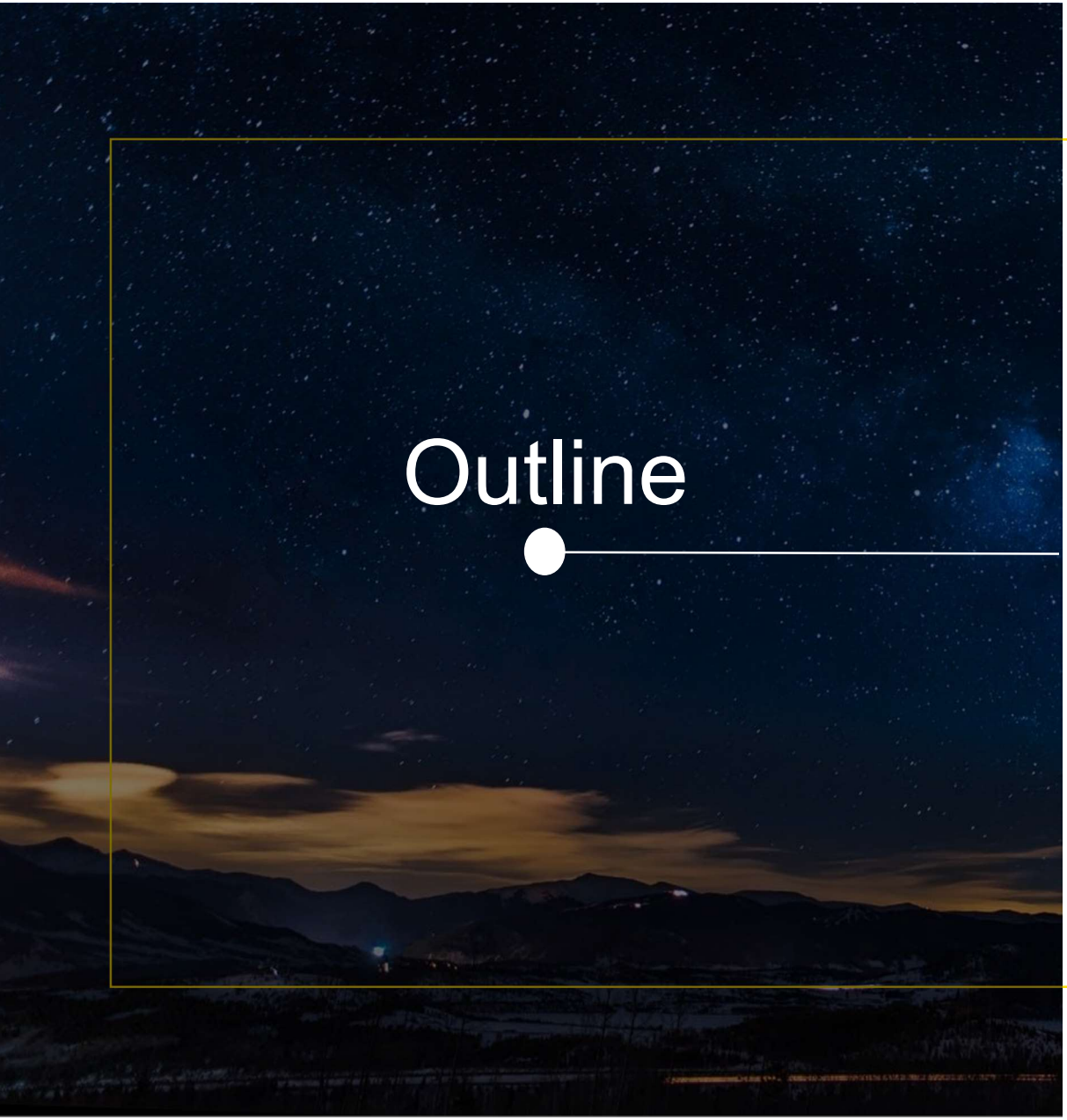

Winning Space Race
with Data Science
TEH RUI LING
06 FEB 2022



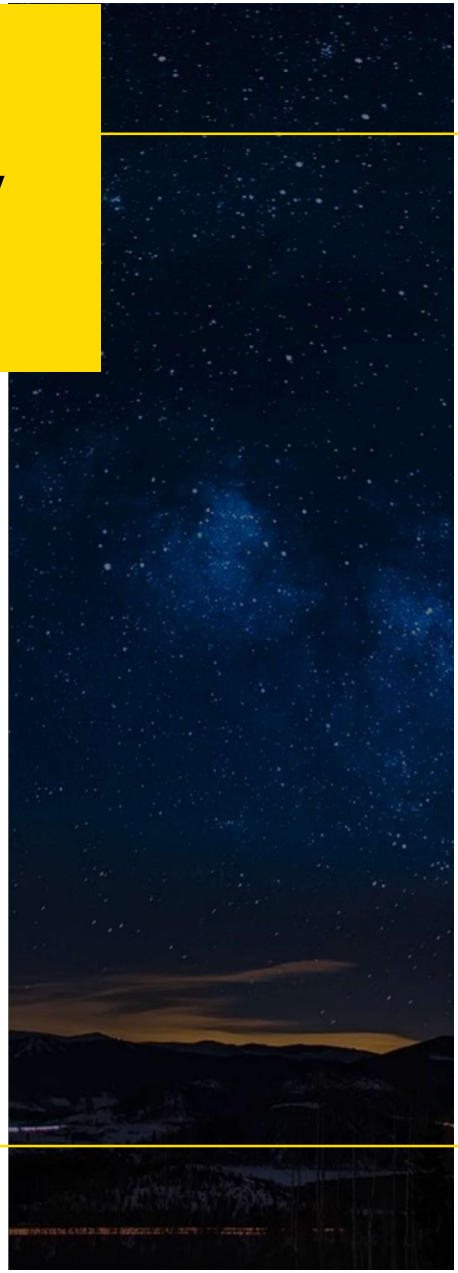
Outline

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

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- Exploratory Data Analysis (EDA)
  - With Data Visualization
  - With SQL
- Building an Interactive Map with Folium
- Building a Dashboard with Plotly Dash
- Predictive Analysis (Classification)



Summary of all results

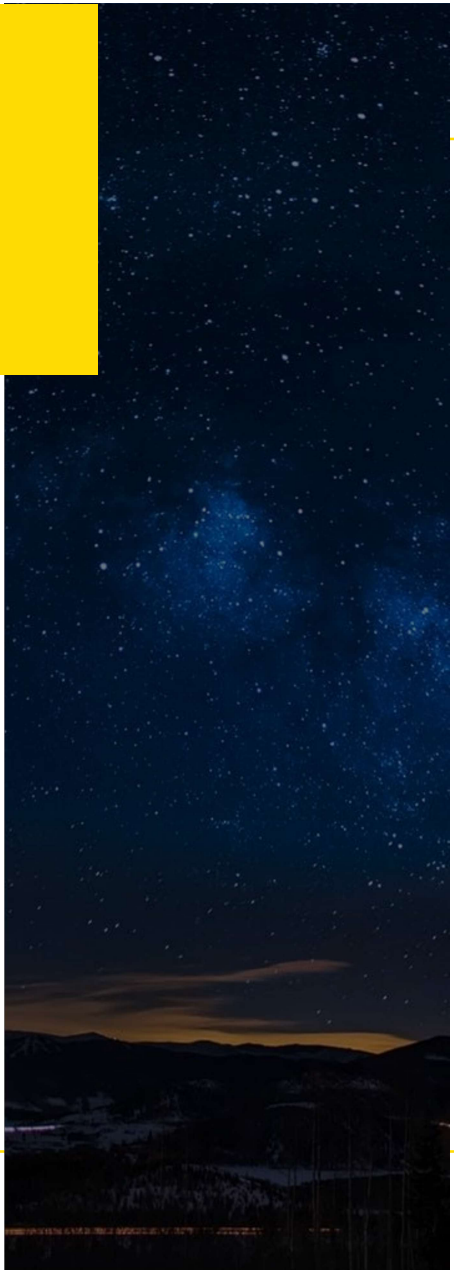
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

Background and context

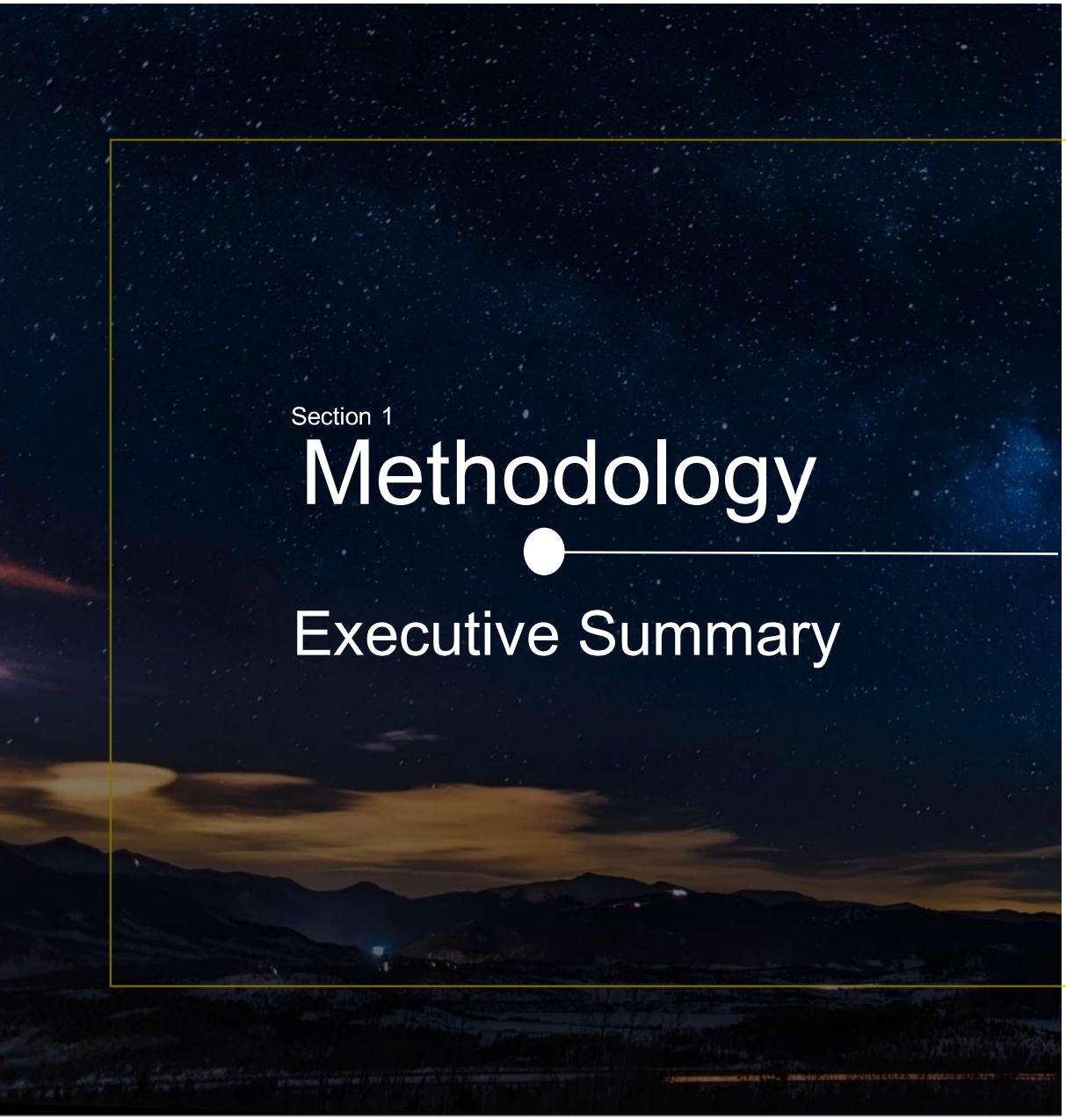
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost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

