presentation]

J --> K[End: Achieve Data Collection Goals]

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
  - *Target Website Identification (Wikipedia)*
  - *HTTP Requests*
  - *HTML Structure Analysis*
  - *Library Selection - BeautifulSoup*
  - *Element Selection (CSS Selectors/XPath) - Iterating through Htl Table Head and Rows*
  - *Data Extraction and Cleaning*
  - *Create a data Frame*
  - *Data Storage into csv*
  - *Error Handling*

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/jupyter-labs-webscraping.ipynb

Place your flowchart of web scraping here

A[Start: Identify Target Website] --> B[Analyze Website HTML Structure]

B --> C[Select Scraping Library (Beautiful Soup, Scrapy, Selenium)]

C --> D[Send HTTP Request to Website]

D --> E[Parse HTML Content]

E --> F[Locate Target Data Elements (CSS/XPath)]

F --> G[Extract Data]

Data Wrangling

- Describe how data were processed
- *Data Discovery and Profiling*
- *Data Cleaning – Identify Null and empty values and remove/replace*
- *Data filtering*
    - *Remove rows with multiple cores*
    - *Convert date utc to datetime*
    - *Filter on date 2020/11/13 and before*
- *Data Feature Engineering*
    - *Cores and Payloads are lists which are replaced with singe column*
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

- You need to present your data wrangling process using key phrases and flowcharts
- A[Start: Subset DataFrame] --> B[Select Columns: 'rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc’]
- B --> C[Remove Rows with Multiple Cores]
- C --> D[Remove Rows with Multiple Payloads]
- D --> E[Extract Single Core Value]
- E --> F[Extract Single Payload Value]
- F --> G[Convert 'date_utc' to Datetime and Extract Date]
- G --> H[Filter Dates: <= 2020-11-13]
- H→ I [Calculate Mean of Payload and Replace NaN with Mean]
- I→J [End: Processed DataFrame]

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Scatter Plot - scatter point chart with x axis to be Payload Mass and y axis to be the launch site, and hue to be the class value (Higher Payload had Less success)
- Scatter Plot - 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 (As the FlightNumber increase, the First stage is more successful)
- Bar Plot – Orbit Vs Their Success rate (To se which Orbits has more success)
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/edadataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- 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

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Markers – They are used to pinpoint specific locations on the map. They are typically indicated by a symbol (often a pin or dot)
- Circles – They highlight a specific area around a point. Circle Markers are often used for visual emphasis or to indicate a radius
- Lines - They connect two or more points on the map. They are used to show paths, routes, or connections
- Explain why you added those objects
- Markers - This allows viewers to immediately visualize the geographical distribution of launch activities
- Lines - Visually represents and allows the viewer to quickly understand the distance from each launch site to the nearest coastline
- Circles – It helps to illustrate the potential impact area or range of influence around each launch site
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Pie Chart for Success
- Scatter Chart for Success
- Explain why you added those plots and interactions
- Pie chart was to see the breakdown of success/failure by each launch_site
- Identify correlation between Payload and Success factor
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

1. We created Numpy array from the Class variable (Success/Failure)
2. We standardized the data followed by Splitting into Test and Training set
3. Created various Models like Logistic regression, SVM, Decision Tree and K-Nearest Neighbors and checked their Accuracy
4. We evaluated model performance using accuracy, precision, and recall, then improved it through hyperparameter tuning and feature engineering
5. To enhance our model, we assessed its metrics against a confusion matrix and refined it by adjusting algorithms and handling class imbalances.

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

https://github.com/sauravcal/applieddatascience/blob/56c5dd0fc7973863e302ab73fbe4a78fb0f34fb5/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

- You need present your model development process using key phrases and flowchart
- Start -->
- Data Acquisition --> Data Preprocessing --> First Model Selection --> Model Training --> Model Evaluation --> Check if Model Acceptable
- If Yes: --> Best Model Selection --> End
- If No: --> Model Improvement --> Re-Train Model to Improv --> Evaluate Model for Improvement --> Check if Model Acceptable
- If Yes: --> Best Model Selection --> End
- If No: --> Alternative Model Selection --> Model Training --> Model Evaluation --> Check if Model Acceptable
- If Yes: --> Best Model Selection --> End
- If No: --> Further iterations or alternative approaches -->
- End

