IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sakshi Shinde
9/9/2023

Outline

Introduction

- Project background and context:
  - SpaceX's achievements include sending spacecraft to the International Space Station (ISS), establishing the Starlink satellite internet constellation, and conducting manned space missions. A key factor contributing to SpaceX's success is its cost-efficiency, with Falcon 9 rocket launches costing substantially less than those of its competitors.
  - In this context, a new player, "Space Y," aims to compete with SpaceX in the commercial space launch market. To do so, Space Y needs to determine the pricing strategy for its rocket launches.
- Problems you want to find answers:
  - First-Stage Recovery Prediction
  - Competitor Strategy Understanding
  - Business Strategy Recommendations
  - Data Continuity and Monitoring

Section
1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API Data Collection - The primary source of data is the SpaceX REST API (Application Programming Interface).The specific API endpoint used for collecting data is /launches/past, which provides information about past SpaceX launches.
  - API Response - When a GET request is made to the API, it responds with data in JSON (JavaScript Object Notation) format.
  - Web Scraping (Alternative): - Python's BeautifulSoup package is used for web scraping to extract Falcon 9 launch records from HTML tables on relevant Wiki pages.
- Perform data wrangling
  - Data Transformation: To make the data suitable for analysis, the json_normalize function is employed.
  - Filtering Data: In cases where the collected data includes records of Falcon 1 launches, filtering is applied to exclude these records.
  - Handling Null Values: Null values in the dataset, particularly in the PayloadMass column, are addressed.

Methodology

- Perform exploratory data analysis (EDA) using visualization and SQL

  Exploratory Data Analysis (EDA) involves visually exploring and summarizing your dataset to gain insights and identify patterns, anomalies, and potential relationships in the data.

- Perform interactive visual analytics using Folium and Plotly Dash

  Interactive visual analytics can provide deeper insights into your data by allowing users to explore and interact with visualizations.

- Perform predictive analysis using classification models

Performing predictive analysis using classification models involves building, tuning, and evaluating models to make predictions about categorical outcomes

Data Collection

- SpaceX REST API Data Collection:
  The primary source of data is the SpaceX REST API (Application Programming Interface).The specific API endpoint used for collecting data is /launches/past, which provides information about past SpaceX launches.

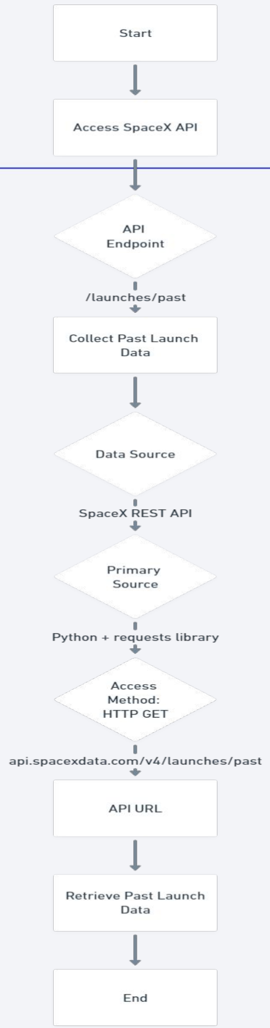
- API Response:
  When a GET request is made to the API, it responds with data in JSON (JavaScript Object Notation) format.

- Web Scraping (Alternative):
  Python's BeautifulSoup package is used for web scraping to extract Falcon 9 launch records from HTML tables on relevant Wiki pages.

Data Collection – SpaceX API

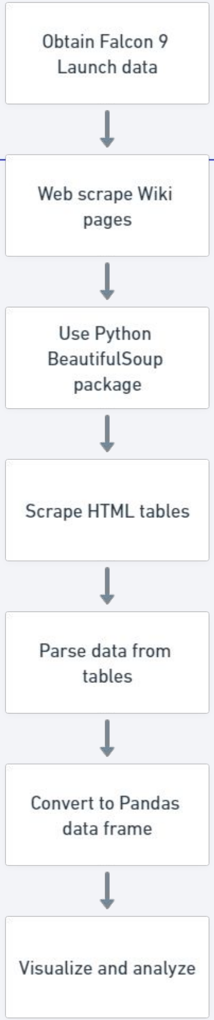
- The primary source of data is the SpaceX REST API (Application Programming Interface). The specific API endpoint used for collecting data is /launches/past, which provides information about past SpaceX launches. The API is accessed using Python and the requests library to perform HTTP GET requests to the API's URL: api.spacexdata.com/v4/launches/past. create a flowchart

- GitHub URL - https://github.com/sakshinde/SpaceY/blob/66db95b1ebbc97e62018b772e834480301199712/Lab%201%20Collecting%20the%20data.ipynb



Data Collection - Scraping

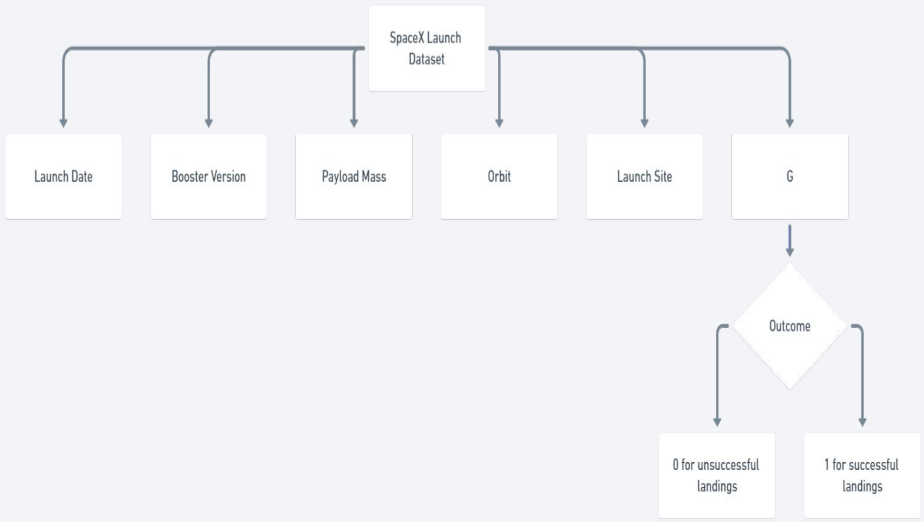
- Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages. In this lesson, you will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Then you need to parse the data from those tables and convert them into a Pandas data frame for further visualization and analysis.
- GitHub URL https://github.com/sakshinde/SpaceY/blob/66db95b1ebbc97e62018b772e834480301199712/lab%202%20web%20scraping.ipynb



Data Wrangling

- The dataset contains information about SpaceX launches, including attributes like launch date, booster version, payload mass, orbit, launch site, and landing outcome (success or failure). The "Outcome" attribute is converted to binary classes: 0 for unsuccessful landings and 1 for successful landings, simplifying classification. This dataset is suitable for analyzing factors influencing successful rocket landings.