Goal of this project:
To create a machine learning pipeline to predict if the Falcon 9 first stage will land successfully.



Problems that we want to find answers

- What are the factors to determine if the rocket will land successfully?
- How to determine the successful landing rate based on the interaction among various features?
- What conditions needed to achieve the best results and ensure a successful landing rate?



Section 1

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scraping from Wikipedia.
- Perform data wrangling
  - Data was processed using one-hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

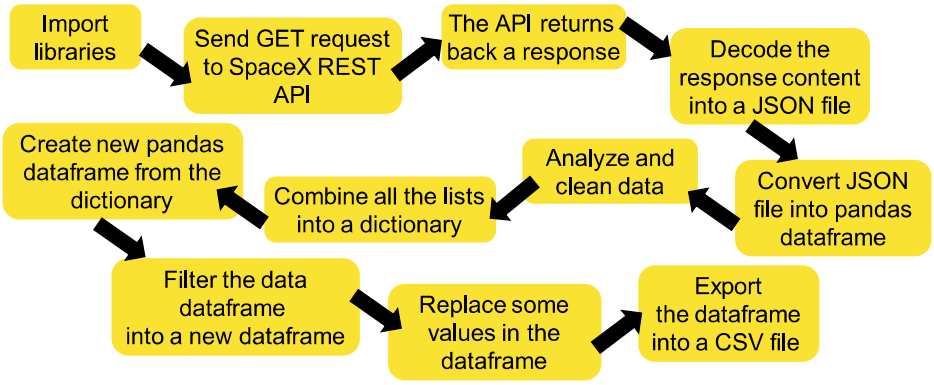
1. We need to identify and gather the available data resources first.
   The resources are:
   - SpaceX launch data
   - Falcon 9 Launch Wikipedia page data

2. Based on the resources, we will need to determine the data collection method.
   a) Using SpaceX REST API, to retrieve the SpaceX launch data.
   b) Using BeautifulSoup, to extract the data from Falcon 9 Launch Wikipedia page.

3. After using the data collection method, we will need to analyze and determine what data needed.

Data Collection – SpaceX API

Flowchart

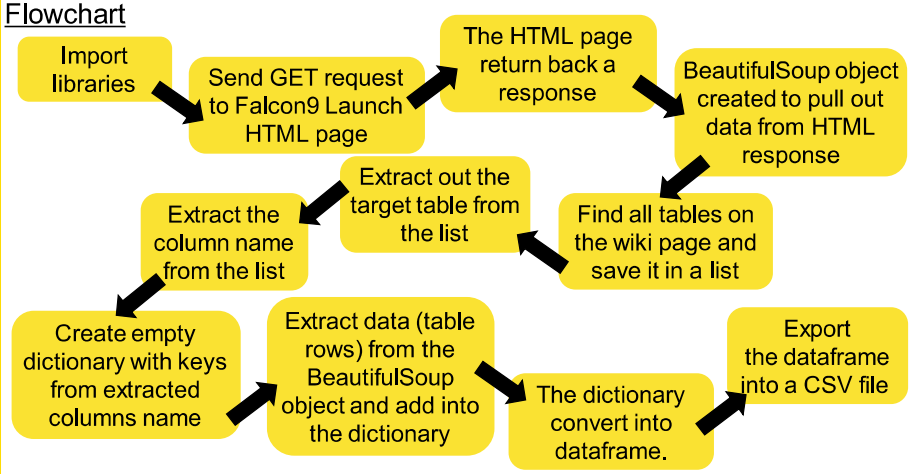


1. Import the necessary libraries.
2. Send a GET request to instruct SpaceX REST API to retrieve the data from the given URL link. The request is communicated via an HTTP message.
3. The SpaceX REST API will return back a response. The response is also communicated via an HTTP message.
4. We decode the response content into a JSON file.
5. We convert the JSON file into pandas dataframe, 'data'.
6. We analyze and clean the data. The data for each column are kept in a list.
   a) We take a subset of the dataframe and store it back to 'data',
   b) Remove rows with multiple cores or payloads,
   c) Convert 'date_utc' to datetime datatype and extract the date only.
   d) A lot of data are IDs – use user-define function to request the APIs to extract information using those IDs.
7. Combine all the lists into a dictionary, 'launch_dict'.
8. Create a new pandas dataframe, 'df_launch', from the dictionary, 'launch_dict'.
9. Filter the data dataframe, 'df_launch', into a new dataframe, 'data_falcon9'.
10. Replace some values (of 'FlightNumber') in the dataframe, 'data_falcon9'.
11. Then export out the dataframe into a CSV file (dataset_part_1.csv).

Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-spacex-data-collection-api-solution.ipynb

Data Collection – Scraping

Flowchart



1. Import the necessary libraries.
2. Send a GET request to instruct the Falcon9 Launch HTML page to retrieve the data. The request is communicated via an HTTP message.
3. The Falcon9 Launch HTML page return back a response. The response is also communicated via an HTTP message.
4. A BeautifulSoup object, 'soup', is created to pull out data from HTML response.
5. Using the 'find_all' function, to find all tables in the BeautifulSoup object, 'soup', and save it in a list, 'html_tables'.
6. Extract out the target table (3rd table on the wiki page) from the list, 'html_tables' and save it to another list, 'first_launch_table'.
7. Extract out the column name from the list, 'first_launch_table'.
8. We create an empty dictionary, 'launch_dict', with keys from extracted columns names from step 7.
9. We extract the launch records data (3rd table's rows) from the BeautifulSoup object, 'soup', and add into the dictionary, 'launch_dict'.
10. The dictionary, 'launch_dict', is converted into pandas dataframe, 'df'.
11. Then export out the dataframe into a CSV file (spacex_web_scraped.csv).

Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-webscraping-solution.ipynb

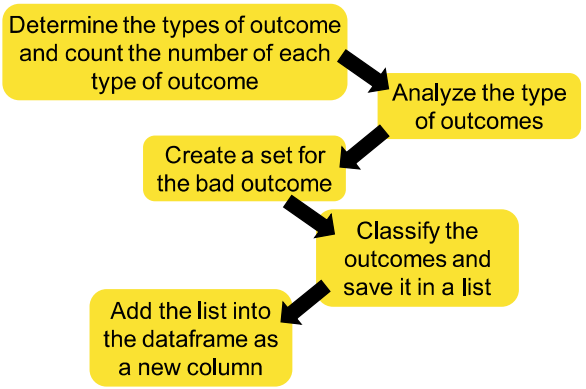
Data Wrangling

Replace missing values with mean value in the 'PayloadMass' column

1. Calculate the mean value of 'PayloadMass' column. The value is saved as 'Mean_Payload' variable.
2. Using replace function, to replace the missing values, 'NaN', with the mean value.

Create a new column to classify the outcome of each launch outcome

1. Using value_counts method, to determine the types of outcomes and count the number of each type of outcome in the 'Outcome' column. Then save the result to 'landing_outcomes' variable.
2. Print out the keys of the 'landing_outcomes' variable and analyze the type of outcomes, whether it is good or bad outcome.
   - Bad outcome (did not land successfully): 'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'.
   - Good outcome (landed successfully): 'True ASDS', 'True Ocean', 'True RTLS'.
3. Create a set of outcomes that did not land successfully, as 'bad_outcome', from the 'landing_outcomes' variable.
4. Using the set, 'bad_outcome', to classify the outcomes in the 'Outcome' column. Then save it in a list, 'landing_class'.
   - Bad outcome label as '0'.
   - Good outcome label as '1'.
5. Add the list, 'landing_class', as a new column in the dataframe, 'df'.



