Winning Space Race with Data Science

Sandip Deb

19th January 2024

Email : debsandip.agt@gmail.com

LinkedIn : www.linkedin.com/in/sandip-deb-22b606b3

Outline

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

Executive Summary

- By following this methodology, we can systematically analyze SpaceX launch data, identify key factors affecting mission success, and derive actionable insights to enhance mission planning and execution.
- In summary, understanding the factors influencing launch success, analyzing launch site performance, payload characteristics, and booster reliability can contribute to informed decision-making and operational improvements for Tesla/SpaceX missions.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We make a get request from SpaceX API https://api.spacexdata.com/v4/
- Perform data wrangling
  - Using helper function we have used the API to extract information using identification numbers.
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - Finding best hyperparameters for SVM, Classification Tree, and Logistic Regression
  - Perform predictions in test date.

Data Collection SpaceX API

- We make a get request to the SpaceX API (https://api.spacexdata.com/v4). We also do some basic data wrangling and formatting. Then Using helper function we have used the API to extract information using identification numbers.

** To make the requested JSON results more consistent, we will use the following static response object for this project:

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

•Request to the SpaceX API

•From the rocket column https://api.spacexdata.com/v4/rockets/ we would like to learn the booster name.

•From the payload Column https://api.spacexdata.com/v4/payloads/ we would like to learn the mass of the payload and the orbit that it is going to.

- From the launchpad column https://api.spacexdata.com/v4/launchpads/ we would like to know the name of the launch site being used, the longitude, and the latitude.

•From cores https://api.spacexdata.com/v4/cores/ we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core which is a number used to separate version of cores, the number of times this specific core has been reused, and the serial of the core.

The data from these requests will be stored in lists and will be used to create a new dataframe and export it to a CSV as dataset_part_1.csv for the next section, but to make the answers consistent.

Data Collection Scraping

Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia

In lab, we performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches.

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Objectives:

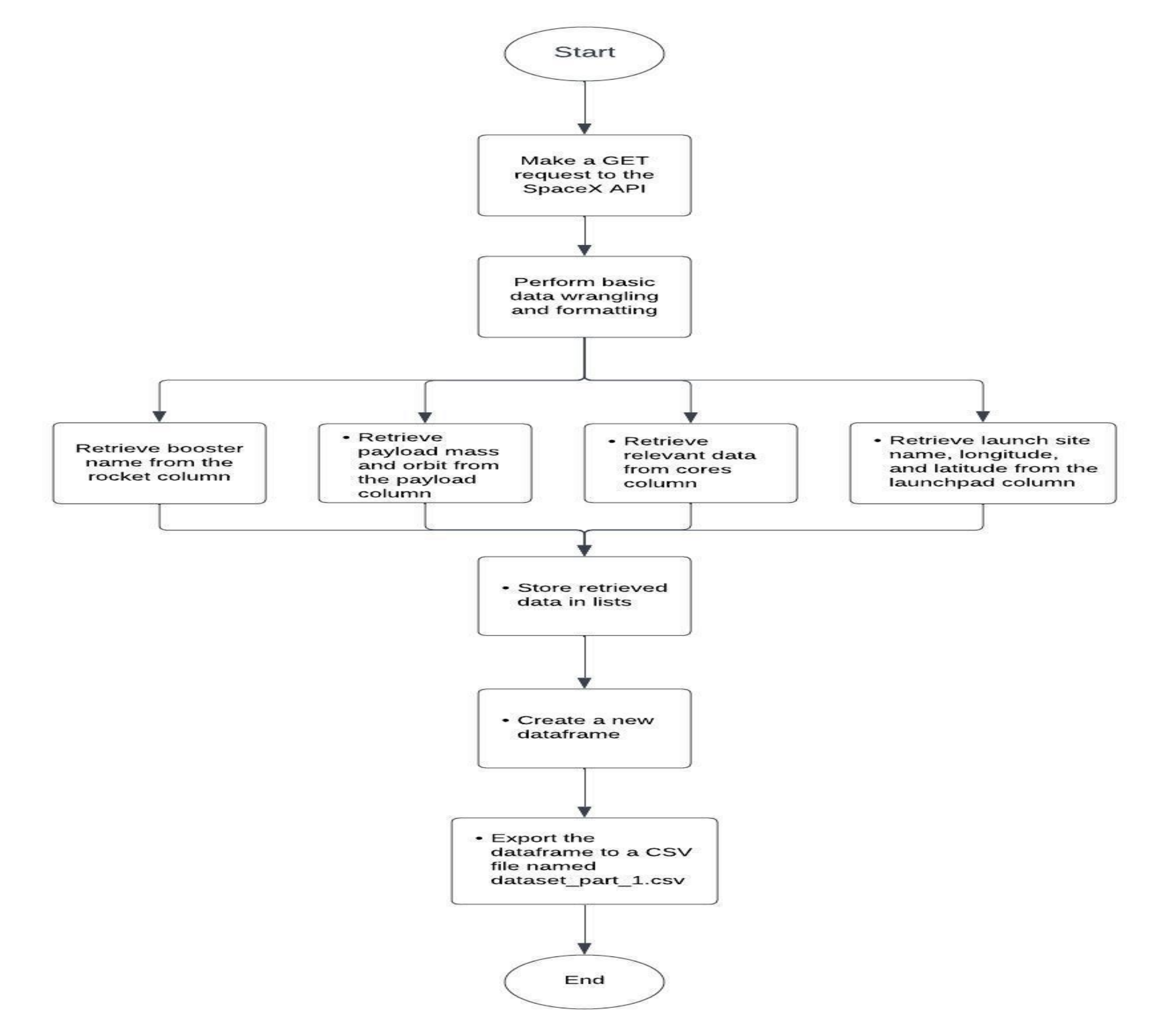
- Web scrap Falcon 9 launch records with BeautifulSoup:
- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame.
- Export data to csv df.to_csv('spacex_web_scraped.csv', index=False)

Data Collection – SpaceX API

- Presenting data collection with SpaceX REST calls using key phrases and flowcharts

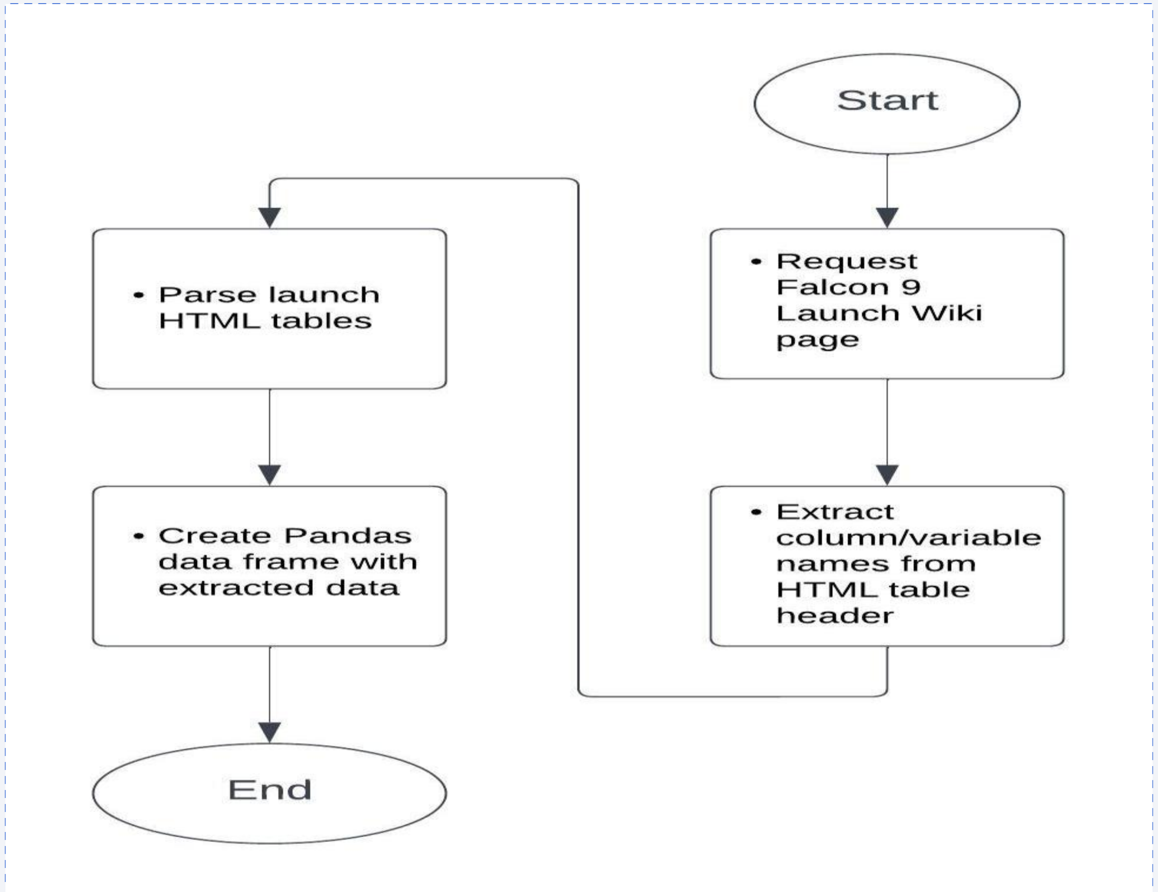
- Added the GitHub URL of the completed SpaceX API calls notebook [L1_jupyter-labs-spacex-data-collection-api.ipynb]

- as an external reference and peer-review purpose



Data Collection - Scraping

- Presenting web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook [L2_jupyter-labs-webscraping.ipynb]
- as an external reference and peer-review purpose



Data Wrangling

- In lab, we determined what would be the label for training supervised models.

- 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; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

