IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Danielle Toffanin
May 19, 2025

Table of Contents

Table of Contents

Table of Contents

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Exploratory data analysis (EDA) using visualizations and SQL
  - Interactive visual analytics using Folium and Plotly Dash
  - Predictive analysis using classification models
- Summary of all results
  - Exploratory data analysis (EDA) results
  - Geospatial analytics
  - Interactive dashboard
  - Predictive analysis using classification models

Introduction

SpaceX has gained worldwide attention for a series of historic milestones.

It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62 million dollars. Other providers cost an upwards of $165 million dollars each. With much of the savings due because SpaceX can reuse the first stage.

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident. If we can determine if the first stage will land, we can determine the cost of a launch.

In essence, we will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection
  - Data was collected using REST API and web scraping
- Data wrangling
  - Data was cleaned by removing null values and filling missing values with the mean
  - Data was transformed using one hot encoding to be applied later on with machine learning models
- Exploratory data analysis (EDA) using visualization and SQL
  - Manipulate and evaluate the SpaceX dataset with SQL queries
  - Visualize relationships between variables, find patterns and trends in the data using Pandas, Matplotlib and Seaborn
  - Determine what attributes are correlated with successful landings
- Interactive visual analytics using Folium and Plotly Dash
  - Analyze launch site geospatial data and proximities to points of interest using Folium
  - Create an interactive dashboard application using Plotly Dash
- Predictive analysis using classification models
  - Build a machine learning pipeline to predict if the first stage booster will land successfully

Data Collection

Data collection refers to the process of gathering information from various sources to be used for analysis, modeling, and decision-making.The following steps were taken to collect the SpaceX data:

Import Libraries	Define helper functions	Request data	Parse data	Define lists	Call functions	Create dictionary	Create dataframe	Filter dataframe
• Requests • Pandas • Numpy • Datetime	• getBoosterVersion • getLaunchSite • getPayloadData • getCoreData	Request the launch data by using the .get() request	Convert the JSON objects into a dataframe by using the .json_normalize() function	Define empty lists to be used to create a new data frame	Apply helper functions to retrieve data and fill the lists	Use lists to create a dictionary	Create a dataframe from the dictionary	Filter the dataframe to include only the Falcon 9 launches

Data Collection – SpaceX REST API

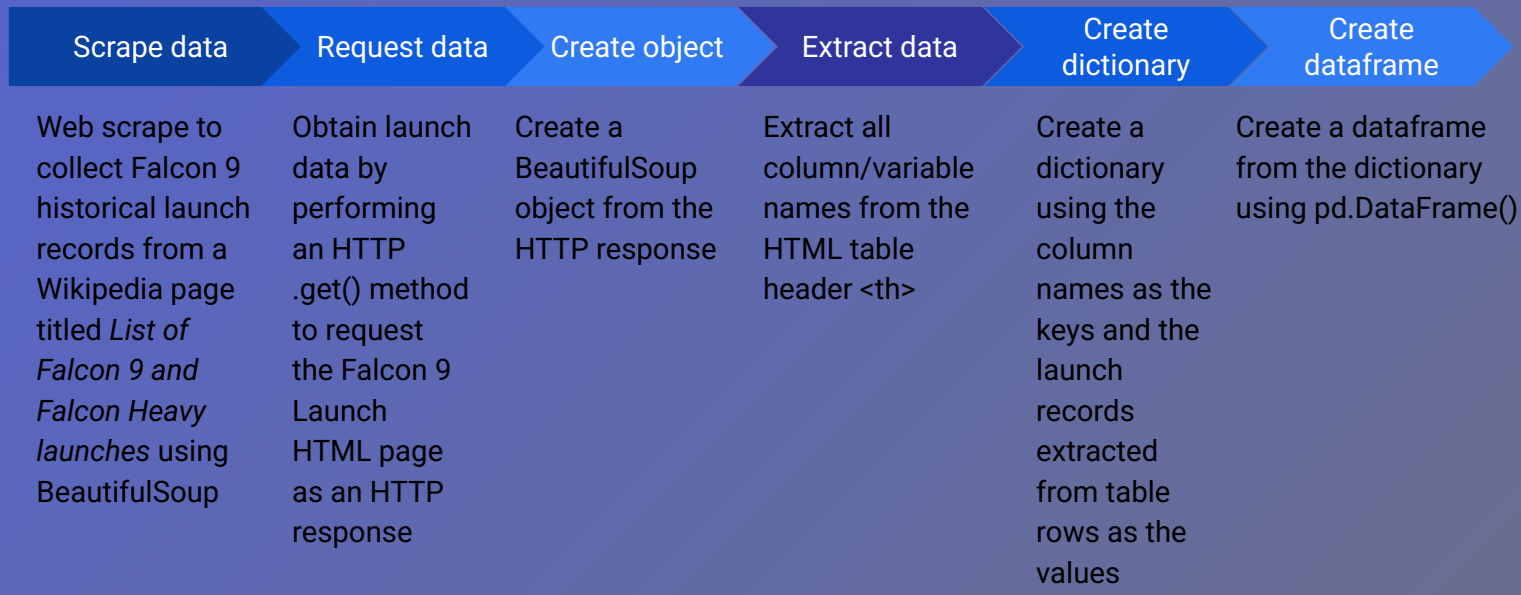
SpaceX launch data is gathered from an API, specifically the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. The following steps were taken to collect the SpaceX data:

Request data	Decode data	Parse data	Preprocess data	Define lists	Call functions	Create dictionary	Create dataframe	Filter dataframe
Request the launch data from the SpaceX REST API by using the .get() request on the URL	Decode the response content as a JSON using .json()	Turn the JSON objects into a dataframe by using the .json_normalize() function	Get information about the launches using the IDs given for each launch to use as columns: ● Rocket ● Payloads ● Launchpad ● Cores ● Flight number ● Date ● Etc.	Define empty lists [ ] to be used to create a new dataframe	Apply helper functions: ● getBoosterVersion ● getLaunchSite ● getPayloadData ● getCoreData to retrieve data and fill the lists	Use lists to create a dictionary using the column names as the keys and the data as the values	Create a dataframe from the dictionary using pd.DataFrame()	Filter the dataframe to include only the Falcon 9 launches

GitHub Link

Data Collection – Web Scraping

Web scraping is the process of extracting data from websites using automated software or scripts. The following steps were taken to collect the SpaceX data:



GitHub Link

Data Wrangling

Data wrangling is the process of transforming raw data into a usable format for analysis. It involves cleaning, structuring, and transforming data to ensure it's accurate, consistent, and ready for analysis.The following steps were taken to wrangle the SpaceX data:

Identify missing values	Determine data types	Missing values	Launch Sites	Orbits	Mission outcomes	One hot encoding
Identify which rows are missing values in the dataset by using .isnull.sum()	Identify which columns are numerical and categorical using .dtypes	● Calculate the mean using .mean() ● Use the mean and the .replace() function to replace np.nan values in the data with the calculated mean	Calculate the number of launches on each site using .value_counts()	Calculate the number and occurrence of each orbit type using .value_counts()	Calculate the number and occurence of mission outcomes per orbit types using .value_counts()	● Convert the categorical data variables from the mission outcomes into binary. ● 0 = unsuccessful first stage landing ●1 = successful first stage landing

GitHub Link

Exploratory Data Analysis (EDA) with Data Visualization

Exploratory Data Analysis (EDA) is a method used to understand the characteristics and structure of a dataset by using visualization and other techniques to identify patterns, anomalies, and potential relationships within the data. It's an iterative process where questions are generated about the data, then visualized, transformed, and modeled, with the results used to refine the initial questions and generate new ones.

Bar Plot

Bar plots are primarily used to compare data across different categories or groups, often showing the frequency or count of occurrences within each category. An example is:

- Success rate vs Orbit Type

Line Plot

Line plots are primarily used to visualize changes or trends in data over time or across different categories. An example is:

- To visualize the launch success yearly trend

Scatter Plot

Scatter plots are primarily used to visualize and analyze the relationship between two variables. Some examples are:

- Flight number vs Pay load mass (Kg)
- Flight number vs Launch Site
- Pay load mass (Kg) vs Launch Site
- Flight number vs Orbit Type
- Pay load mass (Kg) vs Orbit Type

GitHub Link

Exploratory Data Analysis (EDA) with SQL

SQL, or Structured Query Language, is a domain-specific language designed for managing and manipulating data in relational database management systems (RDBMS). It allows users to interact with databases by writing queries to retrieve, insert, update, and delete data. The SQL queries performed on the SpaceX dataset were:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved
6. List the names of the boosters which have successfully landed on a drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failed mission outcomes
8. List all the booster versions that have carried the maximum payload mass
9. List the records which will display the month names, failed landing outcomes on a drone ship, booster versions and launch sites for the months in the year 2015
10. Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates of 2010-06-04 and 2017-03-20, in descending order.

GitHub Link

Build an Interactive Map with Folium

Folium is a Python library used for creating interactive maps, particularly for visualizing geospatial data. The following are the types of map objects created:

- folium.Map() is used to create a map object.
- folium.Circle() is used to create a highlighted circle indicator on a map.
- folium.Popup() is used to create popup labels.
- folium.Marker() is used to create a visual indicator on a map that denotes a point of interest.
- folium.Icon() is used to create an icon on a map.
- folium.PolyLine() is used to create polynomial lines between points.
- MarkerCluster() is used to reduce visual clutter on a map when it contains several markers with very similar or identical coordinates.

