IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Han Hung
25 October 2023

Outline

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

Executive Summary

Several methodologies will be used to collect data like REST APIs and webscraping. After the data has been collected, data wrangling and analysis, or EDA, will be employed to prepare the data for use by the machine learning algorithms. Plots, bar charts, interactive maps, and a dashboard will be built to help find patterns and trends. And finally, several machine learning algorithms will be used to analyze the data and their accuracy scores compared to determine the best method to use.

The data needed are publicly available via SpaceX's REST APIs as well as their Wiki page. EDA was performed and data visualization was done on the results. Kennedy Space Center, or KSC, was identified as the best launch site, and several orbit types were identified as having the best launch outcomes. After training several models using different algorithms, the Decision Tree method performed the best with an 87.5% accuracy.

Introduction

Billionaire Allon Mask wants to start a new company, Space Y, to compete with Space X.

The goal of this project is to determine if the first stage of the rocket will land successfully or not. If the first stage can be recovered, it can be reused. Space Y can save money and charge less per flight.

Currently, Space X charges $62 million per launch. This is because they can land the first stage and reuse it for another mission. Competitors charge $165 million or more for a similar mission because their stages are single use.

To further this goal, a model will be built using Machine Learning to predict if the first stage can be recovered or not.

Another goal is to identify the best site to launch flights.

Allon Mask

SPACE Y

Section 1

Methodology

Methodology

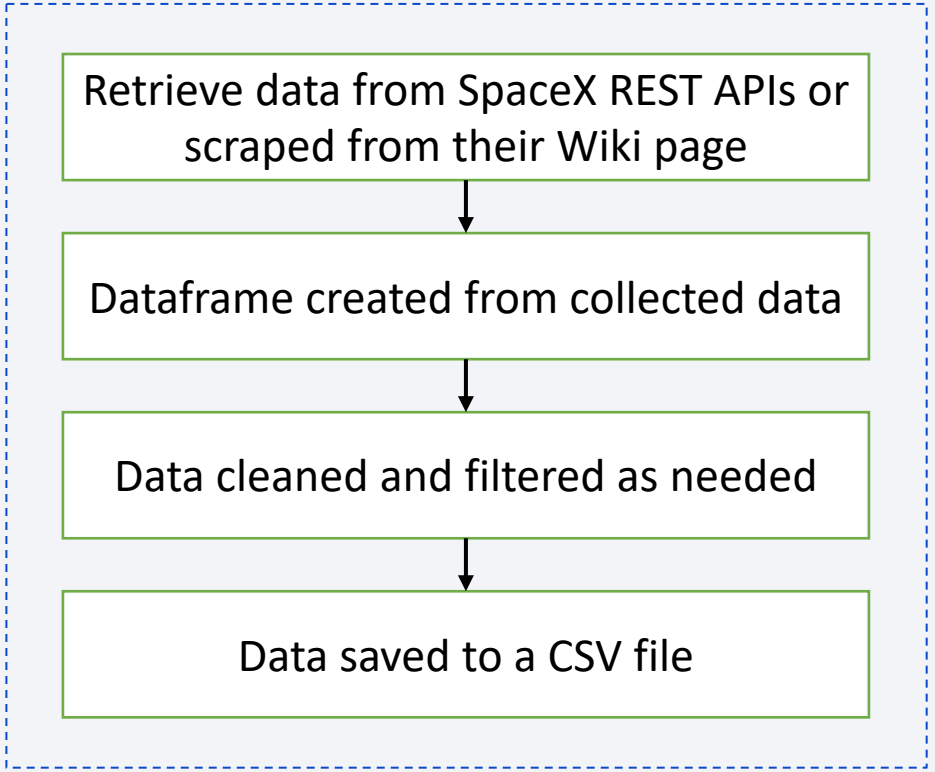
- Data collection methodology:
  - Retrieve data from SpaceX server using their public REST APIs.
  - Retrieve data from their Wiki page.
- Perform data wrangling
  - Categorical values were converted to numeric.
  - Null values were transmuted.
  - Irrelevant columns were dropped.
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Scatter plots, bar charts, and a line plot were created to show patterns and trends.
  - SQL queries were used to extract data for analysis.
- Perform interactive visual analytics using Folium and Plotly Dash
  - An interactive dashboard was created to show launch outcomes by launch site.
  - Folium maps were created to show locations of launch sites as well as launch outcomes and distances to nearest public landmarks.
- Perform predictive analysis using classification models
  - Several models were built using different algorithms.
  - Their accuracy was computed and compared and the one with the best accuracy score was chosen.

Data Collection

The data sets needed for this project reside in a few places. SpaceX has a public REST API server that contains many data. Their Wiki page also contains useful data for this project.

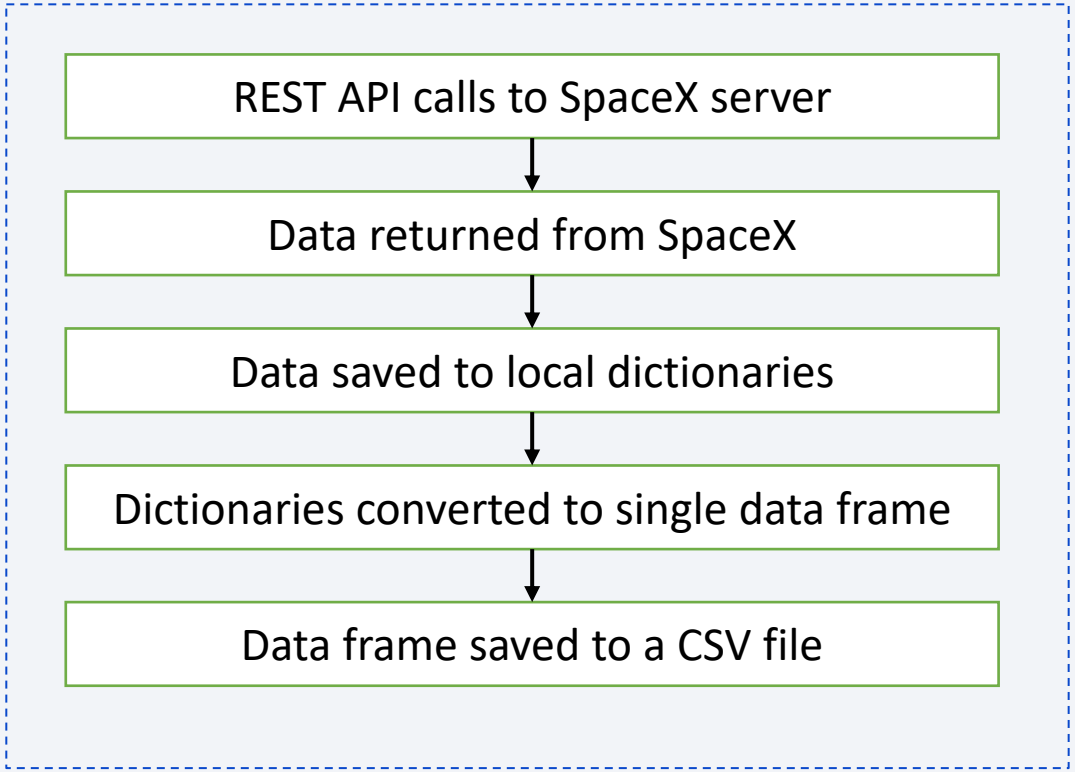
The data was collected using a combination of REST API calls to SpaceX, as well as scraping the data from their Wiki page.

The general flow for data collection is on the right.



Data Collection – SpaceX API

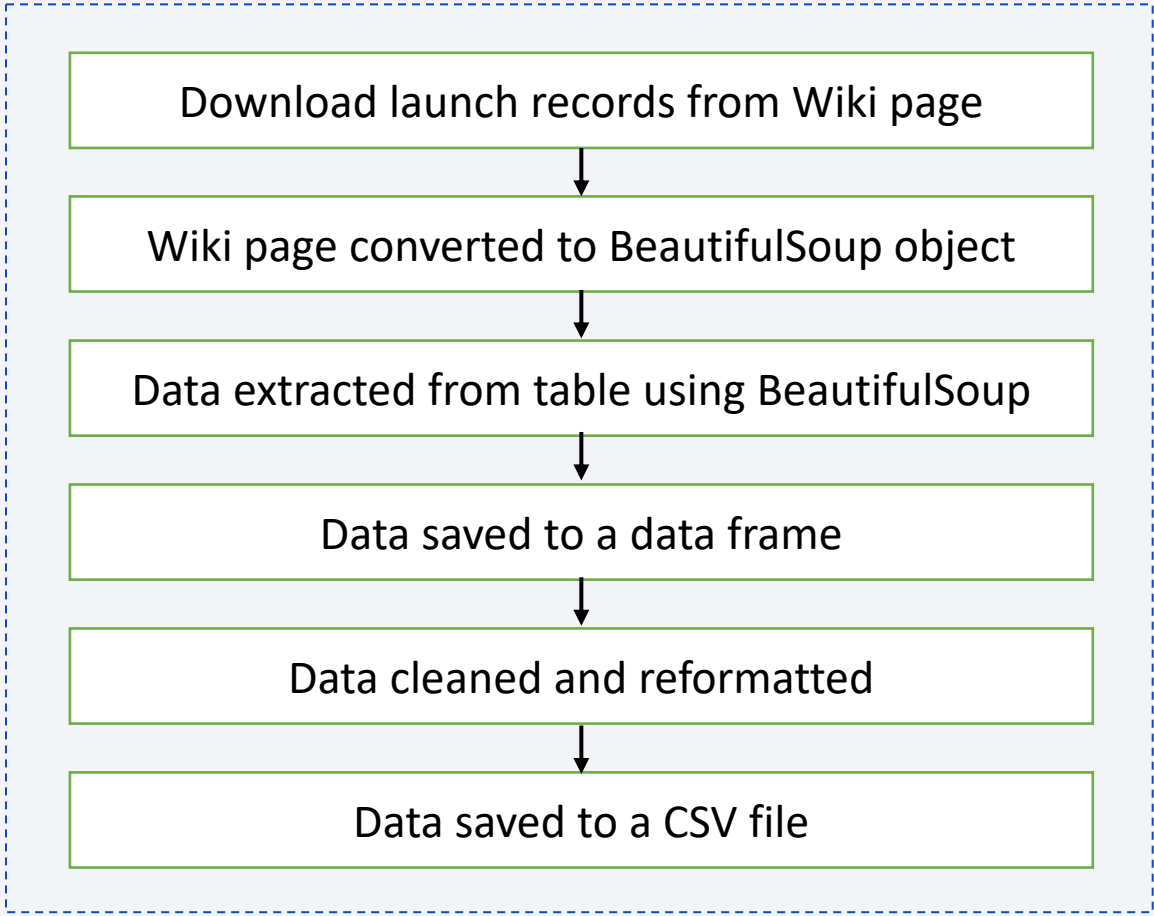
- To the right is a flowchart showing the steps taken to retrieve the data from SpaceX's website.
- This is a list of data retrieved:
  - List of past launches
  - Booster versions
  - Launch sites
  - Payload data
  - Core data like outcome, flight, reuse, etc.
- The GitHub repository for the notebook can be found here.