Links to the notebook in GitHub:

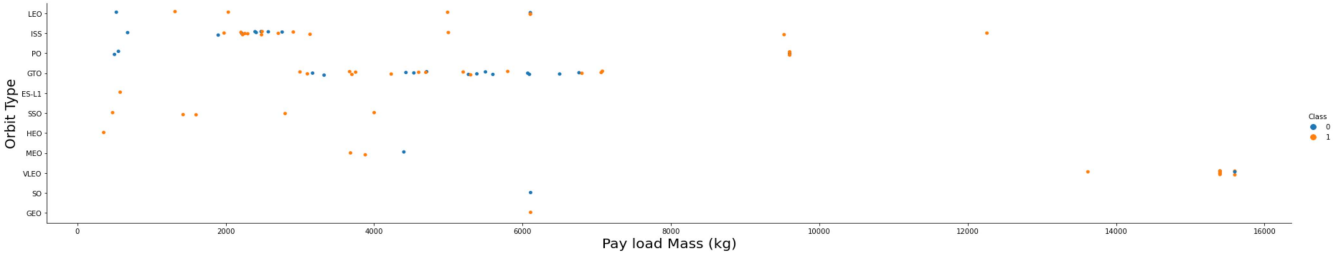
- https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-spacex-data-collection-api-solution.ipynb
- https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling-solution.ipynb

EDA with Data Visualization

Scatter charts were plotted to visualize the relationship between:

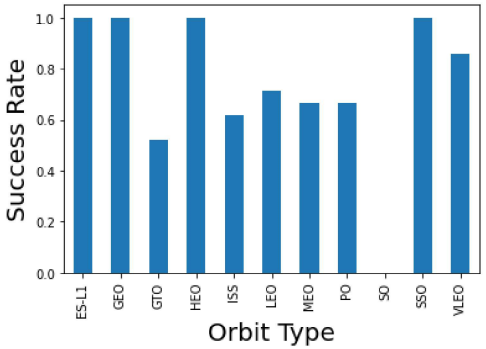
- Flight number vs Payload Mass
- Flight number vs Launch site
- Payload Mass vs Launch site
- Flight number vs Orbit type
- Payload Mass vs Orbit type

Scatter chart was used to show if there is any correlations between the 2 variables.



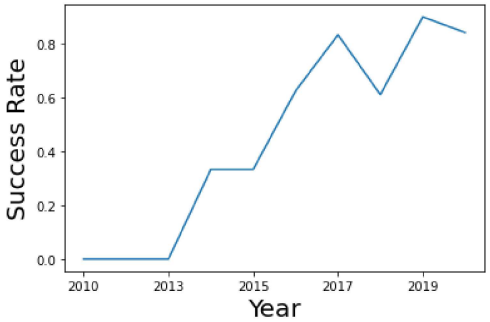
Bar chart were plotted to visualize the relationship between success rate of each orbit type.

Bar chart was used to show each orbit type in a frequency distribution.



Line chart were plotted to visualize the launch success yearly trend.

Line chart was used to show the changes and trends over different time period.



Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-eda-dataviz-solution.ipynb

EDA with SQL

After establishing a connection with the database, the following SQL queries has been executed to get answers from the SpaceX dataset:

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster versions which have carried the maximum payload mass.
- List the failed landing outcomes in drone ship, their booster versions, and launch site names or in year 2015.
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-02, in descending order.

Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/jupyter-labs-eda-sql-coursera-solution.ipynb

Build an Interactive Map with Folium

To visualize the SpaceX launch dataset into an interactive map:

1. We created folium map objects to add circles and text to mark all launch sites on the map.
   So that to give an intuitive insights about where are those launch sites.

2. After that, we created additional color-labeled markers in marker clusters to mark the success/failed launches for each site on the map. This helps to easily identify which launch sites have relatively high success rates.

3. We have also created lines to mark the distances and distance calculated between a launch site to selected proximities.

Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/lab_jupyter_launch_site_location-solution.ipynb

Build a Dashboard with Plotly Dash

To perform interactive visual analytics on SpaceX launch data in real-time for the users, we have added the following input components to interact with the pie chart and scatter point chart in the dashboard:

- A dropdown list, and
- A range slider

The dropdown list allows users to have an option to select either one or all of the Launch Sites' detailed success rate (from the pie chart) and success payload (from the scatter point chart).

The range slider allows users to select the payload range, so as to observes the correlation between that selected payload range and success of the site(s).

The 2 input components helps different users to drill down and filter operational information so data can be viewed from different perspectives or in more details.

Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/Build%20a%20Dashboard%20Application%20with%20Plotly%20Dash.ipynb

Predictive Analysis (Classification)

Model built:

1. After importing the necessary libraries, SpaceX dataset loaded into pandas and then to numpy.
2. Data transformation. - Fit to data, then transform it.
3. Split the data into training and test datasets.
   After the data has been split, we can check the number of test samples we have.
4. To find out which method that performs the best:
   a) We set the type of machine learning algorithm and its parameters to GridSearchCV object.
   b) Using fit method, to train the model. At same time, the GridSearchCV object will conduct an exhaustive search of all parameters for each machine learning algorithm.

Model evaluation:

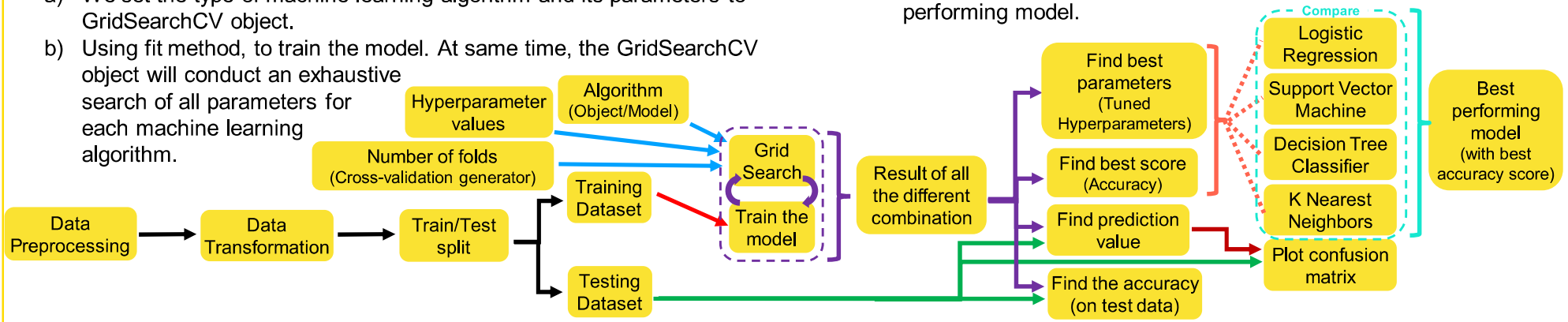
- Calculate the accuracy for each model
- Get the tuned hyperparameters (best parameters) for model
- Plot confusion matrix

Improvement on the model:

- Data preparation – Feature Engineering
- Machine Learning (Modeling) – Algorithm Tuning – Hyperparameter optimization

Finding the best performing classification model:

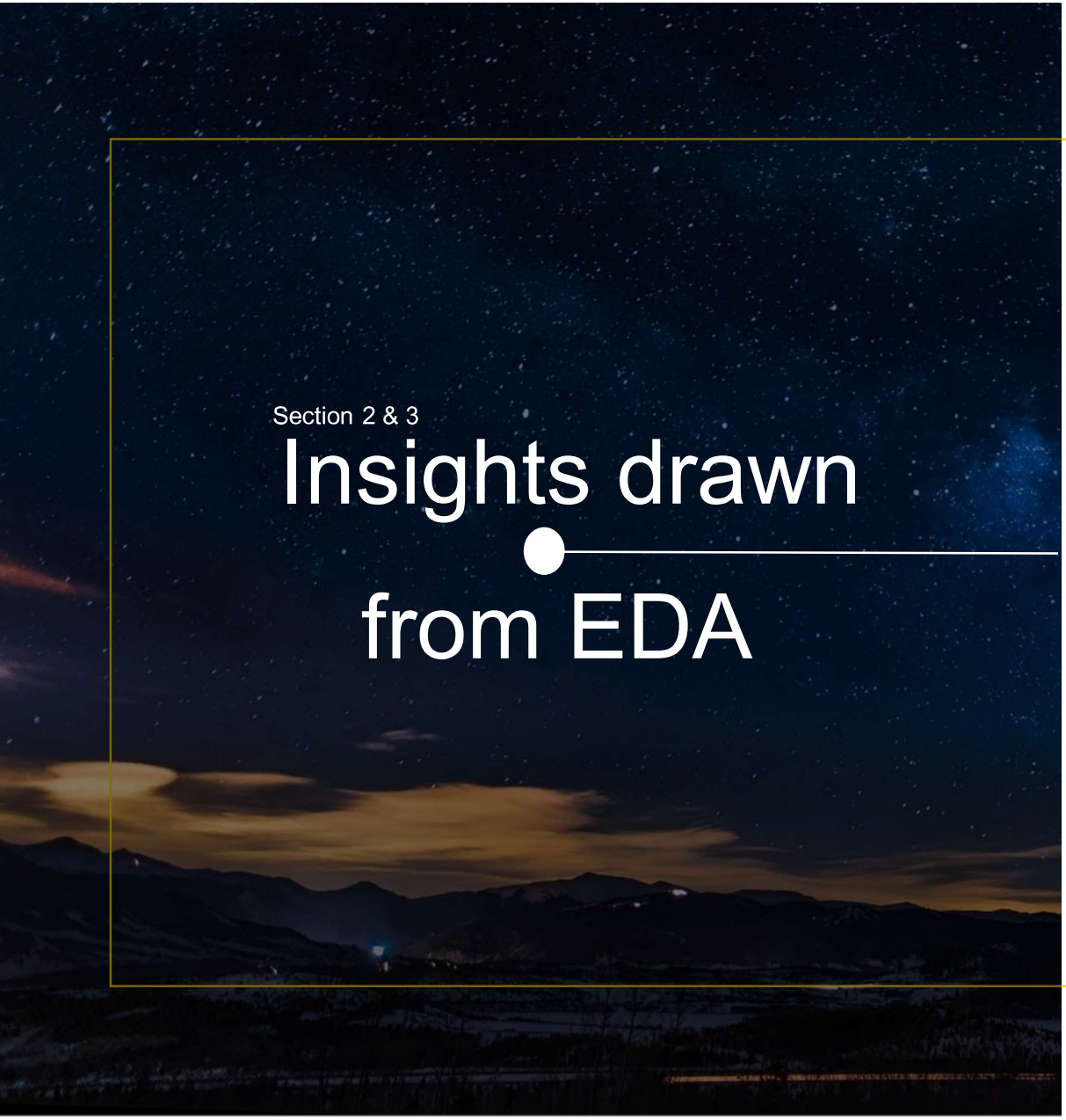
- The model with the best accuracy score (nearest to 1) is the best performing model.