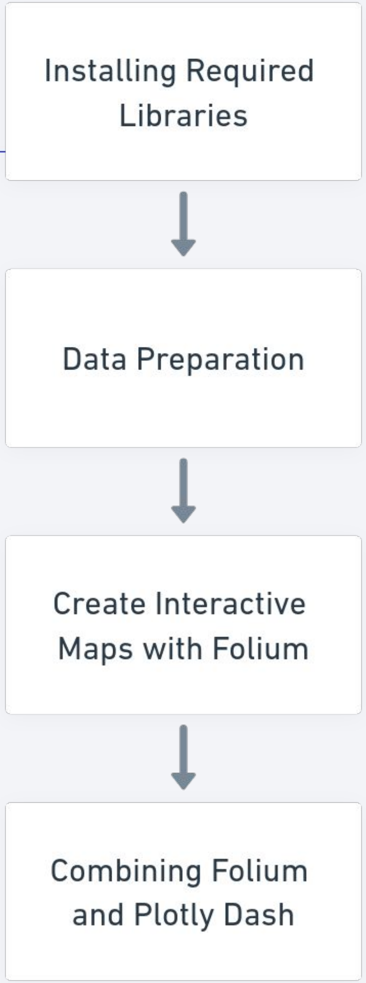
- GitHub URL
  https://github.com/sakshinde/SpaceY/blob/main/lab%203%20data%20wrangling.ipynb



EDA with Data Visualization

- Interactive visual analytics can provide deeper insights into your data by allowing users to explore and interact with visualizations. You can achieve interactive visual analytics using Python libraries like Folium for geographical data and Plotly Dash for creating web-based interactive dashboards

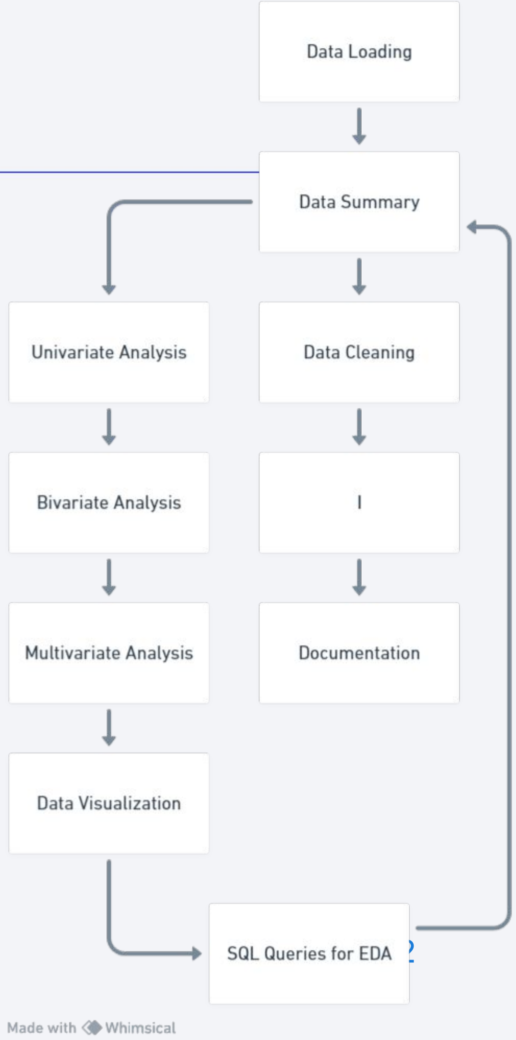
- GitHub URL
  https://github.com/sakshinde/SpaceY/blob/main/lab%204%20EDA%20with%20data%20visualization.ipynb



EDA with SQL

- %load_ext sql
- import csv, sqlite3
  con = sqlite3.connect("my_data1.db")
  cur = con.cursor()
- %sql sqlite:///my_data1.db
- %sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null

- GitHub URL
  https://github.com/sakshinde/SpaceY/blob/main/lab%205%20data%20visualisation%20with%20sql.ipynb



Build an Interactive Map with Folium

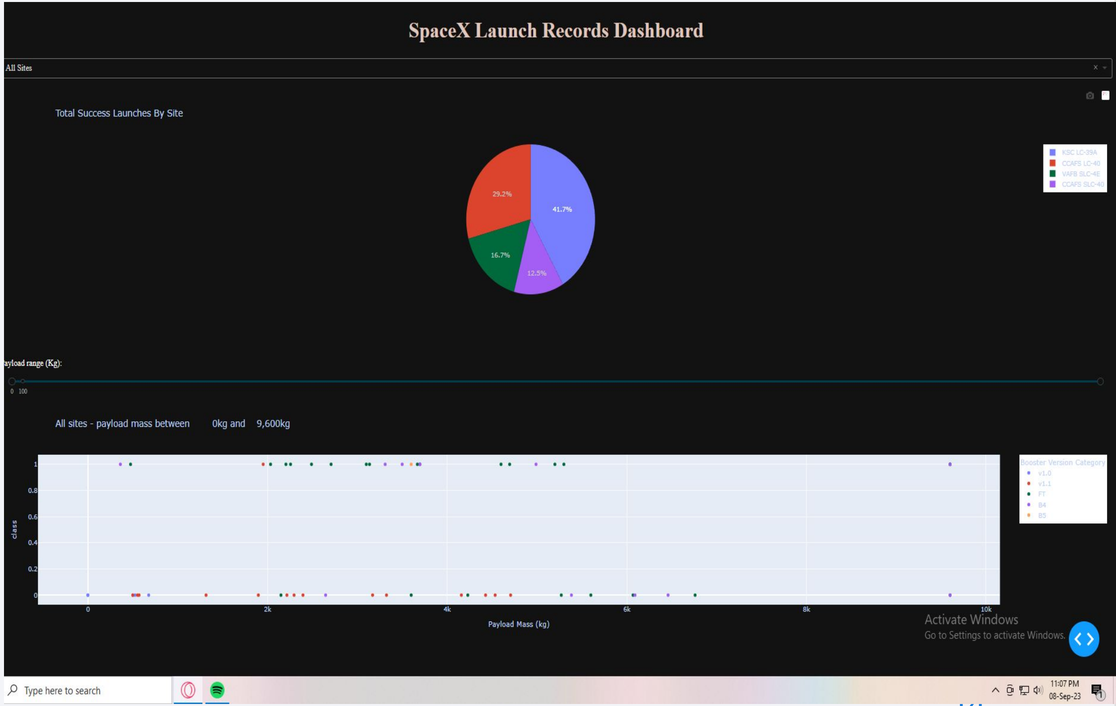
- Use Folium to create interactive maps to visualize geographical data. You can add markers, popups, and other interactive elements.
- folium.Marker([lat1, lon1], popup='Marker 1').add_to(m) folium.Marker([lat2, lon2], popup='Marker 2').add_to(m)

- Interactive maps and dashboards allow stakeholders to explore the SpaceX dataset more effectively. Instead of passive viewing of static graphs, users can actively engage with the data, zooming in, panning, filtering, and interacting with various visualization components.