Data Wrangling steps

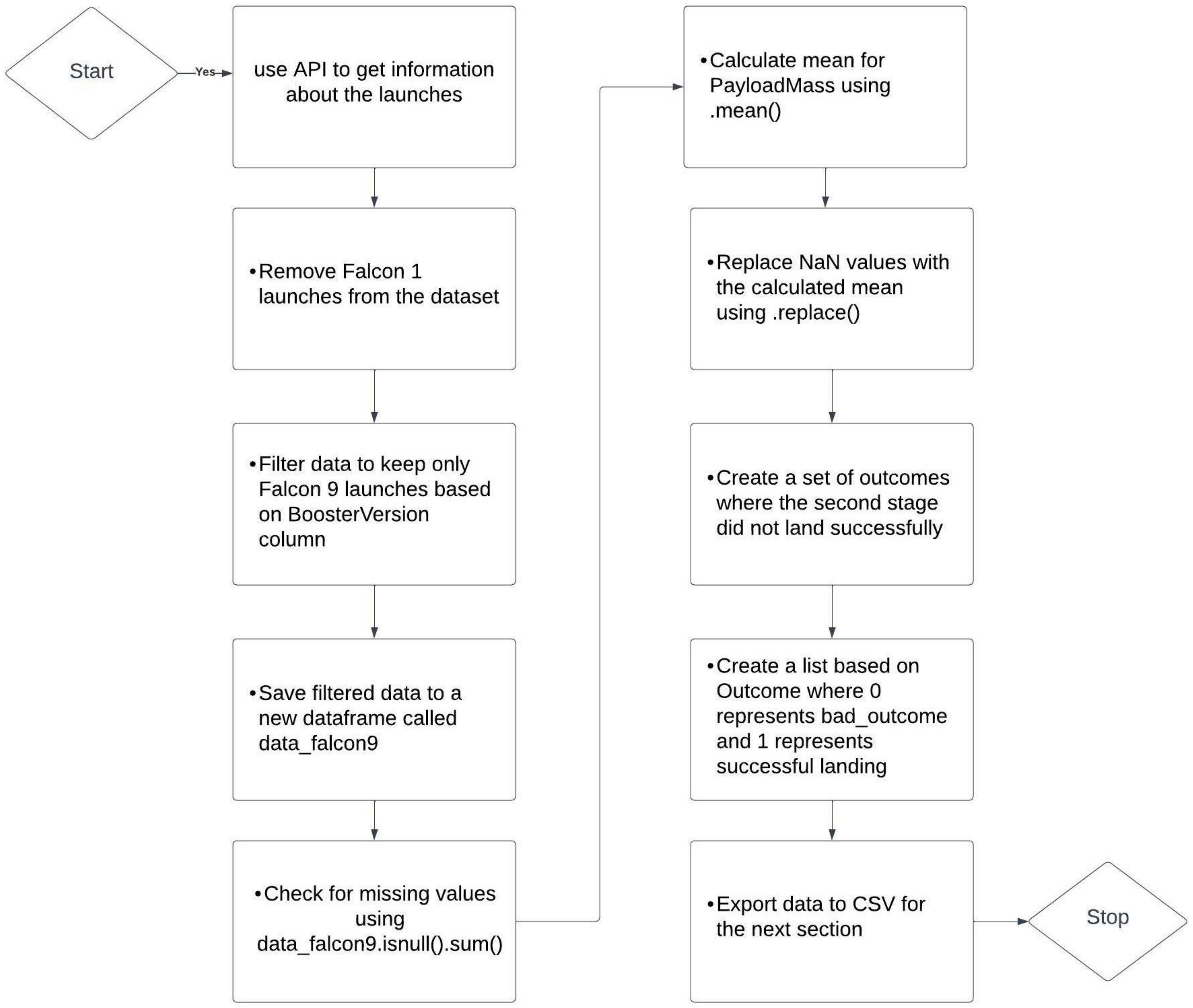
True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed to a drone ship False ASDS means the mission outcome was unsuccessfully landed to a drone ship. None ASDS and None None these represent a failure to land.

- In this lab we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
  - we removed the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the BoosterVersion column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called data_falcon9.

- We have seen that some of the rows are missing values in our dataset using data_falcon9.isnull().sum() command.
- To deal with Missing Values Calculated the mean for the PayloadMass using the .mean(). Then use the mean and the .replace() function to replace np.nan values in the data with the mean you calculated.
- We have created a set of outcomes where the second stage did not land successfully
- Using the Outcome, create a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. Then assign it to the variable landing_class
- This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully
- We export it to a CSV for the next section, but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

Data Wrangling flowchart

- Presentation of data wrangling process using key phrases and flowcharts
- GitHub URL of data wrangling related notebook L3_labs-jupyter-spacex-Data wrangling.ipynb and L1_jupyter-labs-spacex-data-collection-api.ipynb as an external reference and peer-review purpose.



EDA with Data Visualization

- FlightNumber vs. PayloadMass with Launch Outcome Overlay:

  Scatter plot was created to observe the relationship between the flight number and payload mass, overlaying the outcome of the launch. It helps in understanding how the success of the first stage landing relates to the flight number and payload mass.

- FlightNumber vs. LaunchSite with Launch Outcome Color-Coded:

  A categorical plot (catplot) was used to visualize the relationship between the flight number and launch site, with the launch outcome color-coded. This helps in understanding any patterns or trends in the success rates across different launch sites.

- Bar Chart for Success Rate of Each Orbit:

  This bar chart was created to visualize the success rate of each orbit type. It helps in comparing the success rates across different orbits.

- FlightNumber vs. Orbit Type Relationship Visualization:

  This plot was created to observe any relationship between the flight number and orbit type. It helps in understanding if there are any patterns or trends in the choice of orbit over time.

- Payload vs. Orbit Type Relationship Visualization:

  A scatter plot was used to visualize the relationship between the payload and orbit type. It helps in understanding how the payload mass varies across different orbit types.

- Launch Success Yearly Trend Line Chart:

  This line chart was plotted with the year on the x-axis and the average success rate on the y-axis. It helps in understanding the trend of launch success over the years.

** GitHub Link of EDA with data visualization is L5_jupyter-labs-eda-dataviz.ipynb as an external reference purpose

EDA with SQL

- Retrieve the names of the unique launch sites in the space mission using command
  - %sql select distinct(Launch_Site) from SPACEXTABLE
- Retrieve 5 records where launch sites begin with the string 'CCA'
  - %sql select * from spacextable where Launch_Site like 'CCA%' limit 5
- Retrieve the total payload mass carried by boosters launched by NASA (CRS)
  - %sql select sum(PAYLOAD_MASS__KG_) as payload_mass_carried_by_boosters from SPACEXTABLE where Customer = 'NASA (CRS)’
- Retrieve average payload mass carried by booster version F9 v1.1
  - %sql select avg(PAYLOAD_MASS__KG_) as 'payload_mass_carried_by_booster_version_F9_v1.1' from SPACEXTABLE where Booster_Version = 'F9 v1.1'

- Lists the date when the first succesful landing outcome in ground pad was achieved
  - %sql SELECT min(Date) as first_successful_landing_in_ground_pad FROM SPACEXTABLE where Landing_Outcome = 'Success (ground pad)';
- Lists the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
    - %%sql

      SELECT Booster_Version FROM SPACEXTABLE

      WHERE Landing_Outcome = 'Success (drone ship)'

      AND PAYLOAD_MASS__KG_ > 4000

      AND PAYLOAD_MASS__KG_ < 6000;

- Lists the total number of successful and failure mission outcomes

```
%%sql
select Landing_Outcome, count(*) as Count
from spacextable
where Landing_Outcome like 'Success%'
    or Landing_Outcome like 'Failure%'
group by Landing_Outcome
```