GitHub Link

Build a Dashboard with Plotly Dash

Plotly Dash is primarily used for building interactive web applications and dashboards, especially those focused on data visualization and analysis. A dashboard was created with the following:

1. Launch site drop-down list for all launch sites or a specific launch site
2. Pie chart that shows total successful launches for all launch sites or a specific launch site
3. Range slider to select payload in kilograms (Kg) that depicts:
   A) the minimum or starting point, which is set to 0 Kg
   B) the maximum or ending point, which is set to 10,000 Kg
   C) steps, which indicate the slider interval on the slider, with values set at 1,000 Kg
   D) values, which indicate the current selected range.
4. Scatter plot with the x-axis to be the payload and the y-axis to be the launch outcome (i.e., class column: class 1 = successful or class 0 = unsuccessful). In addition, a color-labeled Booster version so that we may observe mission outcomes with different boosters.

GitHub Link

Predictive Analysis (Classification)

Predictive analysis uses statistical models and machine learning algorithms to make predictions about future outcomes based on historical data.The following steps were taken to build, evaluate, improve and determine the best performing classification model:

Model Development

- Load the dataset
- Transform the data (standardize and pre-process)
- Split the data into training and test data sets using train_test_split()
- Decide which type of machine learning algorithm is most appropriate:
  - Logistic Regression
  - Support Vector machines (SVM)
  - Decision Tree Classifier
  - K-nearest neighbors (KNN))
- For each chosen algorithm:
  - Create a GridSearchCV object and a dictionary of parameters
  - Fit the object to the parameters
  - Use the training data set to train the model

Model Evaluation

- For each chosen algorithm use the GridSearchCV object output to:
  - Check which hyperparameters are the best (best_params_)
  - Check the accuracy on the validation data (best_score_)
  - Check the accuracy on the test data (.score)
- Plot and examine the confusion matrix on the test data

Best Classification Model

- Review the accuracy scores for all chosen algorithms
- The model with the highest accuracy score is determined as the best performing model

GitHub Link

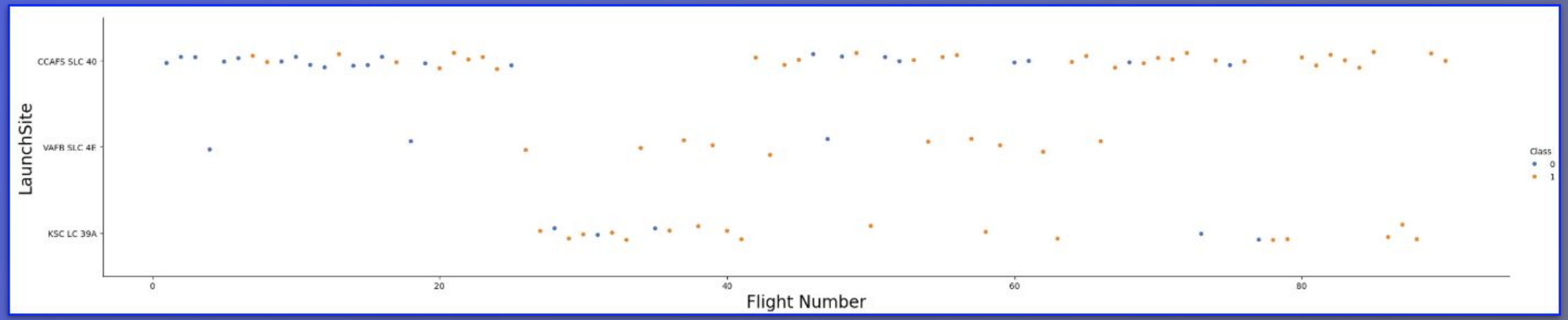
Results

- Exploratory data analysis (EDA) results
    - Both REST API and web scraping are capable of collecting SpaceX data
- Interactive analytics
    - EDA with interactive visualizations provides informative data analysis at a quick glance
    - EDA with SQL is effective for data filtering
    - Plotly Dash is a powerful tool to show instant visual data changes
- Predictive analysis results
    - The Decision Tree Classifier has the best accuracy and best scores of predicting

Section 2

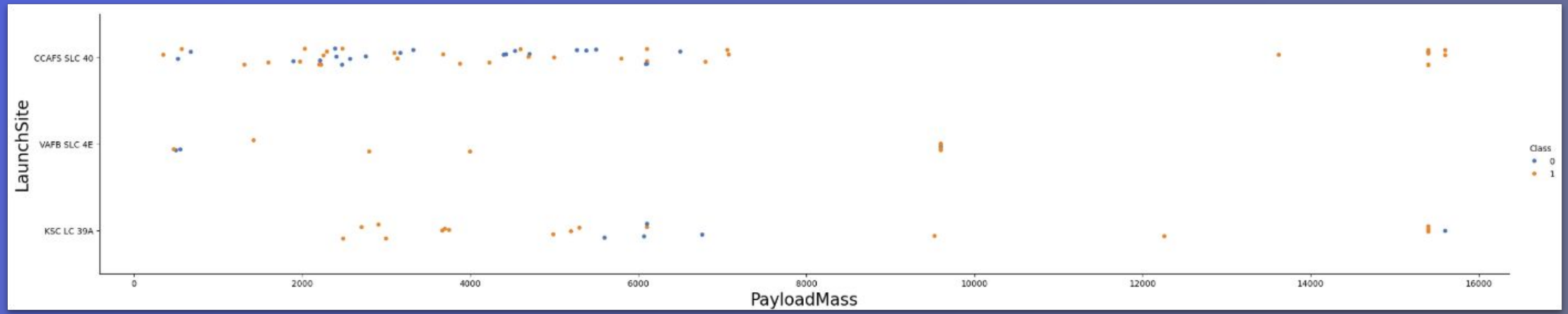
Insights drawn from EDA

Exploratory Data Analysis (EDA) with Data Visualization
Flight Number vs. Launch Site



- This scatter plot shows the relationship between the flight number and launch site.
- Class 0 (blue dot) means the launch outcome was unsuccessful, whereas class 1 (orange dot) means the launch outcome was successful.
- Launch site CCAFS SLC 40, VAFB SLC 4E and KSC LC 39A had 55, 13 and 22 flights, respectively, for a total of 90 flights.
- The majority of the first 25 flights were launched from CCAFS SLC 40 and were mostly unsuccessful.
- The first two flights launched from VAFB SLC 4E were unsuccessful with subsequent flights proving to be successful.
- Flights from KSC LC 39A have been largely successful overall.
- In general, as the number of flights increase, the rate of success increases at each launch site.

Exploratory Data Analysis (EDA) with Data Visualization
Payload vs. Launch Site



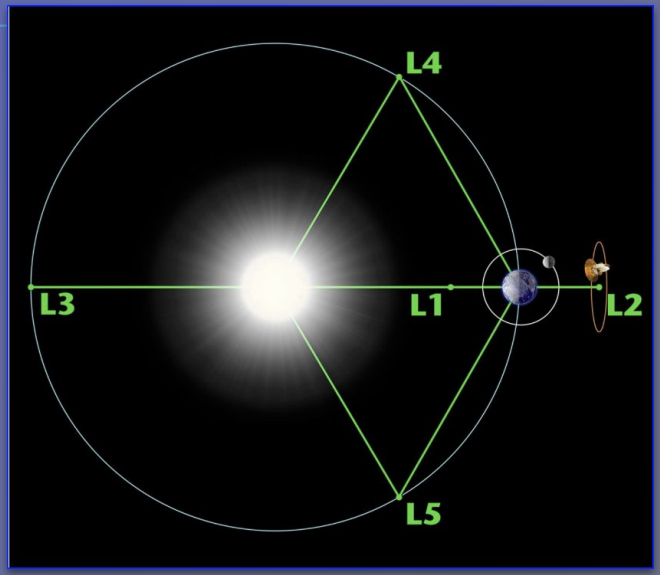
- This scatter plot shows the relationship between the payload mass in Kg and launch site.
- Class 0 (blue dot) means the launch outcome was unsuccessful, whereas class 1 (orange dot) means the launch outcome was successful.
- The flights from CCAFS SLC 40, launched predominately with different payloads ranging from 500 – 7,000 Kg, were equally (un)successful.
- The few flights from VAFB SLC 4E at various payloads were largely successful.
- The flights from KSC LC 39A, launched with various payloads between 2,500 – 15,500 Kg, are were mostly successful.
- There were very few flights that carried over 7,000 Kg. The few that did were mostly successful.
- Overall, there is no clear relationship between payload mass and launch site.