Results

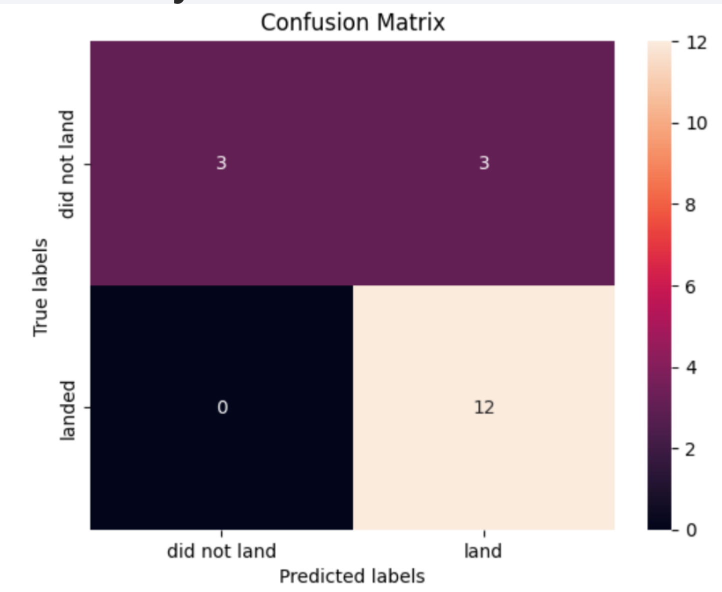
- Exploratory data analysis results
- *Performed data cleaning and normalization*
- *Performed correlation analysis using confusion matrix*

Accuracy for each Model

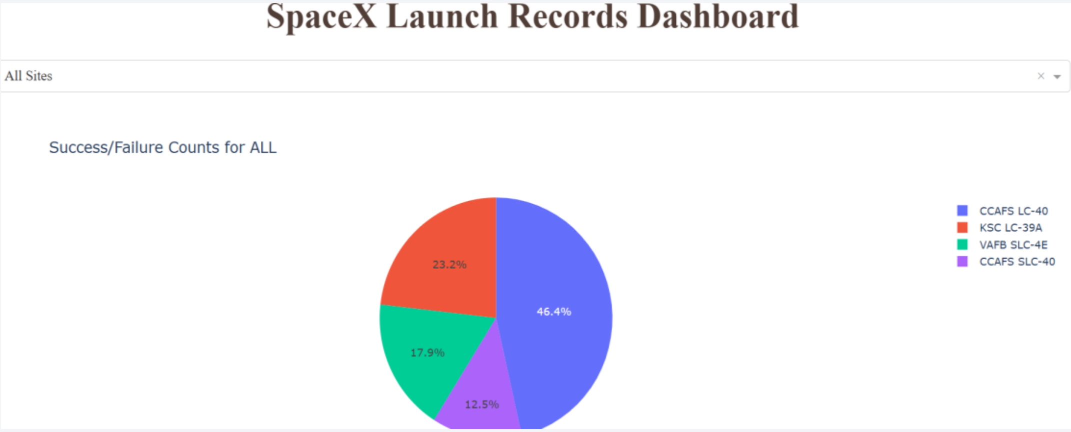
- Logistic Regression: 0.8333
- SVM: 0.8333
- Decision Tree: 0.7778
- k-Nearest Neighbors: 0.8333

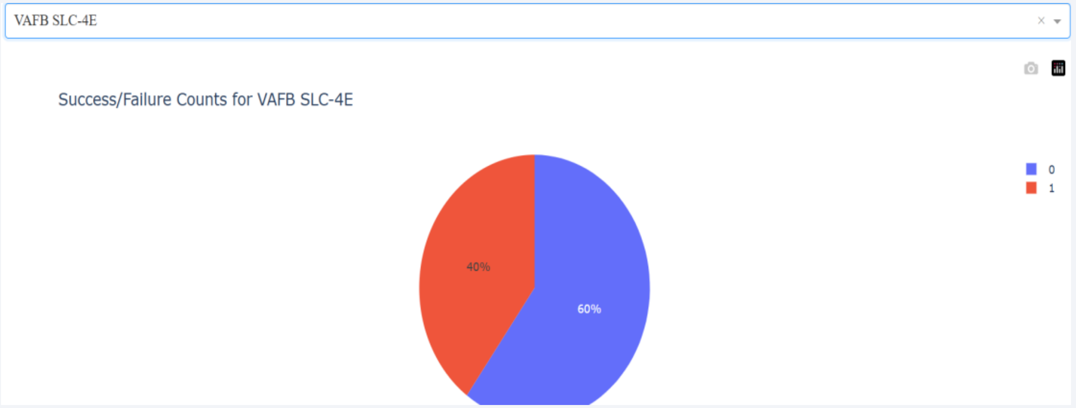
Based solely on accuracy, Logistic Regression, SVM, and k-Nearest Neighbors performed equally well.