Link to the notebook in GitHub: https://github.com/TRL2508/Data-Science-SpaceX-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5-solution.ipynb

Results

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



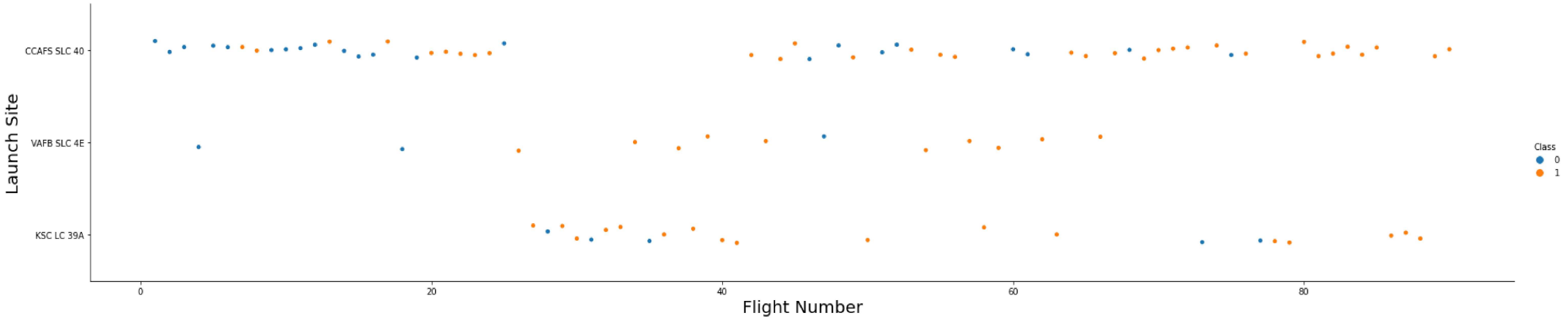
Section 2 & 3

Insights drawn from EDA

- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Success Rate vs. Orbit Type
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type
- Launch Success Yearly Trend
- All Launch Site Names
- Launch Site Name Begin with 'CCA'
- Total Payload Mass
- Average Payload Mass by F9 v1.1
- First Successful Ground Landing Data
- Successful Drone Ship Landing Payload between 4000 and 6000
- Total Number of Successful and failure Mission Outcomes
- Booster Carried Maximum Payload
- 2015 Launch Records
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

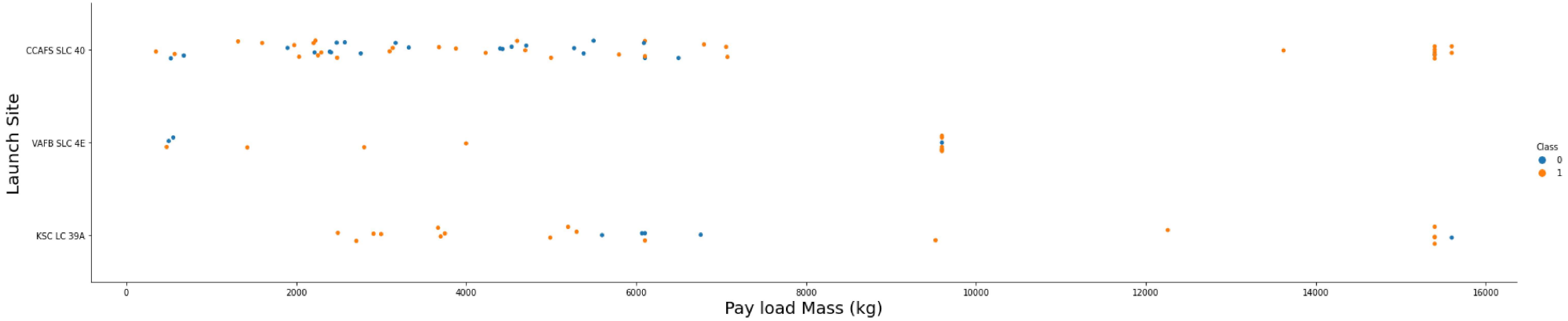
Flight Number vs. Launch Site

The scatter plot shows that the higher the number of flights, the higher the success rate at a launch site.



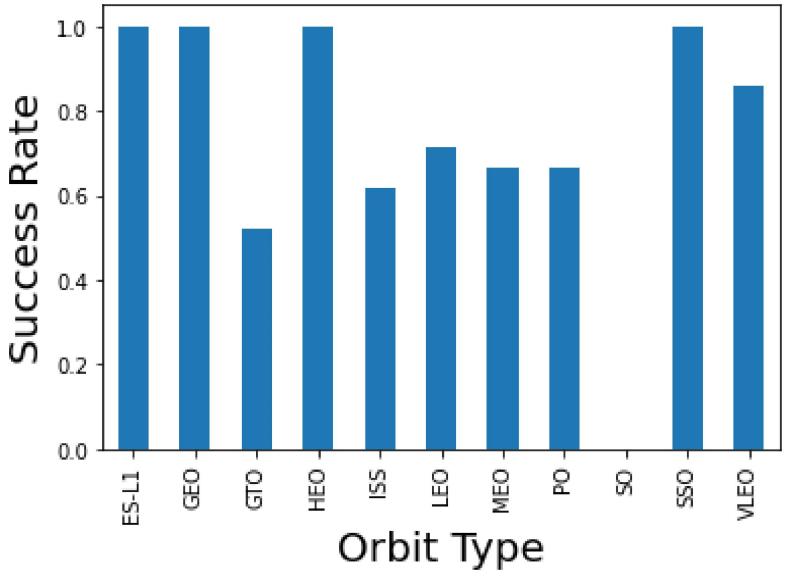
Payload vs. Launch Site

As compared to CCAFS SLC 40 and KSC LC 39A launch sites, VAFB SLC 4E launch site does not have any rocket launched for heavy payload mass (greater than 10,000 kg).



Success Rate vs. Orbit Type

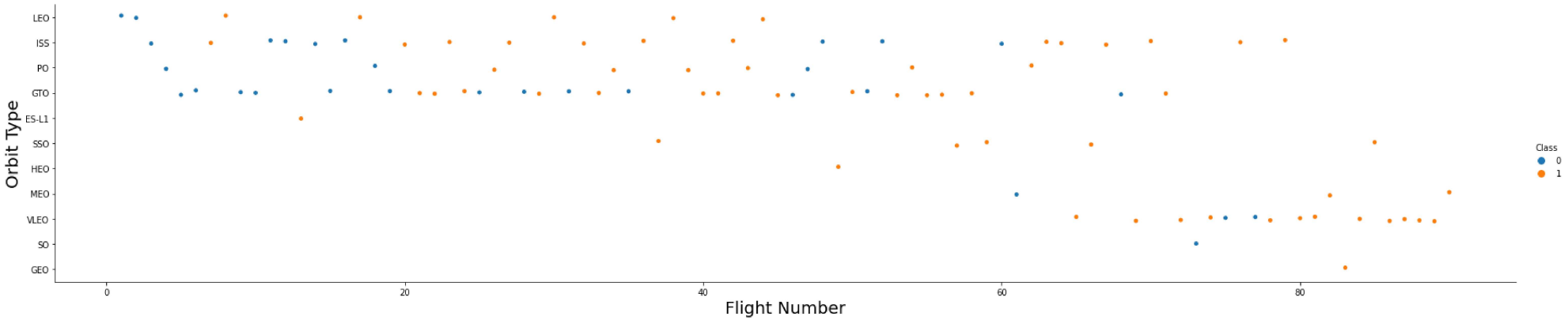
The bar chart shows that ES-L1, GEO, HEO and SSO orbits have the highest success rate.



Flight Number vs. Orbit Type

The scatter plot shows that:

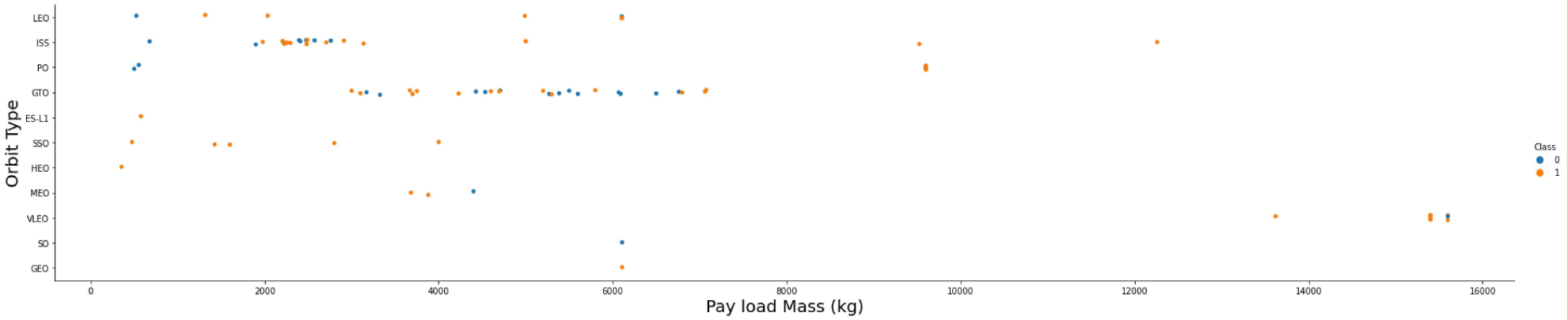
- In the LEO orbit, the success outcome appears to be related to the number of flights.
- There is no relationship between the flight number when in GTO orbit.



