IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kamalakar
12/12/2022

Outline

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

Executive Summary

- In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms.
- The main steps in this project include:
  - Data collection, wrangling, and formatting
  - Exploratory data analysis
  - Interactive data visualization
  - Machine learning prediction
- Our graphs show that some features of the rocket launches have a correlation with the outcome of the launches, i.e., success or failure.
- It is also concluded that decision tree may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

Introduction

In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms.

The main steps in this project include:

Data collection, wrangling, and formatting
Exploratory data analysis
Interactive data visualization
Machine learning prediction
Our graphs show that some features of the rocket launches have a correlation with the outcome of the launches, i.e., success or failure.
It is also concluded that decision tree may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX APIWeb scraping
- Perform data wrangling
  - Pandas and NumPy
  - SQL
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine learning prediction, using
  - Logistic regression
  - Support vector machine (SVM)
  - Decision tree
  - K-nearest neighbors (KNN)

Data Collection

Web scraping

The data is scraped from https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

The website contains only the data about Falcon 9 launches.

We end up with 121 rows or instances and 11 columns or features. The picture below shows the first few rows of the data:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Data Collection - SpaceX API

Step-1

- Make a GET response to the SpaceX REST API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

- Step-2
- Define lists for data to be stored in
- Clean data
- Use these lists as values in a dictionary and construct the dataset

```
# We also want to convert the date_utc to a datetime datatype and then ext
racting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Step-3

- Create Pandas Dataframe

```
# Create a data from launch_dict

data2 = pd.DataFrame(launch_dict)
```

Step 4

Use column names as keys in dictionaries

Convert to Pandas

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df=pd.DataFrame(launch_dict)
```

Data Collection - Scraping

Step-1
Request HTML page

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Step-2
Create BeautifulSoup object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

Fill all the table in the HTML Page

```
html_tables=soup.find_all('table')
```

Step-3
Extract Column Names from tables in HTML Page

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

for i in first_launch_table.find_all('th'):
    if extract_column_from_header(i)!=None:
        if len(extract_column_from_header(i))>0:
            column_names.append(extract_column_from_header(i))
```

Step-4
Use column names as keys in dictionaries
Convert to Pandas

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

Displaying the Landing Outcome Columns

```
# landing_outcomes = values on Outcome column

landing_outcomes = df["Outcome"].value_counts()
landing_outcomes
```

```
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: Outcome, dtype: int64
```

Creating a list [landing class]to check if the booster would land successfully or not

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

def onehot(item):
    if item in bad_outcomes:
        return 0
    else:
        return 1
landing_class = df["Outcome"].apply(onehot)
landing_class
```

Exporting the Data frame to a .csvfile

```
df.to_csv("dataset_part\_2.csv", index=False)
```

EDA with Data Visualization

SCATTER CHARTS

Flight Number vs Launch Site
Payload vs Launch Site
Orbit Type vs Flight Number
Payload vs Orbit Type

BAR CHART

Success Rate vs Orbit Type

LINE CHARTS

Success Rate vs Year

EDA with SQL

- Loading the Dataset using the IBM DB2 Database
- Query the Data using Python
- Performed different queries (10) to understand the dataset better
- Queries included [Displaying: names of unique launch sites, average payload mass carried by booster version etc……

Build an Interactive Map with Folium

- The following steps were taken to visualize the launch data on an interactive map:

1. Mark all launch sites on a map
   - Initialise the map using a Folium `Map` object
   - Add a `folium.Circle` and `folium.Marker` for each launch site on the launch map

2. Mark the success/failed launches for each site on a map
   - As many launches have the same coordinates, it makes sense to cluster them together.
   - Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red.
   - To put the launches into clusters, for each launch, add a `folium.Marker` to the `MarkerCluster()` object.
   - Create an icon as a text label, assigning the `icon_color` as the `marker_colour` determined previously.

3. Calculate the distances between a launch site to its proximities
   - To explore the proximities of launch sites, calculations of distances between points can be made using the `Lat` and `Long` values.
   - After marking a point using the `Lat` and `Long` values, create a `folium.Marker` object to show the distance.
   - To display the distance line between two points, draw a `folium.PolyLine` and add this to the map.