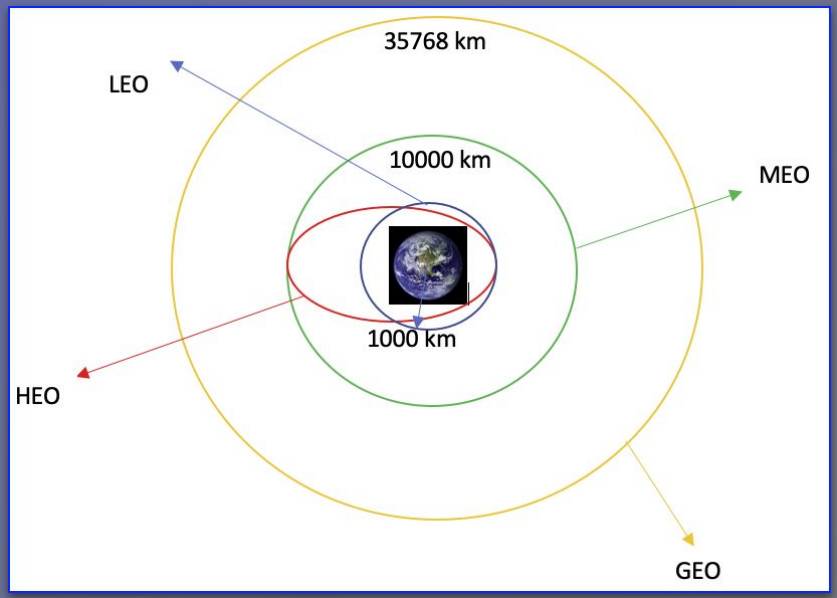
Exploratory Data Analysis (EDA) with Data Visualization

Orbit Type

Each launch aims to an dedicated orbit, and here are some common orbit types:

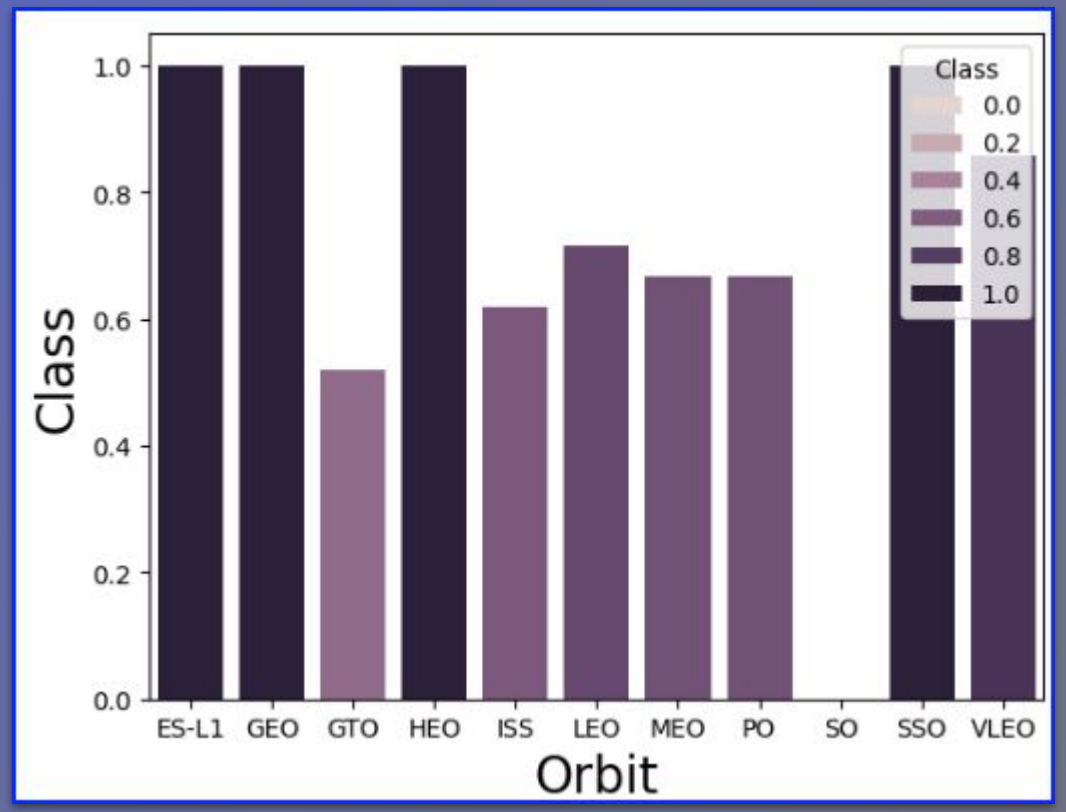
- ES-L1: Earth-Sun Lagrange point 1 (ES-L1). At the Lagrange points the gravitational forces of the two large bodies cancel out in such a way that a small object placed in orbit there is in equilibrium relative to the center of mass of the large bodies. L1 is one such point between the sun and the earth located approximately 1.5 million km (932,000 mi) from Earth.
- GEO: A geostationary orbit, also referred to as a geosynchronous equatorial orbit (GEO) is a circular geosynchronous orbit 35,786 km (22,236 mi) above the Earth's equator following the direction of Earth's rotation.
- GTO: A geosynchronous transfer orbit (GTO) is a high Earth orbit located at 35,786 km (22,236 mi) above Earth's equator.
- HEO: A highly elliptical orbit (HEO), is an elliptic orbit with high eccentricity, usually referring to one around Earth. The point of orbit closest to earth is under 1,000 km (621 mi) and the point of orbit farthest from the earth is 35,786 km (22,236 mi).
- ISS: The International Space Station (ISS) is a modular space station (habitable artificial satellite) in low Earth orbit. It is a multinational collaborative project between five participating space agencies: NASA (United States), Roscosmos (Russia), JAXA (Japan), ESA (Europe), and CSA (Canada).
- LEO: Low Earth orbit (LEO) is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less.
- MEO: Medium earth orbit (MEO). Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 km (22,236 mi). Also known as an intermediate circular orbit.
- PO: A polar orbit (PO) is one type of satellite in which a satellite passes above or nearly above both poles of the body being orbited (usually a planet such as the Earth).
- SSO (or SO): A Sun-synchronous orbit (SSO), also called a heliosynchronous orbit, is a nearly polar orbit around a planet.
- VLEO: Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 450 km.



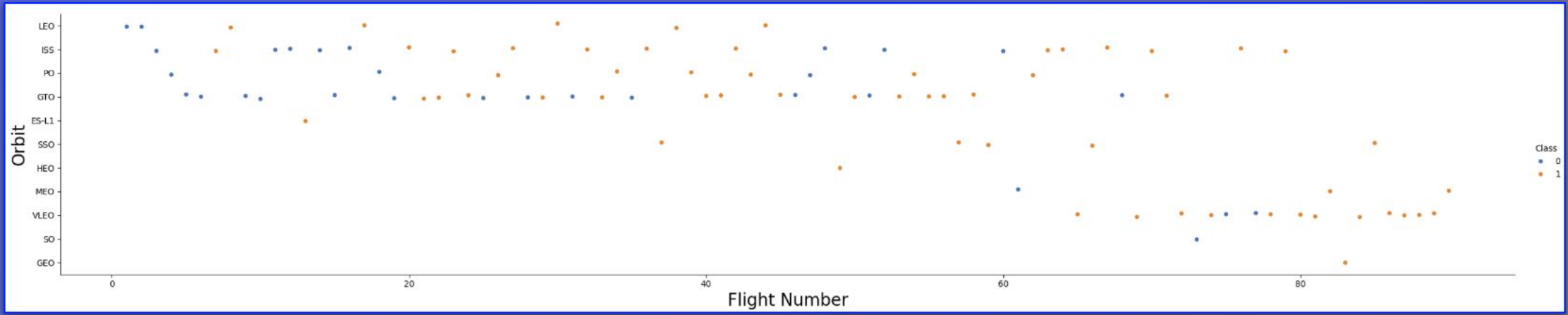


Exploratory Data Analysis (EDA) with Data Visualization
Success Rate vs. Orbit Type

- This bar plot shows the relationship between the success rate and orbit type.
- Class 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0 are the class mean.
- Class 0 means the launch outcome was unsuccessful, whereas class 1 means the launch outcome was successful.
- SSO (sun-synchronous orbit) is technically the same as SO as per the previous slide. There were 5 successful SSO launch outcomes and 1 unsuccessful SO launch outcome for a total of 6. The mean of the SSO and SO launch outcomes should therefore be 0.8.
- The most successful orbit types are:
  - ES-L1 (Earth-Sun Lagrange point 1)
  - GEO (geosynchronous equatorial orbit)
  - HEO (high earth orbit)
- The least successful orbit types are:
  - GTO (geosynchronous transfer orbit)
  - ISS (International Space Station)

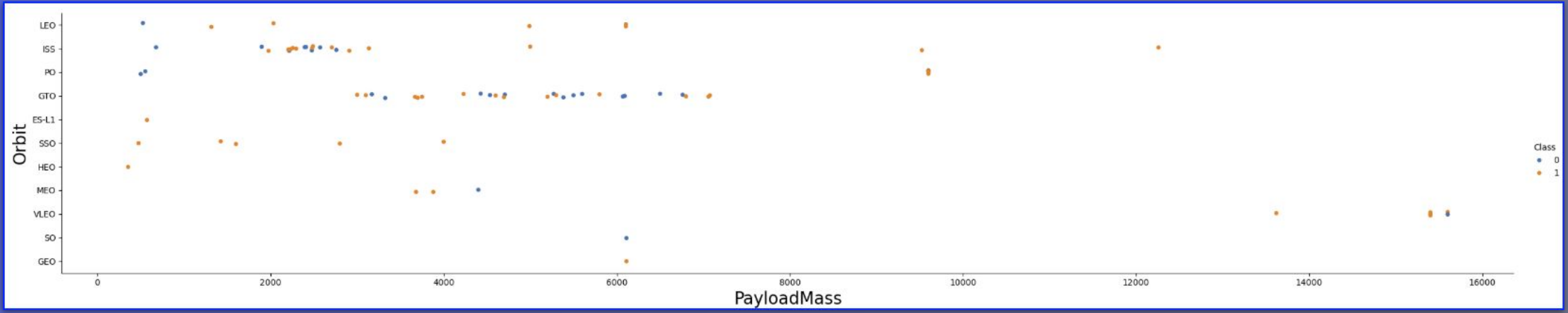


Exploratory Data Analysis (EDA) with Data Visualization
Flight Number vs. Orbit Type



- This scatter plot shows the relationship between the flight number and orbit type.
- Class 0 (blue dot) means the launch outcome was unsuccessful, whereas class 1 (orange dot) means the launch outcome was successful.
- In reference to the prior slide, where we determined ES-L1, GEO and HEO as successful orbit types; all only had 1 flight. Although considered the most successful with a class mean of 1.0, is actually not very good.
- GTO, with the lowest class mean of 0.5, actually had 27 launches, of which, 14 were successful.
- ISS, with the second lowest class mean of 0.6, actually had 21 launches, of which, 13 were successful.
- While there is little relationship between flight number and orbit type, we can see that our interpretation of this scatter plot is more insightful than the previous bar plot.