Payload vs. Orbit Type

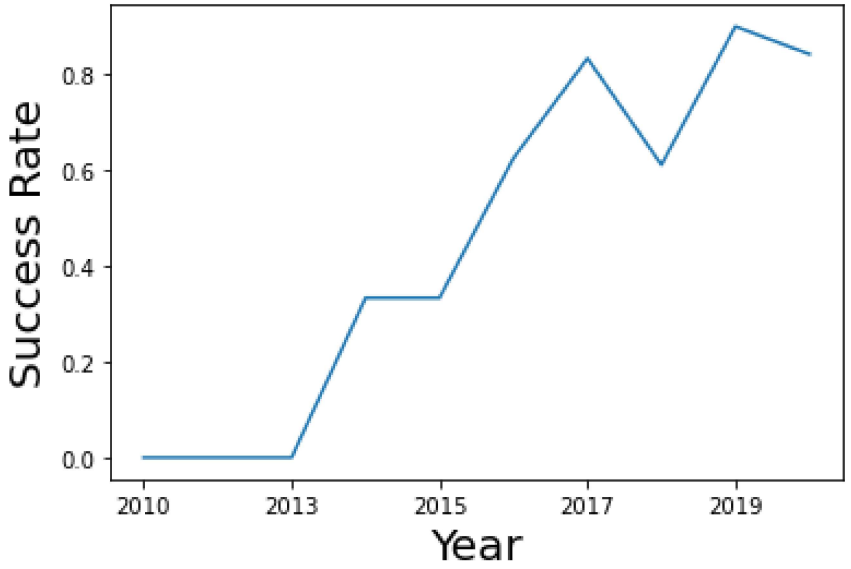
The scatter plot shows that:

- PO, LEO and ISS have a higher successful or positive landing rate with heavy payloads mass.
- GTO cannot be distinguish well as it have both positive and negative landing rate (positive – successful, negative – unsuccessful) plotted there.



Launch Success Yearly Trend

The line chart shows that the success rate since 2013 has been increasing till 2020.



All Launch Site Names

SQL Query:

```
select DISTINCT LAUNCH_SITE
from SPACEXTBL;
```

Result:

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

Explanation:

The 'select DISTINCT' statement is used to return distinct values in the result set.

The SQL query will show the distinct values from the 'LAUNCH_SITE' column in the 'SPACEXTBL' table.

Launch Site Names Begin with 'CCA'

SQL Query:

```
select * from SPACEXTBL
where LAUNCH_SITE
like 'CCA%' LIMIT 5;
```

Explanation:

The 'LIMIT' clause is used to specify the number of records to return.

The SQL query will show the first 5 records where data value in the 'LAUNCH_SITE' column, starts with 'CCA'.

Result:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

SQL Query:

```
select SUM(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD_MASS
from SPACEXTBL
where CUSTOMER = 'NASA (CRS)';
```

Result:

total_payload_mass
45596

Explanation:

The 'SUM()' function returns the total sum of the numeric column.

The SQL query will show the total sum of payload mass where data value in the 'CUSTOMER' column, is 'NASA (CRS)'.

The result column is renamed as 'TOTAL_PAYLOAD_MASS'.

Average Payload Mass by F9 v1.1

SQL Query:

```
select AVG(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_MASS
from SPACEXTBL
where BOOSTER_VERSION = 'F9 v1.1';
```

Result:

avg_payload_mass
2928

Explanation:

The 'AVG()' function returns the average value of the numeric column.

The SQL query will show the average payload mass where data value in the 'BOOSTER_VERSION' column, is 'F9 v1.1'.

The result column is renamed as 'AVG_PAYLOAD_MASS'.

First Successful Ground Landing Date

SQL Query:

```
select MIN(DATE) as DATE
from SPACEXTBL
where LANDING__OUTCOME = 'Success (ground pad)';
```

Result:

DATE
2015-12-22

Explanation:

The 'MIN()' function returns the smallest value of the selected column.

The SQL query will show the minimum date value (earliest date) where data value in the 'LANDING__OUTCOME' column, is 'Success (ground pad)'.

The result column is renamed as 'DATE'.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query:

```
select BOOSTER_VERSION
from SPACEXTBL
where (LANDING__OUTCOME = 'Success (drone ship)')
and (PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000);
```

Result:

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

Explanation:

The 'BETWEEN' command is used to select values within a given range.

The SQL query will show the values from the 'BOOSTER_VERSION' column where:

- Data value in the 'LANDING__OUTCOME' column is 'Success (drone ship)', and
- Data value in the 'PAYLOAD_MASS__KG_' column is between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

SQL Query:

```
select MISSION_OUTCOME, COUNT(*) as TOTAL_COUNT
from SPACEXTBL
GROUP BY (MISSION_OUTCOME);
```

Result:

mission_outcome	total_count
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Explanation:

The 'COUNT()' function returns the number of rows.

The 'GROUP BY' statement groups rows that have the same values into summary rows.

The SQL query will show the values from the 'MISSION_OUTCOME' column and the total number of rows grouped by the same values in the 'MISSION_OUTCOME' column.

The 'COUNT()' result column is renamed as 'TOTAL_COUNT'.

Boosters Carried Maximum Payload

SQL Query:

```
select DISTINCT BOOSTER_VERSION
from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

Explanation:

The 'select DISTINCT' statement is used to return distinct values in the result set.

The 'MAX()' function returns the largest value of selected column.

The subquery (nested query) is a query in another query.

The SQL query will show the distinct values from the 'BOOSTER_VERSION' column where the data value in the 'PAYLOAD_MASS__KG_' column is equal to the maximum value (the subquery is to find the maximum value of 'PAYLOAD_MASS__KG_' column).

Result:

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

2015 Launch Records

SQL Query:

```
select BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME, DATE
from SPACEXTBL
where LANDING__OUTCOME = 'Failure (drone ship)'
and YEAR(DATE) = '2015';
```

Result:

booster_version	launch_site	landing__outcome	DATE
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	2015-04-14

Explanation:

The 'YEAR()' function extract the year part from the specified date value.

The SQL query will show the values from the selected columns ('BOOSTER_VERSION', 'LAUNCH_SITE', 'LANDING__OUTCOME' and 'DATE' columns) where:

- Data value in the 'LANDING__OUTCOME' column is 'Failure (drone ship)', and
- Year value of the date in the 'DATE' column is '2015'.

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

SQL Query:

```
select LANDING__OUTCOME, COUNT(LANDING__OUTCOME) as TOTAL_COUNT
from SPACEXTBL
where DATE BETWEEN '2010-06-04' and '2017-03-20'
group by LANDING__OUTCOME
order by TOTAL_COUNT DESC;
```

Result:

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

Explanation:

The 'COUNT()' function returns the number of rows.

The 'BETWEEN' command is used to select values within a given range.

The 'GROUP BY' statement groups rows that have the same values into summary rows.

The 'ORDER BY' and 'DESC' keywords are used to sort the result-set in descending order.

The SQL query will show the values from the 'MISSION_OUTCOME' column and the total number of rows where:

- Data value in the 'DATE' column is between '2010-06-04' and '2017-03-20',
- Grouped by the same values in the 'LANDING__OUTCOME' column, and
- Based on the total number of rows (of different 'LANDING __ OUTCOME' groups) sort the result rows in descending order.

The 'COUNT()' result column is renamed as 'TOTAL_COUNT'.

Section 4

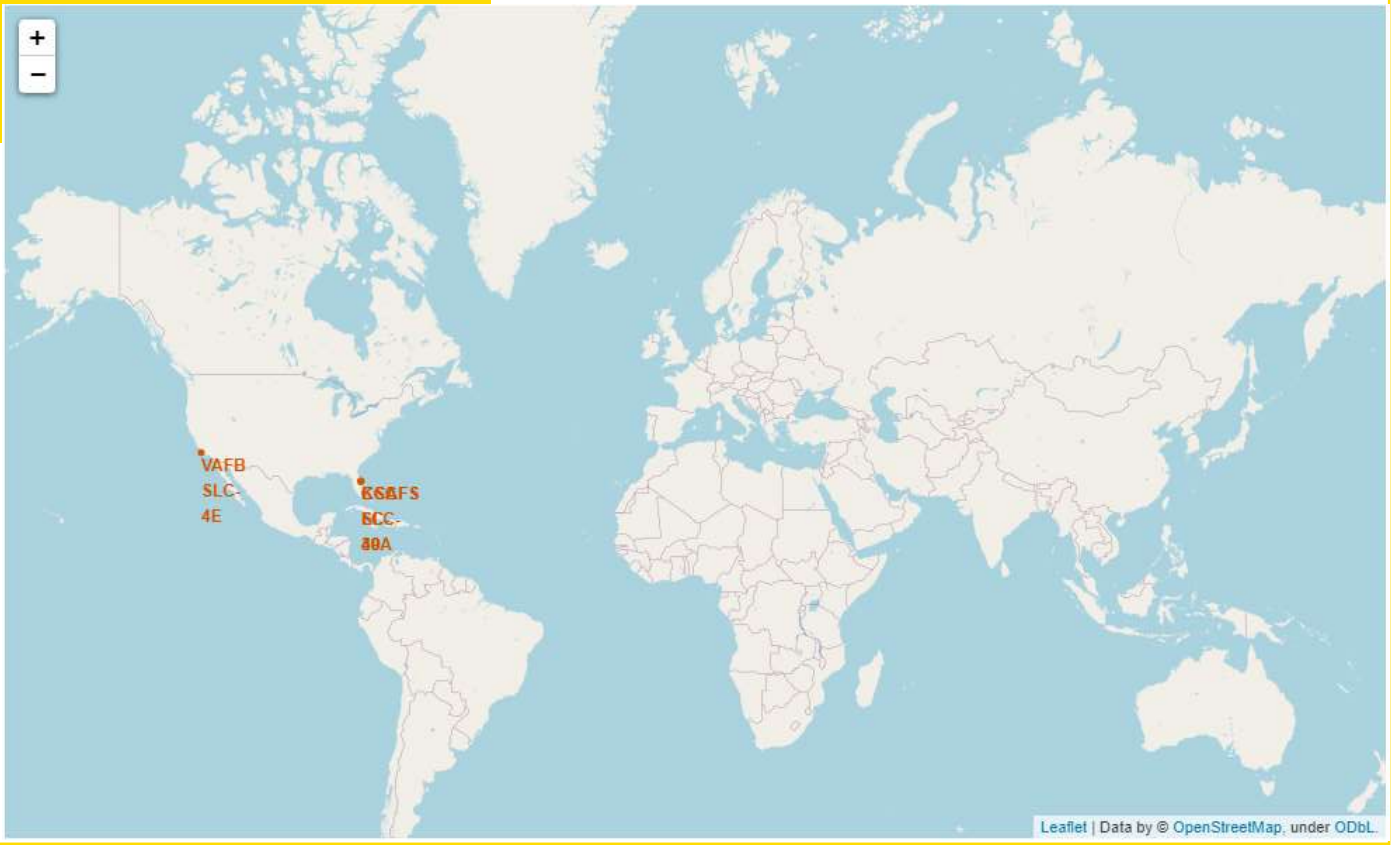
Launch Sites Proximities Analysis

- All Launch Sites' Location Markers on Global Map
- Color-Labeled Launch Outcomes on Map
- Selected Launch Site to Proximities

All Launch Sites' Location Markers on Global Map

All the launch sites are located in either East or West Coasts of the United States.

- VAFB SLC-4E is the only launch site located in California (West Coast of the United States).
- The other launch sites (CCAFS LC-40, CCAFS SLC-40 & KSC LC-39A) are located in Florida (East Coast of the United States).



Color-Labeled Launch Outcomes on Map

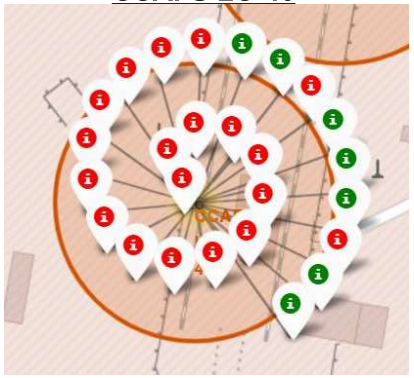
From the color-labeled markers in marker clusters, we are able to easily identify which launch sites have relatively high success rates.

KSC LC-39A has more green color markers, a higher success rate than the other launch sites.

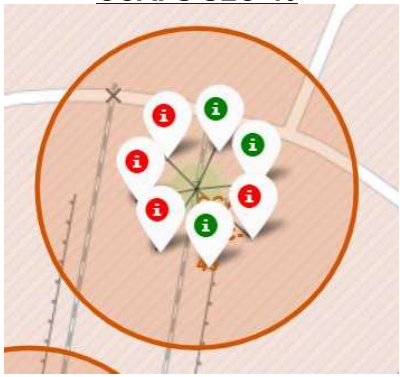
Marker colors represents:

- Green color marker show successful launch outcome.
- Red color marker show failure launch outcome.