- Lists the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_
from spacextable
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable);
```

- Lists the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

  Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

  %%sql

  select Date,substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site

  from spacextable

  where Landing_Outcome = 'Failure (drone ship)'

  and substr(Date,0,5)='2015'

***Github URL of* EDA with SQL, [L4_jupyter-labs-eda-sql-coursera_sqllite.ipynb]

As an external reference.

Build an Interactive Map with Folium

- We have created a folium Map object, with an initial center location(29.559684888503615, -95.0830971930759)to be NASA Johnson Space Center at Houston, Texas.
- Created and added folium.Circle and folium.Marker for each launch site on the site map.
- To enhance the map by adding the launch outcomes for each site, and see which sites have high success rates, which indicates if this launch was successful or not.
- For each launch outcome result in spacex_df data frame, added a folium.Marker to marker_cluster so that from the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

- We have added a MousePosition on the map to get coordinate for a mouse over a point on the map. As such, while we are exploring the map, we can easily find the coordinates of any points of interests (such as railway).
- We calculate the distance between two points on the map based on their Lat and Long values and mark down a point on the closest coastline using MousePosition and calculates the distance between the coastline point and the launch site.
- After obtained its coordinate, create a folium.Marker to show the distance.
- Created a `folium.PolyLine` object using the coastline coordinates and launch site coordinate and similarly we have drawn a line between a launch site to its closest city, railway, highway, etc. You need to use MousePosition to find their coordinates on the map.

** GitHub URL of interactive map with Folium map as an external reference and peer-review purpose is [L6_lab_jupyter_launch_site_location.ipynb]()

Build a Dashboard with Plotly Dash

- We built a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time. This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart. We build this dashboard application via the following tasks.
  - We have four different launch sites, and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate (class=0 vs. class=1). As such, we have created a dropdown menu to let us select different launch sites.
  - Dash callback function is a type of Python function which will be automatically called by Dash whenever receiving an input component updates, such as a click or dropdown selecting event. So, we have added a callback function in spacex_dash_app.py
  - We wants to find if variable payload is correlated to mission outcome. From a dashboard point of view, we want to be able to easily select different payload range and see if we can identify some visual patterns. So, we have Added a Range Slider to Select Payload.

**** GitHub URL of completed Plotly Dash lab [L6_lab_jupyter_launch_site_location.ipynb](), as an external reference and peer-review purpose**

Predictive Analysis (Classification)

To summarize the process of building, evaluating, improving, and finding the best performing classification model:

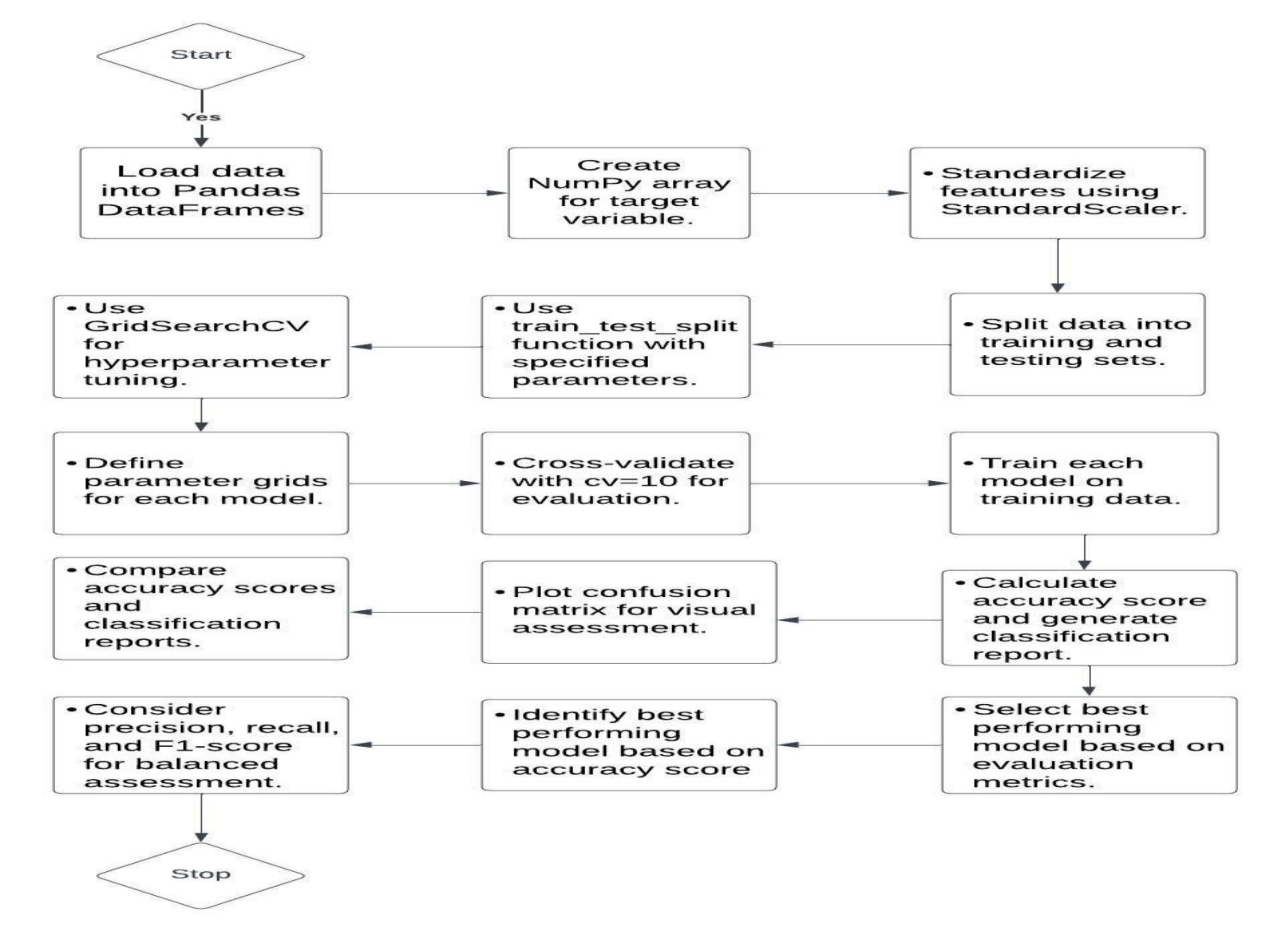
- Data Preprocessing:
  - Loaded the data into Pandas DataFrames.
  - Created a NumPy array for the target variable ('Class') and standardized the features using StandardScaler.
- Splitting Data:
  - Split the data into training and testing sets using train_test_split with a test size of 0.2 and a random state of 2.