https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_1_lab_1_jupyter-labs-spacex-data-collection-api.final.ipynb

Data Collection - Scraping

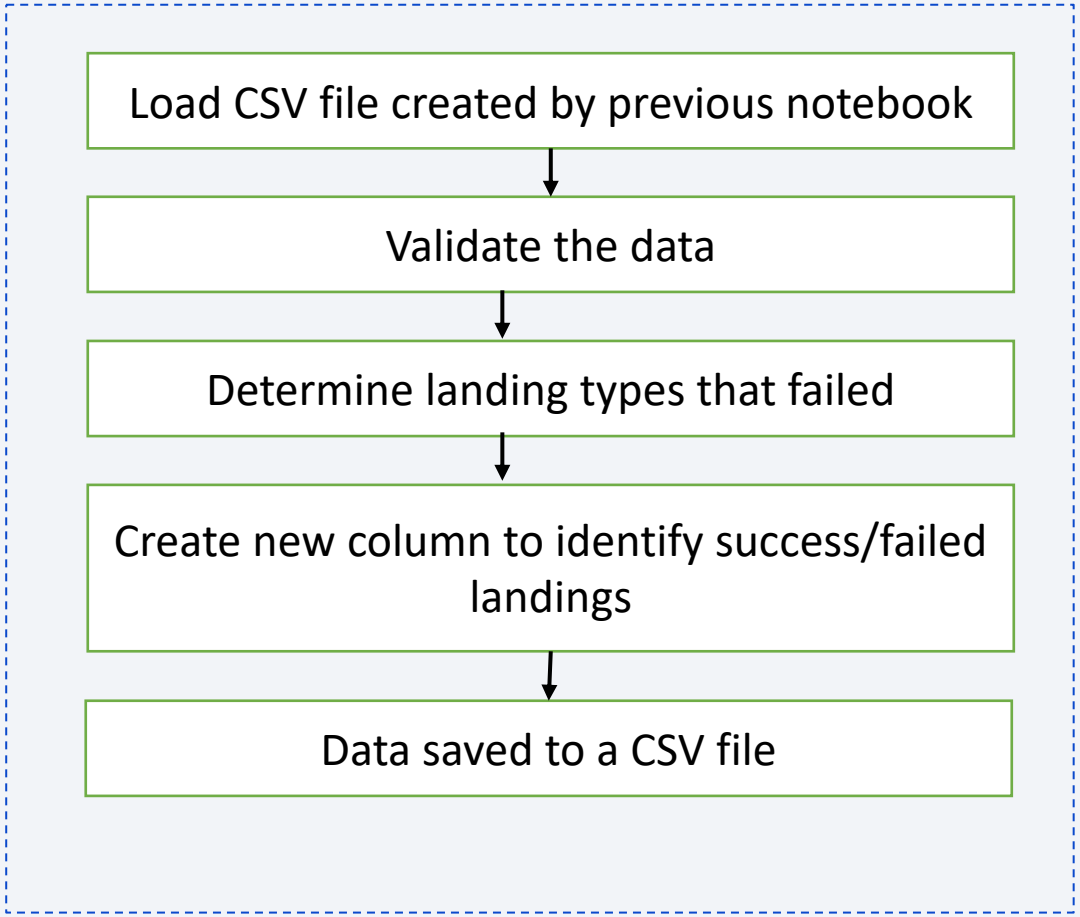
- The records for all of SpaceX launches were located on their Wiki page.
- The page was downloaded and a BeautifulSoup object was created to allow for easy navigation in Python.
- Data from the relevant table was converted into a Pandas data frame.
- The data was cleaned and reformatted.
- Finally, the data frame was saved to a CSV file.
- The GitHub repository can be found here.



https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_1_lab_2_jupyter-labs-webscraping.final.ipynb

Data Wrangling

- Data wrangling is about transforming the raw data into more readily usable formats. For example, converting a date and time string into an actual date and time data type, removing unnecessary columns, identifying null values and either removing them or transmuting them into something useful, converting categorical data into numerical ones, or creating new columns based on existing columns.
- What was done:
  - Replaced null values for payload with the average, known payloads.
  - Create a new 'Class' column based on the landing outcome.
    - 0 represents a failed landing.
    - 1 represents a successful landing.
- Once completed, the data frame was saved to a CSV file.
- The GitHub repository can be found here.



https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_1_lab_3_jupyter-spacex-Data%20wrangling.final.ipynb

EDA with Data Visualization

- Visualizing data is one of the most useful methods to understanding the data. It makes it easier to find patterns, trends, and insights from the data.
- The following table shows the charts and plots that were created.

Data Displayed	Chart or Plot Type	Purpose
Flight Number vs. Payload Mass	Scatter Plot	Payload mass increased over time
Flight Number vs. Launch site	Scatter Plot	SpaceX launches from 4 different locations. This shows which launch site was used the most as well as which flights were successful or not.
Payload Mass vs. Launch Site	Scatter Plot	Shows which launch sites carried how much payload.
Class vs. Orbit Type	Cat Plot	Shows success/failure by orbit type.
Success Rate vs. Orbit Type	Bar plot	Shows which orbit type had the most successful mission outcomes.
Flight Number vs. Orbit Type	Scatter Plot	Shows which orbit type originated from which flight.
Payload mass vs. Orbit Type	Scatter Plot	Shows the payload mass for each orbit type.
Success rate by Year	Line Chart	Shows the success trend over the years.

- The GitHub repository can be found here.

https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_2_lab_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.final.ipynb

EDA with SQL

- SQL is a useful tool because most data in the real-world are stored in databases.
- The following SQL queries were performed:
  - Display the names of the unique launch sites
  - Display 5 records where the launch site begins with 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display the average payload mass carried by booster 'F9 v1.1'
  - List the date when the first successful landing on the ground pad was achieved
  - List the names of the boosters which have success landing on drone ships and have a payload mass greater than 4,000 but less than 6,000
  - List the total number of successful and failed mission outcomes
  - List the names of the booster versions which have carried the maximum payload mass
  - List the records which will display the month names, failure landing on drone ship, booster versions, and launch sites for the year 2015
  - Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order
- The GitHub repository can be found here.

https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_2_lab_1_jupyter-labs-eda-sql-coursera_sqllite.final.ipynb

Build an Interactive Map with Folium

- Folium makes it easy to create interactive maps in a Jupyter notebook that contain markers, lines, icons, etc. to identify points of interest.

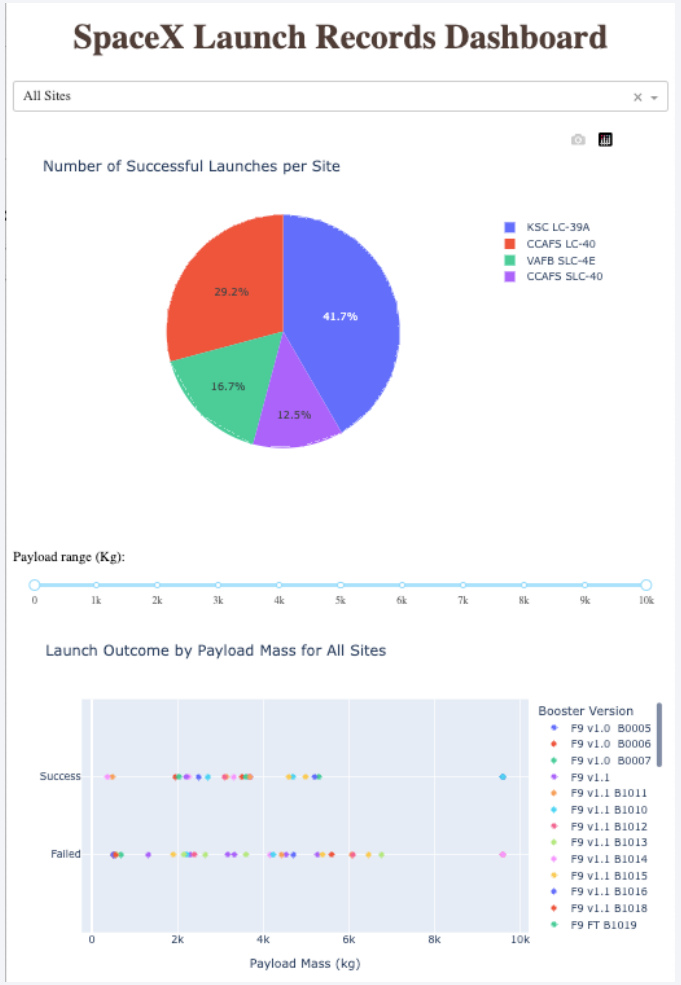
Folium Object	Purpose
Map	This creates the map object where child objects like circles, icons, etc. can be added.
Circle	Add a highlight circle, in map units, around a specific coordinate.
Marker	Add a fixed sized marker object around a specific coordinate.
Popup	Add a popup label to an object when the user clicks on a marker.
CircleMarker	Combination of a circle and marker object.
MarkerCluster	Use this object when there are multiple objects with the same coordinates. It will display the number of items at that coordinate when zoomed out. When zoomed in, it will expand to display all the objects around that coordinate.
PolyLine	Draws a line between two or more coordinates
Icon	Draws an icon on the map at the specified coordinate.