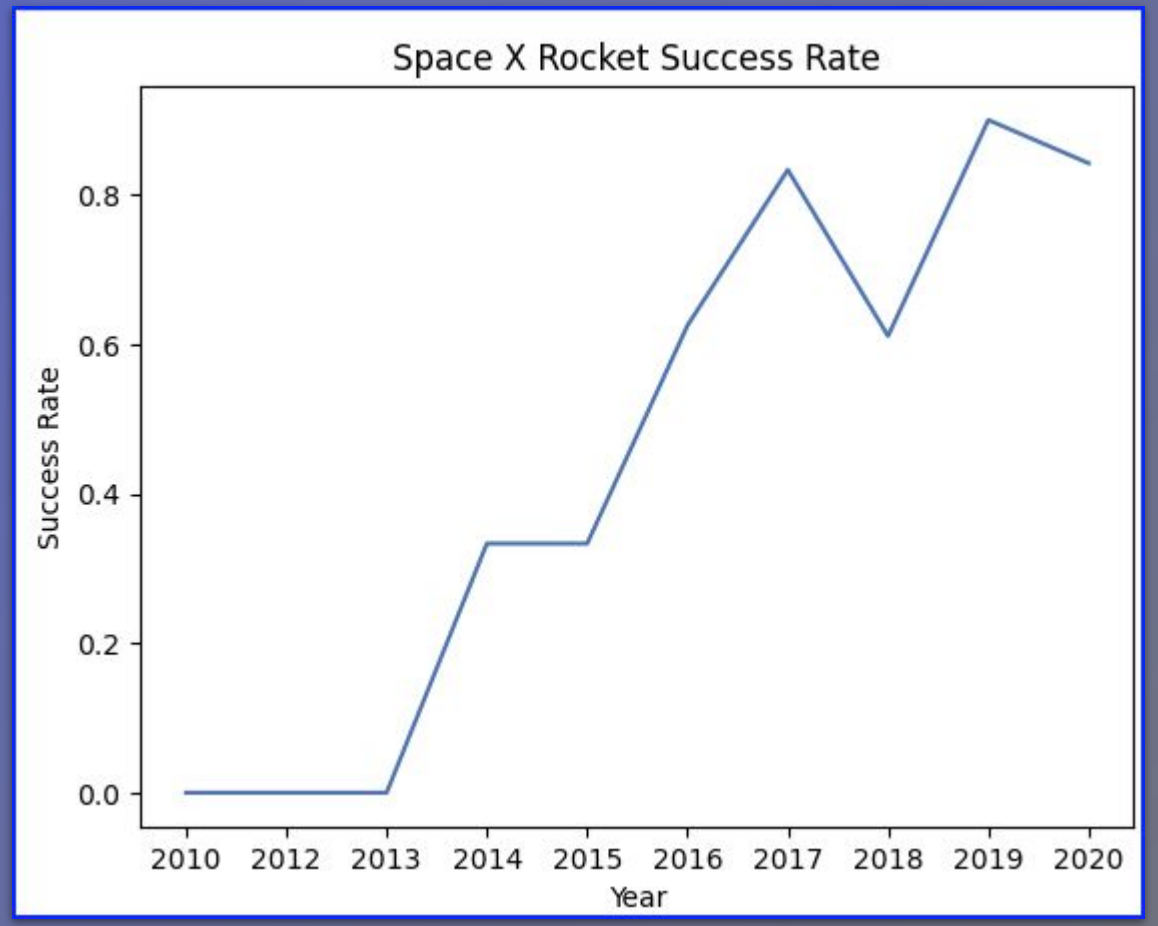
Exploratory Data Analysis (EDA) with Data Visualization
Payload vs. Orbit Type



- This scatter plot shows the relationship between the payload mass in Kg and orbit type.
- Class 0 (blue dot) means the launch outcome was unsuccessful, whereas class 1 (orange dot) means the launch outcome was successful.
- ISS, PO and VLEO orbits were highly successful with payloads over 7,000 Kg, albeit, there were very few flights with payloads over 7,000 Kg.

Exploratory Data Analysis (EDA) with Data Visualization
Launch Success Yearly Trend

- This line plot shows the success rate of landing the Falcon 9 first stage booster across time.
- From 2010-2013, all landing outcomes were unsuccessful.
- Between 2013-2017, the success rate of landing outcomes increased with a small dip in 2018 and 2020.



Exploratory Data Analysis (EDA) with SQL
All Launch Site Names

Display the names of the unique launch sites in the space mission.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

The DISTINCT keyword returns only the unique values from the LAUNCH_SITE column from the SPACEXTABLE

There were 4 unique launch sites in the space mission:

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

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

Exploratory Data Analysis (EDA) with SQL
Launch Site Names Begin with 'CCA'

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

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5
```

SELECT * returns all the data from the SPACEXTABLE database. The WHERE clause limits the query to the Launch_Site column. The LIKE operator is used with the wildcard character '%' after 'CCA' to retrieve all string values beginning with 'CCA'. While the LIMIT clause limits the results to 5 records.

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

Exploratory Data Analysis (EDA) with SQL
Total Payload Mass

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_"), "Customer" FROM SPACEXTABLE WHERE "Customer" LIKE "NASA (CRS)"
```

SELECT returns all the data from columns PAYLOAD_MASS_KG_ and Customer from the SPACEXTABLE database. The WHERE clause limits the query to the Customer column. The LIKE operator is used to retrieve all string values equal to 'NASA (CRS)'. The SUM() function is used to calculate the total payload mass carried by boosters from NASA.

The total payload mass carried by NASA(CRS) boosters was 45,596 Kg.

SUM("PAYLOAD_MASS_KG_")	Customer
45596	NASA (CRS)

Exploratory Data Analysis (EDA) with SQL
Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG("PAYLOAD_MASS__KG_"), "Booster_Version" FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1"
```

SELECT returns all the data from columns PAYLOAD_MASS_KG_ and Booster_Version from the SPACEXTABLE database. The WHERE clause limits the query to the Booster_Version column. The LIKE operator is used to retrieve all string values equal to 'F9 v1.1'. The AVG() function is used to calculate the average payload mass carried by booster version F9 v1.1.

The average payload mass carried by booster version F9 v1.1 was 2,928 Kg.

AVG("PAYLOAD_MASS_KG_")	Booster_Version
2928.4	F9 v1.1

Exploratory Data Analysis (EDA) with SQL
First Successful Ground Landing Date

List the date when the first successful landing outcome on a ground pad was achieved.

```
%sql SELECT MIN("Date"), "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success (ground pad)"
```

SELECT returns all the data from columns Date and Landing_Outcome from the SPACEXTABLE database. The WHERE clause limits the query to the Landing_Outcome column. The LIKE operator is used to retrieve all string values equal to 'Success (ground pad)'. The MIN() function is used to calculate the minimum of the Date column (i.e. the first/oldest date).

The first successful landing outcome on a ground pad was on December 22, 2015.

MIN("Date")	Landing_Outcome
2015-12-22	Success (ground pad)

Exploratory Data Analysis (EDA) with SQL
Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have successfully landed on a drone ship and have payload mass greater than 4000 but less than 6000.

```
%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_", "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

SELECT returns all the data from columns Booster_Version, PAYLOAD_MASS__KG_, and Landing_Outcome from the SPACEXTABLE database. The WHERE clause coupled with the AND operator displays a record if all the conditions are TRUE. The LIKE operator is used to retrieve all string values equal to 'Success (drone ship)'. The BETWEEN operator allows for values between 4000 and 6000 to be selected.

The boosters which have successfully landed on a drone ship with a payload mass between 4,000 and 6,000 Kg were:

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Exploratory Data Analysis (EDA) with SQL
Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failed mission outcomes.

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

SELECT returns all the data from the Mission_Outcome column from the SPACEXTABLE database. The GROUP BY statement groups rows that have the same values into summary rows (i.e., Failure (in flight), Success, Success (payload status unclear)).The COUNT() function returns the number of rows that matches a specified criterion, in this case it's Failure (in flight), Success, Success (payload status unclear).

There was 1 mission failure and 100 successful mission outcomes.