Predictive analysis results



- Interactive analytics demo in screenshots



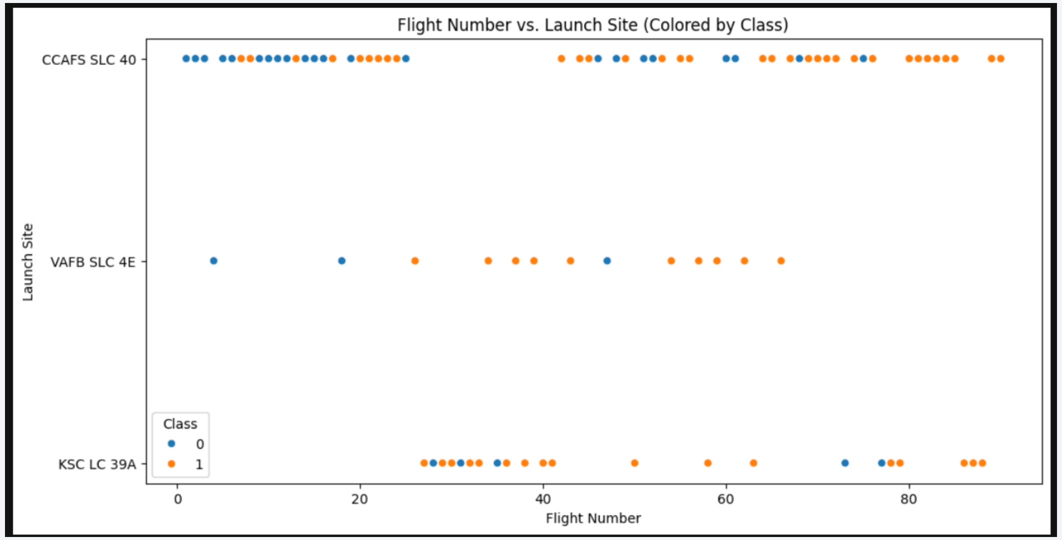


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- plt.figure(figsize=(12, 6)) # Adjust figure size if neededsns.scatterplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df)plt.title("Flight Number vs. Launch Site (Colored by Class)")plt.xlabel("Flight Number")plt.ylabel("Launch Site")plt.show()
- Show the screenshot of the scatter plot with explanations
- The initial Flight numbers were failures.



Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- plt.figure(figsize=(12, 6))sns.scatterplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df)plt.title("Payload Mass (kg) vs. Launch Site (Colored by Class)")plt.xlabel("Payload Mass (kg)")plt.ylabel("Launch Site")plt.show()

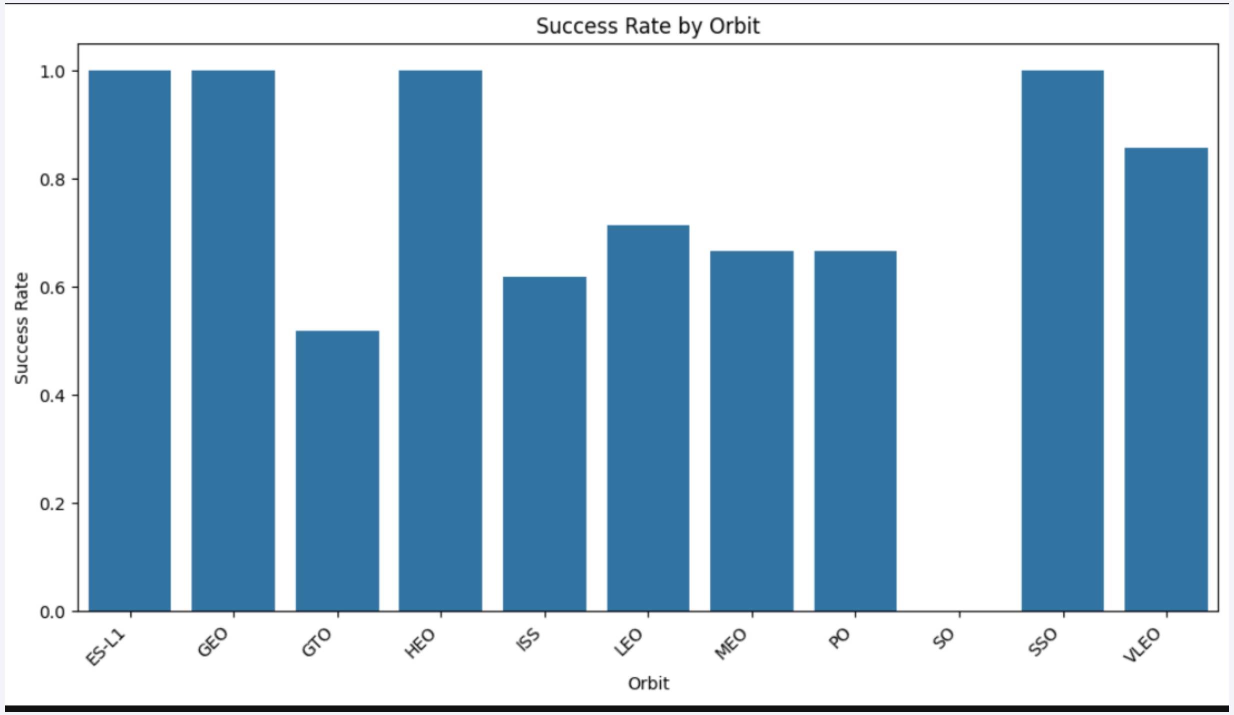
- Show the screenshot of the scatter plot with explanations