- The GitHub repository can be found here.
- Note: When viewing the file in GitHub, the maps were not displayed.

https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_3_lab_1_lab_jupyter_launch_site_location.jupyterlite.final.ipynb

Build a Dashboard with Plotly Dash

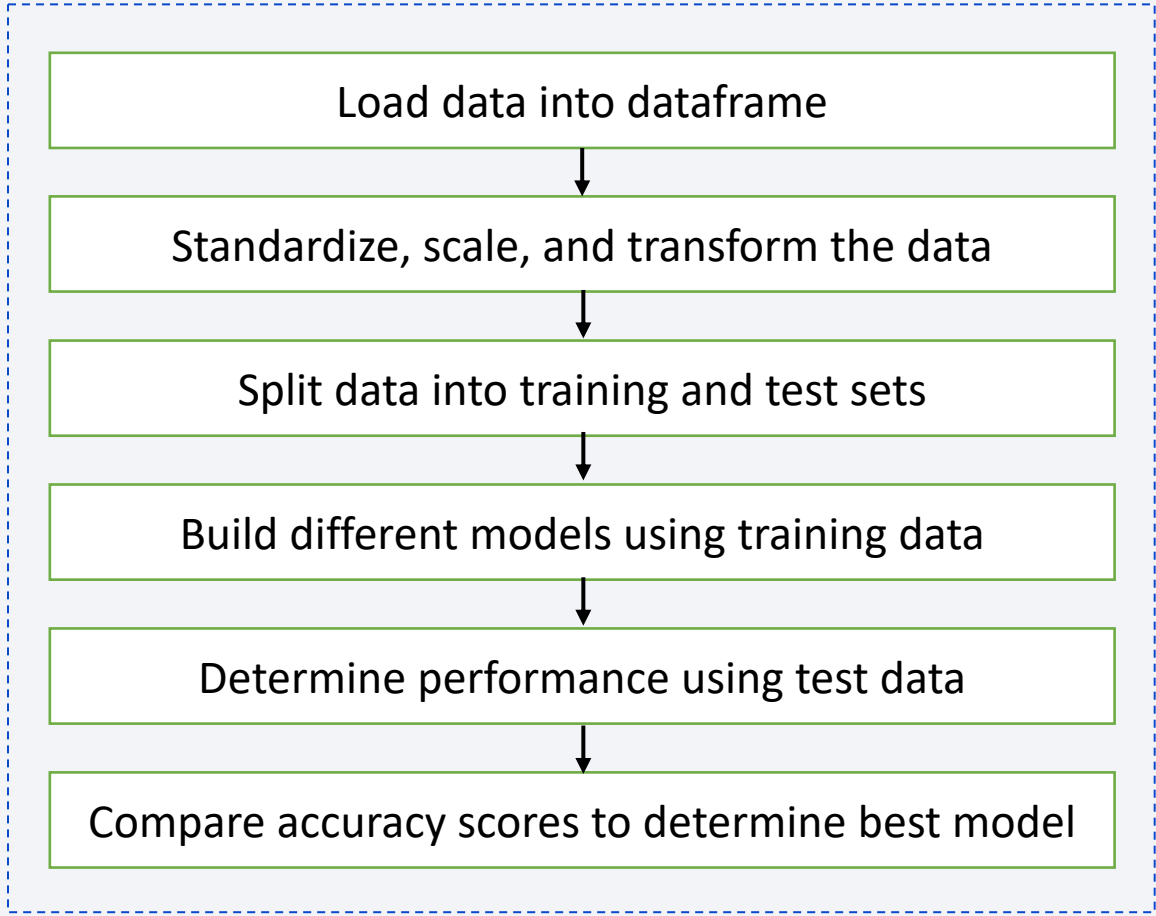
- Dashboards provide an interactive way to present the data.
- Elements used in the dashboard:
  - Dropdown list to select which launch site to display.
  - Range slider to adjust payload mass to display in the scatter plot.
  - Pie chart showing the success to failure ratios for launch sites.
  - Scatter plot showing the launch outcome by payload mass for the selected site or all sites.
- When ‘All’ is selected, the pie chart will display the number of successful launches at each site and the scatter plot will display which booster versions were successful or not for all sites.
- If a site is selected, the pie chart will display the ratio of failed to successful launches for the selected site and the scatter plot will display which booster version was successful or not for the selected site.
- The range slider is used to change the upper and lower range of payload to display in the scatter plot.
- The GitHub repository can be found here.



https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_3_lab_2_spacex_dash_app.final.py

Predictive Analysis (Classification)

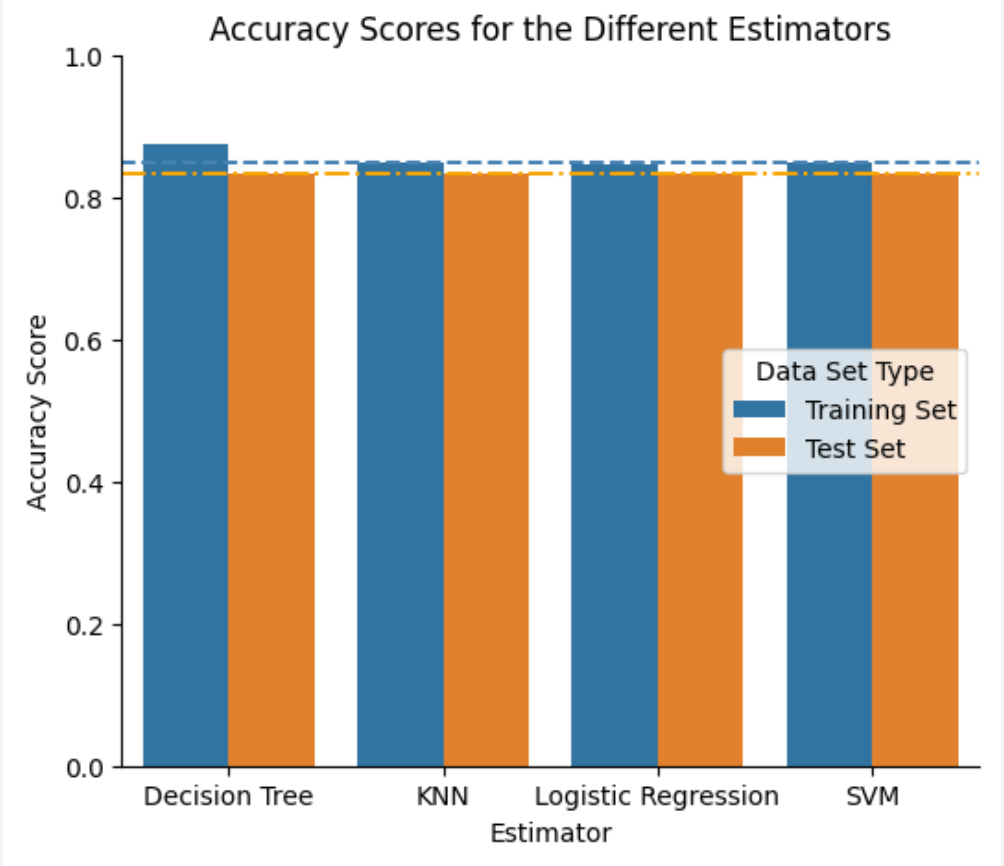
- Once all the data has been collected, the following steps were performed to find the best model:
  - Standardize, scale, and transform the data.
  - Split data into training and test sets using 80/20 split ratio.
  - Build different models using the training data
  - Determine performance for each model using the test data
  - Compare the accuracy scores for each model
  - Pick the model with the best scores.
- See flow chart on the right for an overview of the model development process.
- The GitHub repository can be found here.

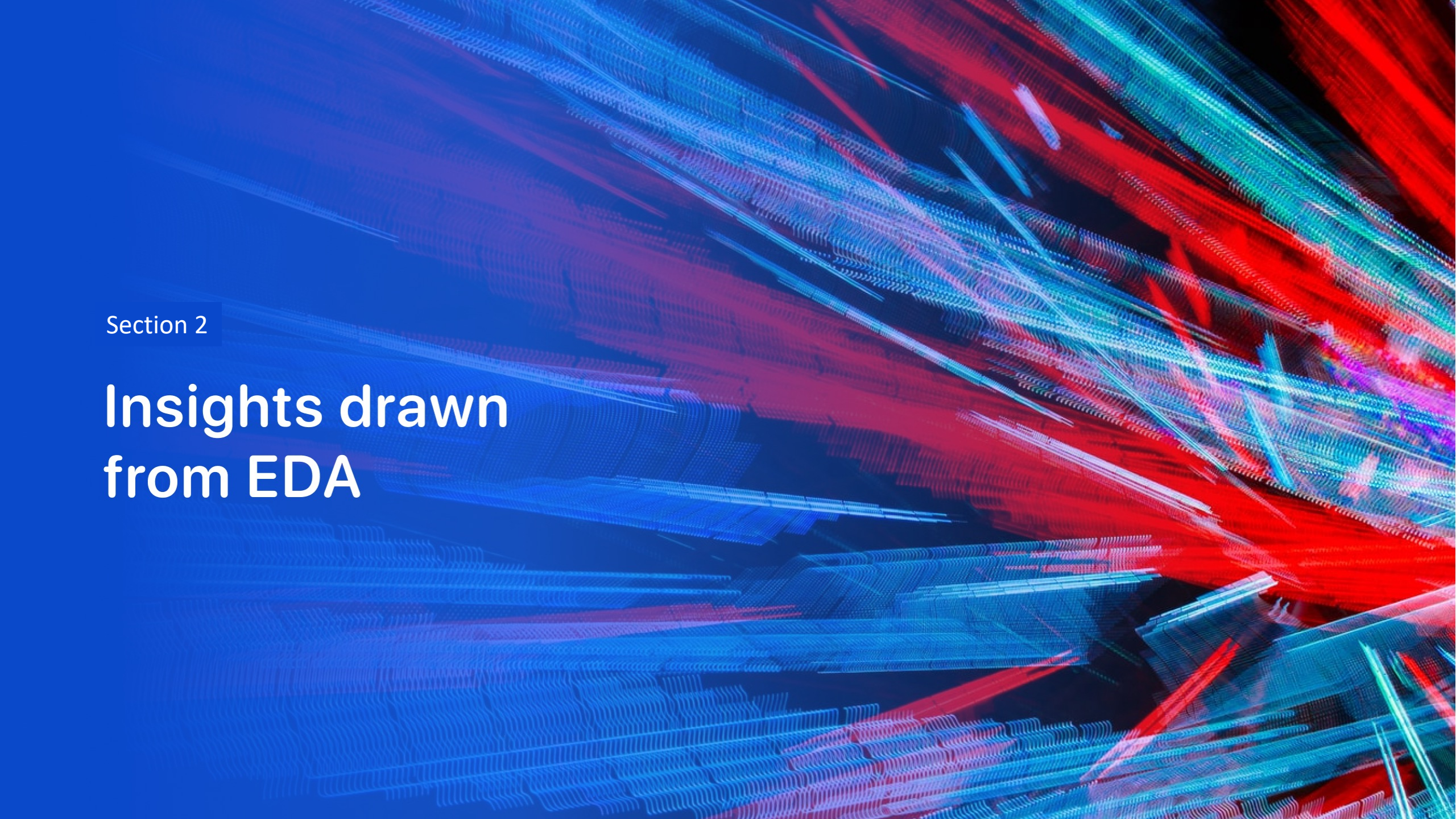


https://github.com/hhung01/data-science-capstone/blob/main/notebooks_and_data/Week_4_lab_1_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.final.ipynb