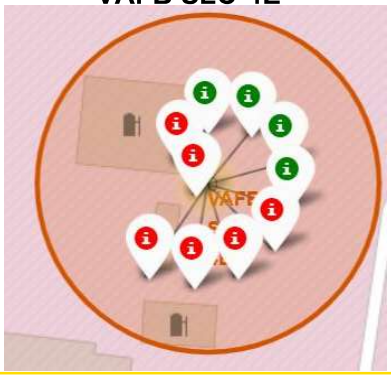
CCAFS LC-40



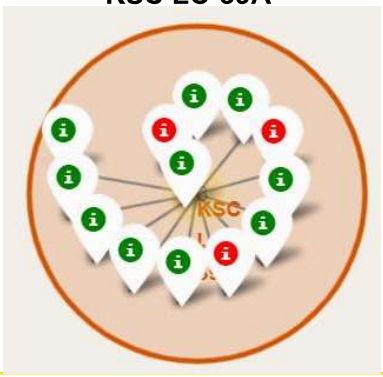
CCAFS SLC-40



VAFB SLC-4E



KSC LC-39A

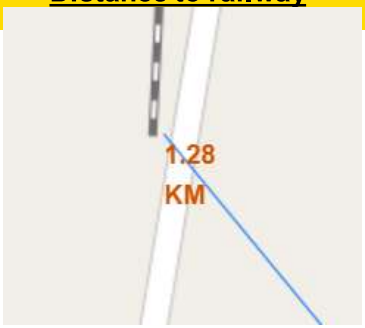


Selected Launch Site to Proximities

After the plot distance lines to the proximities, the launch site (CCAFS SLC-40) is:

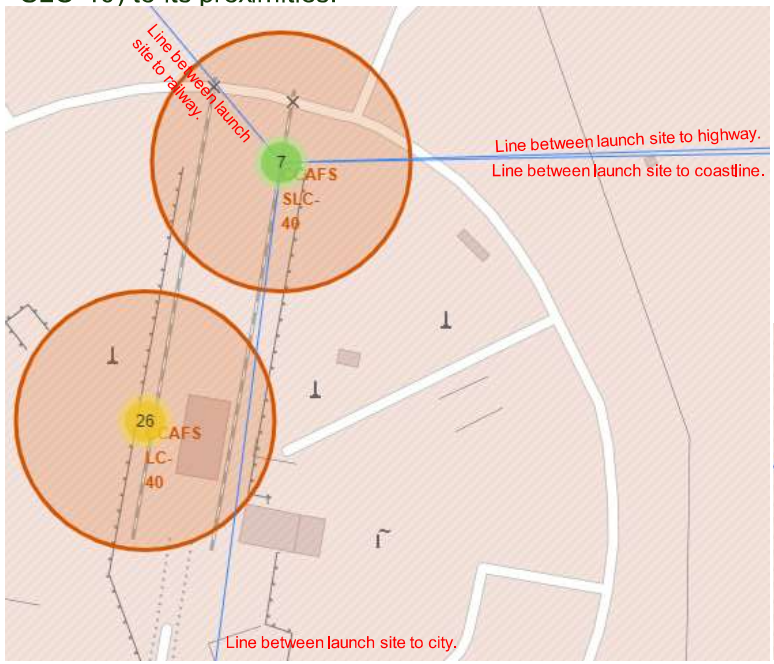
- close to the railway, highway and coastline.
- kept certain distance away from city.

Distance to railway



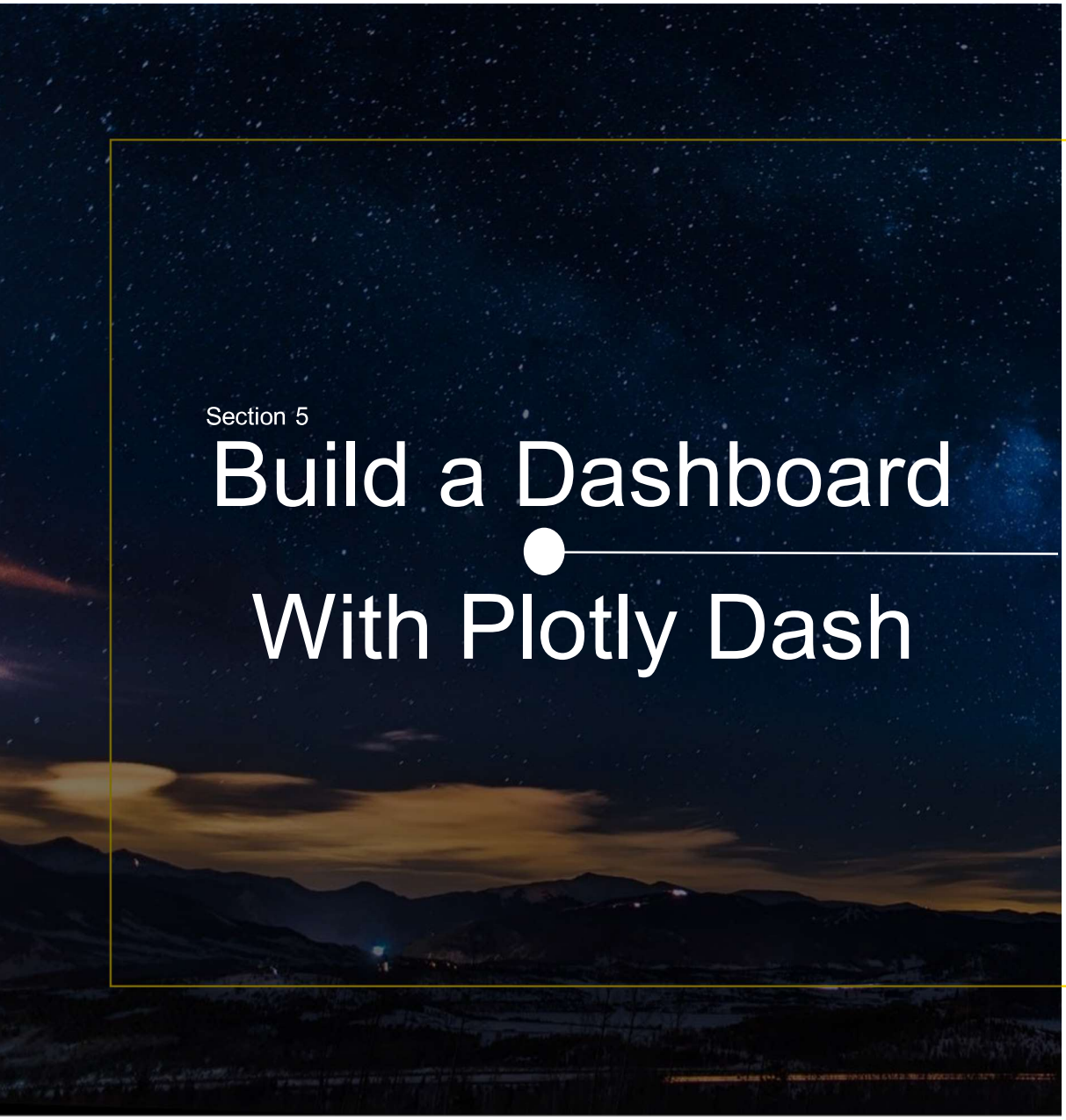
Distance to city

The distances between the selected launch site (CCAFS SLC-40) to its proximities:



Distance to highway

Distance to coastline



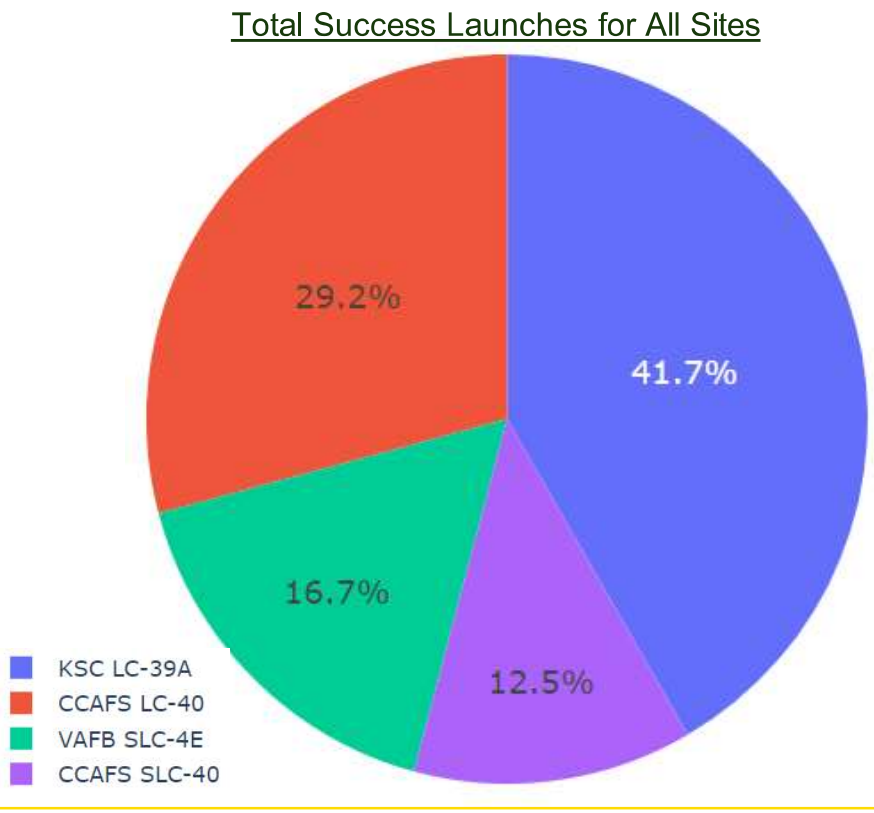
Section 5

Build a Dashboard With Plotly Dash

- Total Launch Success for All Sites in Pie Chart
- Launch Site with Highest Launch Success Ratio in Pie Chart
- Payload vs. Launch Outcome for All Sites, with different payload range in scatter plots

Total Launch Success for All Sites in Pie Chart

The pie chart shows that KSC LC-39A has the highest percentage of successful launches.



Launch Site with Highest Launch Success Ratio in Pie Chart

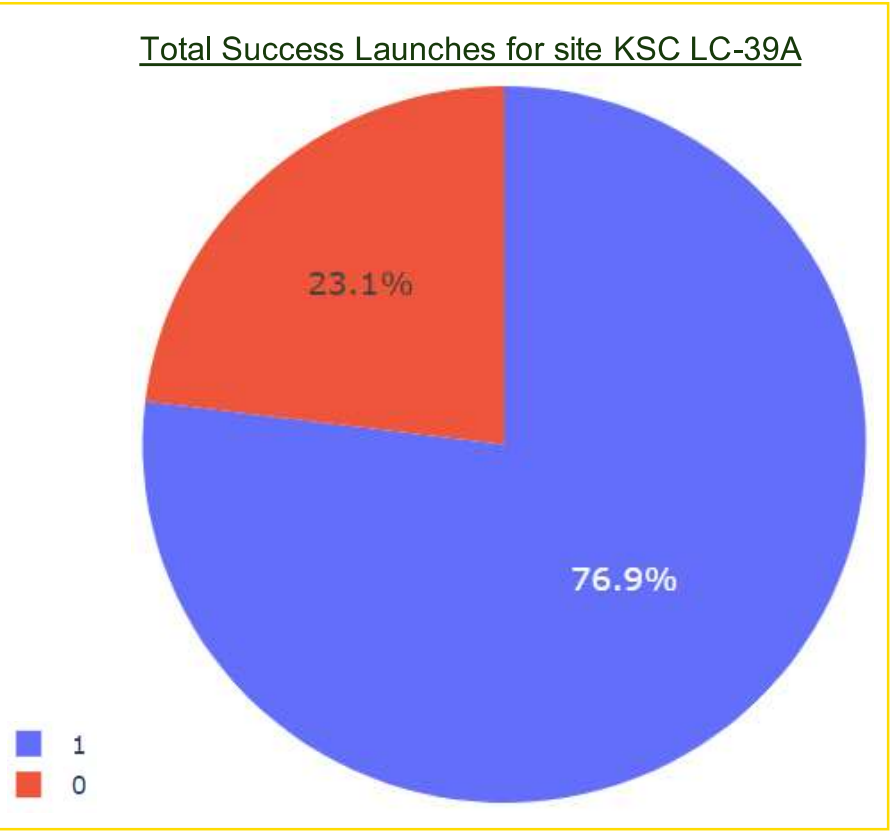
From the dashboard, we can quickly select the specified launch site from the dropdown list to view the percentage of successful and failure launches for the selected site from the pie chart.

Out of all the launch sites, KSC LC-39A has the highest launch success ratio.

Note:

Class '1' meant success launches.

Class '0' meant failure launches.



Payload vs. Launch Outcome for All Sites, with different payload range in scatter plots

From the dashboard, we can quickly select the minimum and maximum payload values in the range slider to view all the successful launches in the specified range.

Comparing the 2 scatter plots, lightweight payload has a higher number of successful launch outcomes.

From the 2 scatter plots, booster version FT have the largest success rate in both lightweight and heavyweight payload ranges. But booster version FT has a higher number of successful launch outcomes with lightweight payload.