- For the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations



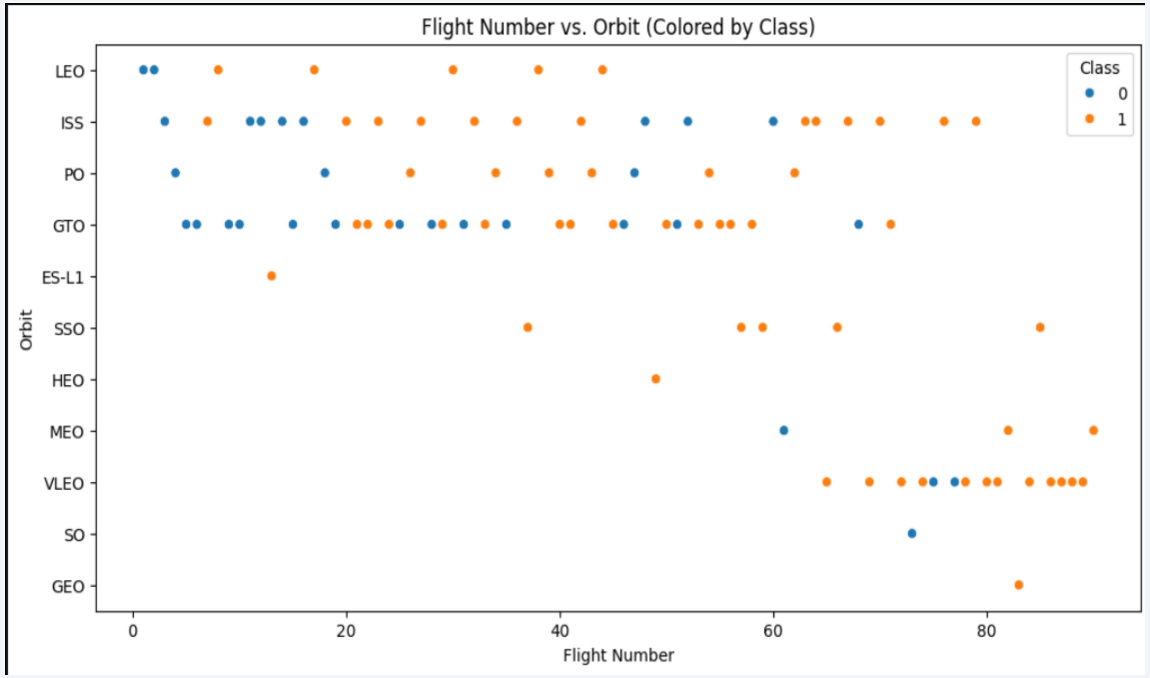

```
orbit_success_rate = df.groupby('Orbit')['Class'].mean().reset_index()

# Create the bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x='Orbit', y='Class', data=orbit_success_rate)
plt.title('Success Rate by Orbit')
plt.xlabel('Orbit')
plt.ylabel('Success Rate')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better readability
plt.show()
```

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

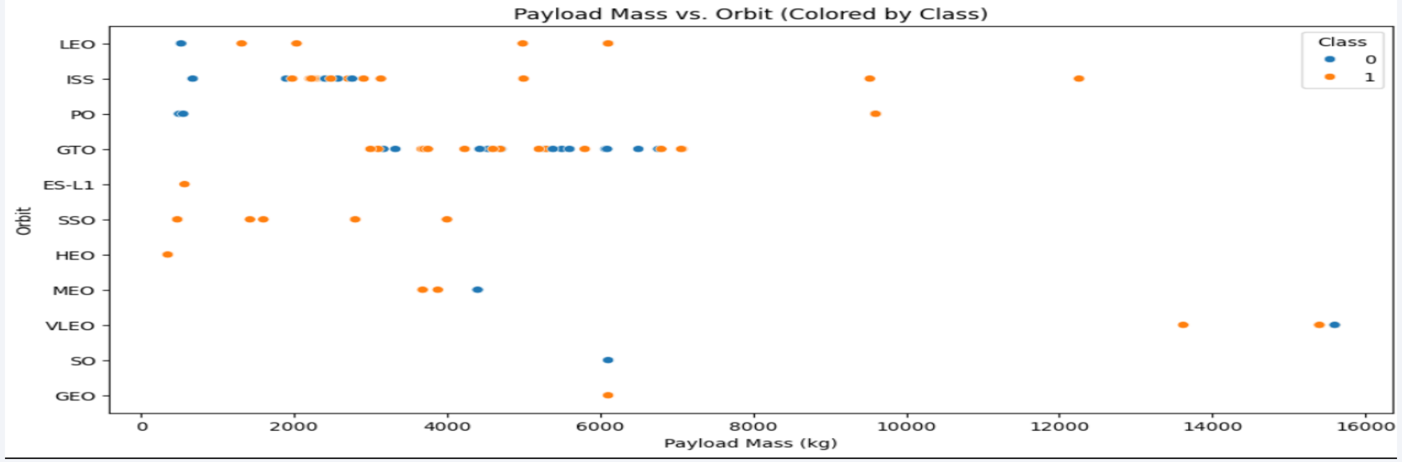
```python
plt.figure(figsize=(12, 6))
sns.scatterplot(x="FlightNumber", y="Orbit", hue="Class", data=df)
plt.title("Flight Number vs. Orbit (Colored by Class)")
plt.xlabel("Flight Number")
plt.ylabel("Orbit")
plt.show()
```