- Model Selection and Hyperparameter Tuning:
  - Used GridSearchCV to perform hyperparameter tuning for Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) classifiers.
  - Defined parameter grids for each model to search through different hyperparameters.
  - Used cross-validation with cv=10 to evaluate each model's performance on the training data.

- Model Training and Evaluation:
    - Trained each model on the training data.
    - Calculated the accuracy score and generated classification reports for each model on the test data.
    - Plotted confusion matrices to visually assess each model's performance in predicting successful and unsuccessful landings.
- Model Comparison and Selection:
    - Compared the accuracy scores and classification reports of all models.
    - Based on the accuracy scores and other evaluation metrics such as precision, recall, and F1-score, selected the best performing model.
- Best Performing Model:
    - Decision Tree Classifier Identified as best performing model based on the accuracy score 0.8714 on test data and other relevant evaluation metrics.

**Overall, the process involved iterative steps of training, evaluation, and fine-tuning models to optimize their performance on the given dataset. The selected model can then be used for future predictions or further analysis as needed.

Presentation of model development process using key phrases and flowchart

GitHub URL of predictive analysis lab, as an external reference and peer-review purpose is [L8_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb]



Results

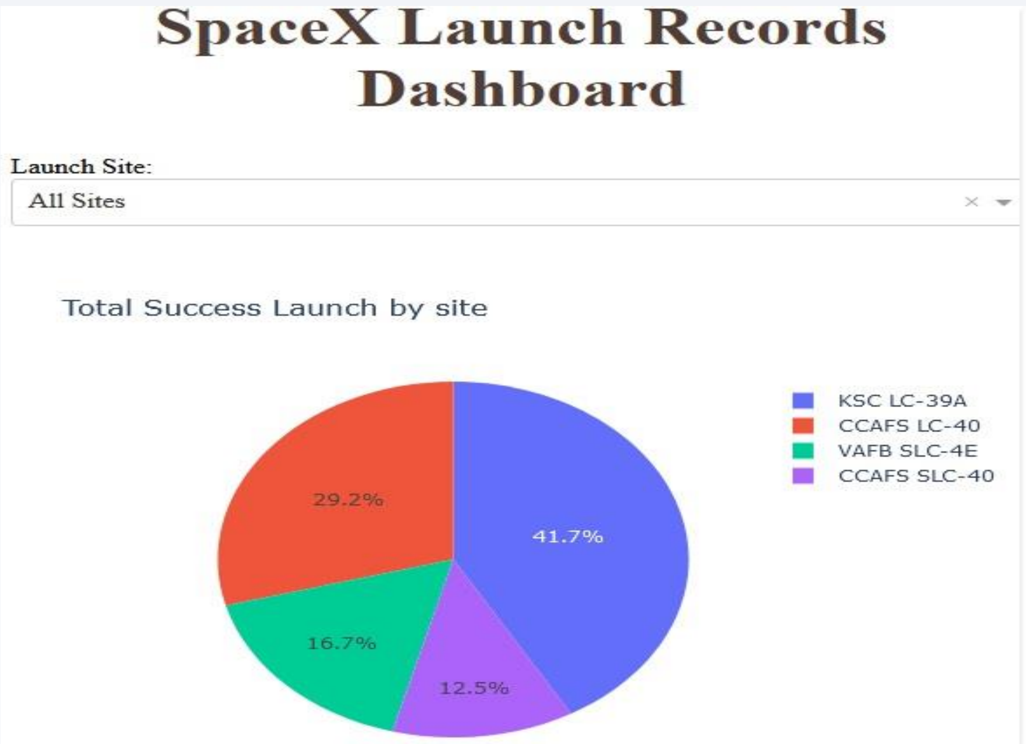
Key EDA results:

- There are Four unique launch sites CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, and CCAFS SLC-40
- Average payload mass for F9 v1.1: 2928.4 kg.
- Total payload mass for NASA (CRS): 45596 kg.
- Success missions: 61, Failures: 10.
- First successful ground pad landing: 2015-12-22.
- Successful drone ship landings with 4000-6000 kg payload.
- Boosters with maximum payload: F9 B5 series.
- Two failed missions in 2015: 'Failure (drone ship)'.
- 8 missions between 2010-06-04 and 2017-03-20.

Predictive analysis results:

- Decision Tree Classifier Identified as best performing model based on the accuracy score 0.8714 on test data and other relevant evaluation metrics.

Interactive analytics demo in screenshots

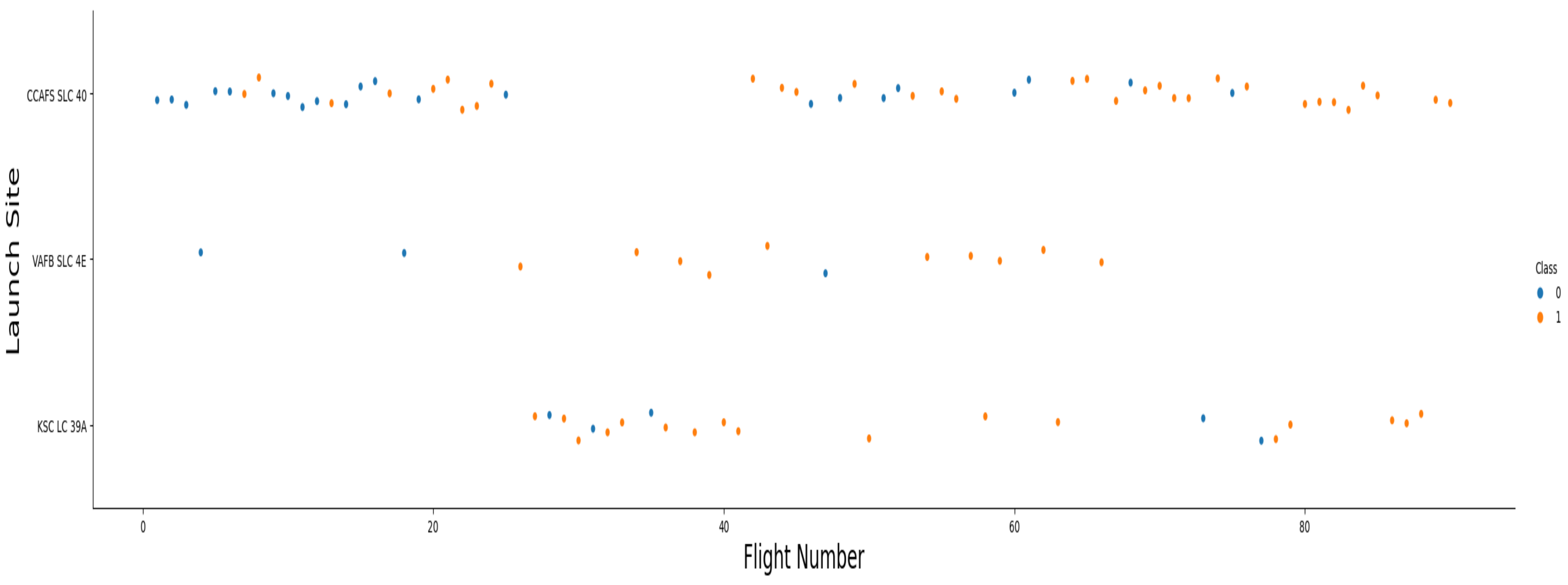


Section 2

Insights drawn from EDA

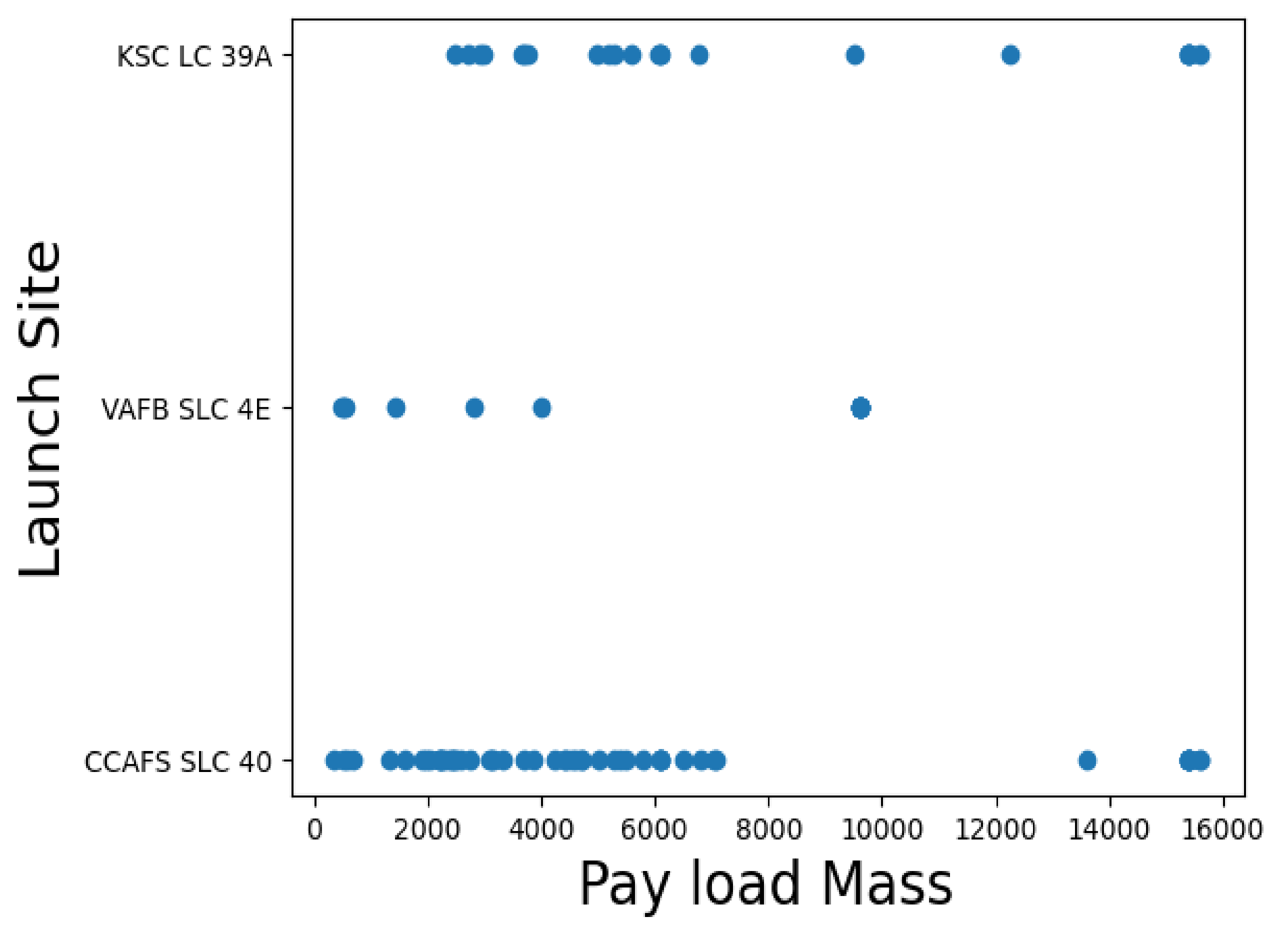
Flight Number vs. Launch Site

We see that as the flight number increases, the first stage is more likely to land successfully from Launch Site. And from CCAFS SLC 40 Launch Site higher the flight number the success rate is higher.



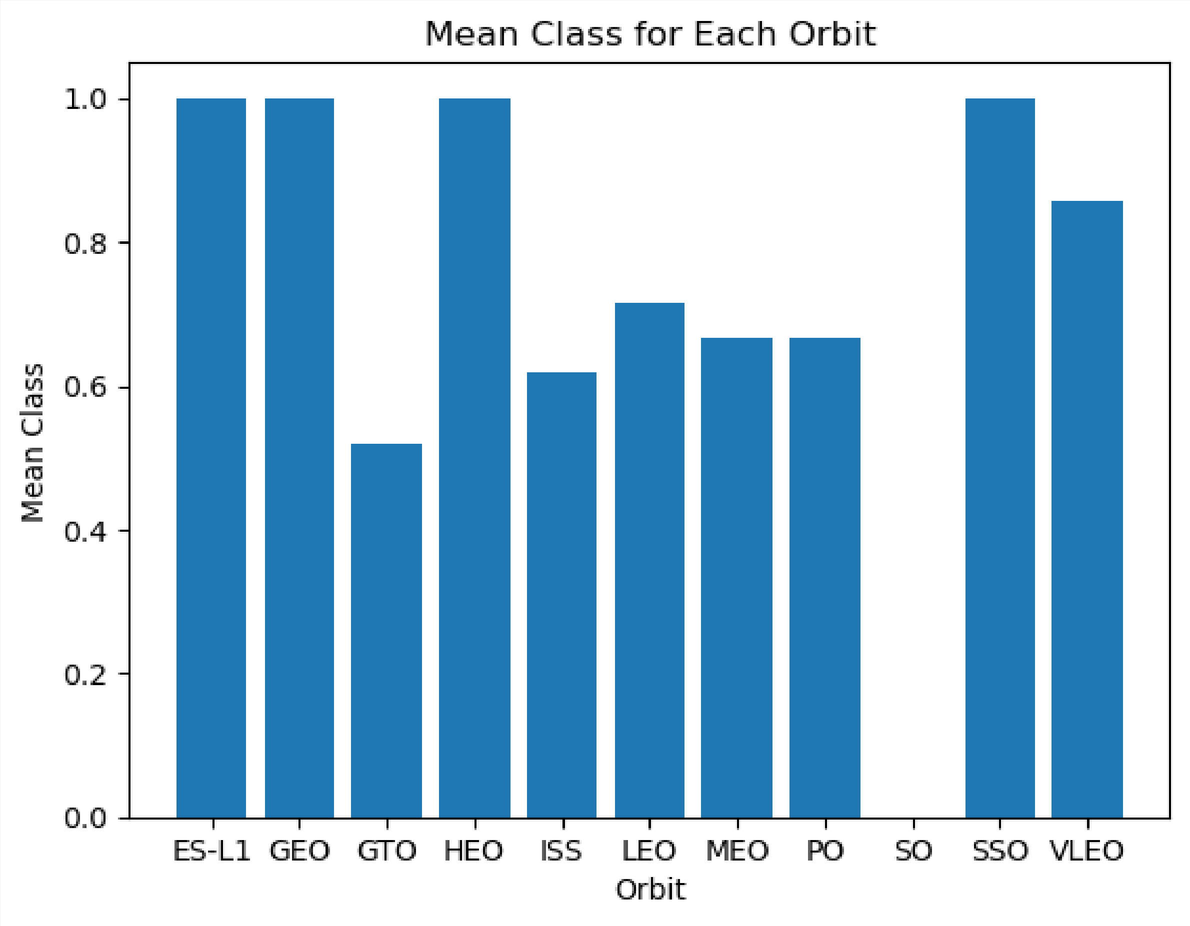
Payload vs. Launch Site

- If we observe Payload Vs. Launch Site scatter point chart, we will find for the VAFB-SLC launch site, there are no rockets launched for heavy payload mass(greater than 10000).



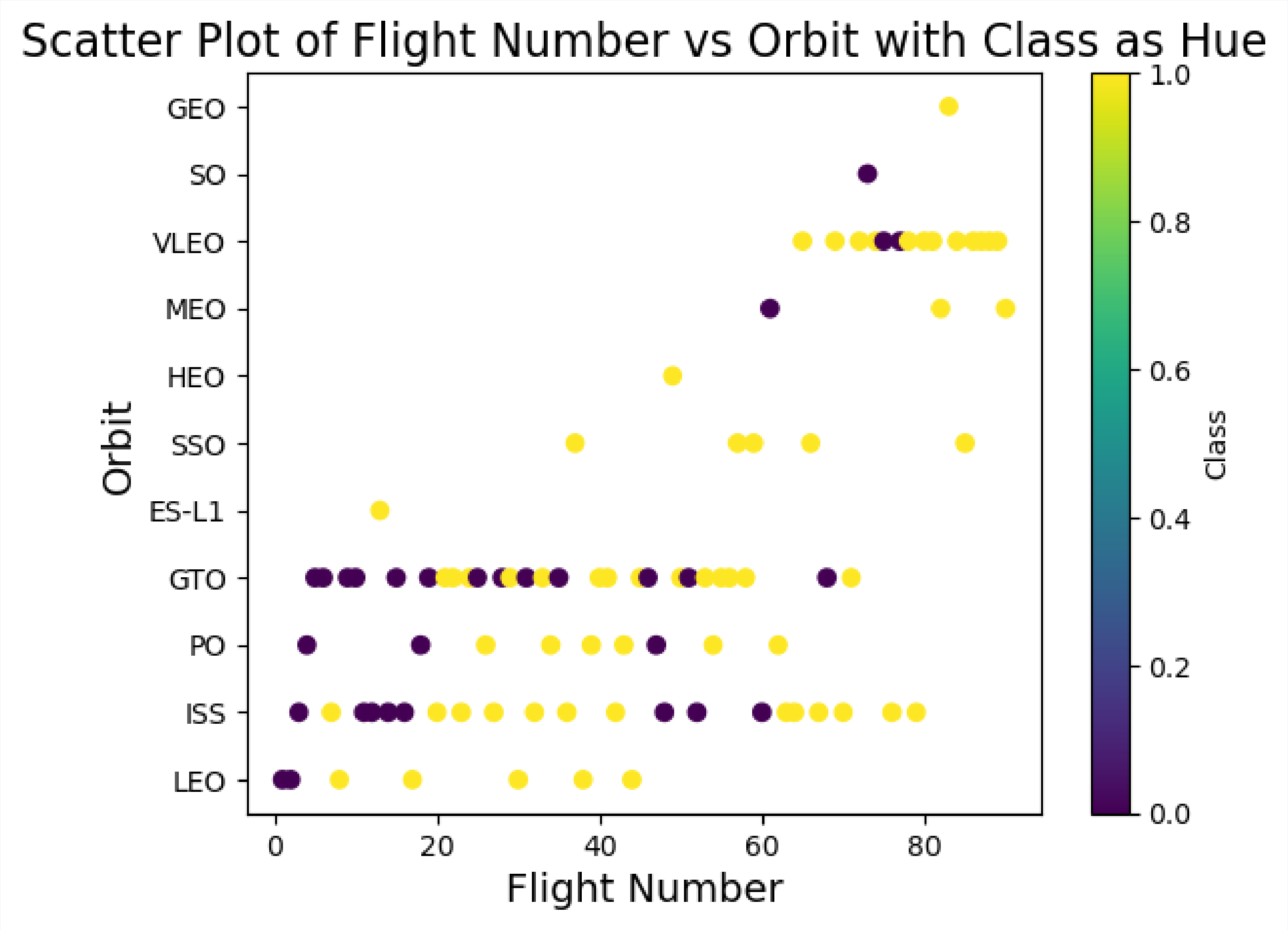
Success Rate vs. Orbit Type

- After analyzing the plot, we find that ES-L1, GEO, SSO orbit has high success rate and orbit SO has zero and GTO has low success rate.



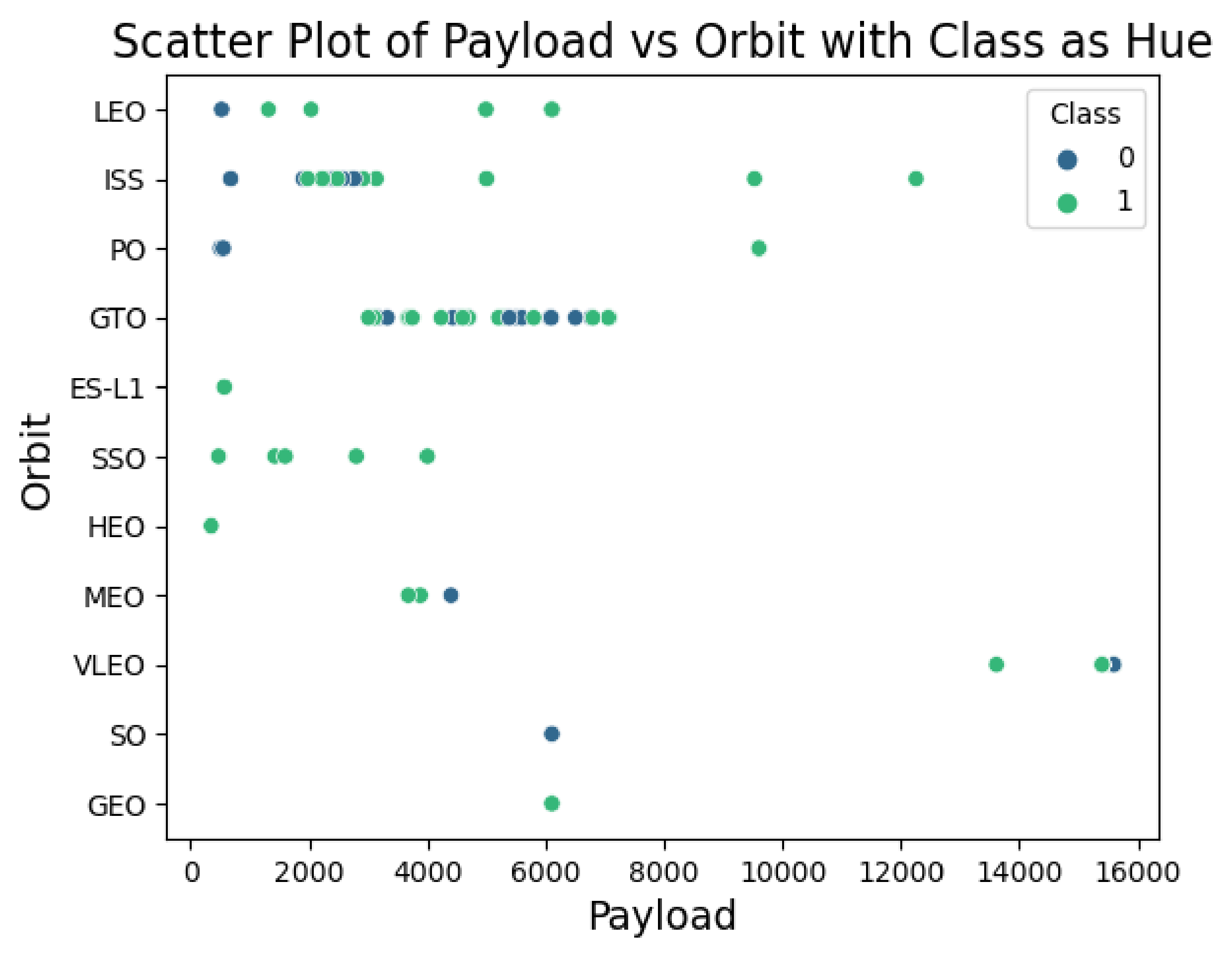
Flight Number vs. Orbit Type

- We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



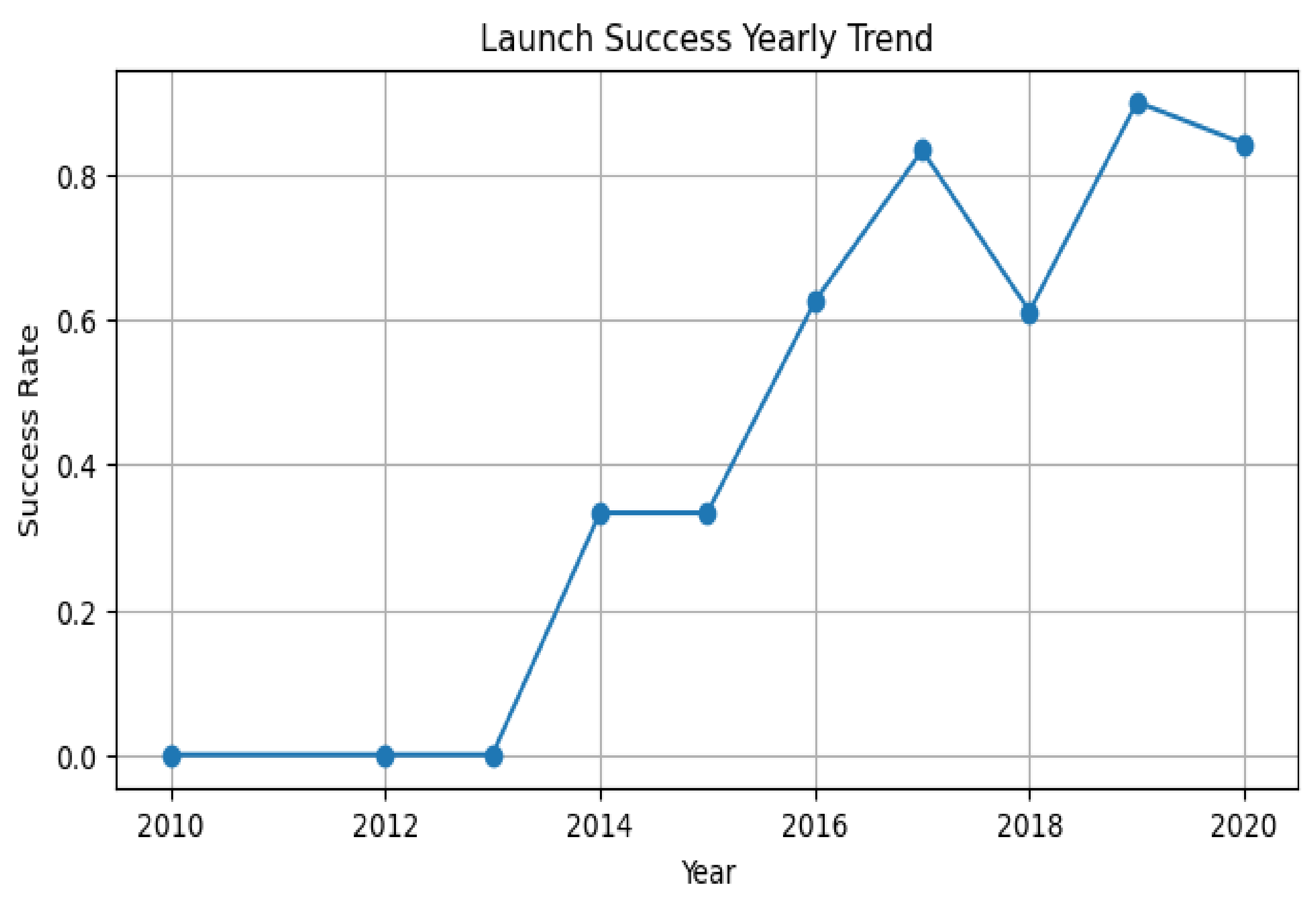
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



All Launch Site Names

This slide presents the unique launch sites utilized by Tesla for their spaceX missions. These launch sites include Cape Canaveral Air Force Station (CCAFS) LC-40, Vandenberg Air Force Base (VAFB) SLC-4E, Kennedy Space Center (KSC) LC-39A, and another site at CCAFS designated as SLC-40.

Query Used for Retrieval Launch Site Name:

```
%sql sqlite:///my_data1.db

df.to_sql("SPACEXTBL", con, if_exists='replace',
index=False,method="multi")

%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null

%sql select distinct(Launch_Site) as Unique_Launch_Sites from SPACEXTABLE
```

Summary:

- This set of commands utilizes SQL magic in Jupyter Notebook to interact with SQLite database named my_data1.db. It first stores a DataFrame named df into a table named SPACEXTBL within the database, replacing the table if it already exists. Then, it creates a new table named SPACEXTABLE by selecting all data from SPACEXTBL where the Date column is not null. Finally, it retrieves distinct launch sites from SPACEXTABLE and presents them as Unique_Launch_Sites.

Launch Site Names Begin with 'CCA'

- We have used query *%sql select * from spacextable where Launch_Site like 'CCA%' limit 5* to retrieve 5 records where launch sites begin with `CCA`

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload mass of **45596** kilograms carried by boosters from NASA
    - We used query *%sql select sum(PAYLOAD_MASS__KG_) as payload_mass_carried_by_boosters from SPACEXTABLE where Customer = 'NASA (CRS)'* to retrieve the information.

Summary:

- This SQL query calculates the total payload mass carried by boosters of type F9 v1 for NASA (CRS) from the table SPACEXTABLE. It sums the PAYLOAD_MASS__KG_ column where the Customer column equals 'NASA (CRS)'. The result is presented as payload_mass_carried_by_boosters.

Average Payload Mass by F9 v1.1

- To retrieve the information “Average Payload Mass by F9 v1.1” we use following sql query.
  - *%sql select avg(PAYLOAD_MASS__KG_) as 'payload_mass_carried_by_booster_version_F9_v1.1' from SPACEXTABLE where Booster_Version = 'F9 v1.1'*
- The result of this SQL query will be the average payload mass carried by booster version 'F9 v1.1', which is **2928.4** kg
- Summary:
  - The SQL code calculates the average payload mass carried by the booster version 'F9 v1.1' from the table named SPACEXTABLE. It selects the average value of the 'PAYLOAD_MASS__KG_' column where the 'Booster_Version' column equals 'F9 v1.1'. The result is named 'payload_mass_carried_by_booster_version_F9_v1.1'.

First Successful Ground Landing Date

- We used below sql query to find the dates of the first successful landing outcome on ground pad:
  - *%sql SELECT min(Date) as first_successful_landing_in_ground_pad FROM SPACEXTABLE where Landing_Outcome = 'Success (ground pad)';*

Summary:

- The SQL query retrieves the earliest date when SpaceX achieved its first successful landing on a ground pad. The result, **"2015-12-22,"** marks a significant milestone in space exploration, highlighting SpaceX's pioneering efforts in reusable rocket technology.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 is:
  - ➢ **F9 FT B1022, F9 FT B1026, F9 FT B1021.2 and F9 FT B1031.2**

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

Summary:

- The SQL query selects the booster version from the SPACEXTABLE where the landing outcome is 'Success (drone ship)' and the payload mass is between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

- **The total count of successful outcomes stands at 61, while the number of failed mission outcomes is 10.**

  We used below sql query to retrieve total number of successful and failure outcomes

  - *%%sql SELECT CASE WHEN Landing_Outcome LIKE 'Success%' THEN 'Success' ELSE 'Failure' END AS Outcome_Type, COUNT(*) AS Count FROM spacextable WHERE Landing_Outcome LIKE 'Success%' OR Landing_Outcome LIKE 'Failure%' GROUP BY Outcome_Type;*

Summary:

The SQL query categorizes landing outcomes as either 'Success' or 'Failure' using a CASE statement. It then counts the occurrences of each outcome type and groups the results accordingly. Finally, it returns the outcome type along with its count.

Total Number of Successful and Failure Mission Outcomes(with breakdown)

- To breakdown total number of successful and failure mission outcome we use below sql query:
  - *%%sql select Landing_Outcome, count(*) as Count from spacextable where Landing_Outcome like 'Success%’ or Landing_Outcome like 'Failure%’ group by Landing_Outcome*

Summery:

The SQL query counts the occurrences of each landing outcome that starts with 'Success' or 'Failure’ in the spacextable. It groups the results by the landing outcome and returns the outcome along with its count.

Landing_Outcome	Count
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass 15600 Kg
  - 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

Query result with a short explanation here:

```
%%sql select Booster_Version, PAYLOAD_MASS__KG_ from spacextable where
PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable);
```

- The SQL query retrieves the booster version and payload mass from the spacextable where the payload mass is equal to the maximum payload mass in the spacextable.

2015 Launch Records

- Below record displays the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Date	month	Landing_Outcome	Booster_Version	Launch_Site
2015-01-10	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

%%sql select Date,substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from spacextable where Landing_Outcome like 'Failure%' and substr(Date,0,5)='2015'

Summary:

The above SQL query selects the date, month, landing outcome, booster version, and launch site from the spacextable where the landing outcome is 'Failure (drone ship)' and the year part of the date is '2015'. It also extracts the month from the date using the substr function.

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

- Landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order is:

Landing_Outcome	Count
Failure (drone ship)	5
Success (ground pad)	3

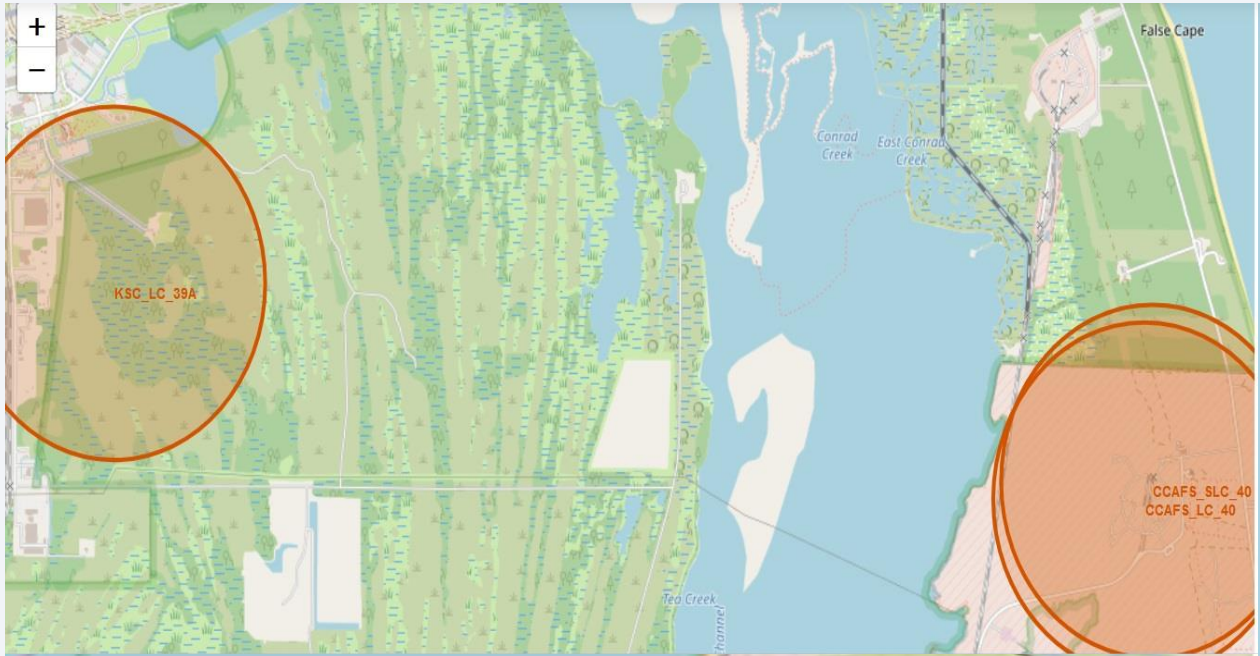
  - *%%sql select Landing_Outcome, count(*) as Count from spacextable where (Landing_Outcome = 'Success (ground pad)' or Landing_Outcome = 'Failure (drone ship)') and Date BETWEEN '2010-06-04' AND '2017-03-20' group by Landing_Outcome order by Count Desc*

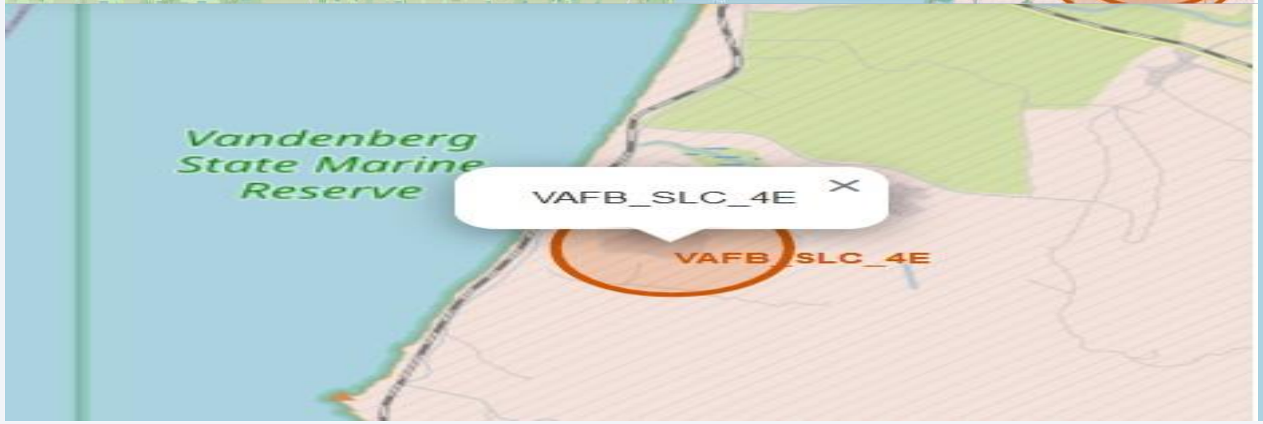
The above SQL query retrieves the count of successful landings on ground pads and failed landings on drone ships from the spacextable between the dates '2010-06-04' and '2017-03-20'. It groups the results by landing outcome, counts the occurrences, and sorts them in descending order based on the count.

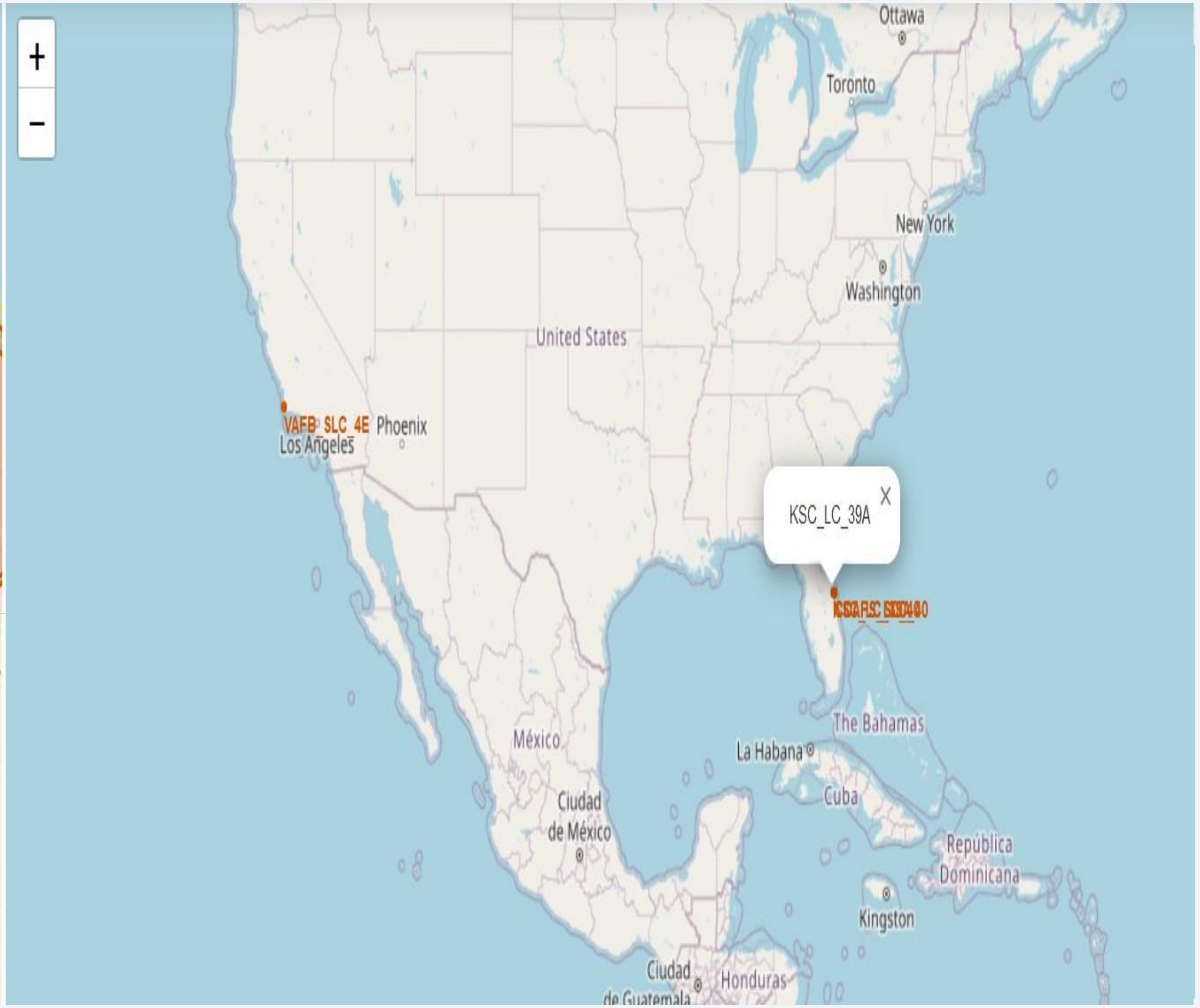
Section 3

Launch Sites Proximities Analysis

All launch sites on a map





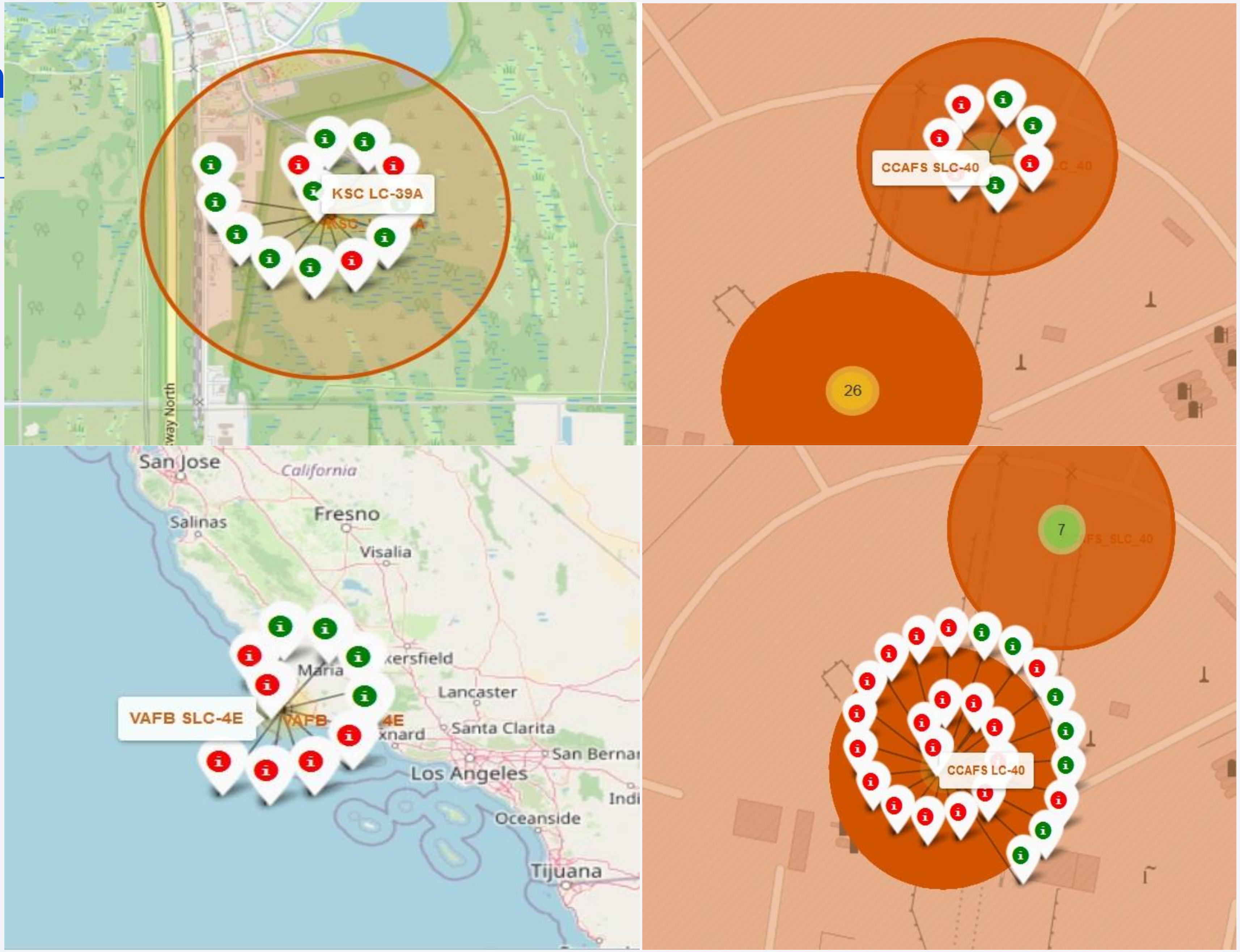


Summary and findings from all launch site's locations

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It also depends on the location and proximities of a launch site.
- Based on these latitude values, all the launch sites are relatively close to the Equator line (closer to 0 degrees latitude). Generally, launch sites closer to the Equator provide certain advantages, such as increased rotational speed due to the Earth's rotation, which can be beneficial for rocket launches.
- CCAFS (Cape Canaveral Air Force Station) SLC-40 and CCAFS LC-40: These launch sites are associated with Cape Canaveral, which is located on the east coast of Florida. Therefore, it is likely that these launch sites are close to the coast. KSC LC-39A: This launch site is associated with Kennedy Space Center (KSC), which is also located on the east coast of Florida. Similar to CCAFS, it is likely to be close to the coast. VAFB SLC-4E: This launch site is associated with Vandenberg Air Force Base (VAFB), which is located on the west coast of California. Therefore, it is likely that this launch site is close to the coast.