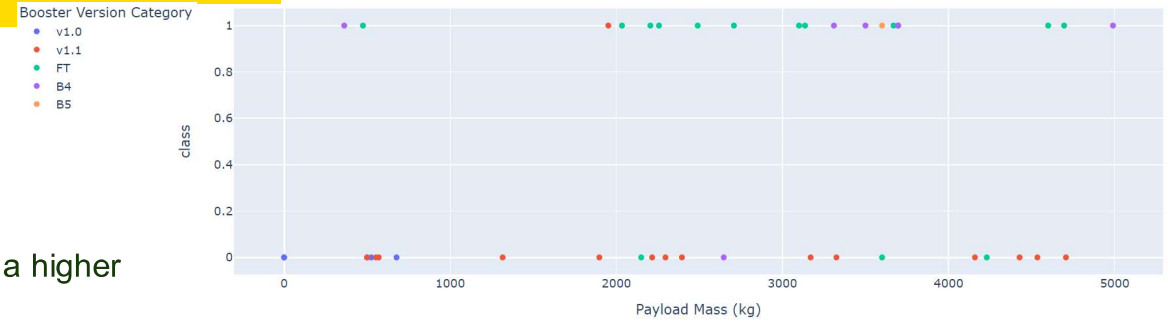
Note:

Class '1' meant success launches.

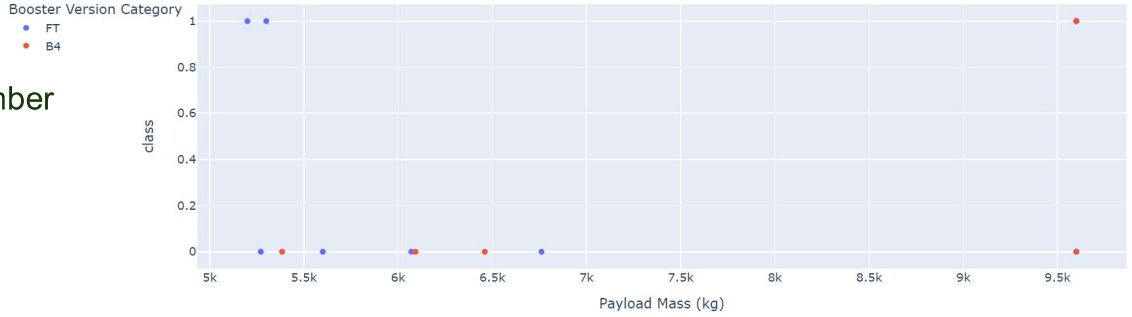
Class '0' meant failure launches.

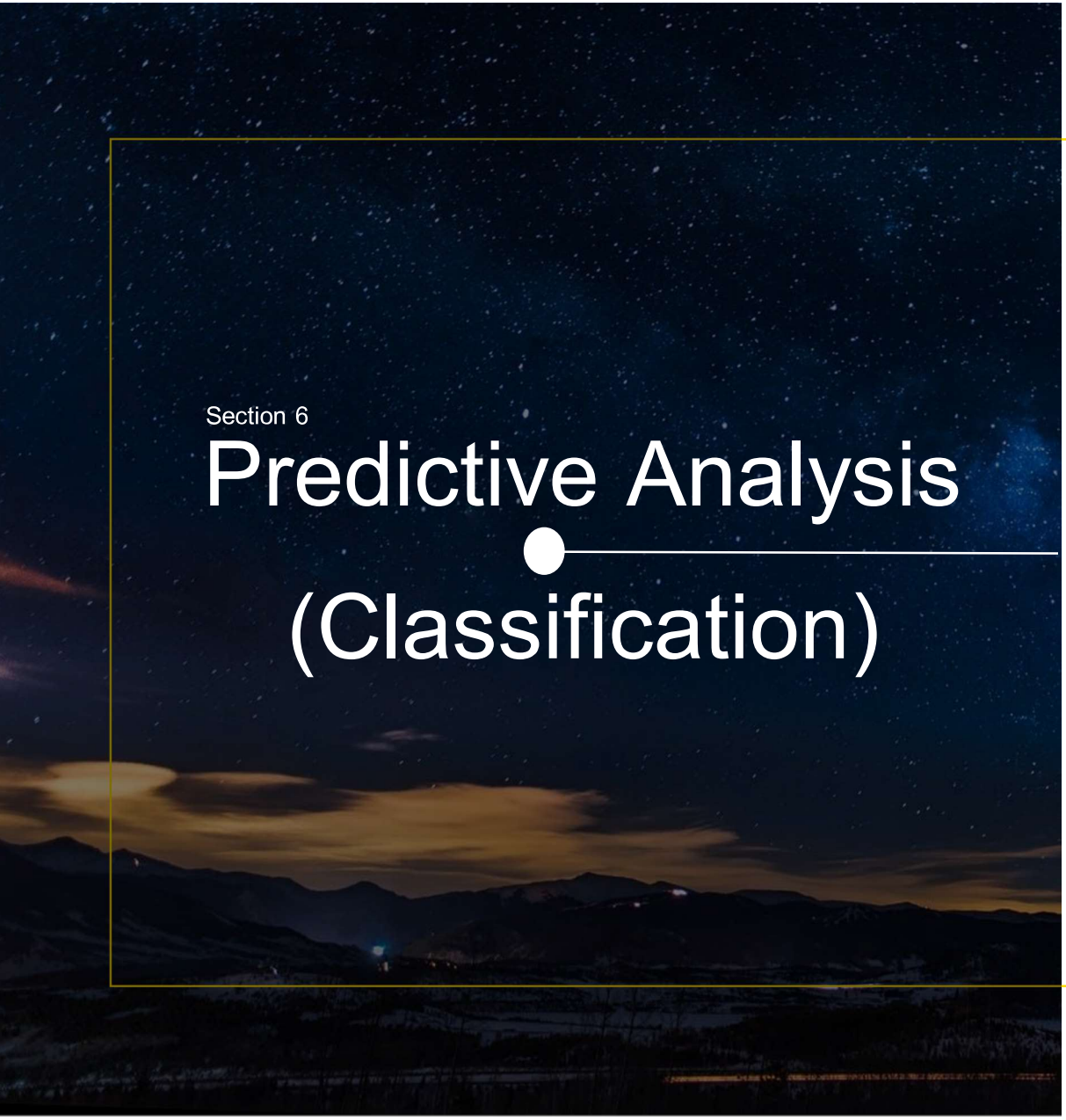
Correlation between Payload and Success for All Sites

Lightweight Payload (Range: 0kg to 5,000kg)



Heavyweight Payload (Range 5,000kg to 10,000kg)





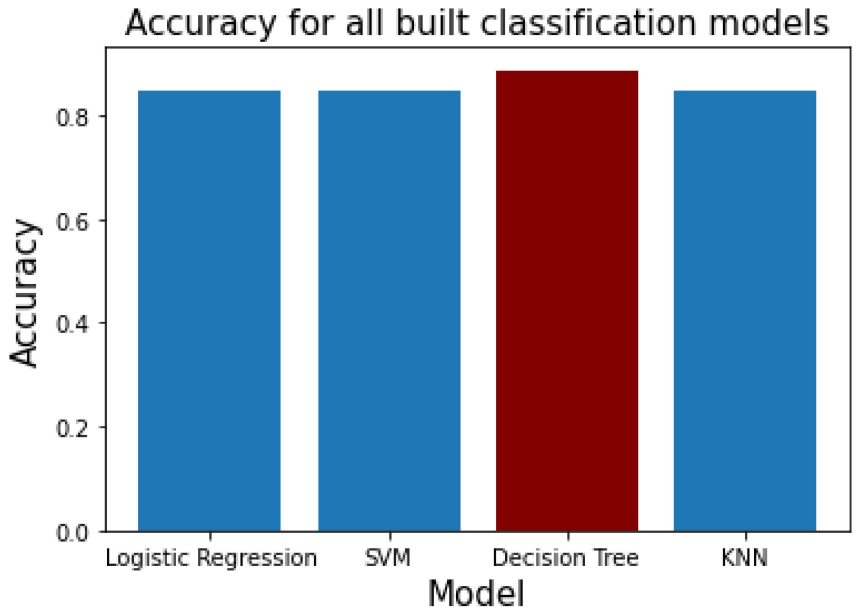
Section 6

Predictive Analysis (Classification)

- Classification Accuracy
- Confusion Matrix

Classification Accuracy

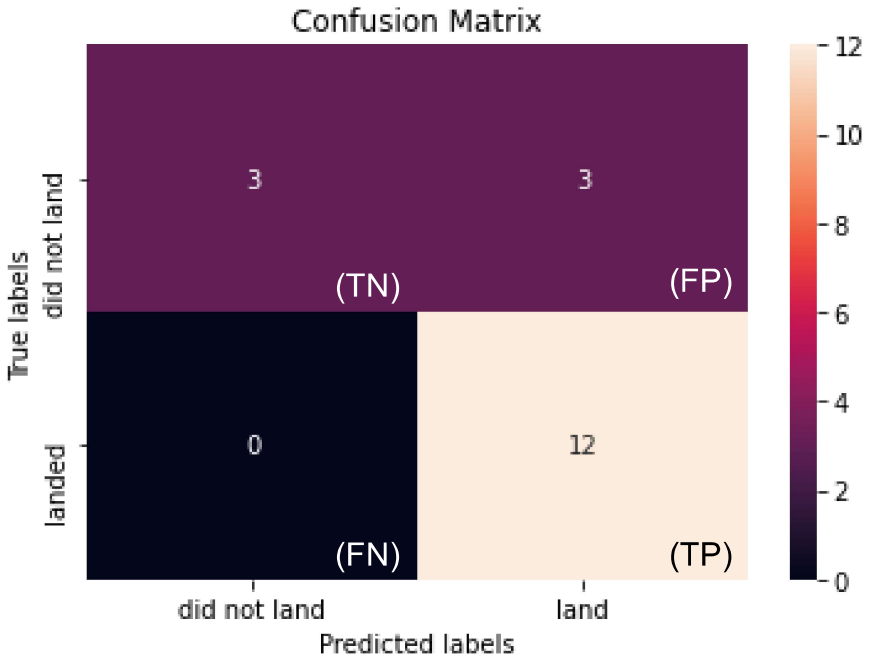
As shown from the bar chart, Decision Tree Classifier model has the highest classification accuracy among all built classification models.



Confusion Matrix

We can use the confusion matrix to evaluate how well the model (Decision Tree Classifier) is performing and what type of errors its making.

From the confusion matrix, we can see that 3 prediction was falsely predicted (false positive). The actual label should be "did not land", but the model predicted "landed".



Note: TN – True Negative, FN – False Negative, TP – True Positive, FP - False Positive

Conclusions

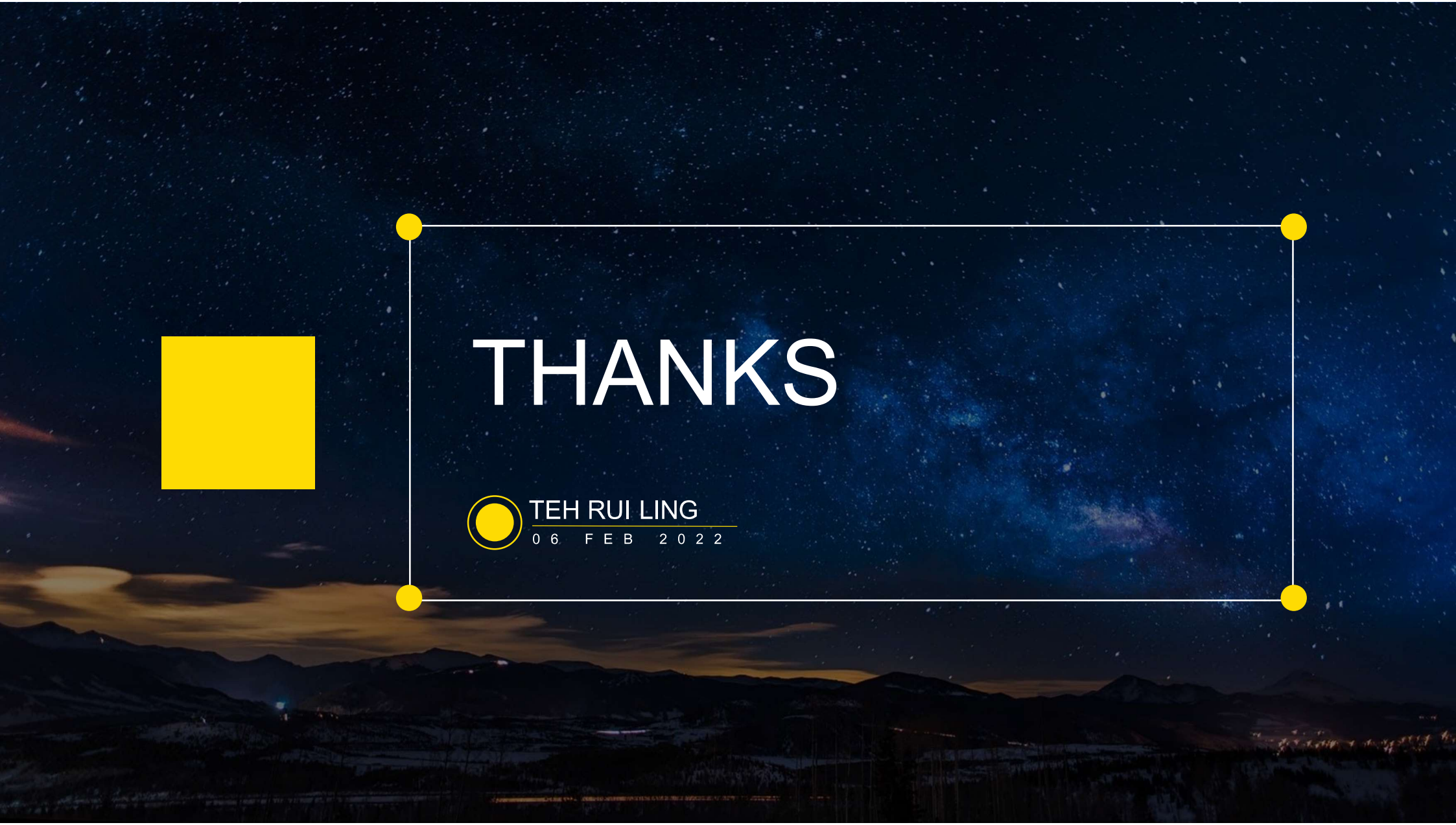
We concluded that:

- ES-L1, GEO, HEO and SSO orbits have the highest success rate.
- The success rate since 2013 has been increasing till 2020.
- KSC LC-39A has the most successful launch outcomes (highest launch success ratio and highest percentage of successful launches as compared to the other launch sites).
- The booster version FT has the largest success rate with lightweight payload.
- The best machine learning algorithm for the SpaceX dataset is the Decision Tree Classifier.

Appendix

All the python files (including SQL queries, charts, notebook outputs, datasets and CSV files are kept in GitHub.

Data-Science-SpaceX-Capstone: https://github.com/TRL2508/Data-Science-SpaceX-Capstone


THANKS
TEH RUI LING
06 FEB 2022