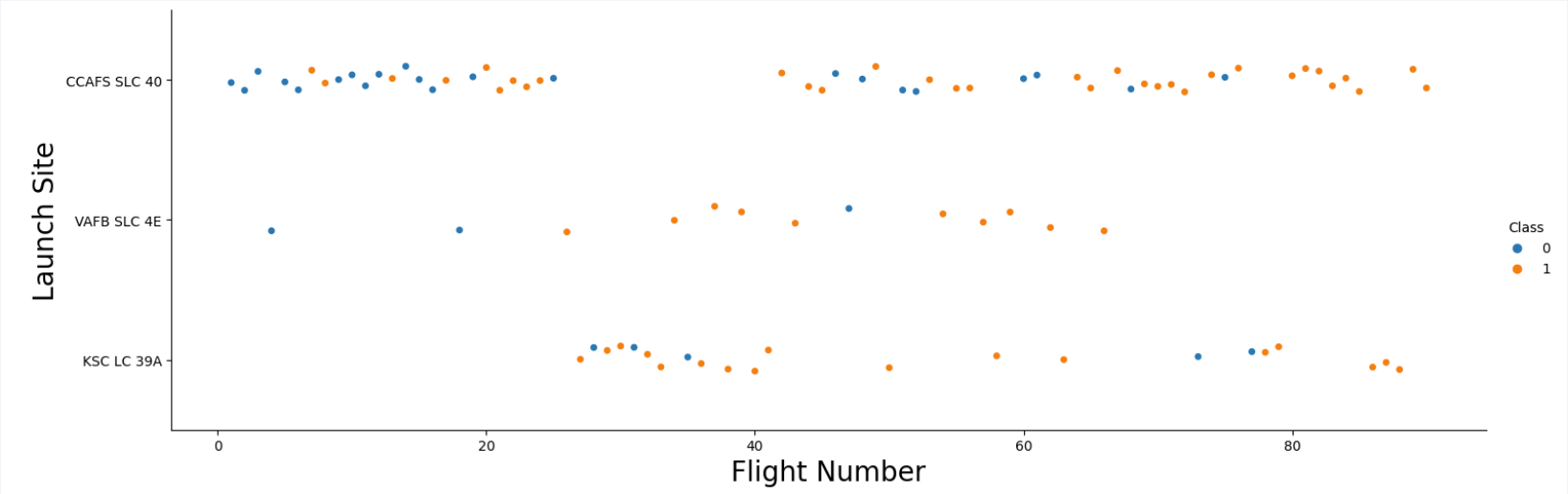
Results

- The following algorithms were evaluated:
  - Logistic Regression (LR)
  - Decision Trees (DT)
  - Support Vector Machine (SVM)
  - K-Nearest Neighbor (KNN)
- After evaluating the different algorithms, LR, SVM, and KNN all produced similar accuracy scores using training data.
- The DT algorithm scored the best using training data.
- All 4 algorithms scored the same using test data.
- Because DT had the bests score, it will be used to predict the launch outcome for future flights.



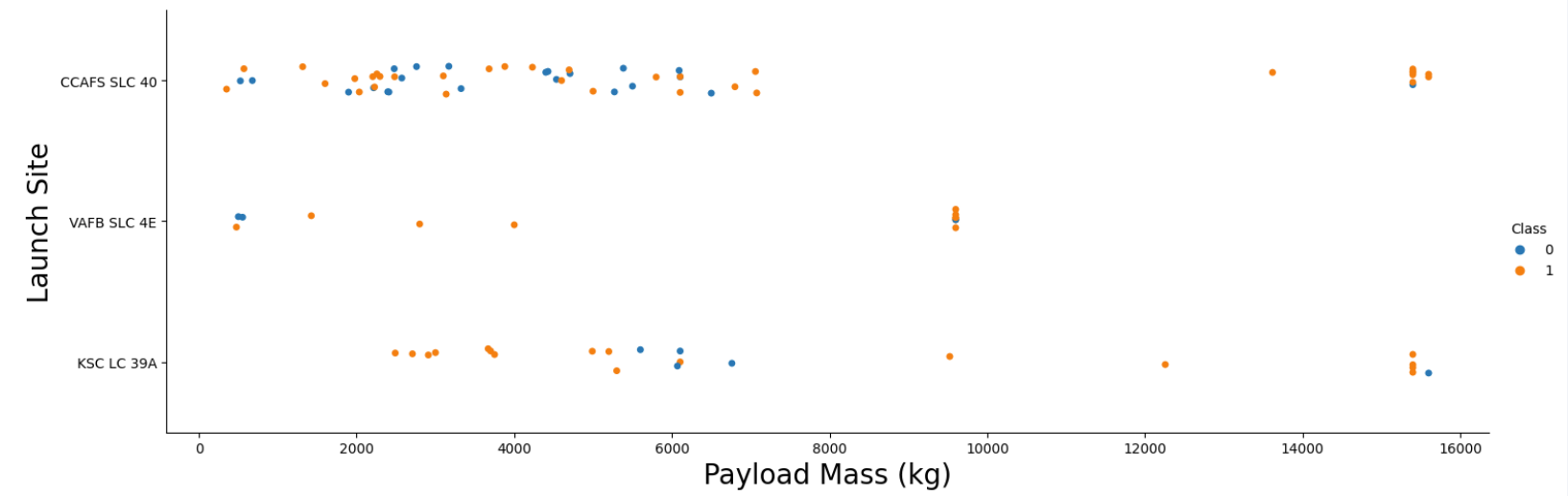

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



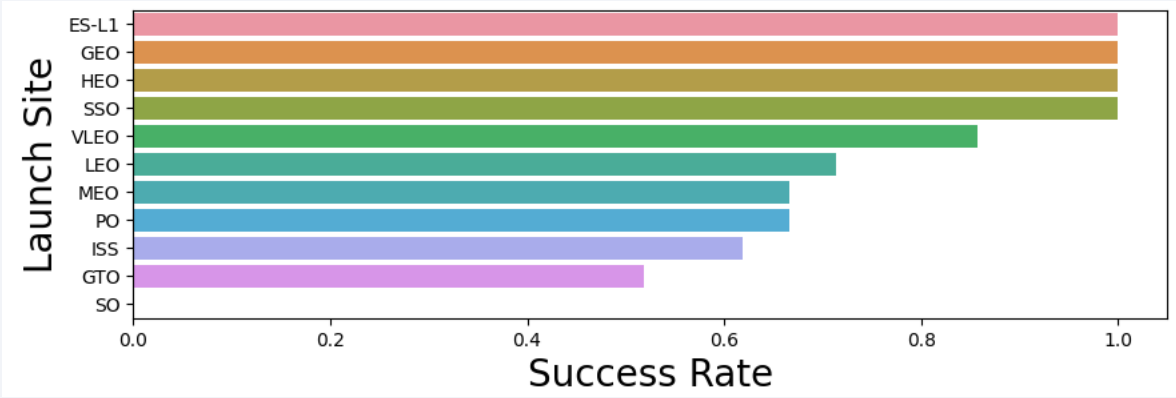
- The blue dots are Class 0, or flights with failed outcomes.
- The orange dots are Class 1, or flights with successful outcomes.
- SpaceX started launching rockets from CCAFS SLC-40, with some launches from VAFB SLC-4E.
- At one point, they switched to using KSC LC-39A before switching back to CCAFS SLC-40 for most of their flights.
- At the beginning, most of the flights had failed outcomes. But their success rate increased over time.

Payload vs. Launch Site



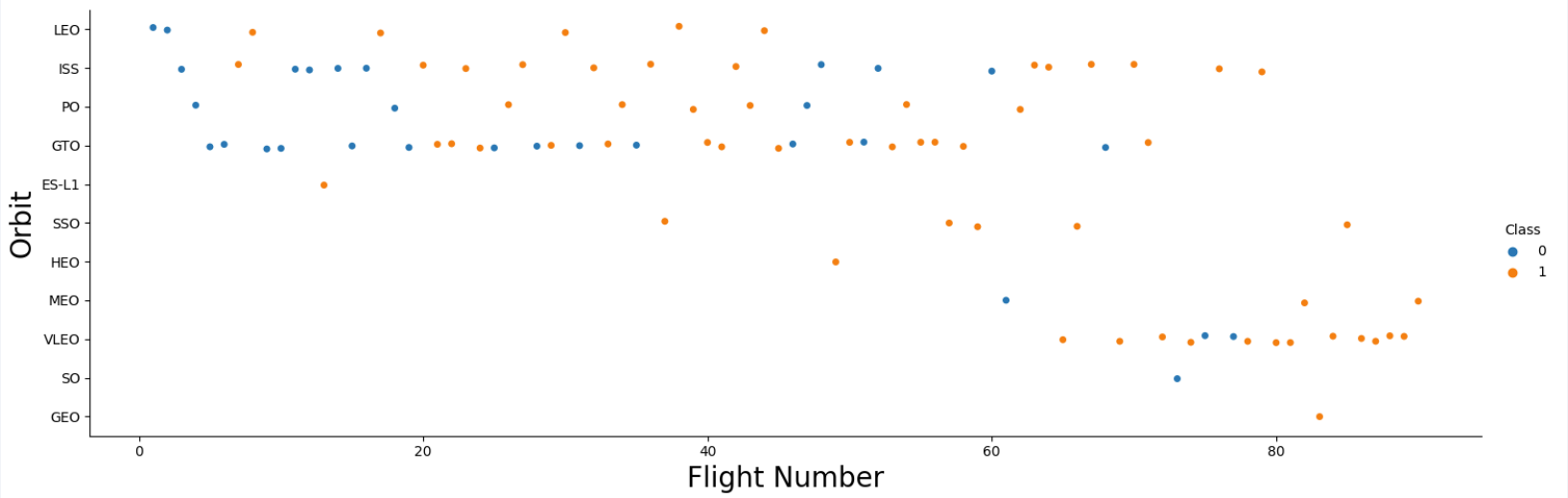
- A lot of flights with payloads less than 8,000 kg originated from CCAFS SLC-40.
- CCAFS SLC-40 and KSC LC-39A launched the heaviest payloads (those greater than 10,000 kg).
- The heaviest payload launched from VAFB SLC-4E is around 9,600 kg.

