Winning Space Race with Data Science

Michael Simony

6. Of December 2025

Outline

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

Executive Summary

Summary of methodologies

- This capstone project demonstrates a complete data science workflow applied to real-world SpaceX launch data. The methodology includes systematic data collection from public APIs and web sources, followed by thorough data wrangling using Python to clean and structure the dataset. Exploratory data analysis (EDA) was performed with visualization tools such as matplotlib and seaborn, as well as SQL queries, to uncover key patterns and trends in launch outcomes, payloads, and site usage.
- Interactive analytics were developed using Folium for mapping launch sites and Plotly Dash for dashboard visualizations, enabling dynamic exploration of the data. For predictive analysis, several classification models were built and evaluated, including Logistic Regression, Support Vector Machines (SVM), and Decision Trees.
- Overall, the project highlights effective data-driven approaches for analyzing and predicting outcomes in space launch operations.

Summary of all results

- Logistic Regression, SVM, and Decision Tree models each achieved high accuracy (about 83%) in predicting SpaceX launch success.
- K-Nearest Neighbors (KNN) performed less well, with lower accuracy.
- The best models reliably identified successful landings, with Logistic Regression showing no false positives for "land" outcomes.

Introduction

This capstone project demonstrates the complete data science workflow applied to a real-world dataset.

The goal is to showcase skills in data collection, wrangling, exploratory analysis, visualization, and predictive modeling using Python, SQL, Folium, and Plotly Dash.

- Project background and context
  The project focuses on analyzing [insert dataset/topic, e.g., “spacex launch data”] to uncover patterns and insights that support data-driven decisions.
- Problems you want to find answers
  - What trends and relationships exist in the data?
  - Can we build predictive models to classify outcomes and improve decision-making?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Gathered data from [source, e.g., public datasets, APIs, company databases].
- Perform data wrangling
  - Cleaned and transformed the data using Python (pandas), addressing missing values, outliers, and formatting issues.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Used Python, pandas, and visualization libraries (matplotlib, seaborn) to explore data distributions, relationships, and trends.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Developed interactive maps with Folium and dashboards with Plotly Dash to present results.
- Perform predictive analysis using classification models
  - Predict outcomes. That approach involves classification models. Typical tasks include Building the model, Training the model, Tuning hyperparameters and Evaluating the model.

Data Collection

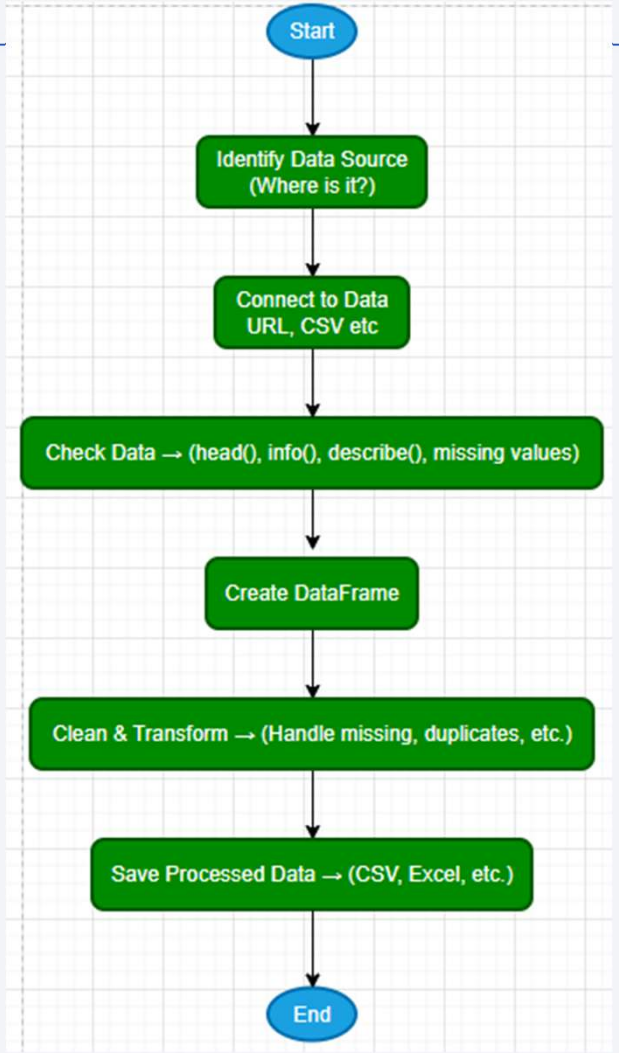
Data Collection is the process of gathering and measuring information on variables of interest, in an established systematic way, to answer research questions, evaluate outcomes, or make informed decisions. It can be done manually or automatically, and the data can come from various sources.

Key Phrases:

What to Collect:

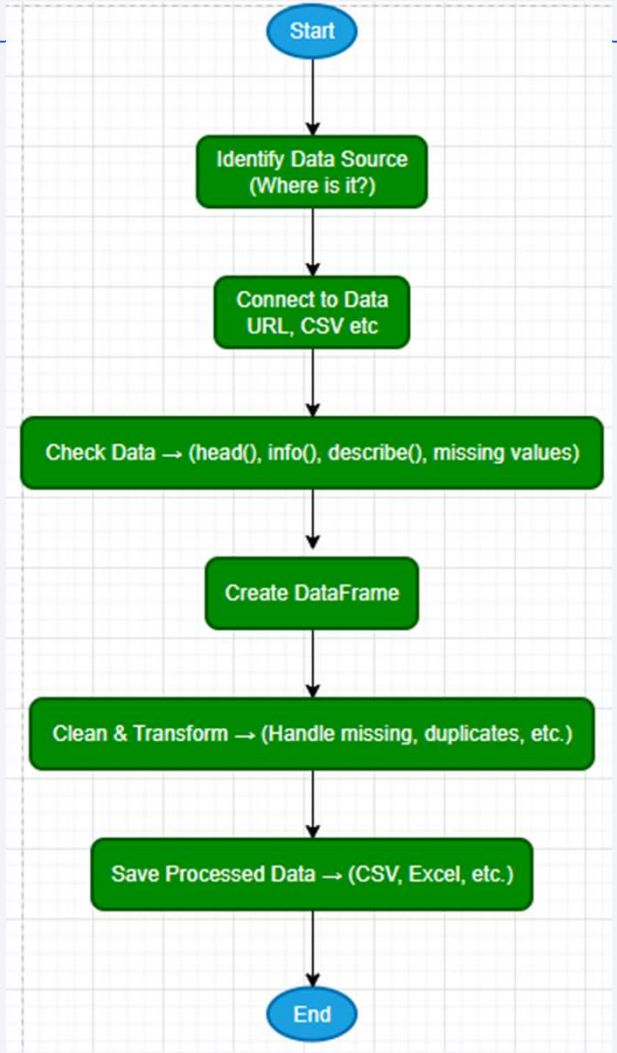
- **User Data:** Demographics, purchase history, saved/liked products, search history, and most visited products.
- **Product Data:** Inventory details, ingredients, popularity, customer ratings.