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

Exploratory Data Analysis (EDA) with SQL
Boosters Carried Maximum Payload

List all the booster versions that have carried the maximum payload mass.

```
%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE) ORDER BY "Booster_Version"
```

A subquery is used here. The second SELECT statement within the parenthesis calculates the maximum payload mass, and this value is used in the WHERE clause.

The ORDER BY keyword sorts the Booster_Version in ascending order by default.

There are 12 booster versions that have carried the maximum payload mass of 15,600 Kg.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

Exploratory Data Analysis (EDA) with SQL
2015 Launch Records

List the records which will display the month names, failed landing outcomes on a drone ship, booster versions and launch sites for the months in the year 2015

```
%sql SELECT SUBSTR("Date", 6, 2) AS MONTH, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Failure (drone ship)" AND SUBSTR("Date",0,5)='2015'
```

SELECT returns all the data from the Date, Landing_Outcome, Booster_Version and Launch_Site columns from the SPACEXTABLE database. The WHERE clause coupled with the AND operator displays a record if all the conditions are TRUE. The LIKE operator is used to retrieve all string values equal to ‘Failure (drone ship)’. The SUBSTR() function extracts a substring from a string. The first SUBSTR() starts at position 6 and extracts 2 characters from the Date column. Since date was formatted as YYYY-MM-DD, or 2015-01-10, this extracts MM, or 01. Note the ‘-’ counts as a character, The second SUBSTR() starts at position 0 and extracts 5 characters from the Date column, or YYYY, but must also equal 2015. The AS command is used to rename a column or table with an alias.

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

Exploratory Data Analysis (EDA) with SQL
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 dates of 2010-06-04 and 2017-03-20, in descending order.

```
%sql SELECT "Date", "Landing_Outcome", COUNT("Landing_Outcome") FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT("Landing_Outcome") DESC
```

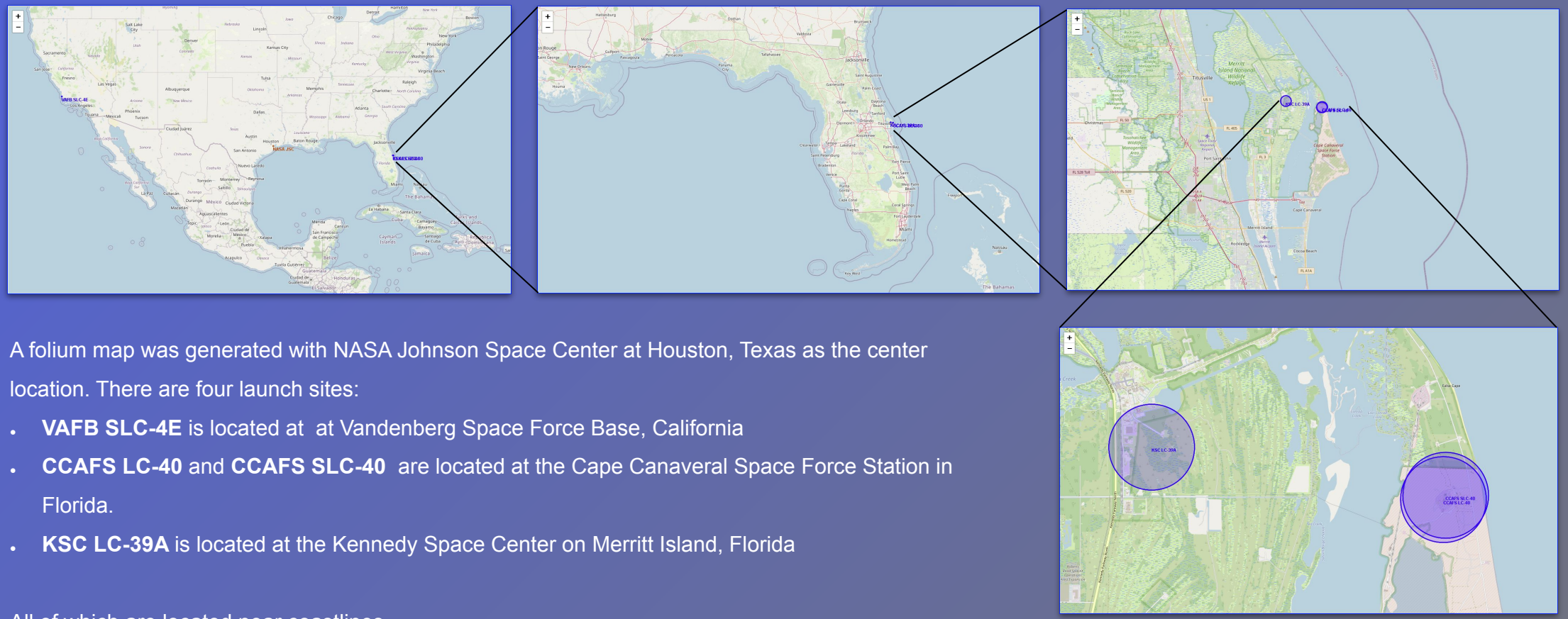
SELECT returns all the data from the Date and Landing_Outcome columns from the SPACEXTABLE database. The WHERE clause coupled with the AND operator displays a record if all the conditions are TRUE. The GROUP BY statement groups rows that have the same values into summary rows (i.e., No attempt, Success (drone ship), Failure (drone ship), Success (ground pad), Controlled (ocean), Uncontrolled (ocean), Failure (parachute), and Precluded (drone ship)). The COUNT() function returns the number of rows that matches a specified criterion. The BETWEEN operator allows for dates between June 4, 2010 and March 20, 2017 to be selected. The ORDER BY COUNT clause sorts the landing outcomes in descending order using the DESC keyword.

Date	Landing_Outcome	COUNT("Landing_Outcome")
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Build an Interactive Map with Folium
All launch sites on a map

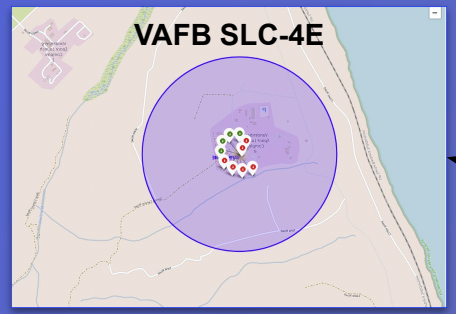


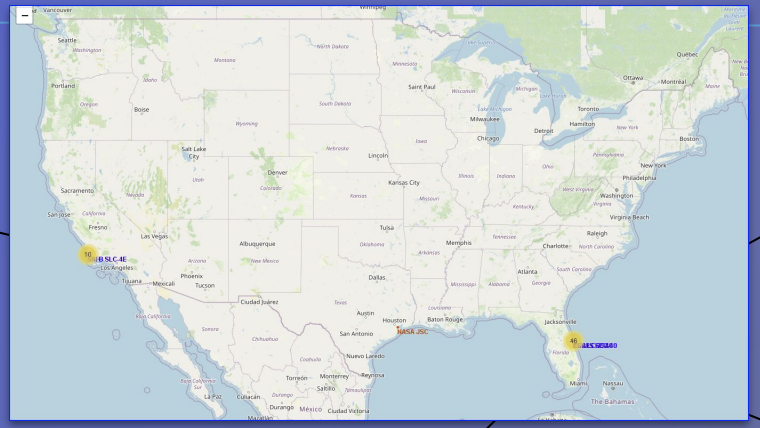
A folium map was generated with NASA Johnson Space Center at Houston, Texas as the center location. There are four launch sites:

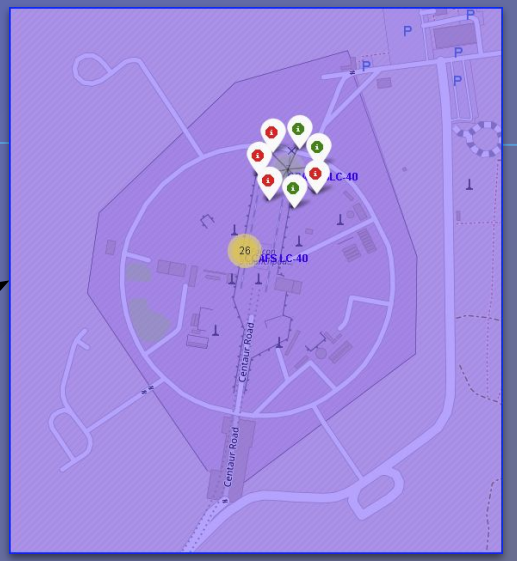
- **VAFB SLC-4E** is located at at Vandenberg Space Force Base, California
- **CCAFS LC-40** and **CCAFS SLC-40** are located at the Cape Canaveral Space Force Station in Florida.
- **KSC LC-39A** is located at the Kennedy Space Center on Merritt Island, Florida

All of which are located near coastlines.