- GitHub URL https://github.com/sakshinde/SpaceY/blob/main/lab%206Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

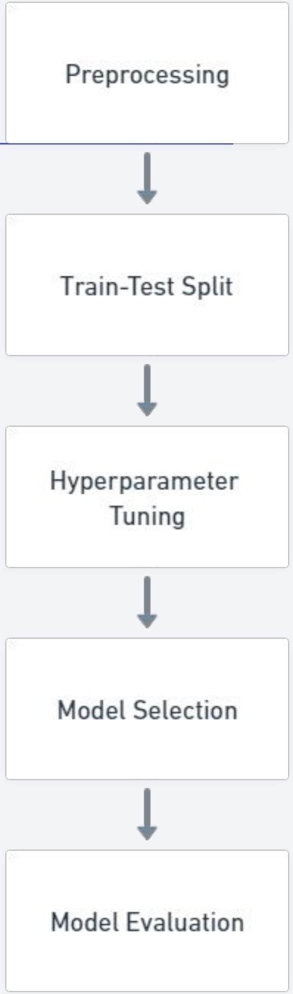
Build a Dashboard with Plotly Dash

- The dashboard includes a launch site drop-down for site selection, a success pie chart displaying success rates, a range slider for payload mass selection, and a success-payload scatter chart.
- These components allow users to filter and explore launch data, analyze success rates by site, and assess the relationship between payload mass and success. They enhance data exploration and support informed decision-making.



Predictive Analysis (Classification)

- To build the best classification model:
- 1. Preprocessed and split the data.
- 2. Selected classification algorithms.
- 3. Trained models with default hyperparameters.
- 4. Evaluated initial models.
- 5. Tuned hyperparameters using techniques like Grid Search.
- 6. Improved models by updating hyperparameters.
- 7. Evaluated refined models on testing data.
- 8. Selected the best-performing model based on evaluation metrics.
- 9. Considered deployment if the model meets criteria.
- 10. Ongoing monitoring and maintenance for real-world use.
- GitHub URL https://github.com/sakshinde/SpaceY/blob/main/lab%206Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb



Results

- Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.

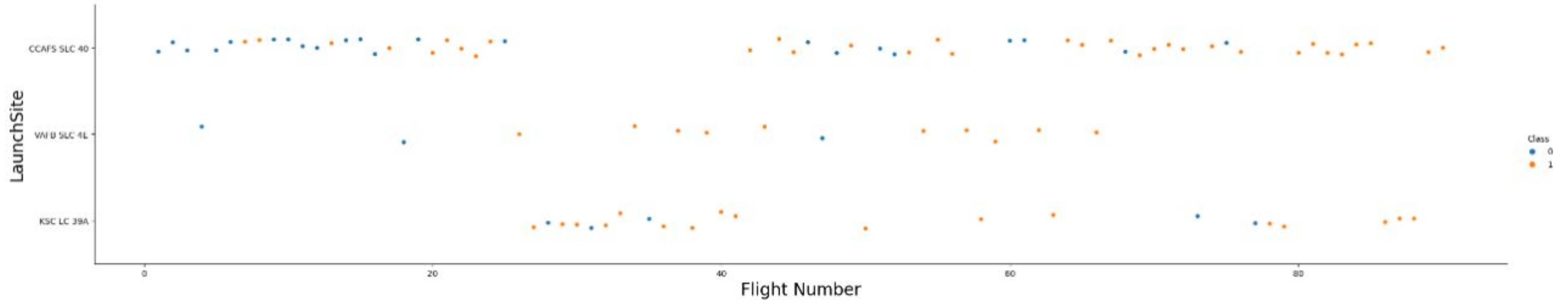
- Most launches happens at east cost launch sites.

- Predictive Analysis showed that Decision Tree Classifier is the best model to predict successful landings, having accuracy over 87% and accuracy for test data over 94%

Section
2

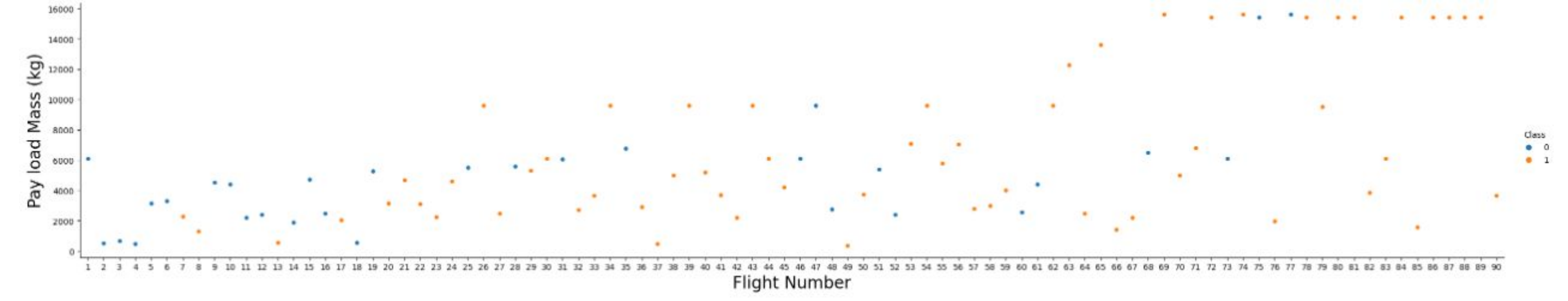
Insights drawn from EDA

Flight Number vs. Launch Site



- "Based on the analysis depicted in the plot above, it is evident that the most successful launch site in recent times is CCAFS SLC 40, with a significantly high rate of successful launches. Following closely in second place is VAFB SLC 4E, and in third place is KSC LC 39A.
- Furthermore, the data shows a positive trend in the overall success rate of SpaceX launches over time. This indicates an improvement in the reliability and success of Falcon 9 missions."
- This revised statement provides a clear and concise summary of the findings from the plot.

Payload vs. Launch Site



- Payloads exceeding 9,000kg, which is roughly equivalent to the weight of a school bus, exhibit an exceptional success rate in Falcon 9 launches.
- It is worth noting that payloads exceeding 12,000kg appear to be feasible primarily from the launch sites at CCAFS SLC 40 and KSC LC 39A."
- This revised statement retains the key information while providing clarity and conciseness.