- **Sources:** Databases, transaction logs, user interactions.
- **Time-Consuming:** Yes. Collecting and identifying relevant data sources can be **labor-intensive.**



Data Collection – SpaceX API

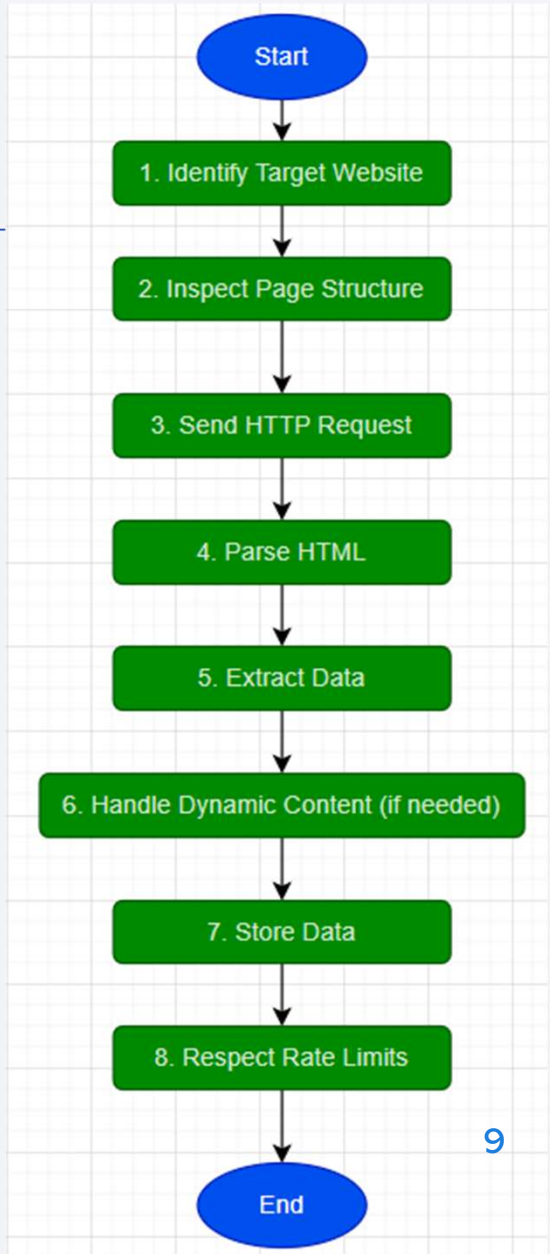
- Define the URL
- Get the data
  response=requests.get(static_json_url)
- Check the data
  response.status_code (=200)
- Convert data to DataFrame
  data =
  pd.json_normalize(response.json())
- See the DataFrame Head
  data.head(5)
- File Name: jupyter-labs-spacex-data-collection-api
- [MichSimo1962/testpro]



Data Collection - Scraping

Key Phrases

- Define Target URL
- Review legal and rights
- Inspect Page Structure
- Send HTTP request
- Parse HTML
- Extract Data
- Store Data
- GitHub URL

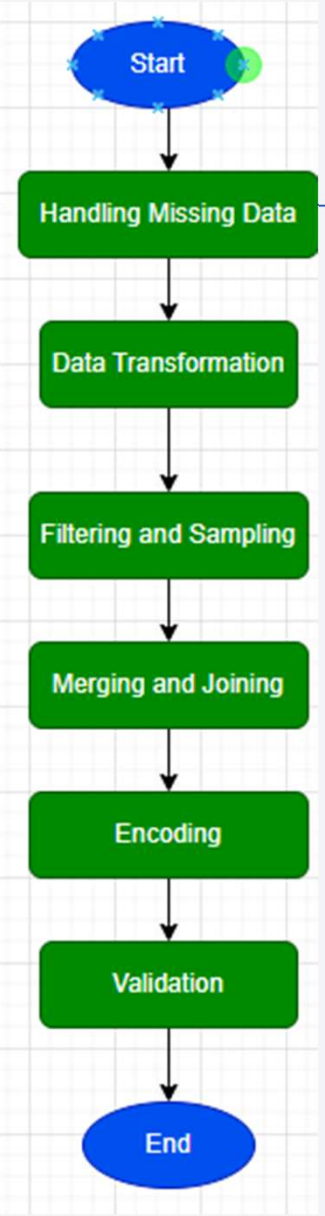


Data Wrangling

Data Wrangling refers to the process of cleaning, transforming, and organizing raw data into a structured and usable format for analysis. It is a critical step in any data science workflow because real-world data often contains inconsistencies, missing values, and irrelevant information.