- Based on the general geographical locations of these launch sites, it is likely that they are in proximity to the coast.
- Coastal launch sites provide safety, flexibility, and efficiency for space missions. Proximity to the ocean allows for versatile launch trajectories, enhances safety by directing rocket debris away from populated areas, and enables access to multiple orbital inclinations. Coastal locations also offer favorable weather conditions and logistical advantages.

Success/failed launches for each site on the map

- Now we have enhanced the map by adding the launch outcomes for each site, and see which sites have high success rates. the class column indicates if this launch was successful or not
- We have created markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
- From the color-labeled markers, we can be able to easily identify/visualize which launch sites have relatively high success rates.



Distances between a launch site to its proximities

Summary and Findings from proximities of launch sites

- We have explored and analyzed the proximities of launch sites, we can easily find the coordinates of railways, highways, coastline and cities.
- launch sites in close proximity to railways which is 0.98 km, highways which is 0.58, coastline which is 0.88 km and far from cities which is approx. 51.52 km.

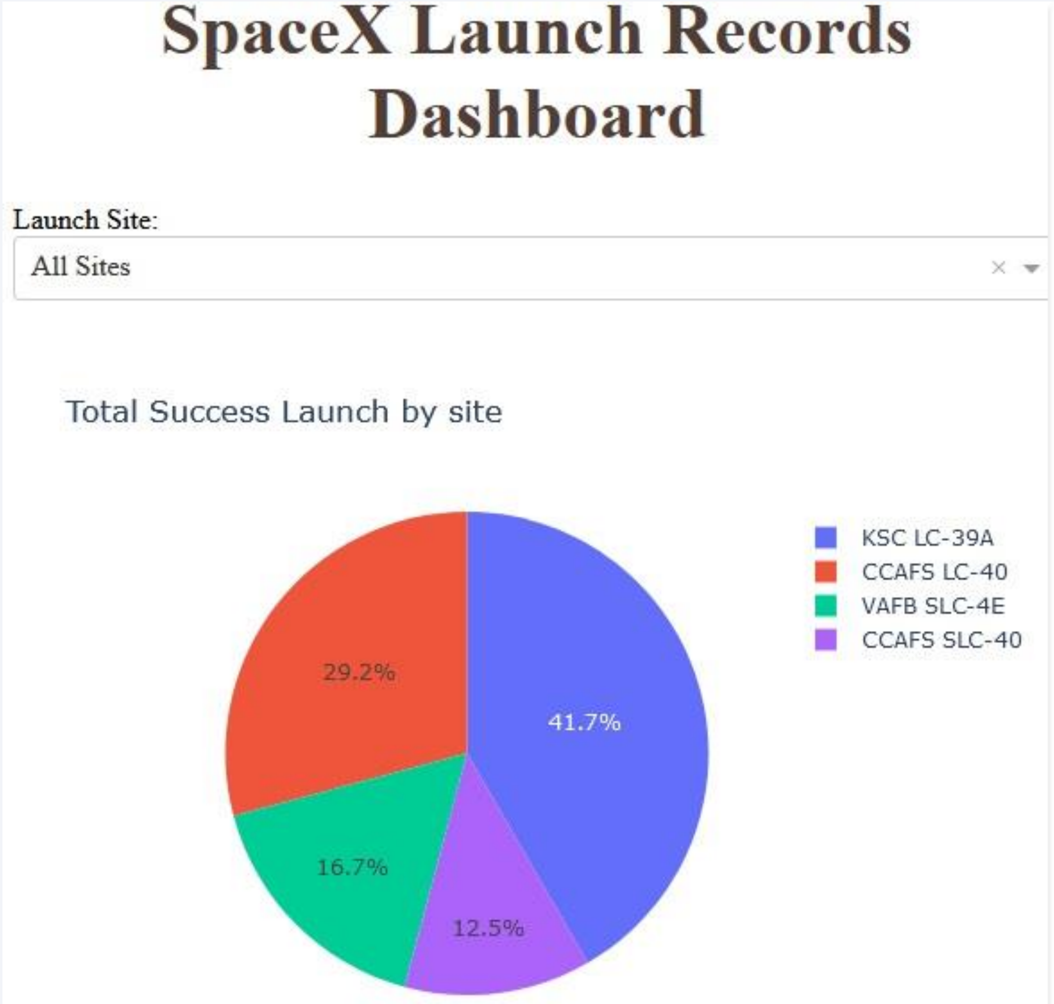
❖ **So, Selecting optimal launch sites involves criteria like close proximity to railways and highways for logistics, proximity to coastlines for safe launches over water, and being far from cities to ensure public safety. Balancing these factors optimizes transportation and minimizes risks during space launches.**

Section 4

Build a Dashboard with Plotly Dash

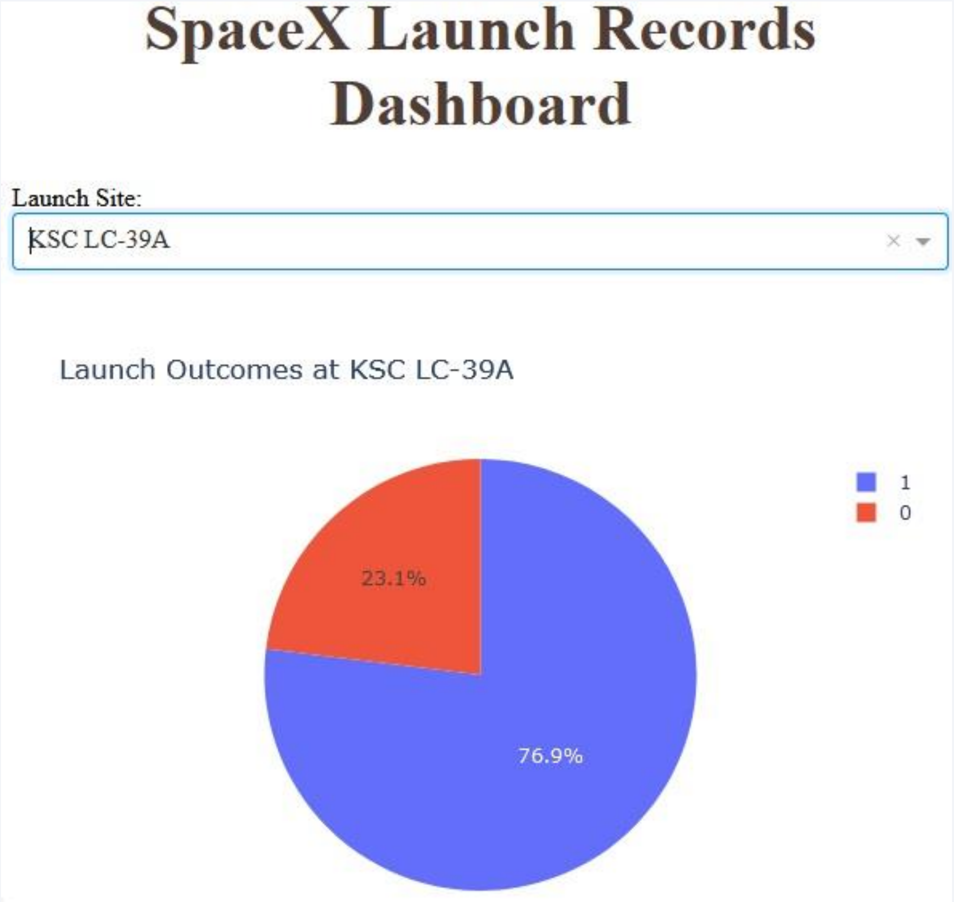
SpaceX Launch sites success Ratio

❖ This pie chart clearly shows that Launch Site KSC LC 39A has highest success ratio with 41.7 percent while, CCAFS LC-40 holds 29.2 percent, VAFB SLC-4E and CCAFS SLC40 holds 16 percent and 12.5 percent respectively.



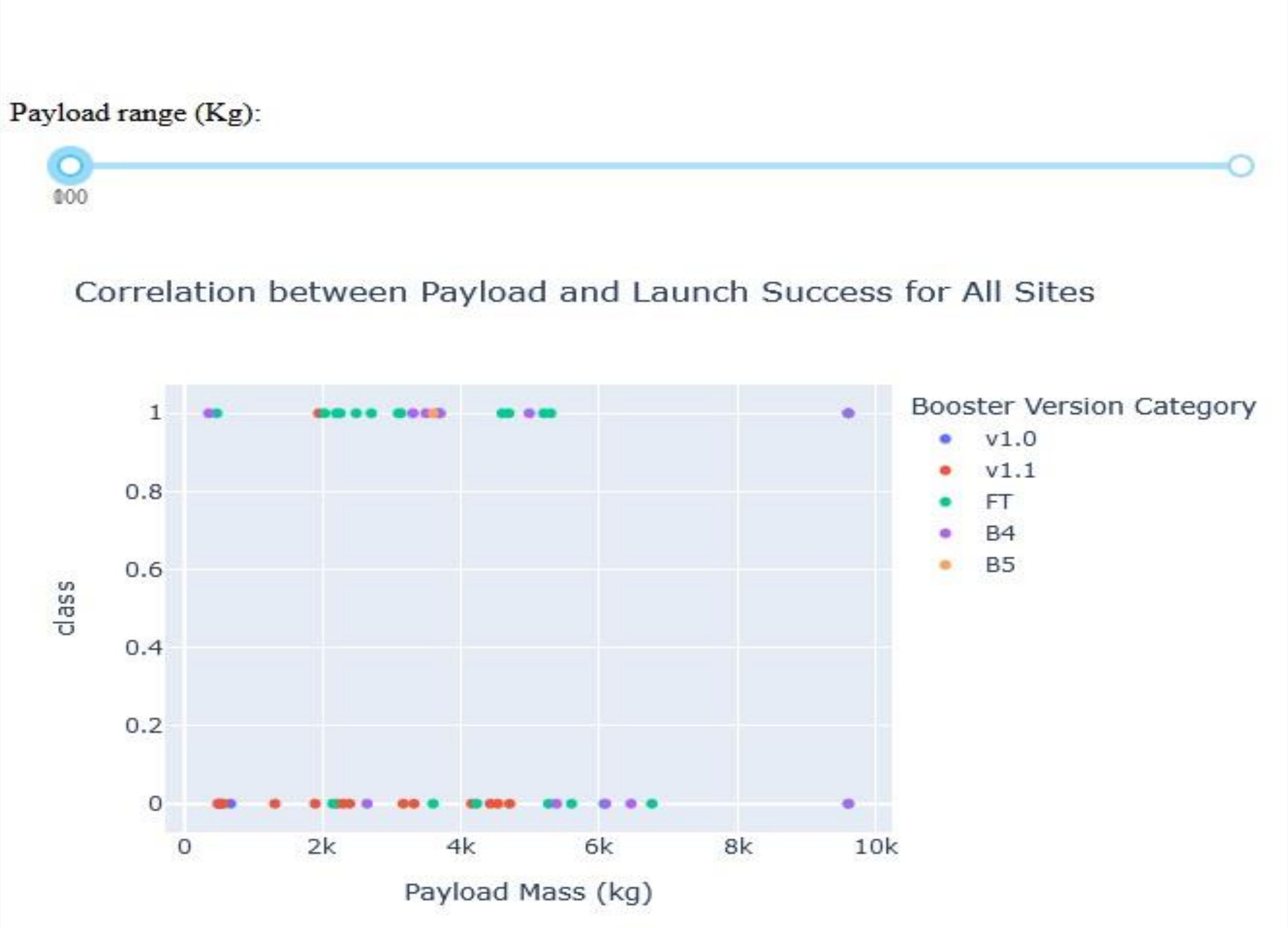
KSC LC-39A's success

- ❖KSC LC-39A's success is attributed to its equatorial location, robust infrastructure, extensive experience, favorable weather, and diverse mission support. These factors ensure efficient launches, refined processes, and higher success rates, compared to other sites.



Payload Vs Launch Outcome Scatter Plot

- Payloads within the 2k to 6k range consistently achieve high launch success rates, indicating optimal performance in this payload bracket.
- Conversely, payloads ranging from 5.5k to 10k exhibit lower success rates, suggesting potential challenges or constraints within this payload range.
- Noteworthy is the observation that the F9 Booster version FT consistently demonstrates the highest success rates compare to other booster, indicating robust performance and reliability associated with this booster version.



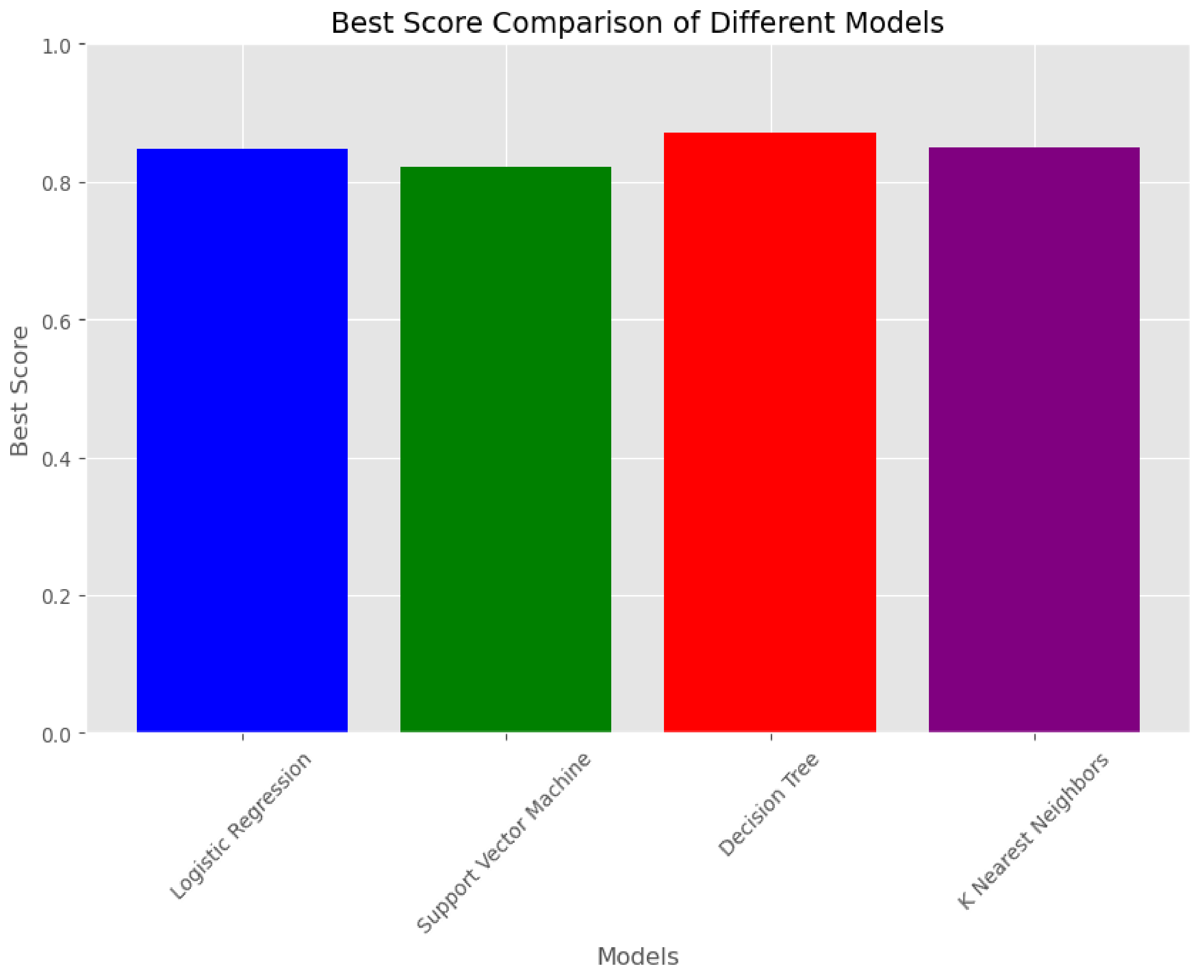
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Model	Best Parameters	Best Score