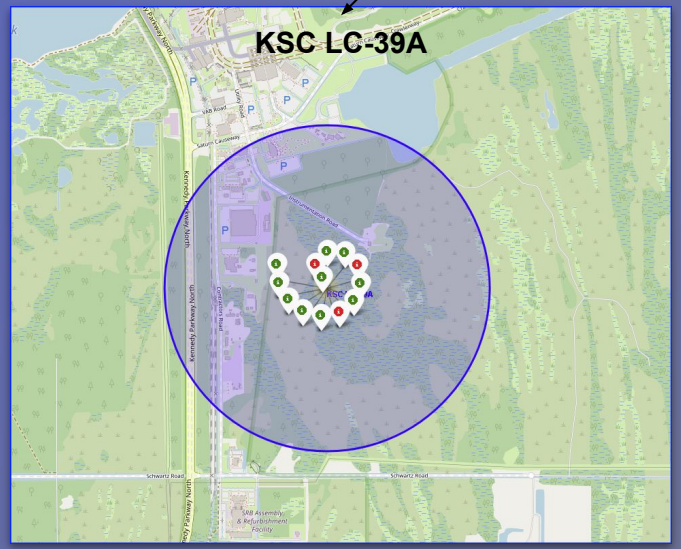
Build an Interactive Map with Folium
Launch outcome clusters

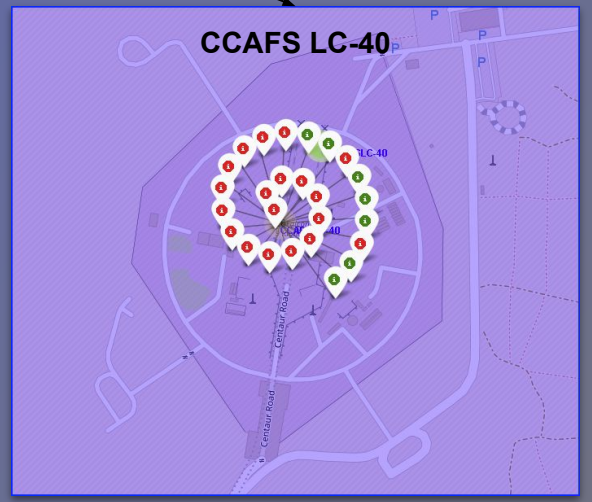






- Launches have been grouped into clusters.
- **Green** markers are for successful launches.
- **Red** markers are for unsuccessful launches.





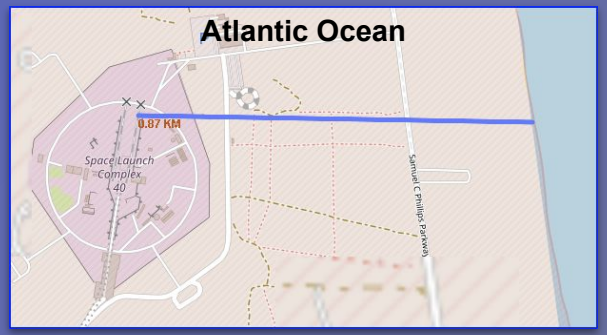
Build an Interactive Map with Folium
Proximity of launch sites to points of interest

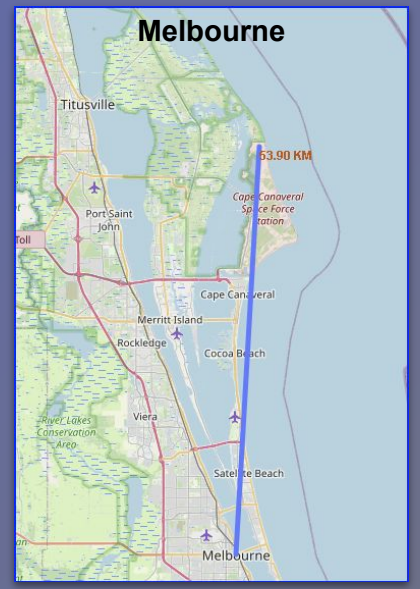
Using launch site CCAFS SLC-40 as an example, we can understand more about the placement of launch sites.

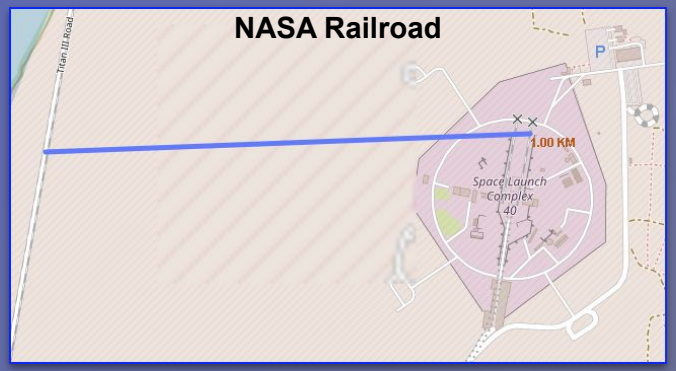
The distance from CCAFS SLC-40 to a point of interest, such as a coastline, railway, highway and city using latitude and longitude coordinates, was calculated.

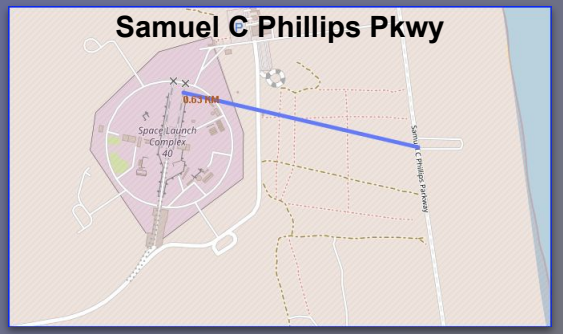
Distance from **CCAFS SLC-40** to:

- **Atlantic Ocean** 0.87 km (0.54 miles)
- **NASA Railroad** 1.00 km (0.62 miles)
- **Samuel C Phillips Pkwy** 0.63 km (0.39 miles)
- **Melbourne** 53.90 km (33.49 miles)









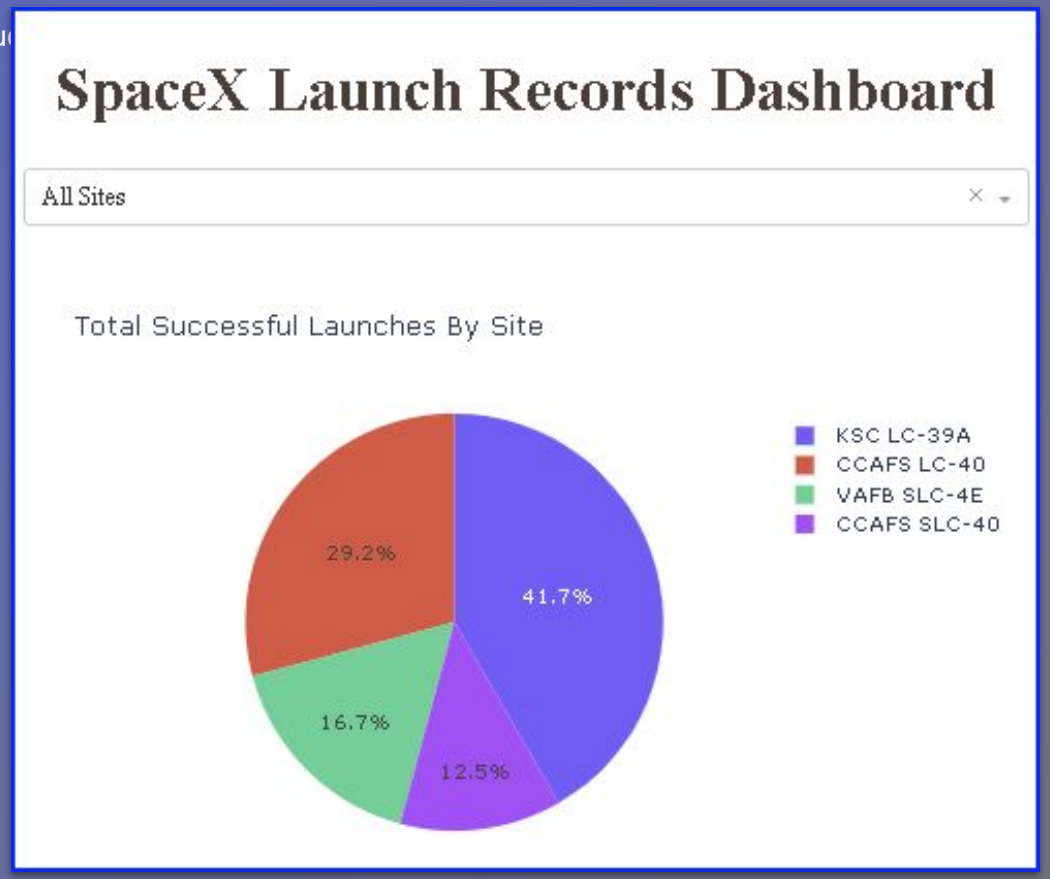
Section 4

Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash
Pie chart of launch success count for all sites

This is a pie chart of the successful launches for all launch sites. There were 24 total su
launches.

- Launch site KSC LC-39A had the most successful launches with 41.7%.
- CCAFS LC-40 came in second with 29.2%.
- Next was VAFB SLC-4E with 16.7%.
- Lastly, CCAFS SLC-40 with 12.5%.