Key Activities in Data Wrangling

- **Handling Missing Data:** Replace null values with mean, median, or most frequent values.
- **Data Transformation:** Convert data types, normalize values, and create new derived features.
- **Filtering and Sampling:** Remove irrelevant records (e.g., filtering out Falcon 1 launches in SpaceX data).
- **Merging and Joining:** Combine data from multiple sources such as APIs and web scraping.
- **Encoding:** Apply techniques like one-hot encoding for categorical variables.
- **Validation:** Ensure data consistency and integrity before analysis.
- GitHub URL



EDA with Data Visualization

- Scatter plot transform abstract numbers into visual patterns and making it easier to interpret relationships.
- Scatter (Flight Number/Launch Site) – CCAFS have most flights. Class 0 and 1. Easy to quickly get an overview for the three Launch Sites
- Scatter (Payload Mass/ Launch Site – CCAFS and KSC have the heavy flights. This chart show clearly how two Sites are used for heavy weight.
- Bar chart(success rate/each orbit type) – four have most success, one no flight. This kind of chart make it easy to show patterns and differences between groups.
- Line Plot (Year/Succes) – shows that the success has increased over the years.
- https://github.com/MichSimo1962/testpro/blob/main/edadataviz.ipynb

EDA with SQL

- %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
- %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5;
- %sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
- %sql SELECT AVG("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
- %sql SELECT MIN(DATE) AS Frist_Success_Ground FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
- %sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;
- %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Mission_Outcome;
- %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
- %sql SELECT Date,Booster_Version,Launch_Site, substr(Date, 6,2) FROM SPACEXTABLE WHERE substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)';
- %sql SELECT Landing_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY Total DESC;
- https://github.com/MichSimo1962/testpro/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

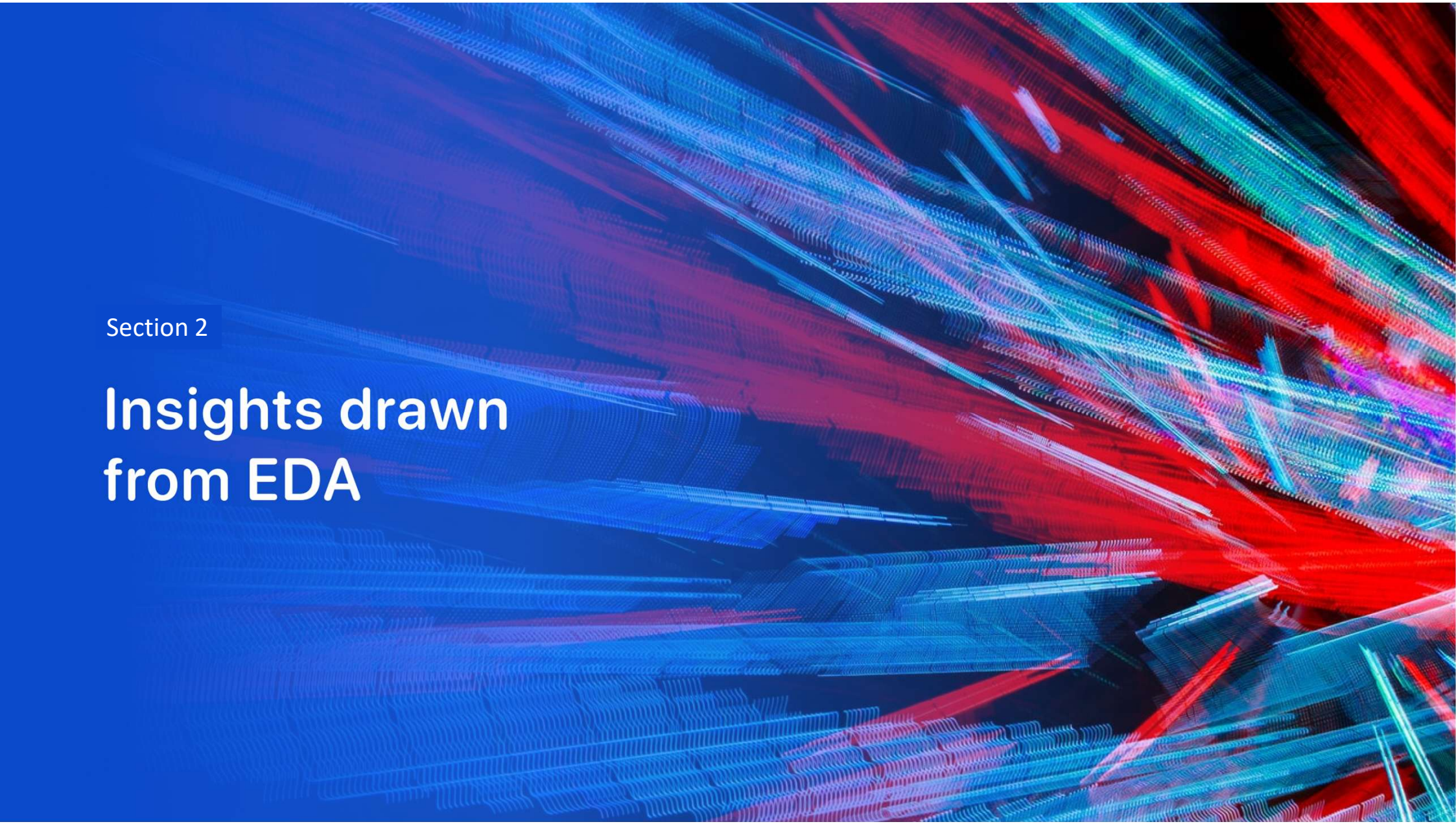
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

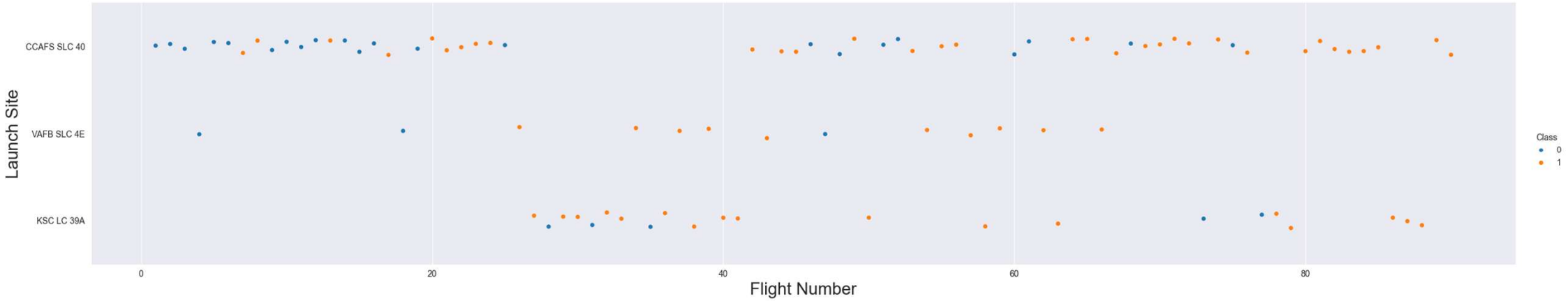
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