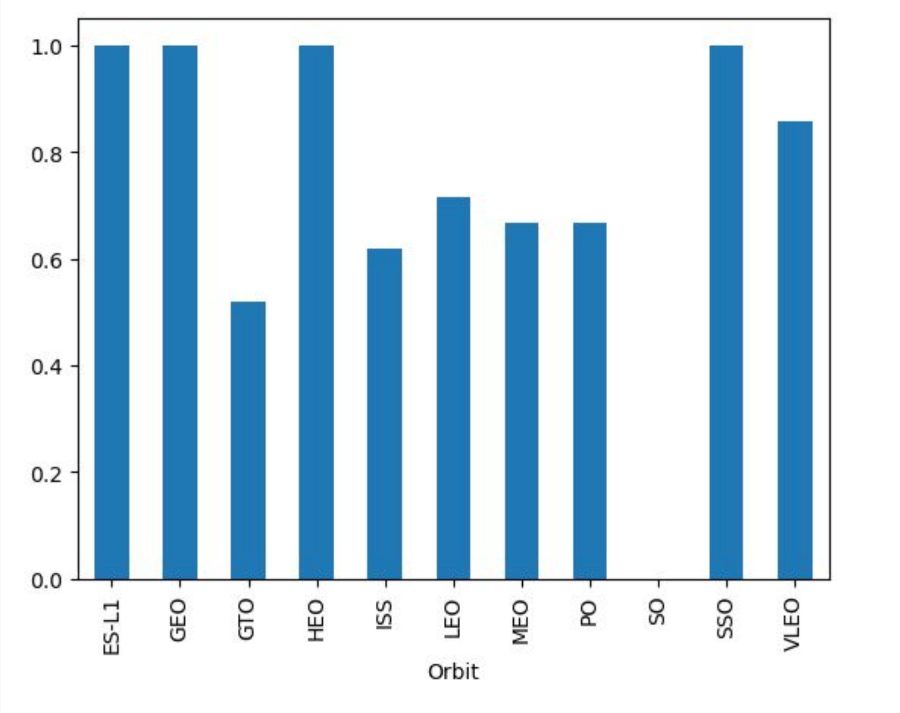
Success Rate vs. Orbit Type

The highest success rates are observed for launches into the following orbits:

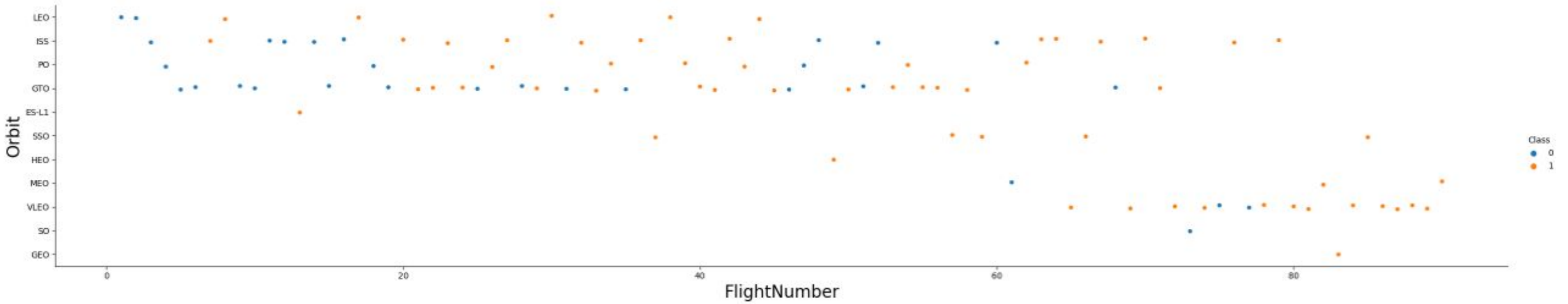
- ES-L1 (Earth-Sun Lagrange Point 1)
- GEO (Geosynchronous Earth Orbit)
- HEO (Highly Elliptical Orbit)
- SSO (Sun-Synchronous Orbit)

Additionally, orbits with notably high success rates include:

- VLEO (Very Low Earth Orbit) with a success rate exceeding 80%
- LFO (Low Earth Orbit) with a success rate exceeding 70%

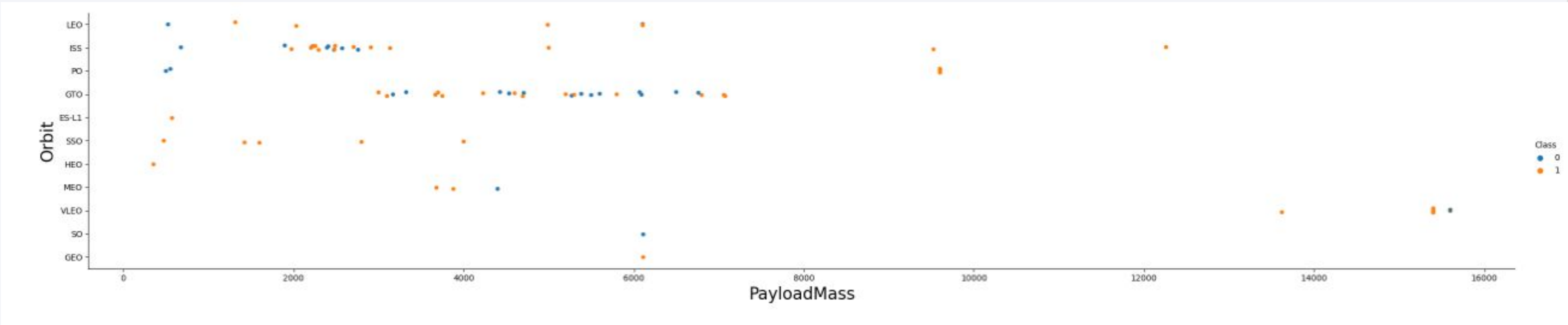


Flight Number vs. Orbit Type



- Interestingly, the success rate for all orbits has shown improvement over time, indicating a positive trend in mission reliability.
- Notably, the VLEO (Very Low Earth Orbit) appears to be a promising new business opportunity, as evidenced by the recent increase in the frequency of launches to this orbit.