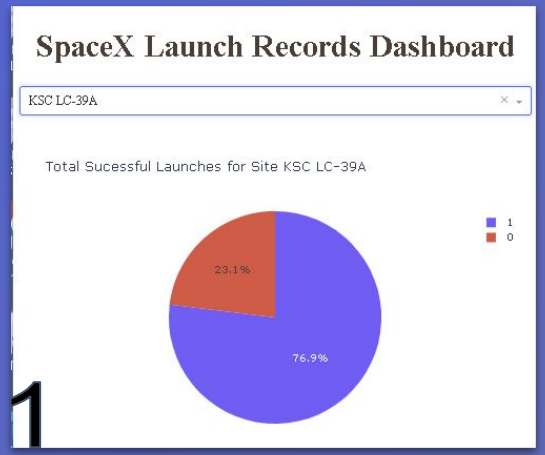


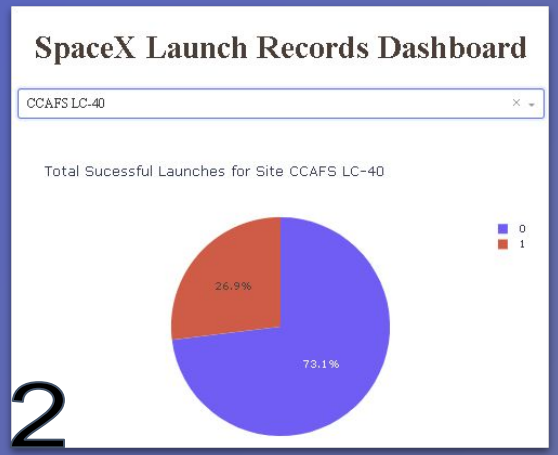
Build a Dashboard with Plotly Dash

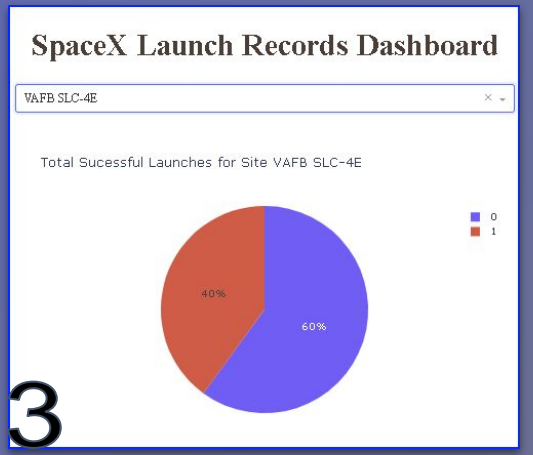
Pie chart for the launch site with the highest launch success ratio

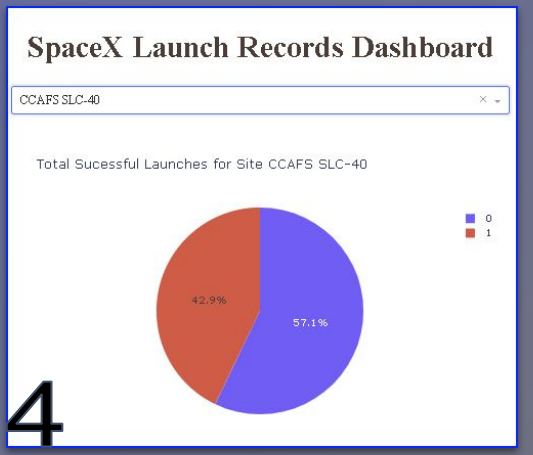
These pie charts show the launch success ratios for all launch site separately.

1. Launch site KSC LC-39A had the highest launch success ratio with 76.9%.
2. Second was CCAFS LC-40 with 73.1%
3. Next was VAFB SLC-4E with 60%
4. Last was CCAFS SLC-40 with 57.1%



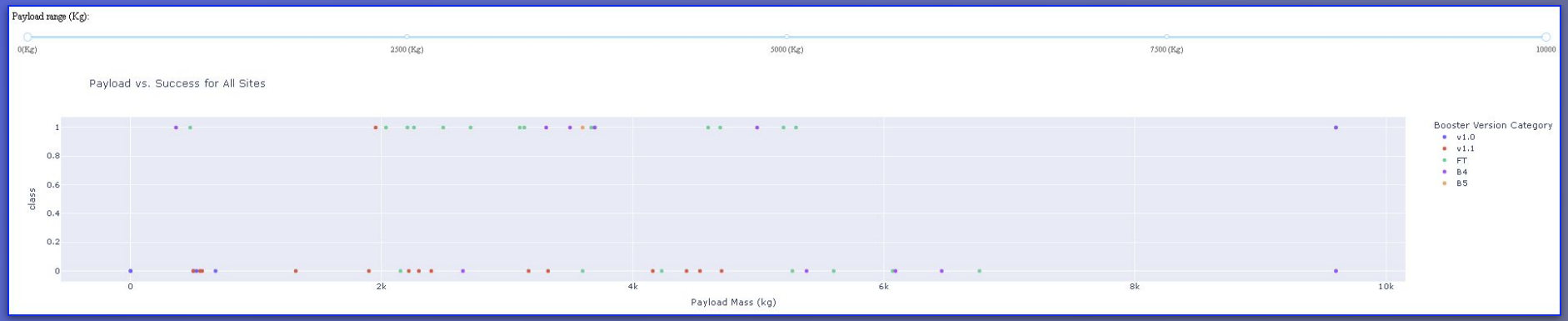




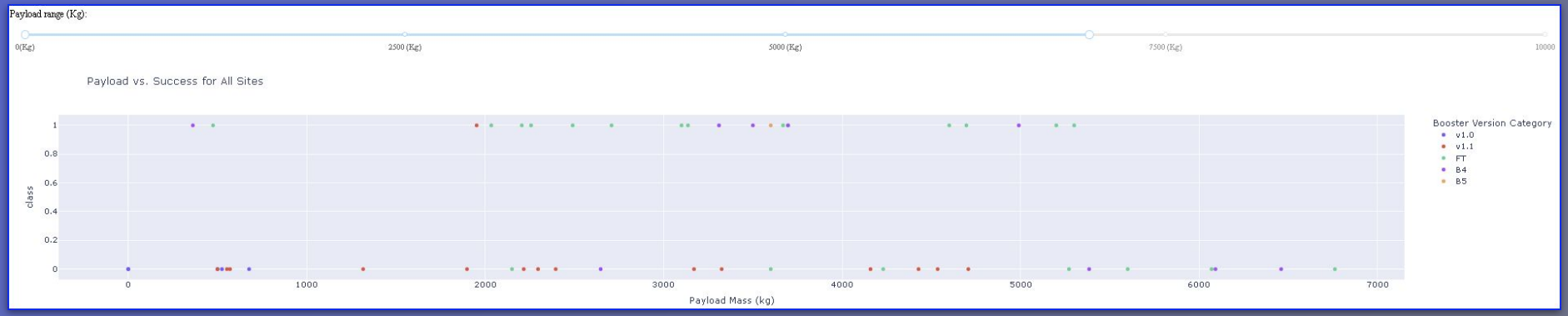


Build a Dashboard with Plotly Dash
Payload vs. Launch Outcome scatter plot and range slider

- These scatter plots show the (un)successful launches by payload and booster version.
- Class 0 means the launch outcome was unsuccessful, whereas class 1 means the launch outcome was successful.
- Per the data, the majority of launches carry payloads under 7,000 kg

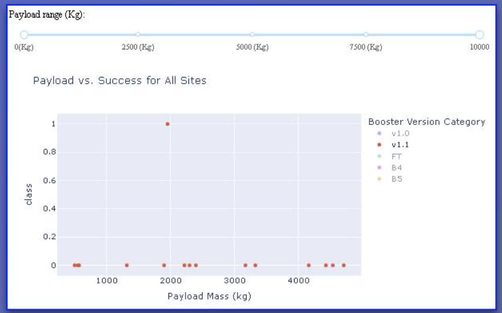


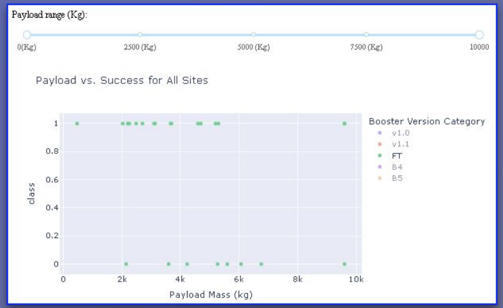
0 Kg - 10,000 Kg



0 Kg - 7,000 Kg

Build a Dashboard with Plotly Dash
Booster Version





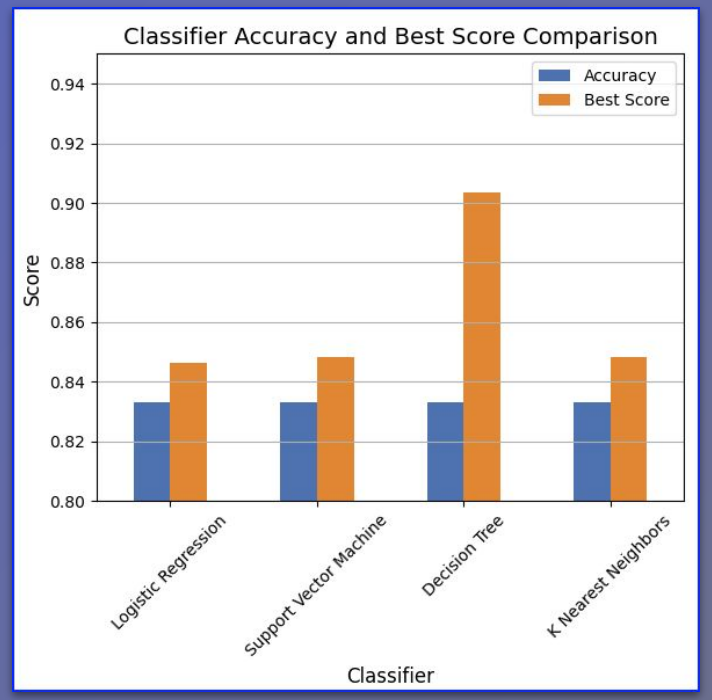
- These scatter plots show the (un)successful launches by payload and booster version.
- Class 0 means the launch outcome was unsuccessful, whereas class 1 means the launch outcome was successful.
- Boosters v1.0 and v1.1 were predominately unsuccessful at all payload ranges.
- Boosters FT and B4 seemed somewhat equally split between successful and unsuccessful launches; having more success with payloads under 7,000 kg .
- The B5 booster succeeded in it’s one and only launch with a 3,600 kg payload.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