Success Rate vs. Orbit Type



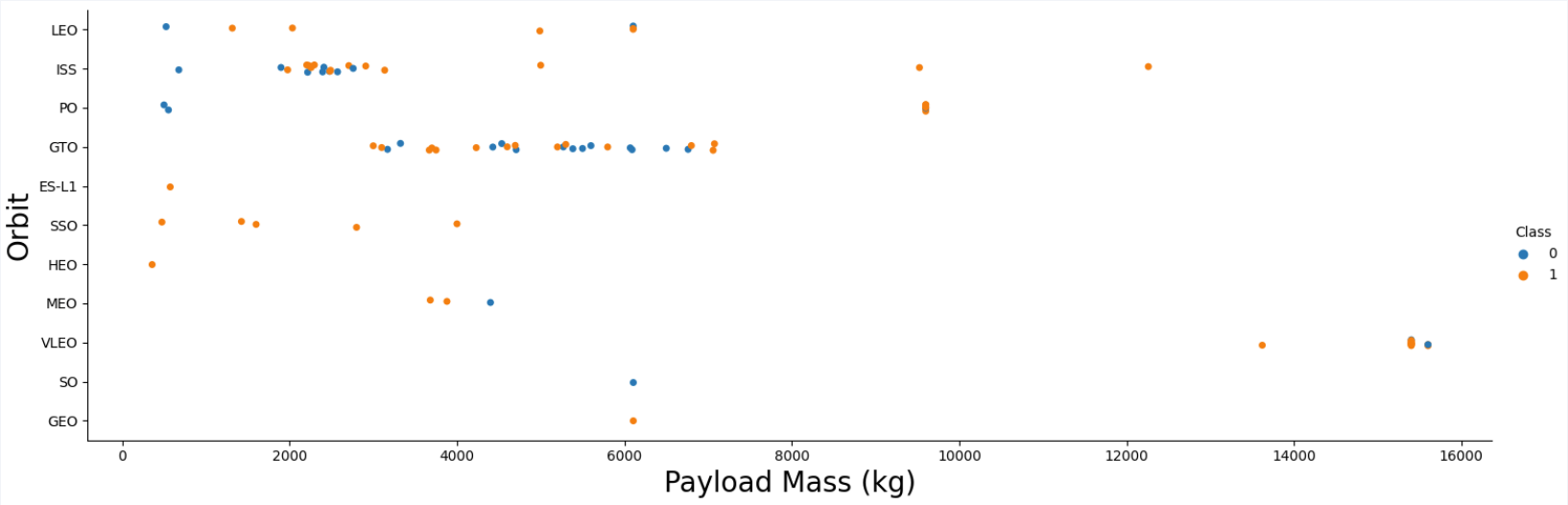
- The orbit types with the most successful flights were ES-L1, GEO, HEO, and SSO with a success rate of 100%.
- The least successful orbit type is SO, with a success rate of 0%.

Flight Number vs. Orbit Type



- At the beginning, most flights had orbit types of LEO, ISS, PO, and GTO.
- Over time, other orbit types were added.
- Later flights concentrated on VLEO orbit type.

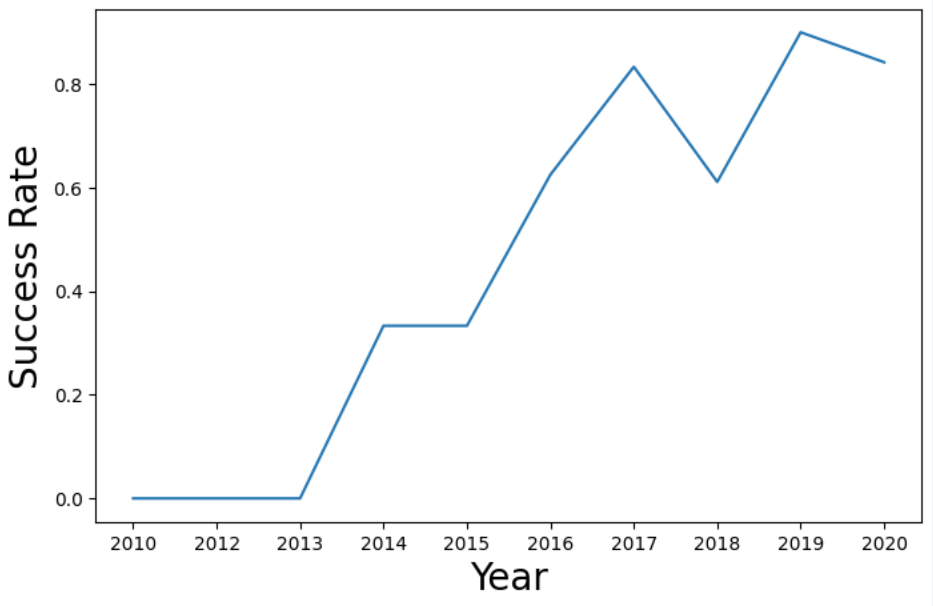
Payload vs. Orbit Type



- Flights to the ISS typically carry between 2,000 to 3,000 kg per flight. Although some flights carried as much as 12,000 kg.
- VLEO orbits tend to carry heavier payloads compared to LEO ones.
- Payloads for GTO orbits range from around 3,000 to 7,000 kg.

Launch Success Yearly Trend

- Between 2010 and 2013, there were no successful launches.
- Starting from 2013, the success rate increased on an annual basis.
- There was a dip in the success rate around 2018 but recovered in 2019 and dipped again slightly in 2020.



All Launch Site Names

- The names of all the unique launch sites are:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

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

- The SQL query used to retrieve this data is:

```
%sql SELECT DISTINCT (Launch_Site) FROM SPACEXTBL ORDER BY Launch_Site
```

- **`DISTINCT`** instructs SQL to only return unique values of **`Launch_Site`** from the table.
- **`ORDER BY`** instructs SQL to order the output by **`Launch_Site.`** The default order is in ascending order.

Launch Site Names Begin with 'CCA'

- The 5 records where the launch site begins with ‘CCA’ are:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The SQL used to retrieve this is:

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

- **`LIKE`** combined with ‘**`CCA%`**’ tells SQL to search the column **`Launch_Site`** for values that begins with ‘**`CCA`**’.
- **`LIMIT`** tells SQL to return the first 5 records it finds.

Total Payload Mass

- The total payload carried by boosters from NASA is:

Sum of Payload Mass (KG)
45596

- The SQL query used to retrieve this data is:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS 'Sum of Payload Mass (KG)' FROM SPACEXTBL WHERE Customer='NASA (CRS)'
```

- The **`SUM()`** function tells SQL to add up all the values in the **`PAYLOAD_MASS_KG_`** column.
- The **`WHERE`** clause tells SQL to only include rows where the **`Customer`** is '**`NASA (CRS)`**'

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- The SQL query used to calculate this is:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE '%F9 v1.1%'
```

- The **`AVG()`** function tells SQL to get the average of all the values in the **`PAYLOAD_MASS_KG_`** column.
- The **`WHERE`** clause tells SQL to limit the results where the **`Booster_Version`** contains the pattern **`'%F9 v1.1%`**'. The '**`%`**' on both sides tell the **`LIKE`** clause to match anywhere in the string.

First Successful Ground Landing Date

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

Date
2015-12-22

- The SQL query used to determine this is:

```
SELECT MIN(Date) AS 'Date' FROM SPACEXTBL WHERE Landing_Outcome='Success (ground pad)'
```

- The **`MIN()`** function is used to find the minimum value in the specified column.

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are:

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

- The SQL query used to retrieve this data is:

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

- The '****' tells the **%sql** parser that it is a multi-line command.
- The **WHERE** clause tells SQL to return records where the **Landing_Outcome** is equal to '**Success (drone ship)**' and the **PAYLOAD_MASS_KG_** is greater than 4,000 and less than 6,000 kg, exclusive.

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failed mission outcomes are:

Mission Outcome	Count
Failure	3
Success	98

- The SQL query used to retrieve this information is:

```
%sql SELECT 'Success' AS 'Mission Outcome', COUNT(*) AS Count FROM SPACEXTBL WHERE Mission_Outcome='Success' \
     UNION SELECT 'Failure' AS 'Mission Outcome', COUNT(*) AS Count FROM SPACEXTBL WHERE Mission_Outcome!='Success'
```

- The SQL query consists of 2 separate **`SELECT`** statements.
- The **`COUNT()`** function tells SQL to count the number of records matching the condition in the **`WHERE`** clause.
- **`UNION`** tells SQL to combine the results from the two **`SELECT`** statements into one table.

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass are listed on the right.
- The SQL query used to retrieve this information is:

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

- The **`MAX()`** function was used in the subquery to retrieve the maximum payload.
- A subquery was needed because of the way the SQL processor works.
- This value was then used to compare against the **`PAYLOAD_MASS_KG_`** for all records.

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

2015 Launch Records

- The list of failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 is:

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

- The SQL query used to retrieve this information is:

```
%sql SELECT SUBSTR(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site \
     FROM SPACEXTBL WHERE SUBSTR(Date,0,5)='2015' AND Landing_Outcome='Failure (drone ship)'
```

- The function **`SUBSTR(Date, 6, 2)`** was used to extract the month from the Date column to be returned with the results.
- The function **`SUBSTR(Date, 0, 5)`** was used to extract the year from the Date column for comparison in the **`WHERE`** clause.