Payload vs. Orbit Type

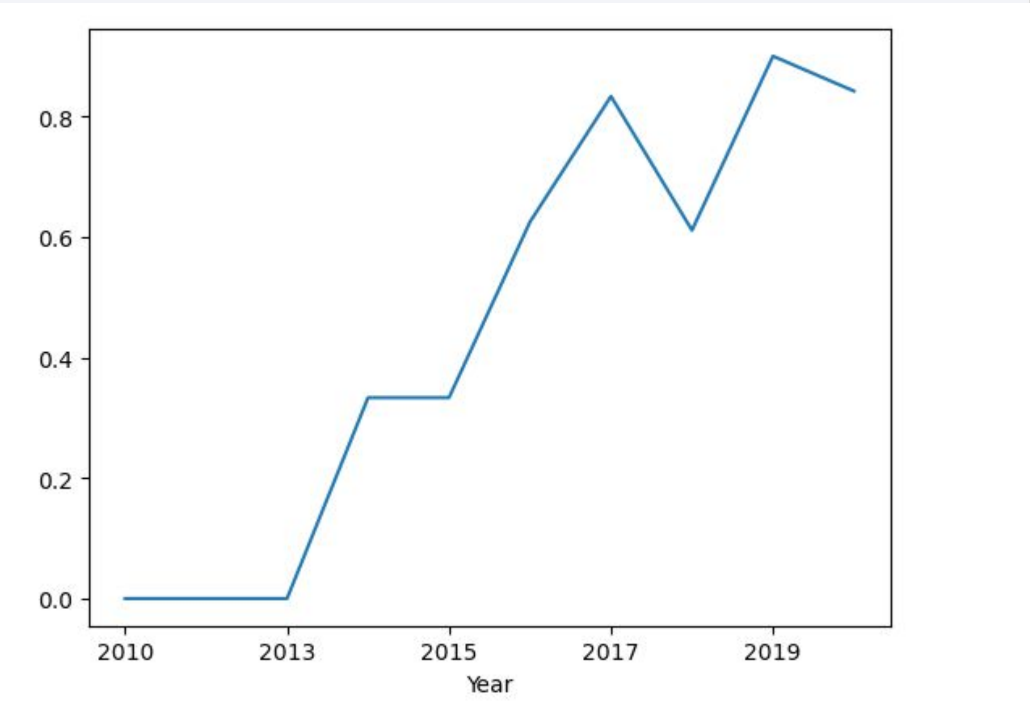


- It appears that there is no significant correlation between payload size and the success rate for launches into Geosynchronous Transfer Orbit (GTO).
- Remarkably, the International Space Station (ISS) orbit exhibits the broadest range of payload sizes and maintains a commendable success rate.
- It's worth noting that there have been relatively few launches to orbits such as SO (Supersynchronous Orbit) and GEO (Geosynchronous Earth Orbit)

Launch Success Yearly Trend

- The success rate began to show a significant increase starting in 2013 and continued to rise until 2020.

- It appears that the initial three years represented a period of adjustments and technological improvements



All Launch Site Names

- Based on the data, there are four distinct launch sites, each identified by selecting unique occurrences of 'launch_site' values from the dataset.

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

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
- This SQL query extracts unique values from the "LAUNCH_SITE" column in the "SPACEXTBL" table. It then sorts these unique values in ascending order based on their natural order (alphabetically, in this case) using the "ORDER BY" clause with "1" as shorthand for the first column. This query provides a list of launch sites without duplicates, presented in alphabetical order.

Total Payload Mass

TOTAL_PAYLOAD
111268

- sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD LIKE '%CRS%';

- This SQL query computes the sum of payload masses (column "PAYLOAD_MASS__KG_") from the "SPACEXTBL" table. It specifically selects records where the "PAYLOAD" column contains the substring 'CRS'. The result of this sum is given an alias "TOTAL_PAYLOAD." Essentially, it's calculating the combined payload mass for records associated with missions that have 'CRS' in their payload description.

Average Payload Mass by F9 v1.1

AVG_PAYLOAD
2928.4

- sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
- This SQL query calculates the average payload mass for records in the "SPACEXTBL" table where the "BOOSTER_VERSION" column is equal to 'F9 v1.1'. Here's an explanation:
- SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD: This part of the query selects the "PAYLOAD_MASS__KG_" column and calculates the average of its values. The result is assigned the alias "AVG_PAYLOAD."
- FROM SPACEXTBL: This specifies the table from which the data is being selected, which is the "SPACEXTBL" table in this case.
- WHERE BOOSTER_VERSION = 'F9 v1.1': This is the conditional part of the query. It filters the rows from the table where the "BOOSTER_VERSION" column is equal to 'F9 v1.1'. The average payload mass is calculated only for records associated with this specific booster version.

First Successful Ground Landing Date

first_success_gp
2015-12-22

- By filtering data by successful landing outcome on ground pad and getting the minimum value for date it’s possible to identify the first occurrence, that happened on 12/22/2015.
- This SQL query calculates the earliest date of a successful ground pad landing, assigning it the alias "FIRST_SUCCESS_GP." Here's a brief explanation:
- 1. `SELECT MIN(DATE) AS FIRST_SUCCESS_GP`: This part of the query selects the minimum (earliest) date from the "DATE" column and labels it as "FIRST_SUCCESS_GP."
- 2. `FROM SPACEXTBL`: Specifies the source table as "SPACEXTBL."
- 3. `WHERE LANDING__OUTCOME = 'Success (ground pad)'`: It filters the rows in the table where the "LANDING__OUTCOME" column indicates a successful landing on a ground pad. The query identifies and returns the earliest date with this specific outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success
- This SQL query retrieves distinct values from the "BOOSTER_VERSION" column of the "SPACEXTBL" table. It selects only those rows where the "PAYLOAD_MASS__KG_" falls within the range of 4000 to 6000 and the "LANDING__OUTCOME" is 'Success.' However, the query appears to be incomplete. Here's the explanation of what it does so far:
- SELECT DISTINCT BOOSTER_VERSION: This part of the query selects distinct values from the "BOOSTER_VERSION" column.
- FROM SPACEXTBL: Specifies the source table as "SPACEXTBL."
- WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success': Filters the rows from the table where the "PAYLOAD_MASS__KG_" falls within the range of 4000 to 6000 and the "LANDING__OUTCOME" is 'Success.'

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

Total Number of Successful and Failure Mission Outcomes

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

- This SQL query provides a summary of mission outcomes by counting the occurrences of each unique "MISSION_OUTCOME" value in the "SPACEXTBL" table. It then presents the results in ascending order based on the "MISSION_OUTCOME" values. Here's an explanation:
- 1. `SELECT MISSION_OUTCOME, COUNT(*) AS QTY`: This part of the query selects two columns:
- - "MISSION_OUTCOME": Represents the mission outcome.
- - "COUNT(*) AS QTY": Calculates the count of records for each unique "MISSION_OUTCOME" value and assigns it an alias "QTY."
- 2. `FROM SPACEXTBL`: Specifies the source table as "SPACEXTBL."
- 3. `GROUP BY MISSION_OUTCOME`: Groups the records based on the "MISSION_OUTCOME" column. This groups records with the same "MISSION_OUTCOME" value together.
- 4. `ORDER BY MISSION_OUTCOME`: Orders the results in ascending order based on the "MISSION_OUTCOME" values.
- The result of this query will provide a list of mission outcomes along with the quantity (count) of records associated with each outcome.

Boosters Carried Maximum Payload

- This SQL query retrieves distinct values from the "BOOSTER_VERSION" column of the "SPACEXTBL" table. It selects only those rows where the "PAYLOAD_MASS__KG_" is equal to the maximum payload mass value found in the entire "SPACEXTBL" table. Here's an explanation:
- 1. `SELECT DISTINCT BOOSTER_VERSION`: This part of the query selects distinct values from the "BOOSTER_VERSION" column.
- 2. `FROM SPACEXTBL`: Specifies the source table as "SPACEXTBL."
- 3. `WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)`: This condition filters the rows where the "PAYLOAD_MASS__KG_" is equal to the result of a subquery. The subquery `(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)` calculates the maximum payload mass found in the entire "SPACEXTBL" table.
- The result of this query will provide distinct booster versions associated with launches that had the maximum payload mass.

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE)

- This SQL query retrieves data from the "BOOSTER_VERSION" and "LAUNCH_SITE" columns of the "SPACEXTBL" table. It filters the rows based on two conditions:
- LANDING__OUTCOME = 'Failure (drone ship)': This condition selects rows where the "LANDING__OUTCOME" column is equal to 'Failure (drone ship)'.
- DATE_PART('YEAR', DATE): This part of the query extracts the year from the "DATE" column using the DATE_PART function.