- Show the screenshot of the scatter plot with explanations

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

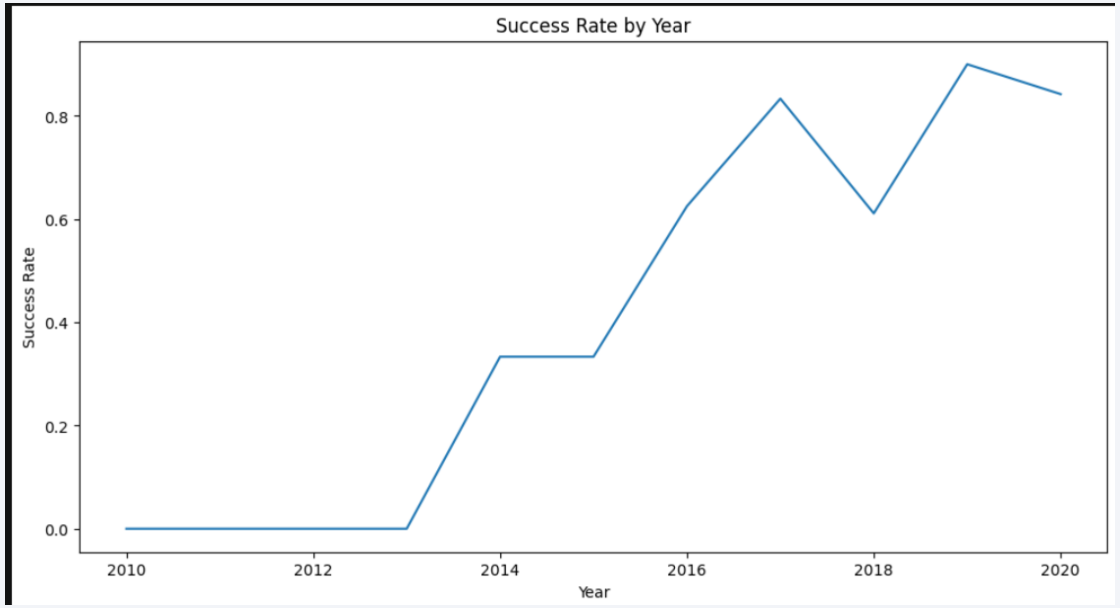



```
plt.figure(figsize=(12, 6))
sns.scatterplot(x="PayloadMass", y="Orbit", hue="Class", data=df)
plt.title("Payload Mass vs. Orbit (Colored by Class)")
plt.xlabel("Payload Mass (kg)")
plt.ylabel("Orbit")
plt.show()
```

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations




```
# Calculate the success rate for each year
year_success_rate = df.groupby('Date')['Class'].mean().reset_index()

# Convert 'Date' column to numeric for plotting
year_success_rate['Date'] = pd.to_numeric(year_success_rate['Date'])

# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
plt.figure(figsize=(12, 6))
sns.lineplot(x='Date', y='Class', data=year_success_rate)
plt.title('Success Rate by Year')
plt.xlabel('Year')
plt.ylabel('Success Rate')
plt.show()
```

All Launch Site Names

- Find the names of the unique launch sites
- %%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE

- Present your query result with a short explanation here
- The Query removes duplicate occurrence of the Launch sites where a site has [illegible] multiple times

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- %%sqlSELECT *FROM SPACEXTBLWHERE Launch_Site LIKE 'CCA%'LIMIT 5

- Present your query result with a short explanation here
- The Query returns first 5 rows where Launch Site name starts with 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, 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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- %%sqlSELECT SUM("PAYLOAD_MASS__KG_")FROM "SPACEXTABLE"WHERE "CUSTOMER" = 'NASA (CRS)';

- Present your query result with a short explanation here
- The query is summing up all payload where Customer is NASA (CRS)