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

- The count of landing outcomes between the date 2010-06-04 and 2017-03-20 is displayed to the right.
- The SQL query used to retrieve this data is:

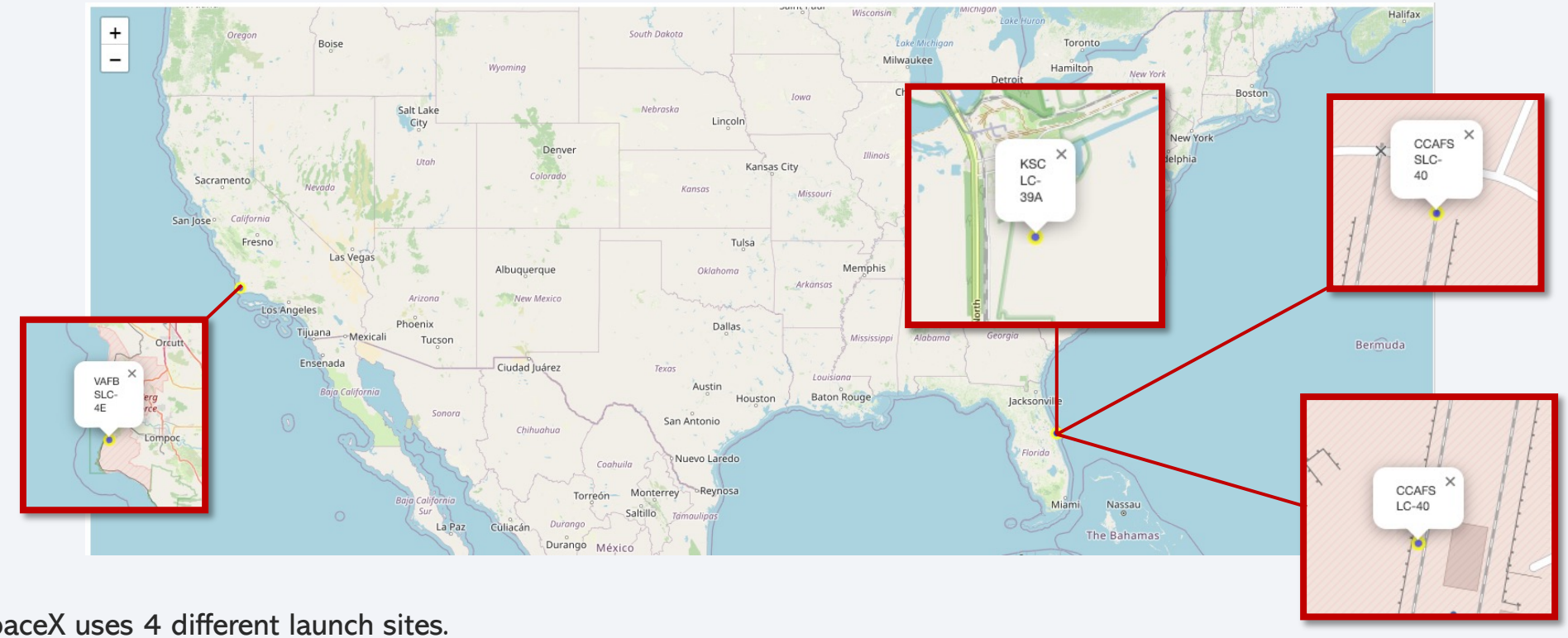
```
%sql SELECT Landing_Outcome, Count(*) AS Count FROM SPACEXTBL \
     WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' \
     GROUP BY Landing_Outcome ORDER BY Count DESC
```

- The **BETWEEN** in the **WHERE** clause tells SQL to return records between the 2 specified dates, inclusive.
- **GROUP BY** tells SQL to group all records by **Landing_Outcome**.
- The **Count()** function tells SQL to count the records grouped by **Landing_Outcome**.
- **ORDER BY** tells SQL to order the results by **Count** in descending order.

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

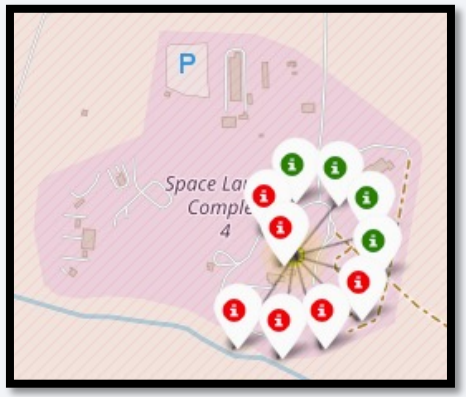

Section 3
Launch Sites
Proximities Analysis

Map of All Launch Sites

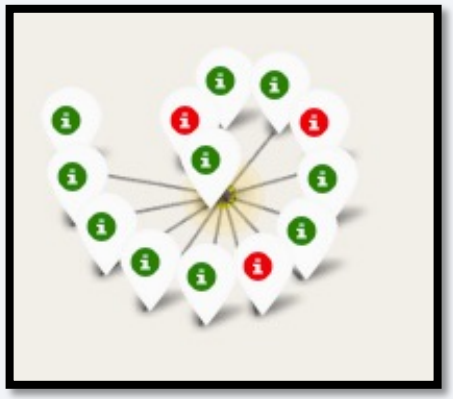


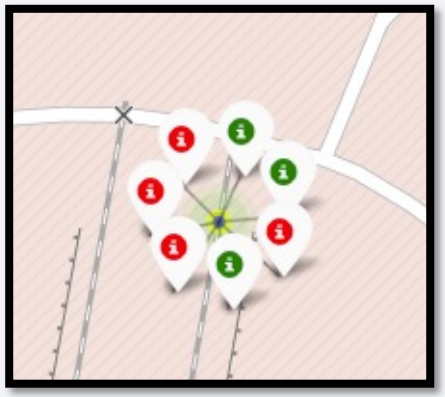
- SpaceX uses 4 different launch sites.
- 3 of the launch sites are located in Florida at the Kennedy Space Center (KSC) and Cape Canaveral (CCAFS).
- Kennedy Space Center and Cape Canaveral are right next to each other on the east coast of Florida.
- The one launch site in California is located at the Vandenberg Air Force Base (VAFB) Space Launch Complex (SLC).