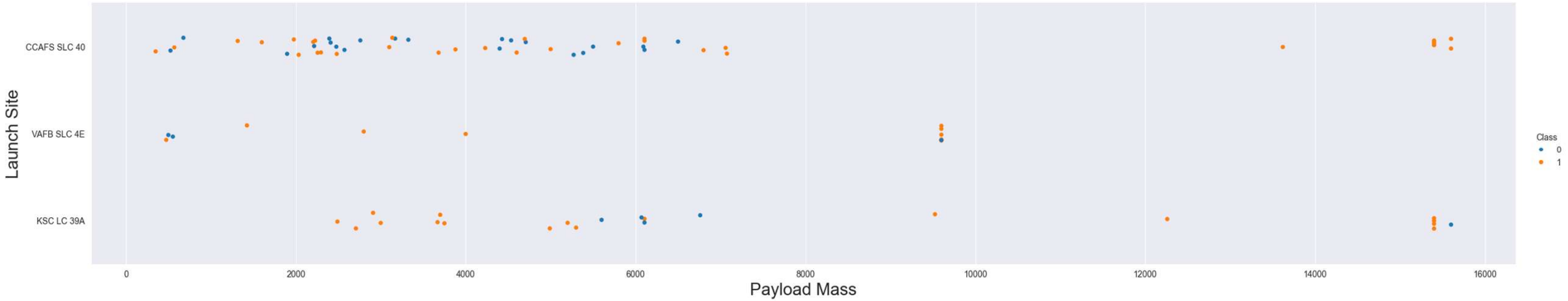


Class 1 Dominance in Later Flights:

- For CCAFS SLC 40 and KSC LC 39A, Succes flight becomes more prevalent in higher flight numbers, suggesting improvements or changes in operations over time.

- Launch Site Usage:
  - CCAFS SLC 40 is used consistently across all flight numbers.
  - KSC LC 39A is used more frequently in later flight numbers.
  - VAFB SLC 4E is used less frequently and mostly for failed flights