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- SELECT AVG("PAYLOAD_MASS__KG_")FROM "SPACEXTBL"WHERE "Booster_Version" = 'F9 v1.1';
- Present your query result with a short explanation here

AVG("PAYLOAD_MASS__KG_")
2928.4

- The Average of payload all the rows is taken where Booster_Version = 'F9 v1.1’

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %%sqlSELECT MIN("Date")FROM "SPACEXTBL"WHERE "Landing_Outcome" = 'Success (ground pad)';

- Present your query result with a short explanation here

MIN("Date")
2015-12-22

- The First successful landing was in 22/12/2015 which is taken by the oldest date using MIN function and Outcome = Success

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
- *%%sqlSELECT "Booster_Version"FROM "SPACEXTBL"WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;*
- Present your query result with a short explanation here
- All Booter version is listed where the payload is between 4000 and 6000 is identified uwing < > operators.

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
- %%sqlSELECT "Mission_Outcome", COUNT(*) AS "Outcome_Count"FROM "SPACEXTBL"WHERE "Mission_Outcome" like ('Success%') or "Mission_Outcome" like ('Failure%')GROUP BY "Mission_Outcome";
- Present your query result with a short explanation here
- Grouping by mission outcome shows the count under each category

Mission_Outcome	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
- %%sqlSELECT DISTINCT "Booster_Version"FROM "SPACEXTBL"WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM "SPACEXTBL");

Present your query result with a short explanation here

Identify Max Payload and compare it against individual payloads and list the Booster_version.

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
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
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
- %%sqlSELECT substr("Date", 6, 2) AS "Month", "Landing_Outcome", "Booster_Version", "LaunchSite"FROM "SPACEXTBL"WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)';
- Present your query result with a short explanation here

Month	Landing_Outcome	Booster_Version	"LaunchSite"
01	Failure (drone ship)	F9 v1.1 B1012	LaunchSite
04	Failure (drone ship)	F9 v1.1 B1015	LaunchSite

- Extract the month and check for Landing outcome as Failure on drone ship for Year 2015.The year is extracted from the Date column.

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
- %%sqlSELECT "Landing_Outcome", COUNT(*) AS "Outcome_Count"FROM "SPACEXTBL"WHERE "Date" BETWEEN '2010-06-04' AND '2017-03 20'GROUP BY "Landing_Outcome"ORDER BY "Outcome_Count" DES
- Present your query result with a short explanation here
- Rank is done using grouping and counting by Landing outcome,

followed by sorting on the count for each group.