Launch Outcomes



VAFB SLC-4E


KSC LC-39A

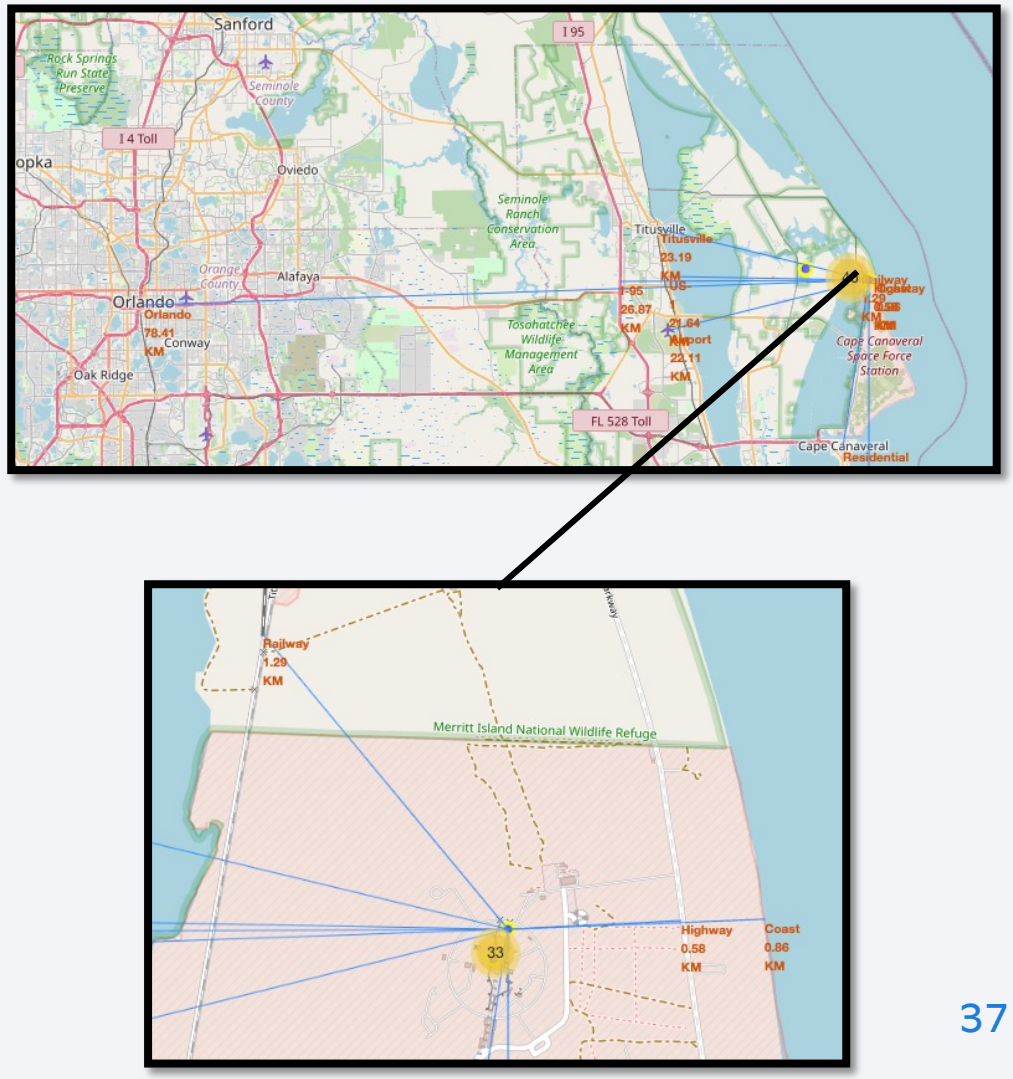

CCAFS SLC-40

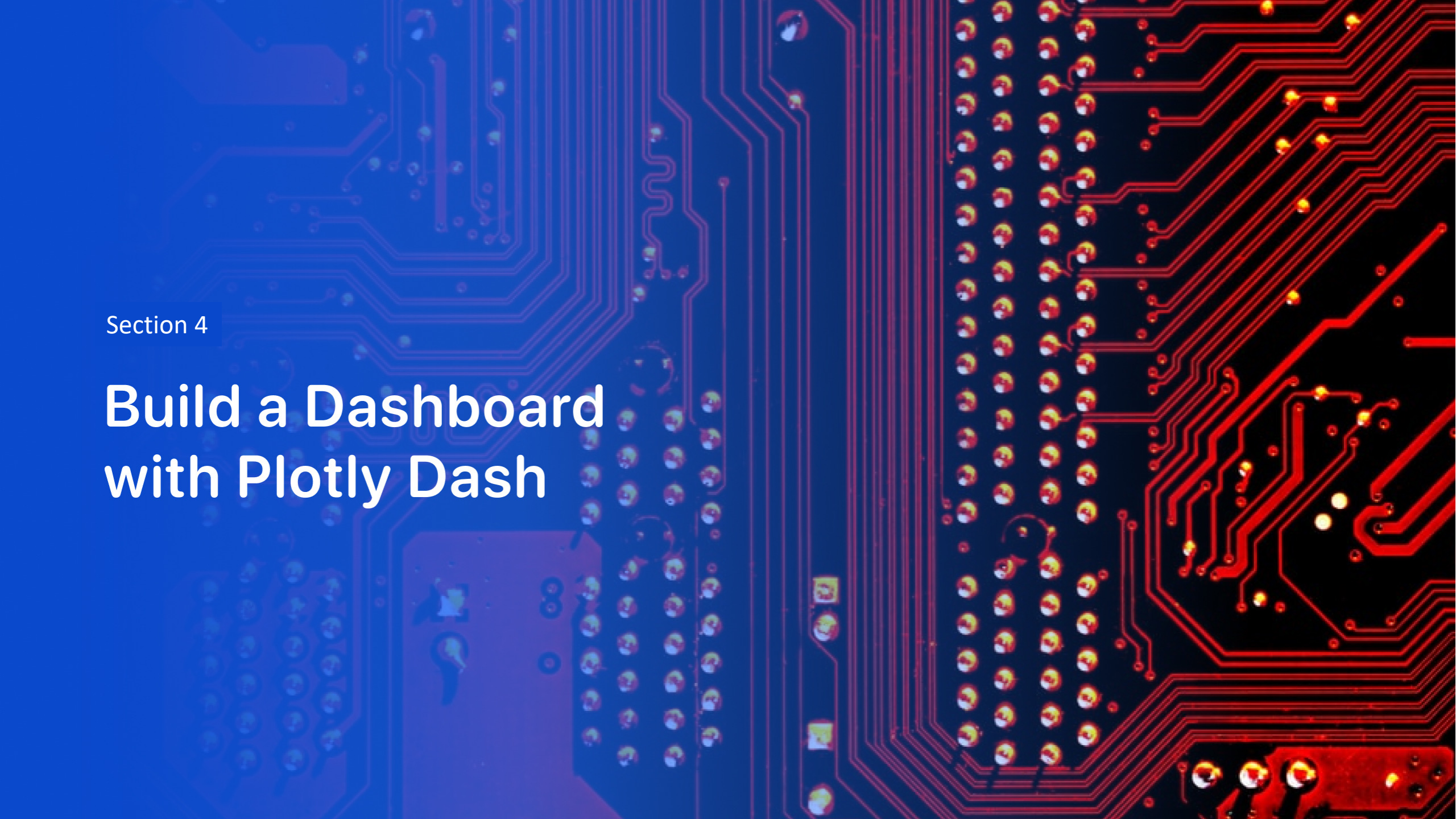
CCAFS LC-40

- The images above show the launch outcomes for all the flights from each launch location.
- The number of markers around each site indicates the number of launches made from that site.
- A red marker indicates a failed outcome while the green marker indicates a successful outcome.
- From the images above we can easily conclude the following:
  - CCAFS SLC-40 launched the least number of flights.
  - CCAFS LS-40 launched the most number of flights.
  - KSC LC-39A launched the most number of successful flights
  - CCAFS LC-40 has the most flights that failed.

Proximity Map for CCAFS SLC-40

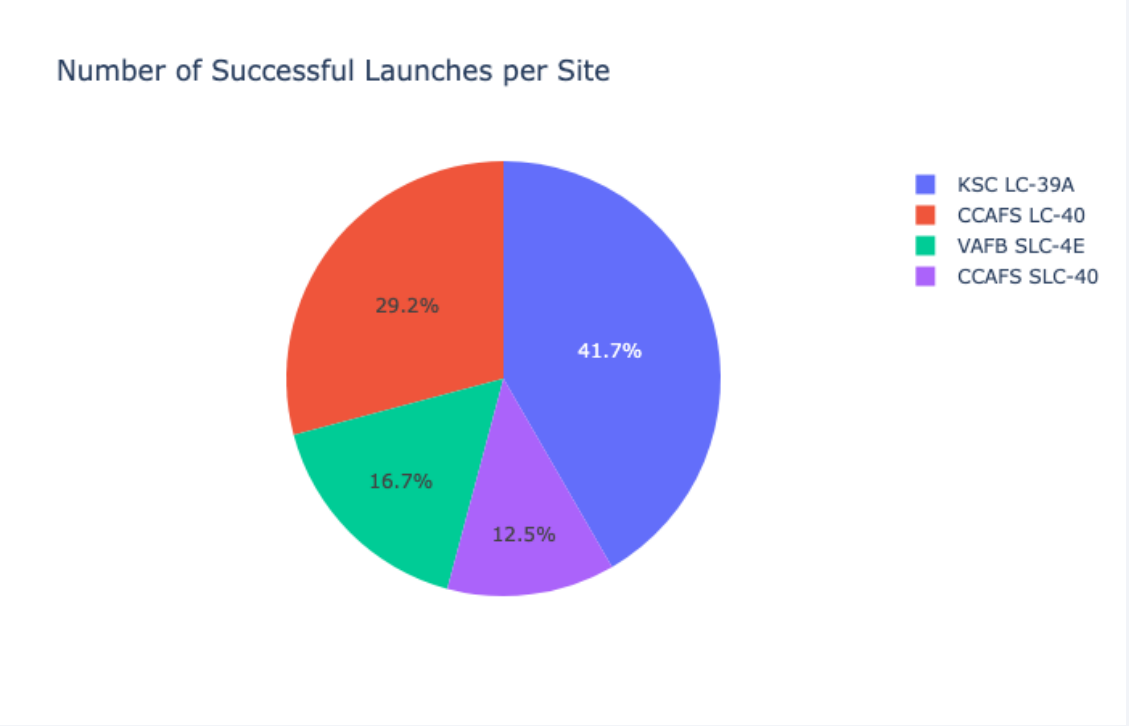
- The image to the right is a proximity map showing distances to several points of interest like Orlando (nearest major city), nearest airport, major highways, coast, and other locations.
- The lower image displays a zoomed in portion showing the 3 closest points of interest to the launch site.
- The coastline and highway are less than 1 km away.
- The nearest railway is about 1.29 km away. However, that railway is only used by Cape Canaveral to transport rockets from the hangars to the launch pads. So, it does not carry civilian passengers.
- The nearest airport, Space Coast Regional Airport, is around 22.11 km away.
- The nearest major city, Orlando, is around 78.41 km away.




Section 4
Build a Dashboard with Plotly Dash

Number of Successful Launches per Site

- The pie chart shows all the launch sites and their share of all the successful launches.
- Each wedge represents a site and the percentage of successful launches from that site.
- The legend indicates the colors associated with each launch site.
- For example, site KSC LC-39A is blue and has the the largest share of successful launches with 41.7%.

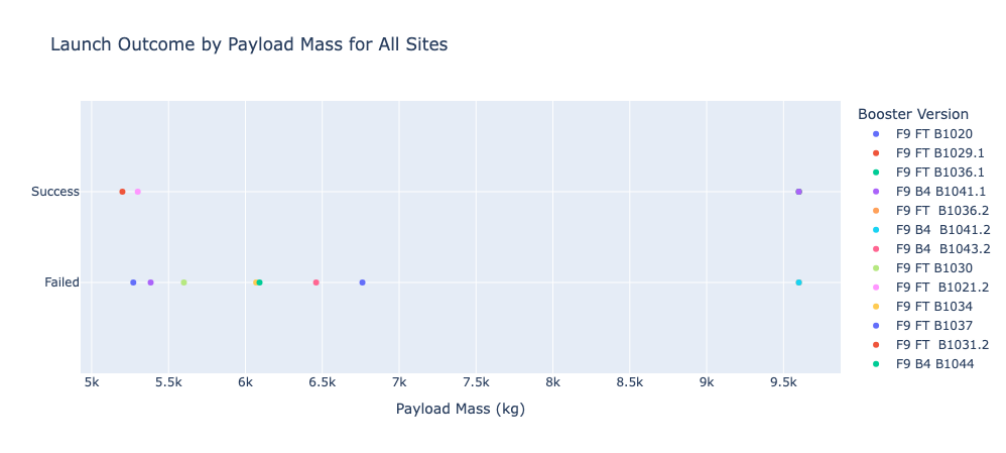


Pie Chart of the Site with the Highest Success

- From the pie chart showing all launch sites, it was determined that KSC LC-39A had the most successful launches.
- After selecting that site in the dropdown, the pie chart shows that 76.9% of launches had successful outcomes.

Scatter Plots of Payload vs. Launch Outcome

- The scatter plot on the left shows the launch outcomes for payloads between 0 and 5,000 kg for all sites.
- The scatter plot on the right shows the launch outcomes for payloads between 5,000 kg and 10,000 kg for all sites.
- As the payload mass increases, there are more failed launch outcomes.

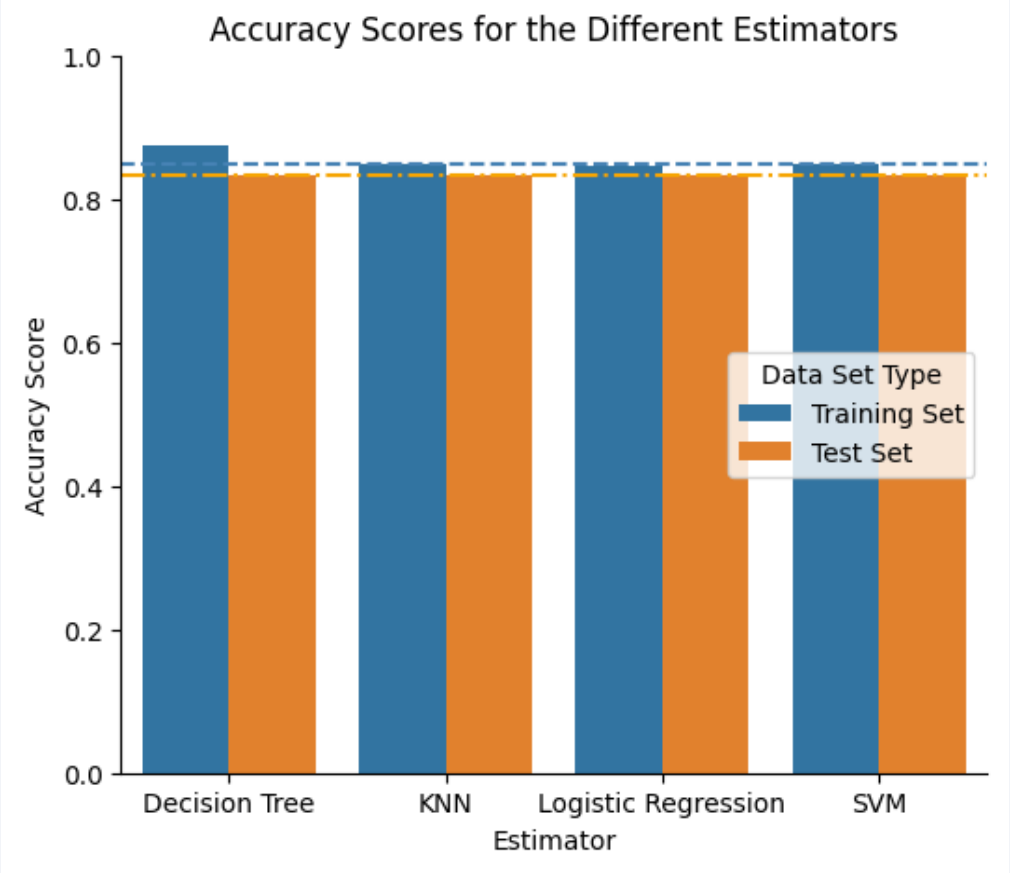


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Different models were built using the following algorithms:
  - Logistic Regression (LR)
  - Decision Tree (DT)
  - Support Vector Machine (SVM)
  - K-Nearest Neighbor (KNN)
- KNN, LR, and SVM produced nearly identical accuracy scores using both training and test data.
- DT scored the best using training data and scored the same as the others using test data.
- DT will be used to predict landing outcomes for future flights.