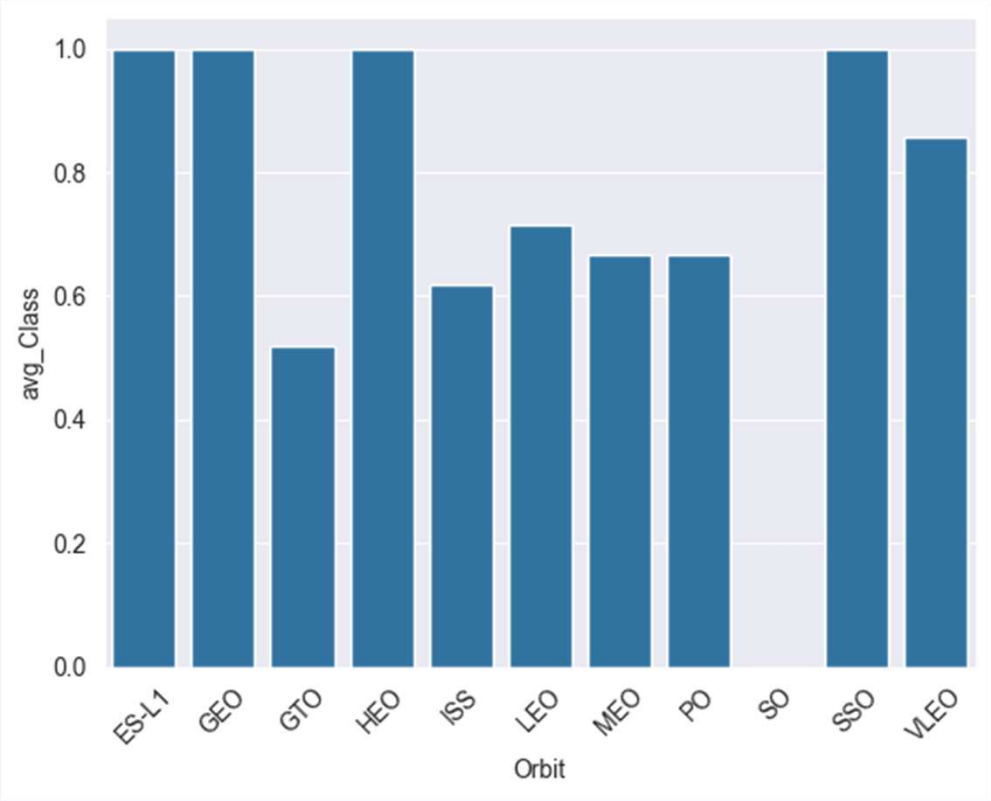
Payload vs. Launch Site



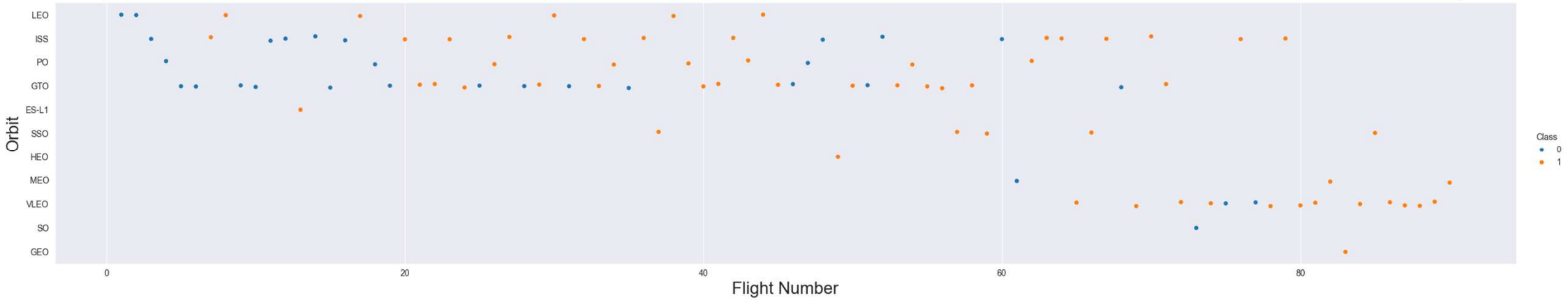
- Payload Mass and Launch Site Usage:
  - CCAFS SLC 40 is versatile, handling a wide range of payload masses for both classes.
  - VAFB SLC 4E is used less frequently and primarily for lower payload masses.
  - KSC LC 39A is predominantly used for higher payload masses, especially those above 10,000 kg.
- Class Distribution:
  - For CCAFS SLC 40, both classes are distributed across the payload mass range, indicating a balanced usage for different mission types or outcomes.
  - For KSC LC 39A, class 1 (orange) is more prevalent for higher payload masses, suggesting it might be preferred for heavier or more critical payloads.
- CCAFS has most launches, but none at same size than VAFB at maximum.

Success Rate vs. Orbit Type

- ES-L1, GEO, GTO, HSO and SSO have the highest average classification performance.
- LEO, MEO and PO have moderate performance
- ISS and SO have the lowest performance.
- There is a noticeable variability in classification performance across different orbits.

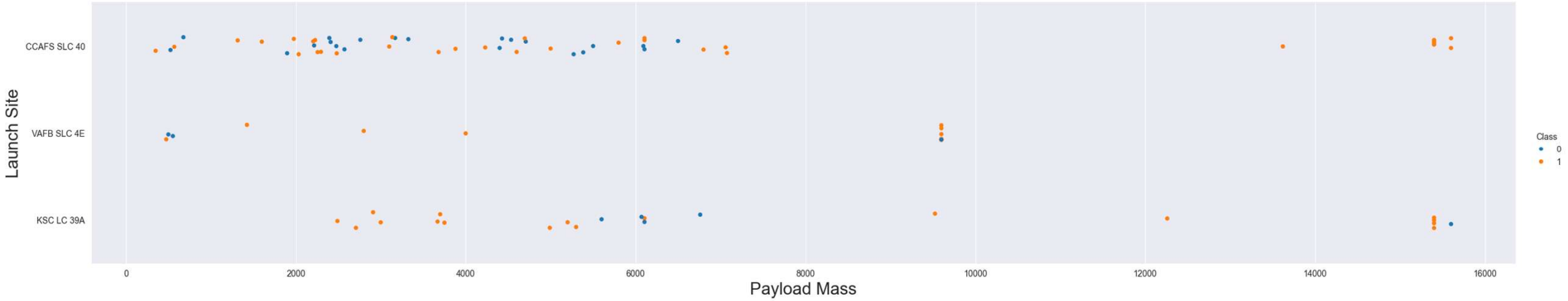


Flight Number vs. Orbit Type



- LEO, ISS, PO, GTO, SSO, and HEO are frequently used across a wide range of flight numbers, indicating their commonality or importance in various missions.
- ES-L1, MEO, and GEO are less frequently used, with data points scattered across fewer flight numbers.
- VLEO and SO are predominantly class 1 in higher flight numbers, which might indicate improvements, specific mission requirements, or operational changes over time.

Payload vs. Orbit Type



- CCAFS SLC 40 is versatile, handling a wide range of payload masses for both classes, indicating its flexibility for various mission types.
- VAFB SLC 4E is used less frequently and primarily for lower payload masses, suggesting it may be specialized for lighter payloads.
- KSC LC 39A is predominantly used for higher payload masses, especially those above 10,000 kg, indicating its capability to handle heavier payloads.

Launch Success Yearly Trend

- From 2010 to 2013, there were no successful launches, indicating initial challenges or a period of development and testing.
- Starting in 2014, there is a noticeable improvement in launch success rates, suggesting advancements in technology, processes, or operational efficiency.
- From 2017 onwards, the success rate significantly increases, peaking in 2019 with a 100% success rate. This indicates a period of high reliability and success in launches.
- While there are slight fluctuations in 2018 and 2020, the overall trend from 2017 onwards remains high, suggesting sustained improvements and consistency in launch success.

All Launch Site Names

- The names of the unique launch sites

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

- The query that has been used is this one:
  %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
  Useful for identifying all unique launch sites in the dataset, which can help in filtering, grouping, or analyzing launch data by site.

Launch Site Names Begin with 'CCA'

- Here are 5 records where launch sites begin with `CCA`

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)

- The query that has been used is this one:
  %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5;
  Useful for quickly inspecting data related to launch sites containing "CCA" (e.g., Cape Canaveral Air Force Station) without retrieving the entire dataset.

Total Payload Mass

- The total payload carried by boosters from NASA is:

Total_Payload_Mass
45596

- The query that has been used is this one:
  %sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
  Useful for analyzing the cumulative payload mass launched for NASA's Commercial Resupply Services (CRS) missions, providing insight into the total cargo or equipment sent to space for these missions.

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1:

Total_Payload_Mass
2928.4

- The query that has been used is this one:
  %sql SELECT AVG("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
  Useful for analyzing the cumulative payload mass launched for F9 v1.1 Booster Version missions, providing insight into the average Payload mass sent to space for these missions.

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad was:

Frist_Success_Ground
2015-12-22

- The query that has been used is this one:

```
%sql SELECT MIN(DATE) AS Frist_Success_Ground FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';
```

Useful for analyzing the dates for all the missions, providing insight using the MIN function for sent to space for these missions and had a outcome with succes.