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

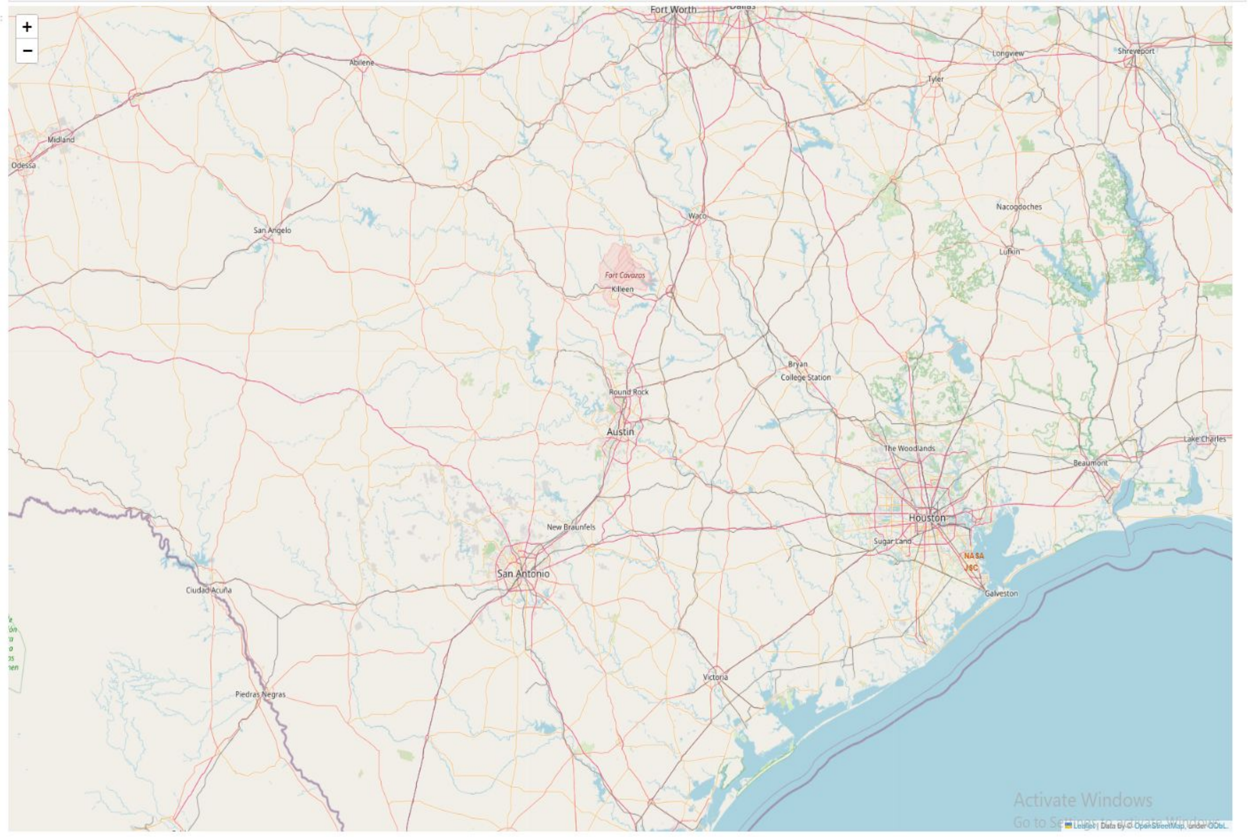
- sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDE
- This SQL query provides a summary of landing outcomes by counting the occurrences of each unique "LANDING__OUTCOME" value in the "SPACEXTBL" table for a specified date range. SELECT LANDING__OUTCOME, COUNT(*) AS QTY: This part of the query selects two columns:
    - "LANDING__OUTCOME": Represents the landing outcome.
    - "COUNT(*) AS QTY": Calculates the count of records for each unique "LANDING__OUTCOME" value within the specified date range and assigns it an alias "QTY."
- FROM SPACEXTBL: Specifies the source table as "SPACEXTBL."
- WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20': This condition filters the rows based on a date range between '2010-06-04' and '2017-03-20'.