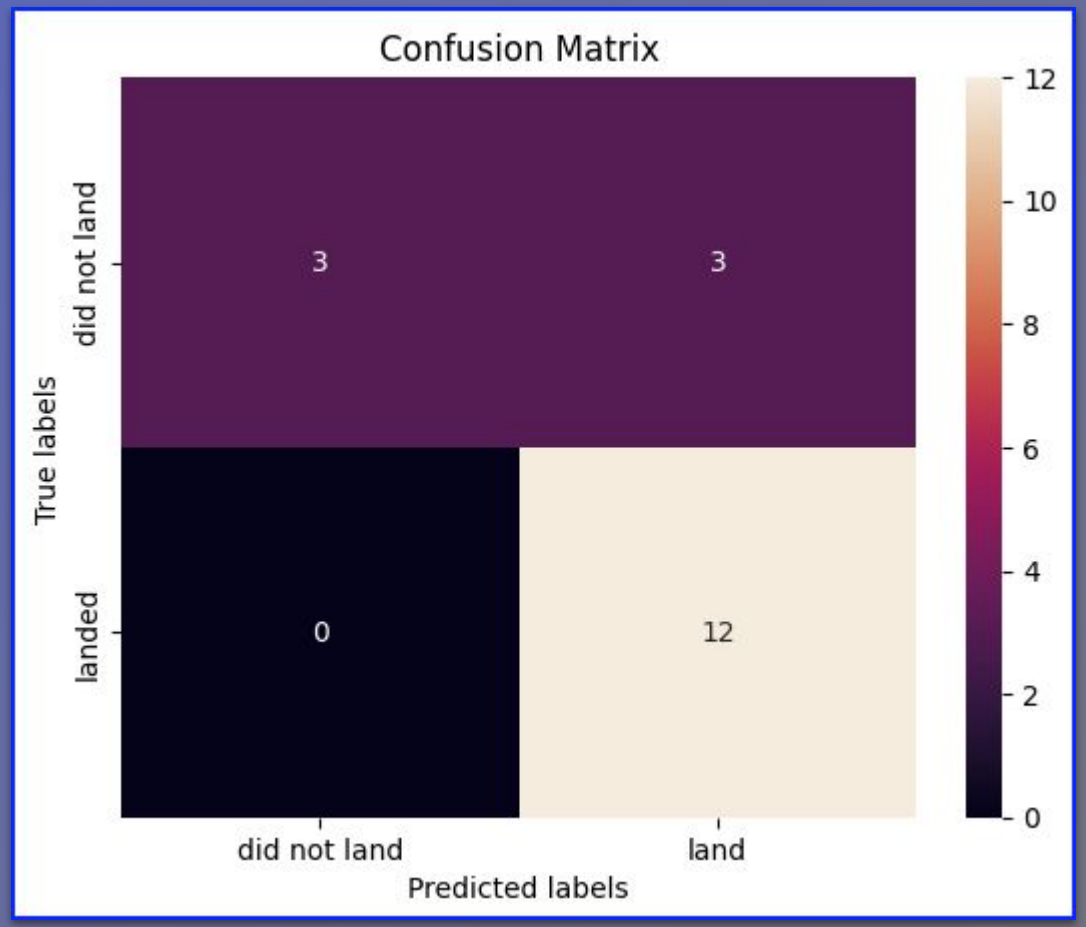
- This bar plot shows the accuracy and best score results for:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K Nearest Neighbors (KNN)
- All 4 classification models have an **accuracy score** of **0.833**
- The classification model with the highest **best score** is the **Decision Tree** classifier at **0.904**



	Accuracy	Best Score
Logistic Regression	0.833333	0.846429
Support Vector Machine	0.833333	0.848214
Decision Tree	0.833333	0.903571
K Nearest Neighbors	0.833333	0.848214

Confusion Matrix

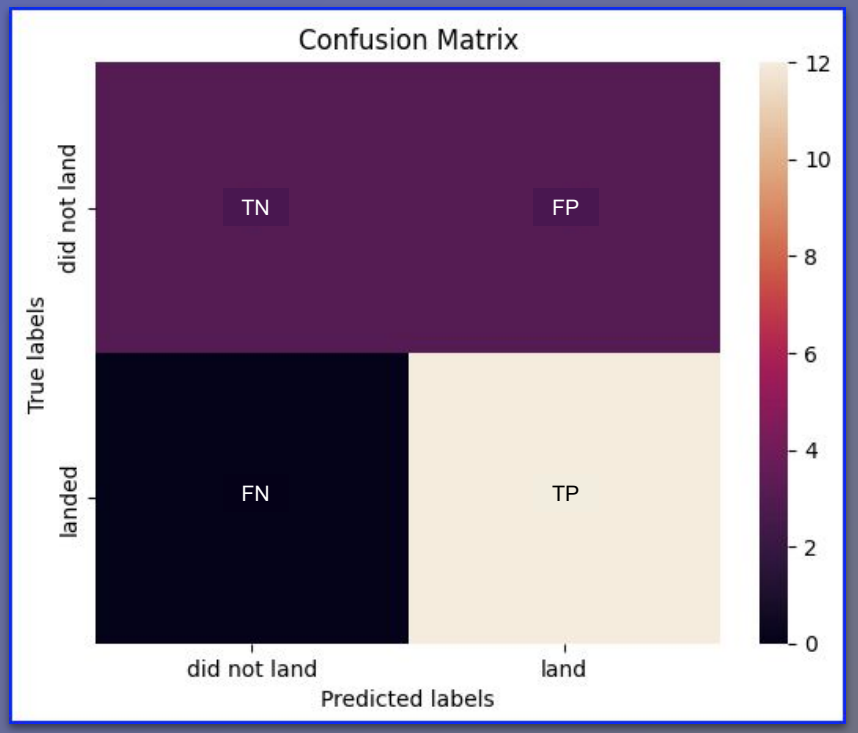
- This is a confusion matrix. It is a table that summarizes the performance of a classification model by comparing its predicted labels to the true labels. It visually represents the different types of predictions the model makes, including correct predictions (true positives and true negatives) and incorrect predictions (false positives and false negatives).
- As previously shown, the best performing classification model is the Decision Tree classifier.
- This shows that:
  - 12 launches were correctly predicted to have successfully landed (true positive)
  - 3 launches were incorrectly predicted to successfully land (false positive)
  - 3 launches were correctly predicted to land unsuccessfully (true negative)
  - There were no false negatives



Confusion Matrix

Metric	Focuses On	Measures...
Precision	Correct Positives	Of the items labeled positive, how many were actually correct?
Recall	Complete Positives	Of all the actual positives, how many did we catch?
F1 Score	Balance	The harmonic mean of Precision & Recall (good when you care about both)
Accuracy	Overall Correctness	How many total predictions were correct (positive and negative)?

- Precision = TP / (TP + FP) = 12 / (12 + 3) = 12 / 15 = **0.80 or 80%**
- Recall (a.k.a. True Positive Rate (TPR)) = TP / (TP + FN) = 12 / (12 + 0) = 12 / 12 = **1.00 or 100%**
- F1-Score = 2 x (Precision x Recall) / (Precision + Recall) = 2 x (0.80 x 1.00) / (0.80 + 1.00) = **0.89 or 89%**
- Accuracy = (TP + TN) / (TP + TN + FP + FN) = (12 + 3) / (12 + 3 + 3 + 0) = **0.83 or 83%**



- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

Conclusions

- As the number of flights increase, the success rate also increases, with most early flights being unsuccessful.
  - From 2010-2013, all landing outcomes were unsuccessful (class = 0).
- Between 2013-2017, the success rate of landing outcomes increased with small dips in 2018 and 2020.
- While it may initially appear that the most successful orbit types were: ES-L1, GEO and HEO, and the least successful orbit types were: GTO and ISS. Upon further analysis, ES-L1, GEO and HEO only appeared to have been so successful because they all only had 1 launch. GTO and ISS were actually much more successful.
- Out of the 4 launch sites, launch site KSC LC-39A had the most successful launches at 41.7%. It also had the highest success ratio at 76.9%.
- The majority of launches carried payloads under 7,000 kg.
- Out of the 5 different boosters, boosters FT and B4 were the most successful
- The best performing classification model was the Decision Tree with an accuracy of 0.8334 and a best score of 0.9036

Appendix

All code and Jupyter Notebooks can be found on my GitHub account.

GitHub repo: https://github.com/dyung0/IBM-Data-Science-Capstone-Project-2025

Thank you!