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:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

- The query that has been used is this one:

  %sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;

  Here the column Landing_Outcome is used to selected the Succes critical together with the Payload_Mass_kg limited between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes:

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

- The query that has been used is this one:

  %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Mission_Outcome;

  The COUNT function is used on column “Mission_Outcome” and it returns the number of launches for each unique launch site and then it is order in groups.

Boosters Carried Maximum Payload

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

- The query that has been used is this one:

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

  It retrieves the distinct booster versions from a table called SPACEXTABLE for the record(s) where the PAYLOAD_MASS__KG_ (payload mass in kilograms) is equal to the maximum payload mass in the same table.

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:

Date	Booster_Version	Launch_Site	substr(Date, 6,2)
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	01
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	04

- The query that has been used is this one:

  %sql SELECT Date,Booster_Version,Launch_Site, substr(Date, 6,2) FROM SPACEXTABLE WHERE substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)';

  This command retrieves specific information from a table called SPACEXTABLE for all missions that happened in 2015 and had a landing outcome of "Failure (drone ship)".

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

- The query that has been used is this one:

  %sql SELECT Landing_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY Total DESC;

  This command counts how many times each landing outcome occurred in the SPACEXTABLE and then sorts the results from most to least common.

Landing_Outcome	Total
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1


Section 3
Launch Sites
Proximities Analysis

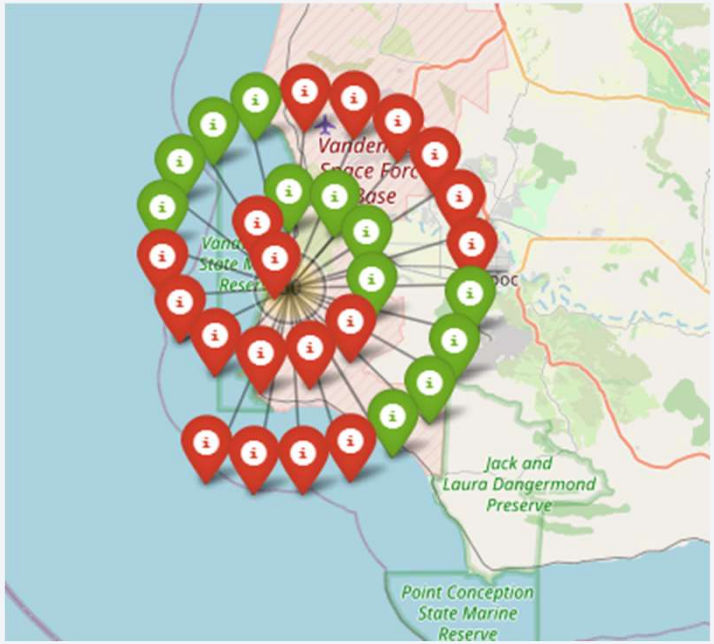
Map show the location for the Launch Sites

- All launch sites on a map

- The four Launch Sites are visible with blue markers on the map. There of
- Three of them are placed in the same area, near Orlando.
  The last one is placed near Los Angeles.
  The little table shows the coordinate.

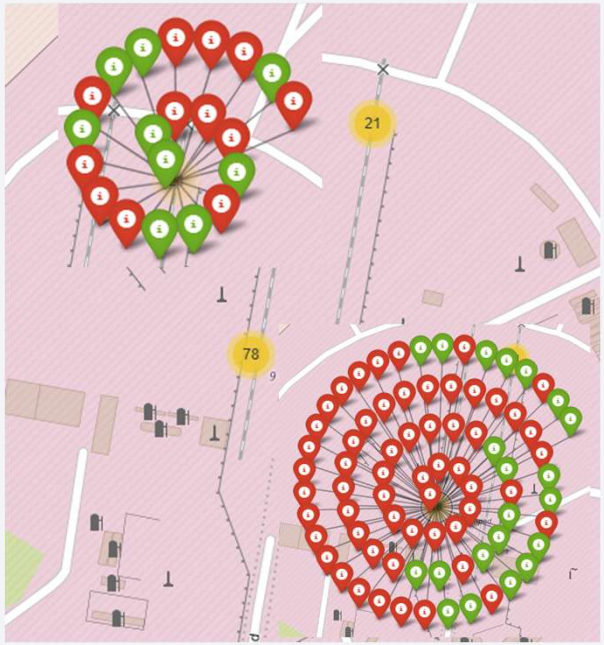
	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