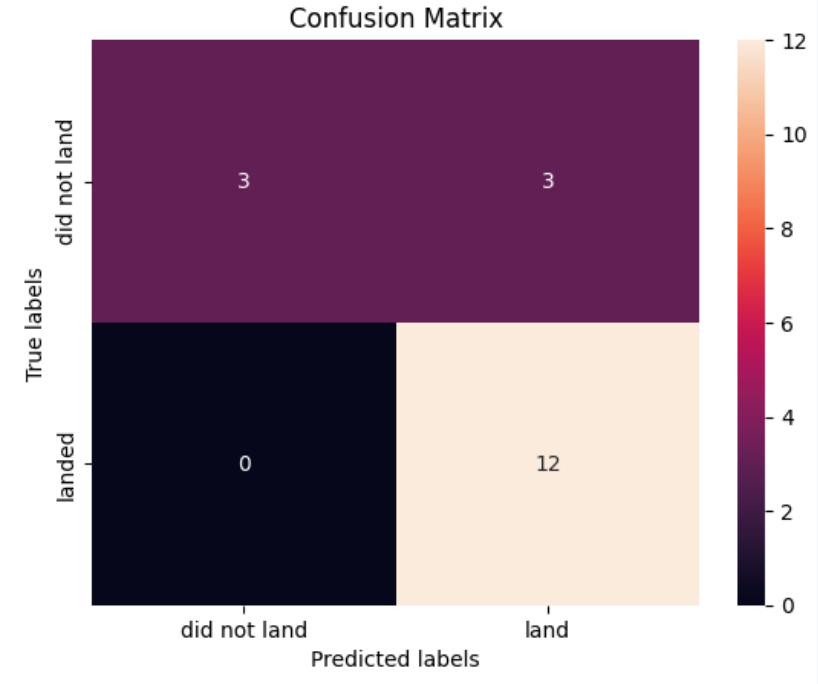


Confusion Matrix

- A confusion matrix is a table that compares the predicted values against the true values. It displays the number of predictions that matched the true values as well as the number that did not.
- The table is split into 4 quadrants.

Quadrant	Description
Upper left	True Negative = Model correctly predicted a failed landing
Lower left	False Negative = Model incorrectly predicted a failed landing
Upper right	False Positive = Model incorrectly predicted a successful landing
Lower right	True Positive = Model correctly predicted a success landing

- A Negative outcome means the rocket failed to land.
- A Positive outcome means the rocket landed successfully.
- For the two upper quadrants, there were 6 records where the flight failed to land. The Decision Tree correctly predicted that 3 would fail to land but determined that 3 would succeed, when in fact they also failed.
- For the two lower quadrants, there were 12 records where the flight landed successfully. The Decision Tree algorithm correctly predict all 12.

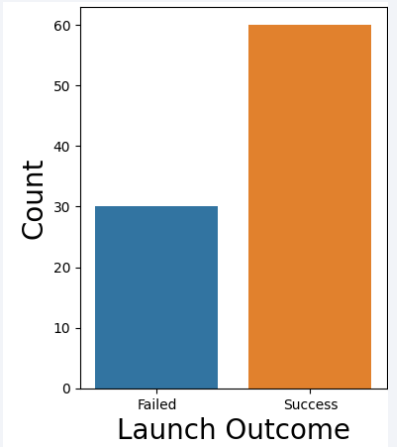


Confusion Matrix for the Decision Tree Algorithm

Conclusions

- The success rate for landing is 66.7% or 2 out of every 3 flights will land successfully.
- The four orbit types with the highest success rates were ES-L1, GEO, HEO, and SSO.
- Decision Tree Algorithm is the best model to use for predicting the landing outcome of future flights with an accuracy rate of 87.5%.
- Launch site KSC LC-39A had the best mission success rate at 76.9%.

Success rate is 66.67%



	Orbit	Class
0	ES-L1	1.0
1	GEO	1.0
3	HEO	1.0
9	SSO	1.0

Algorithm	Accuracy
Decision Tree	0.875000
KNN	0.848214
SVM	0.848214
Logistic Regression	0.846429

Appendix

- Google Colab did not require installing some libraries. There were installed already.

```
# NOTE!!!
#
# The following lines are not needed in Google Colab

# !pip3 install beautifulsoup4
# !pip3 install requests
```

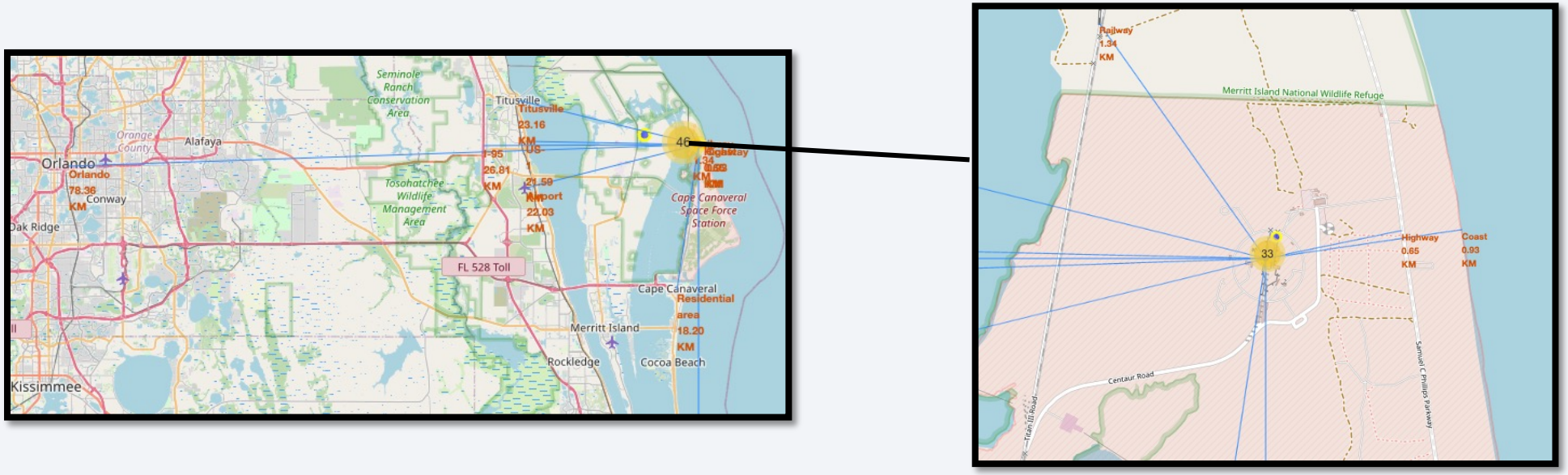
```
Requirement already satisfied: beautifulsoup4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.11.1)
Requirement already satisfied: soupsieve>1.2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from beautifulsoup4) (2.3.2.post1)
Requirement already satisfied: requests in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (2.29.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests) (2023.5.7)
```

```
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

Appendix

- Here is a proximity map for CCAFS LC-40.

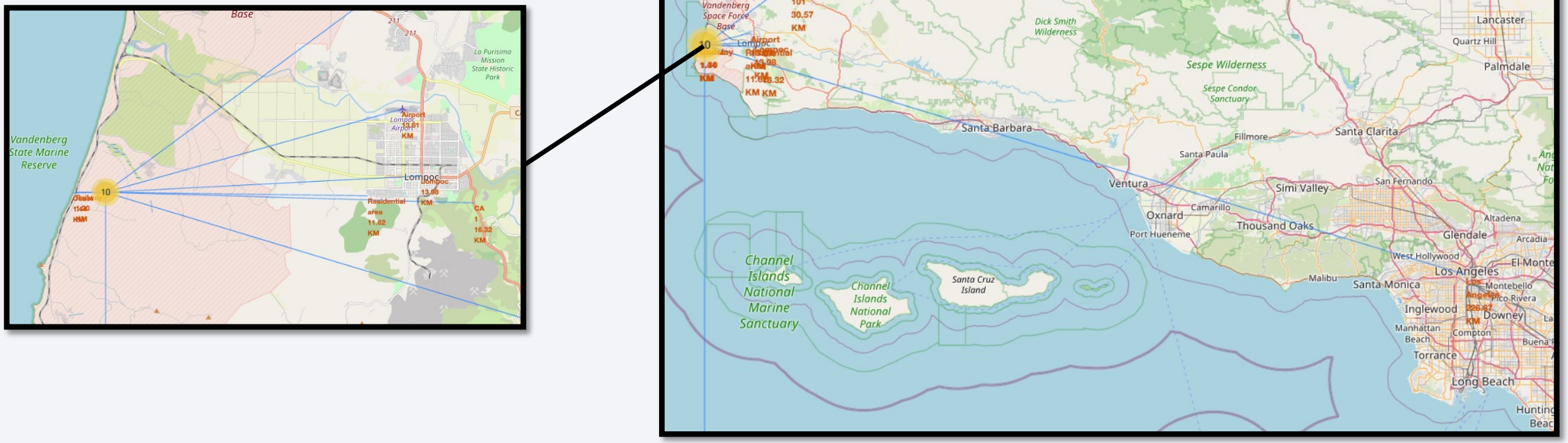


Appendix

- Here is another for KSC LC-39A.

Appendix

- Here is another for VAFB SLC-4E.



Thank you!