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

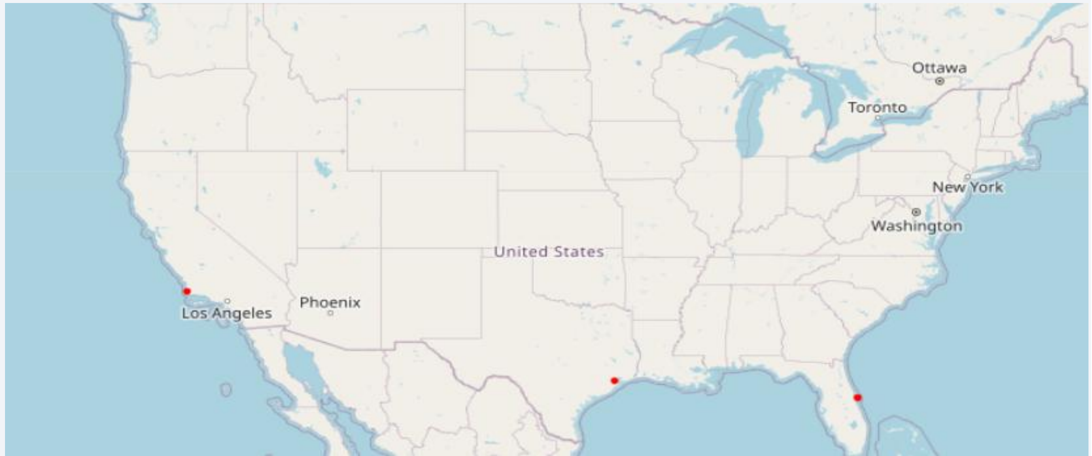
Section 3

Launch Sites Proximities Analysis

Launch Sites for SpaceX

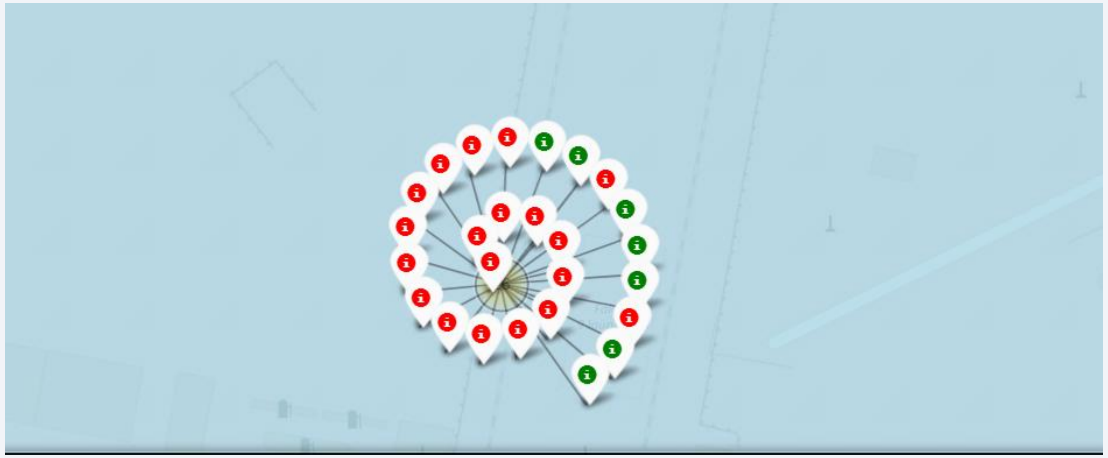
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- *Launch sites are marked in red dot.*
- Explain the important elements and findings on the screenshot
- The launch has taken place from 3 geographical

locations and multiple sites within those locations.



Launch Site with Outcome

- 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
- The Map shows the outcome Success/Failure of a Geographical region on zooming.

Launch site with Prxomity to Habitat

- 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

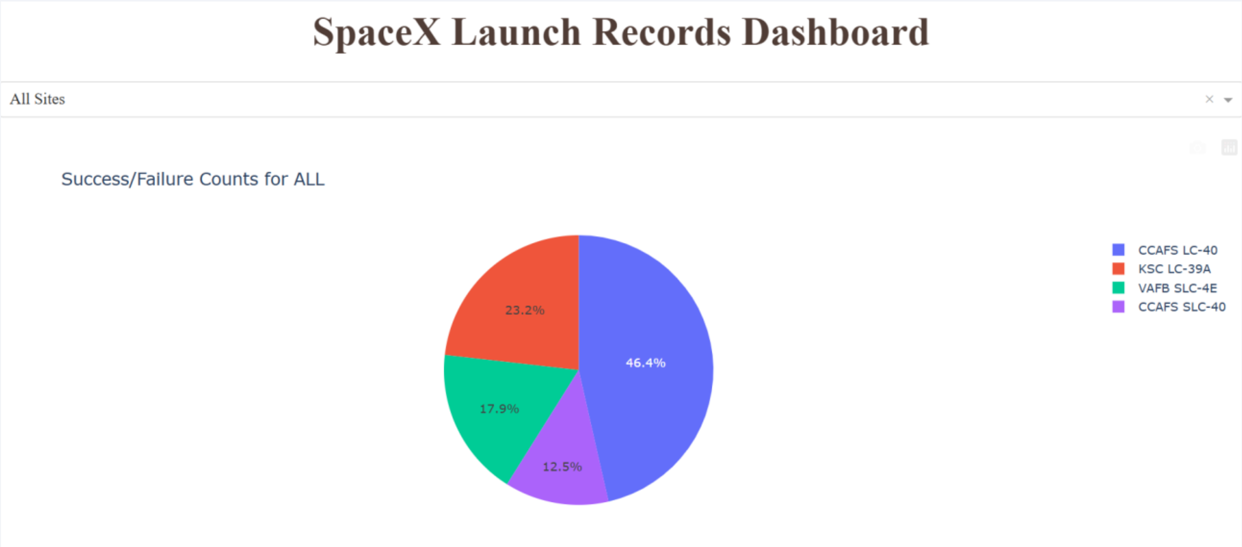
- Explain the important elements and findings on the screenshot
- The map shows the distabce between the railway, highline, coastline coordinate (lat, lon) from the launch site (lat,lon) in km

Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

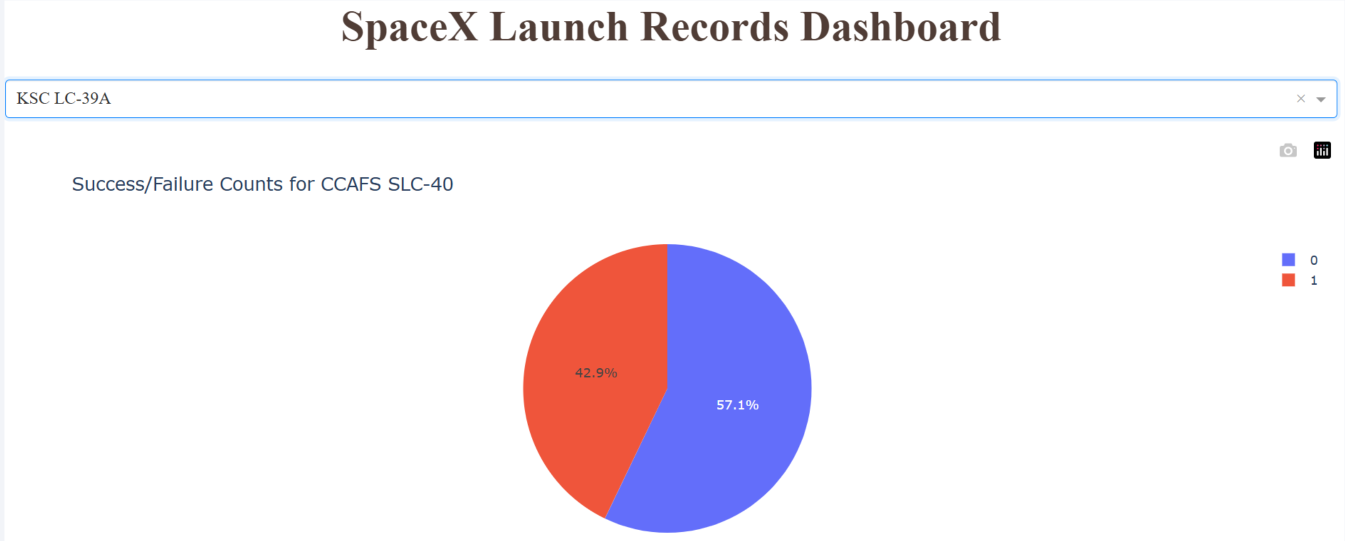
- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart



- Explain the important elements and findings on the screenshot
- We can se 4 distinct launch sites with Success/Failure count

SpaceX Launch Records – Highest Success

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio



- Explain the important elements and findings on the screenshot
- The Highest success is for KSC Location. 57%

Payload and Success Dashboard

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.
- With Higher payload, there is more Failure as seen by adjusting the Slider.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy
- *Decision Tree Model has the Highest Classification Accuracy as it has the Highest score.*
- *If we are looking at 'Accuracy Results' the Logistic Regression, SVM, and k-Nearest Neighbors models all have the highest classification* accuracy (0.876785714285714)

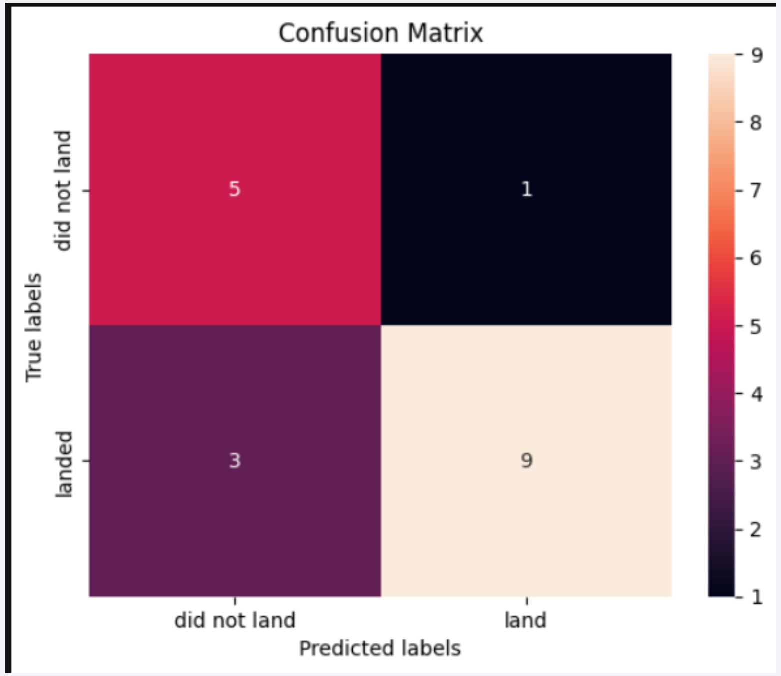
Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- Best Performing Model is Decision Tree
- Confusion Matrix of Decision Tree, can

be seen from the landed. Incorrect Predictions

are 0, 3 which ae low in comparison to 9,5

which are correct.



Conclusions

- The Decision Tree model demonstrated the highest predictive performance, achieving a score of approximately 0.877, suggesting its superior capability in classifying launch outcomes within the dataset
- While the Support Vector Machine (SVM) and k-Nearest Neighbors (KNN) models exhibited comparable scores around 0.848, the Logistic Regression model presented a slightly lower score of approximately 0.846.
- The Payload is directly correlated to the Success as more Payload is adding to failures. For initial launch with Low payloads, the success rate was Higher
- Launch site proximity to geographical features like Railway line, Main Highway, Cities had no connection to its success.

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!