The success/failed launches for each site



VAFB

KSC

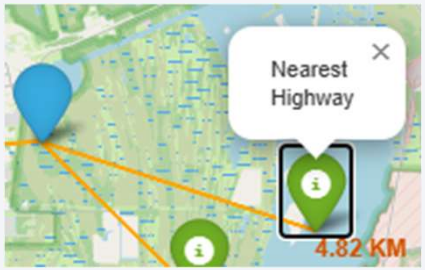


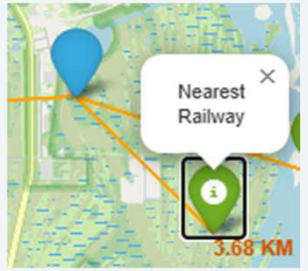
CCAFS LC & CCAFS SLC

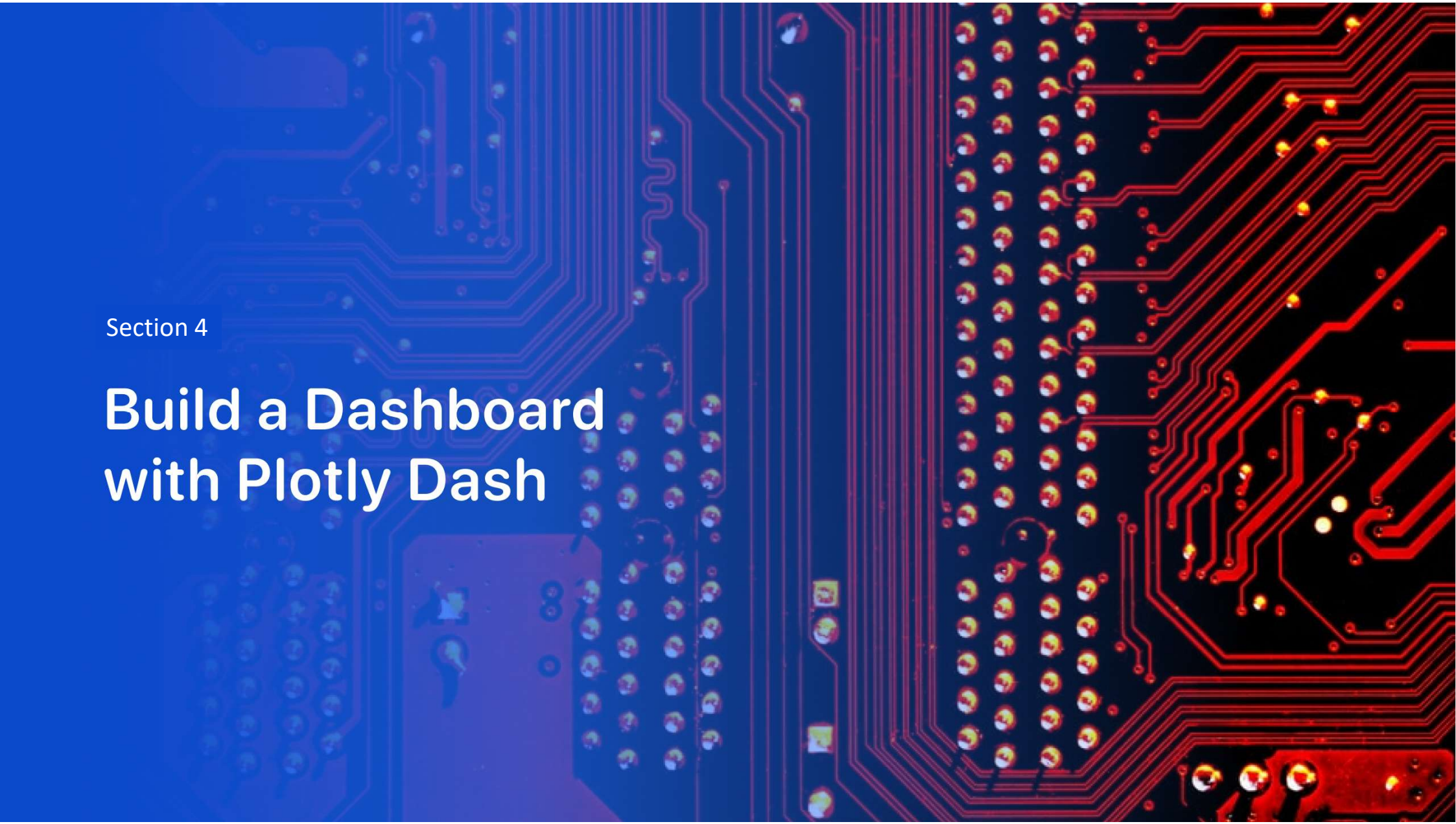
- VAFB has nearly 50/50 success/failed, while KSC has most succuss launches for all four sites. The two CCAF sites seems to be more like test sites with many failed launches.

Distances between a launch site to its proximities

- On the three screenshots, the distance are showed in km to Orlando, above and range 71.76 km, and nearest Highway and Railway.

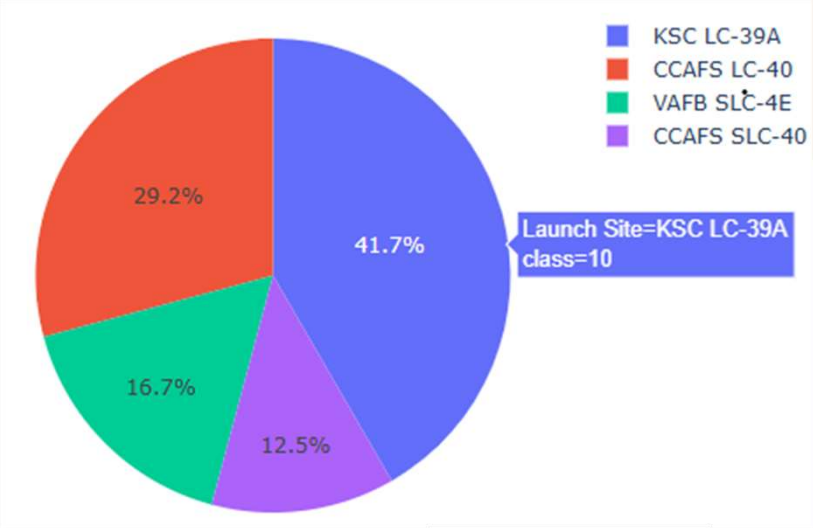


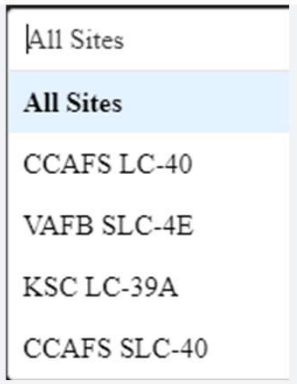



Section 4
Build a Dashboard with Plotly Dash

Launch Site Drop-down

- KSC LC-39A Dominance: The largest portion of the dataset, 41.7%, is attributed to the KSC LC-39A launch site, which is classified as class=10. This indicates that KSC LC-39A is the most frequently used or significant site in this dataset.
- Focus on KSC LC-39A: If this dataset represents launch frequency, resource allocation, or success rates, KSC LC-39A should be a primary focus for analysis, maintenance, or investment.
- Class 10 Significance: The class designation (class=10) for KSC LC-39A might indicate a specific category or priority level, which could be important for further investigation or decision-making.





<Dashboard Screenshot 2>

- 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

Payload and Success for all Sites

- Most successful launches occur with payloads under 5,000 kg.
- B4 and FT booster versions have the highest success rates.
- B4, in particular, is highlighted as successful for a payload of 3,696.55 kg.