0	Logistic Regression	{'logreg__C': 0.01, 'logreg__penalty': 'l2', '...	0.846429
1	Support Vector Machine	{'svm__C': 0.03162277660168379, 'svm__gamma': ...	0.821429
2	Decision Tree	{'tree__criterion': 'gini', 'tree__max_depth':...	0.871429
3	K Nearest Neighbors	{'knn__algorithm': 'auto', 'knn__n_neighbors':...	0.848214

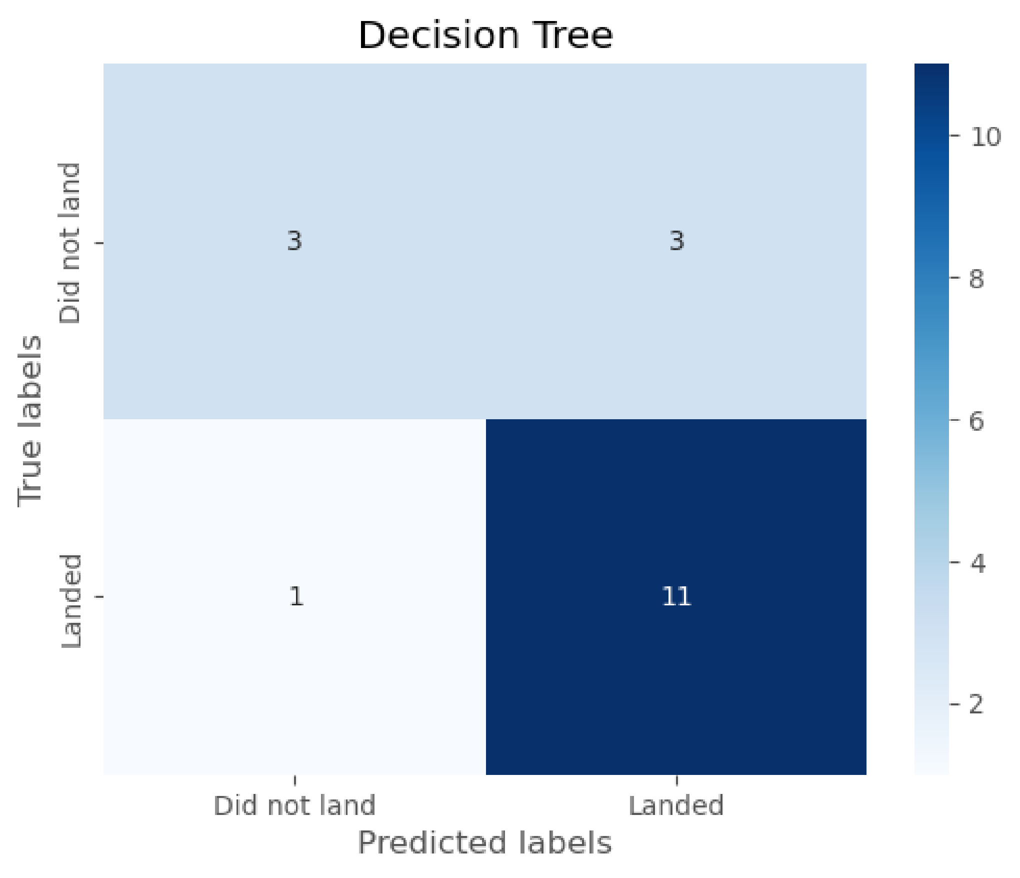
❖ The Decision Tree model achieved the highest best score of 0.8714 among all models. Therefore, based on this criterion, the Decision Tree model performs better compared to Logistic Regression, Support Vector Machine, and K Nearest Neighbors. However, it's important to note that model performance can vary based on different evaluation metrics, datasets, and problem contexts. So, the choice of the best-performing model may ultimately depend on your specific requirements and considerations.



Confusion Matrix

- The confusion matrix for the Decision Tree model is:
  - True Negatives (TN): 3
  - False Positives (FP): 3
  - False Negatives (FN): 1
  - True Positives (TP): 11

- It indicates that the model correctly classified 3 instances as negative and 11 instances as positive but misclassified 3 instances as positive and 1 instance as negative.



Conclusions

- Success rate increases with higher flight numbers, particularly at CCAFS SLC 40 Launch Site.
- AFB-SLC launch site has no launches for heavy payload masses (>10000).
- In LEO orbit, success appears to be related to the number of flights, while in GTO orbit, there's no apparent relationship.
- Success rate has generally increased since 2013, with a stable period in 2014, and a noticeable rise after 2015.
- Launch success rates are influenced by various factors such as payload mass, orbit type, and proximity to the Equator.
- Coastal launch sites like CCAFS, KSC, and VAFB offer safety, flexibility, and logistical advantages for space missions.
- Optimal launch site selection considers factors like proximity to railways, highways, coastlines, and distance from cities to ensure efficient transportation and minimize risks.
- KSC LC 39A exhibits the highest success ratio, attributed to its equatorial location, robust infrastructure, favorable weather, and extensive experience.

- Payloads in the 2k to 6k range consistently achieve high success rates, while payloads from 5.5k to 10k show lower success rates.
- The F9 Booster version FT demonstrates the highest success rates, indicating its reliability and robust performance compared to other boosters.
- By utilizing Decision Tree classification algorithm, we can predict whether a SpaceX launch will result in a successful or unsuccessful landing outcome based on historical data. This model could then be used for various purposes, such as optimizing landing procedures, assessing mission success probabilities, and enhancing overall operational efficiency.

Appendix

- Including all relevant document, python code, data sets, project documentation and project overview to GitHub repository and the link is:

https://github.com/debsandipagt/Space-X-Falcon-9-First-Stage-Landing-Prediction

Thank you!