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

Section 3

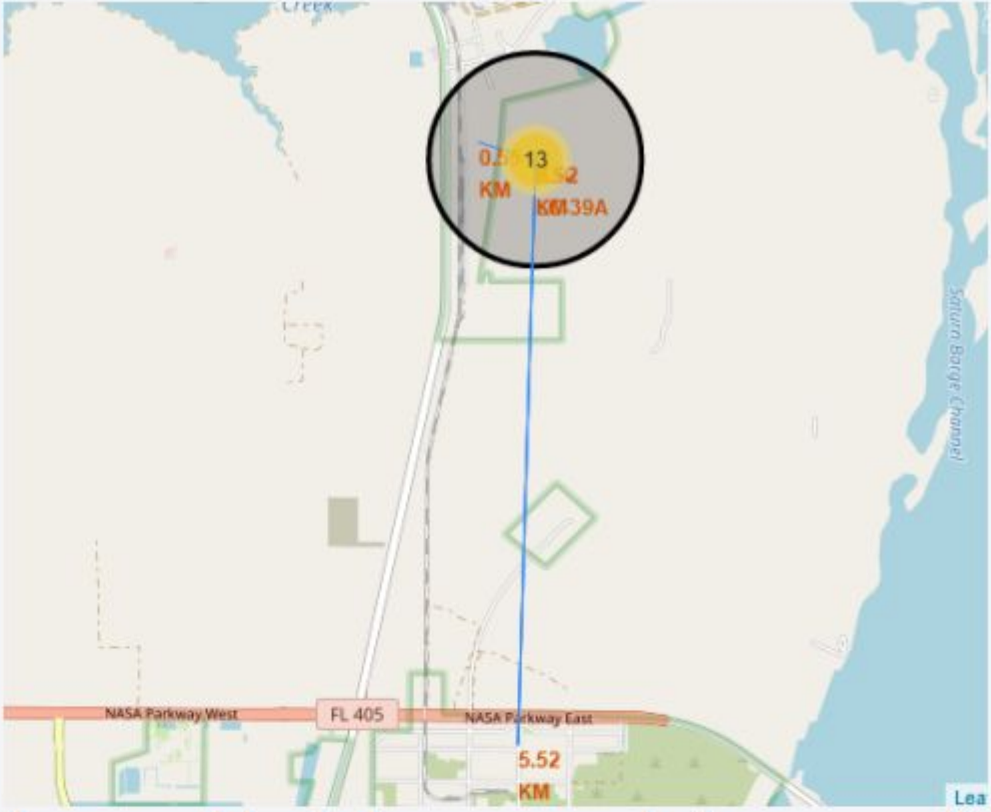
Launch Sites Proximities Analysis

All launch sites



Launch Outcomes by Site

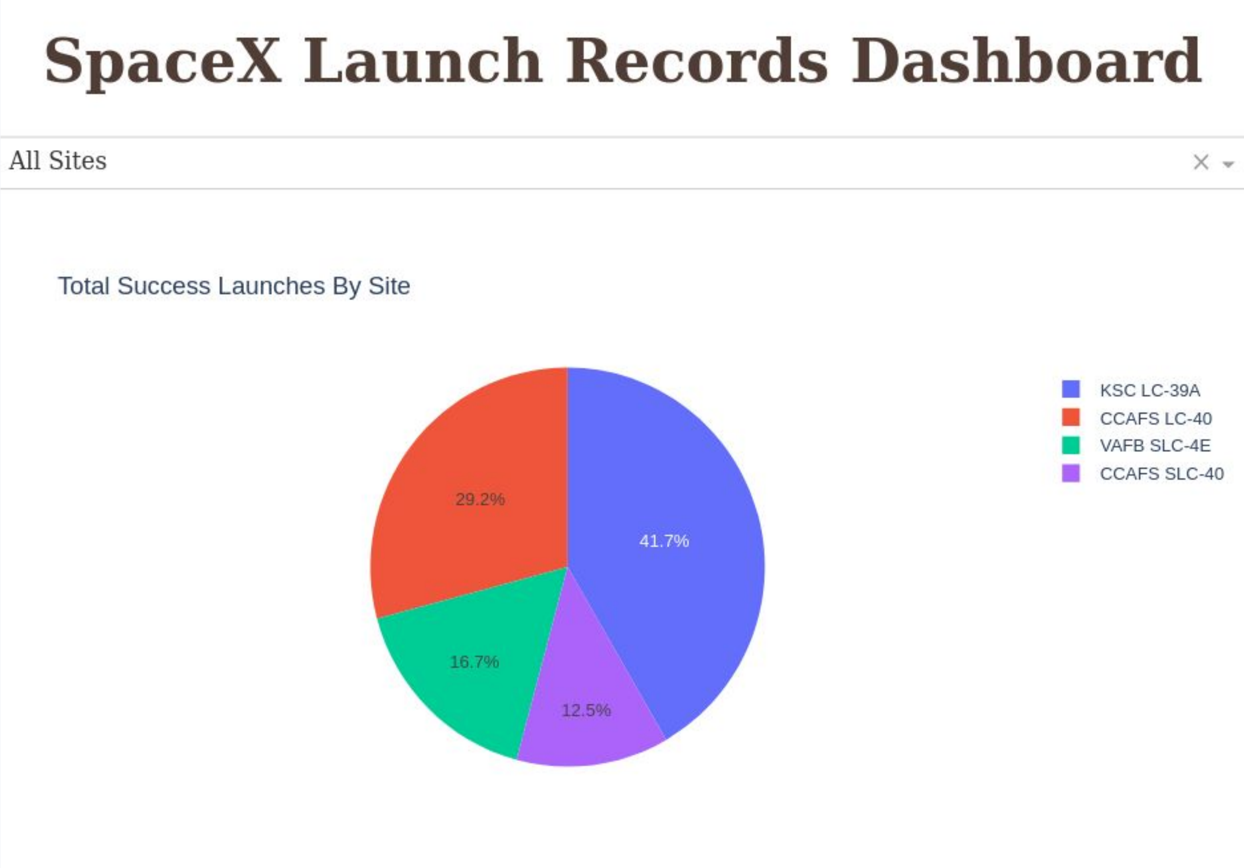
Logistics and Safety



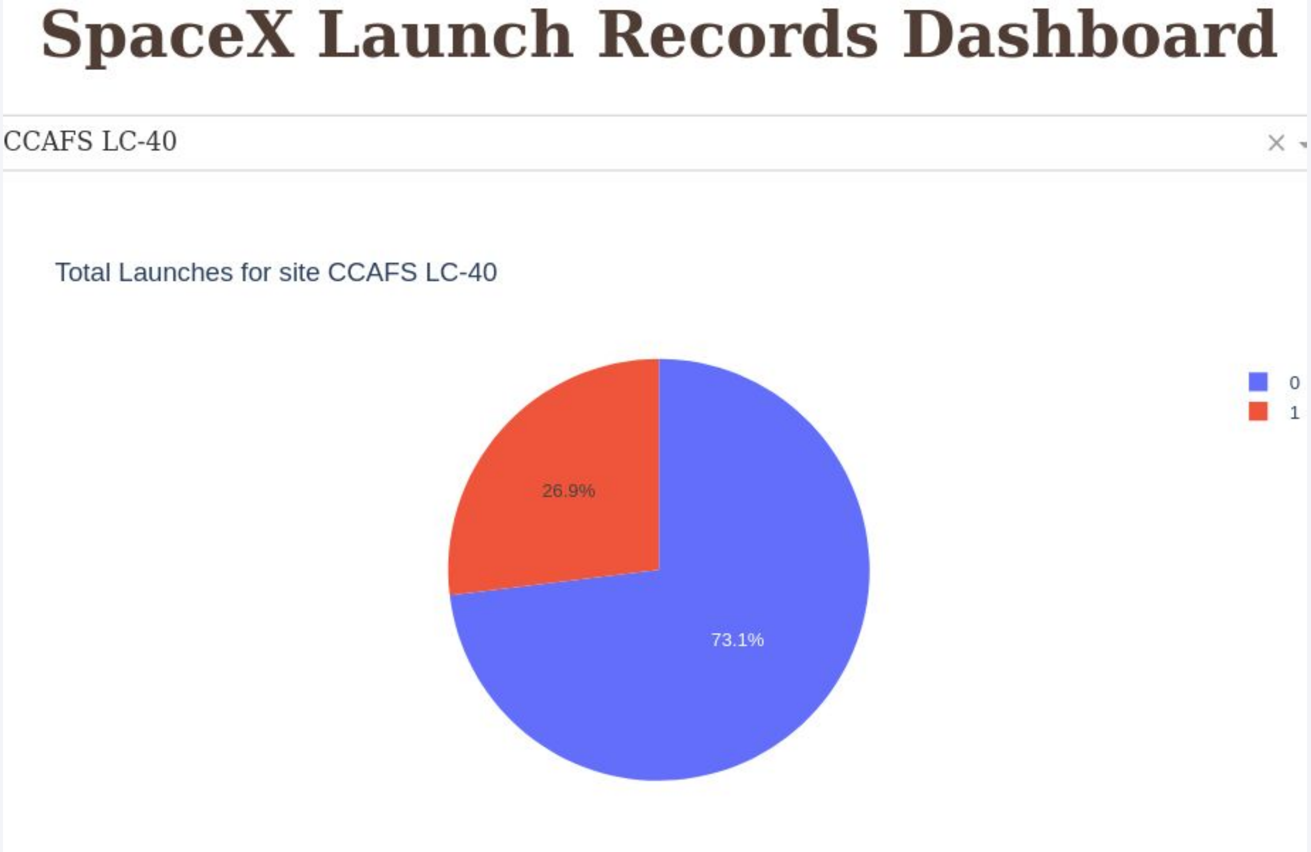
Section
4

Build a Dashboard with Plotly Dash

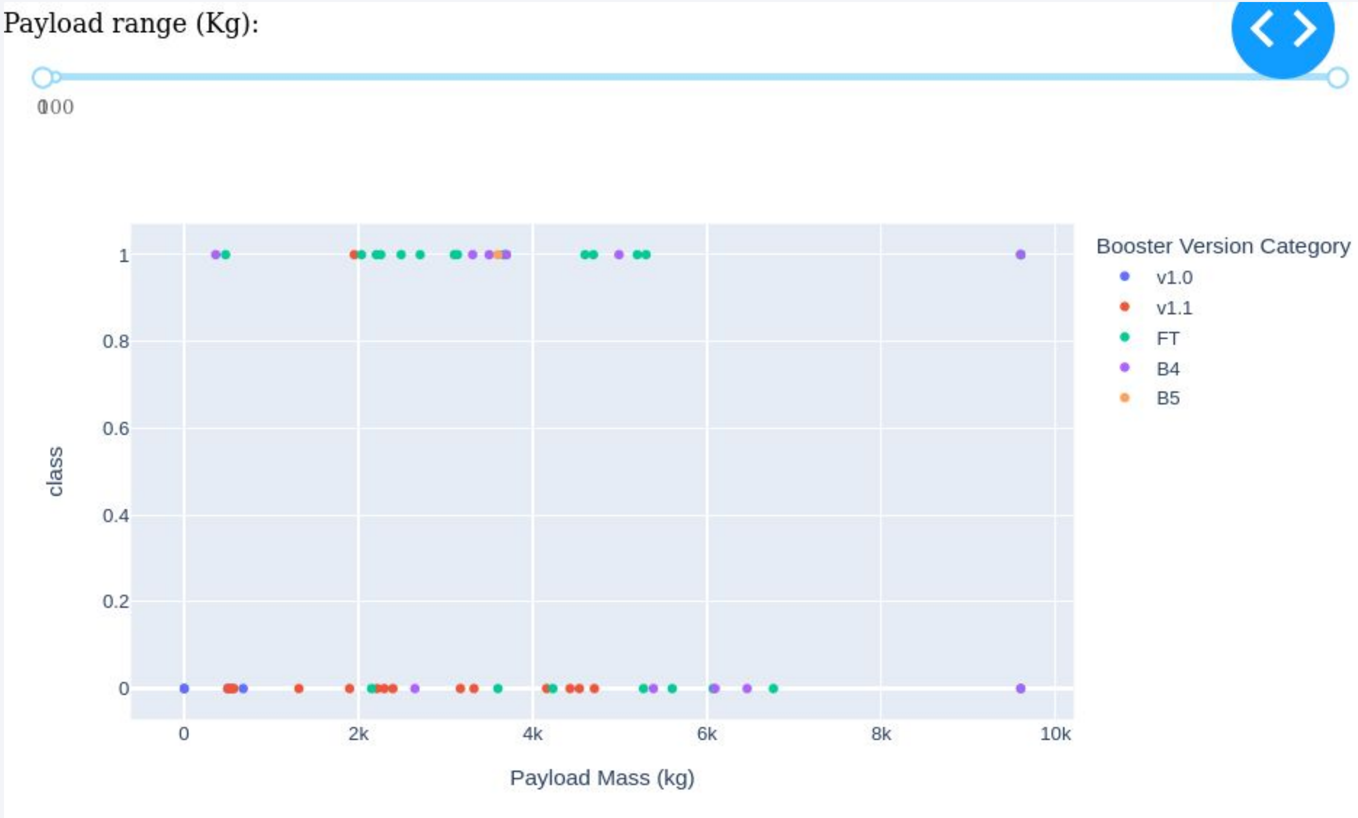
Successful Launches by Site



Launch Success Ratio for KSC LC-39A



Payload vs. Launch Outcome

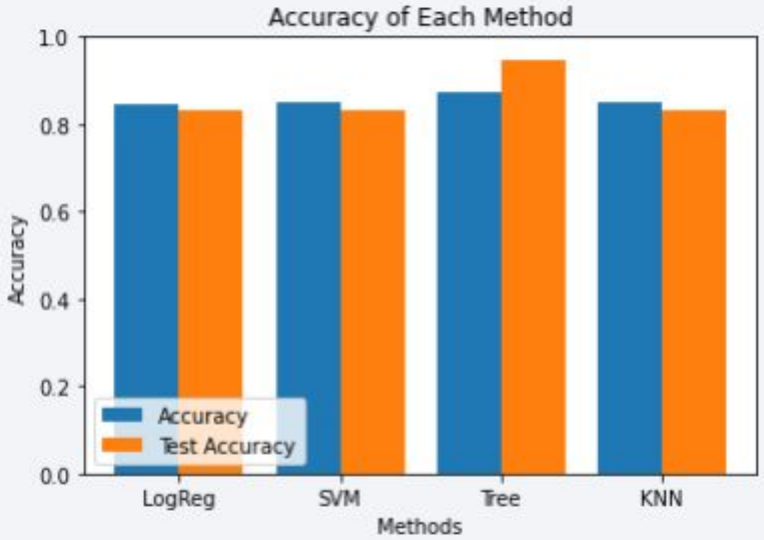


Section 5

Predictive Analysis (Classification)

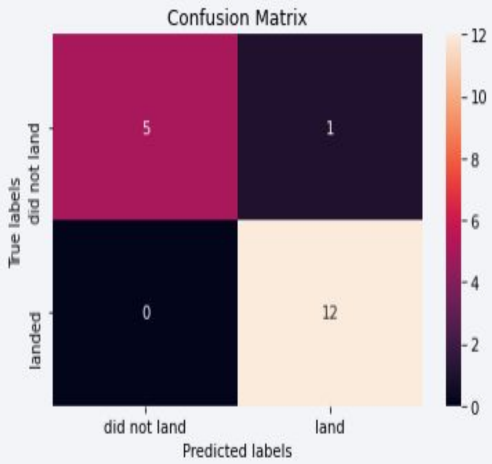
Classification Accuracy

- Four different classification models were tested, and their respective accuracies are displayed in the plot.
- Notably, the Decision Tree Classifier emerged as the top-performing model, boasting an accuracy exceeding 87%



Confusion Matrix

- The Confusion Matrix of the Decision Tree Classifier underscores its accuracy by revealing a substantial number of true positives and true negatives in comparison to the false instances



Conclusions

- "In conclusion, the analysis of SpaceX launch data and the application of classification models have yielded valuable insights.
- Notably, the Decision Tree Classifier demonstrated exceptional accuracy, exceeding 87%, making it a robust choice for predicting the success of Falcon 9 first stage landings.
- The Confusion Matrix further validates its performance by showcasing a significant number of true positives and true negatives.
- These findings are instrumental in enhancing our understanding of SpaceX missions and optimizing future launch predictions."

Thank you!