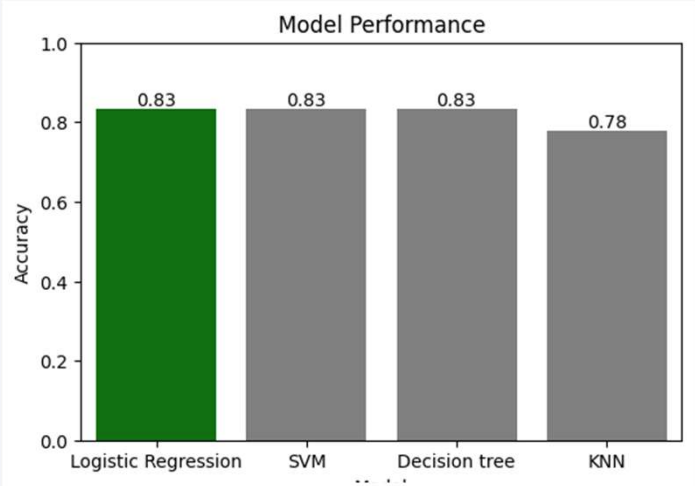
Section 5

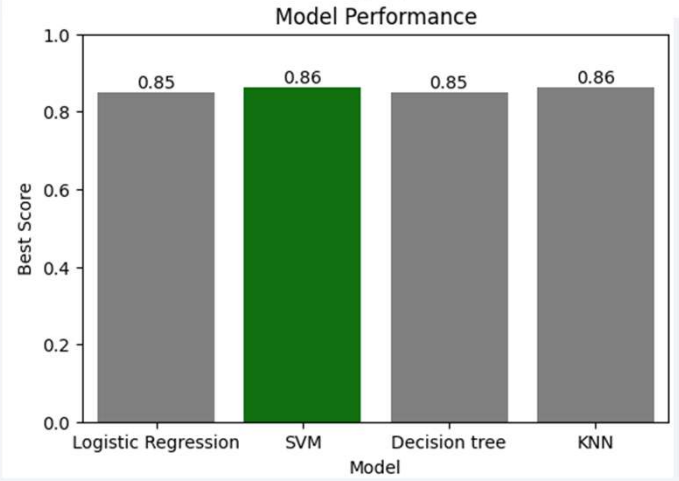
Predictive Analysis (Classification)

Classification Accuracy

Conclusion

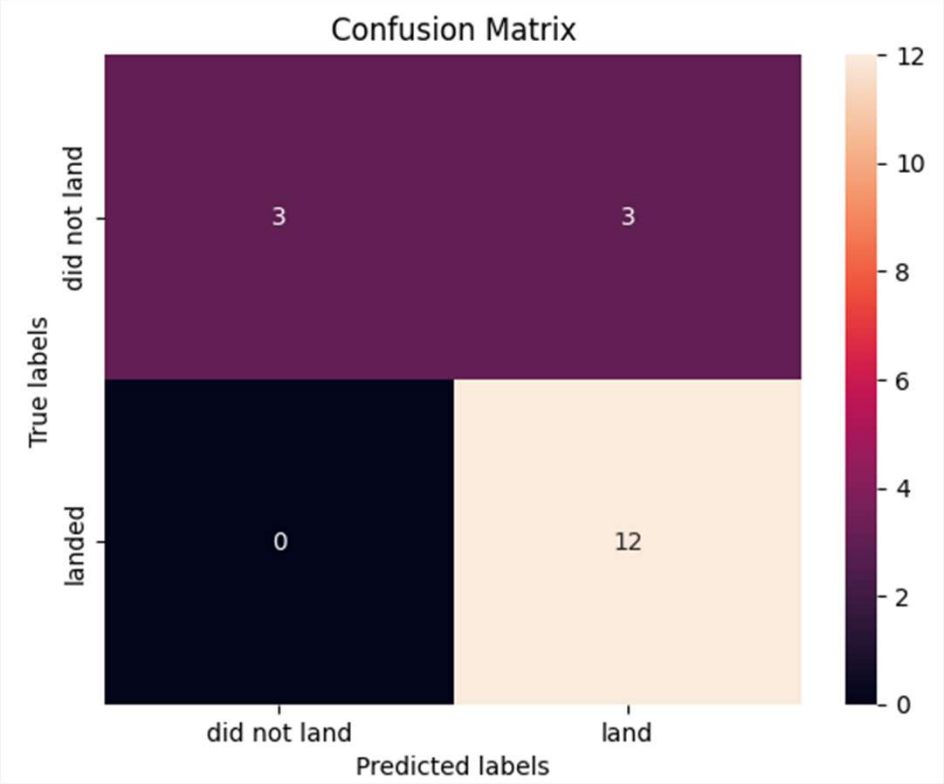
- Logistic Regression, SVM, and Decision Tree are equally effective for this dataset, each achieving an accuracy of 83%.
- KNN is less accurate, with a performance of 78%.
- If model accuracy is the primary criterion, Logistic Regression, SVM, or Decision Tree would be the preferred choices.
- The Bar Chart below is made on Best Score instead and give another result.





Confusion Matrix

- Looking at the Accuracy, the best Method: Logistic Regression with accuracy: 0.8333333333333334
- High Accuracy for "land": The model is very good at identifying successful landings, with 12 true positives and 0 false positives.
- Moderate Performance for "did not land": The model correctly identified 3 true negatives, but it also misclassified 3 instances as "did not land" when they were actually "land" (false negatives).
- No False Positives: The model did not incorrectly predict any "land" outcomes as "did not land," which is a strong point for avoiding false alarms in this context.
- Overall Performance: The model performs well for the "land" class but has room for improvement in correctly identifying the "did not land" class.



Conclusions

- In the last exercise, one can optimize the four models and thereby achieve greater accuracy.
- The first three Logistic Regression, SVM and Decision Tree give almost the same accuracy, whereas KNN is somewhat lower in accuracy when using "Accurate“

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!