Build a Dashboard with Plotly Dash

PLOTLY DASH

- Creating an interactive dashboard with Pie charts and Scatter Plots/Graphs
- Pie chart
- Used to show distribution of successful launches across all launch sites
- Shows success/failure ratio for each individual site
- Scatter plot
- Shows us how success varies across different launch sites, payload mass and booster version

Predictive Analysis (Classification)

Model Evaluation

- To prepare the dataset for model development:
  - Load dataset
  - Perform necessary data transformations (standardise and pre-process)
  - Split data into training and test data sets, using `train_test_split()`
  - Decide which type of machine learning algorithms are most appropriate
- For each chosen algorithm:
  - Create a `GridSearchCV` object and a dictionary of parameters
  - Fit the object to the parameters
  - Use the training data set to train the model

Model Evaluation

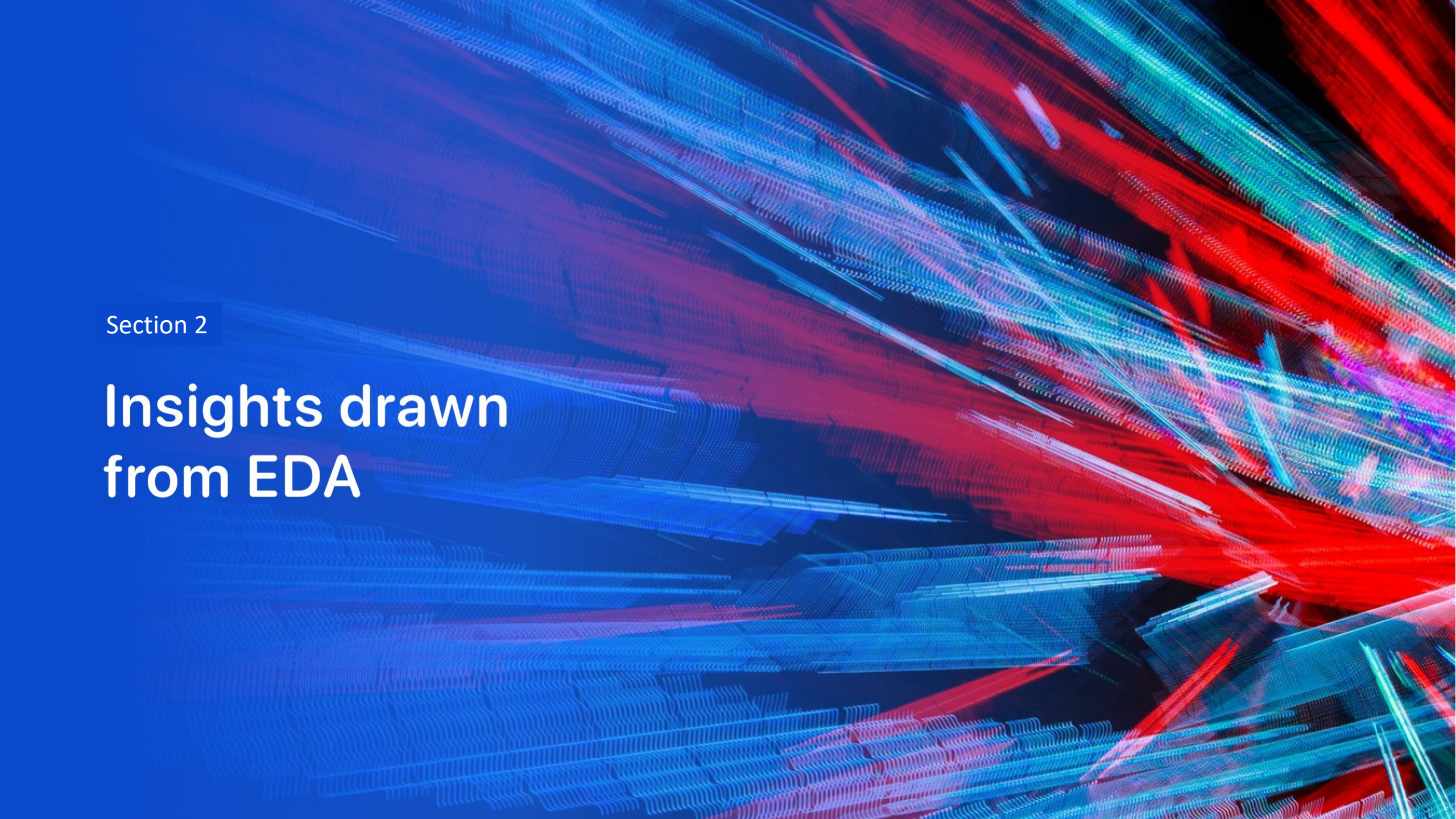
For each chosen algorithm:
Using the output GridSearchCV object:
Check the tuned hyperparameters (best_params_)
Check the accuracy (score and best_score_)
Plot and examine the Confusion Matrix

Best Fit Classification

- Review Accuracy Score
- Check which accuracy score is the highest to determine the best performing model

Results

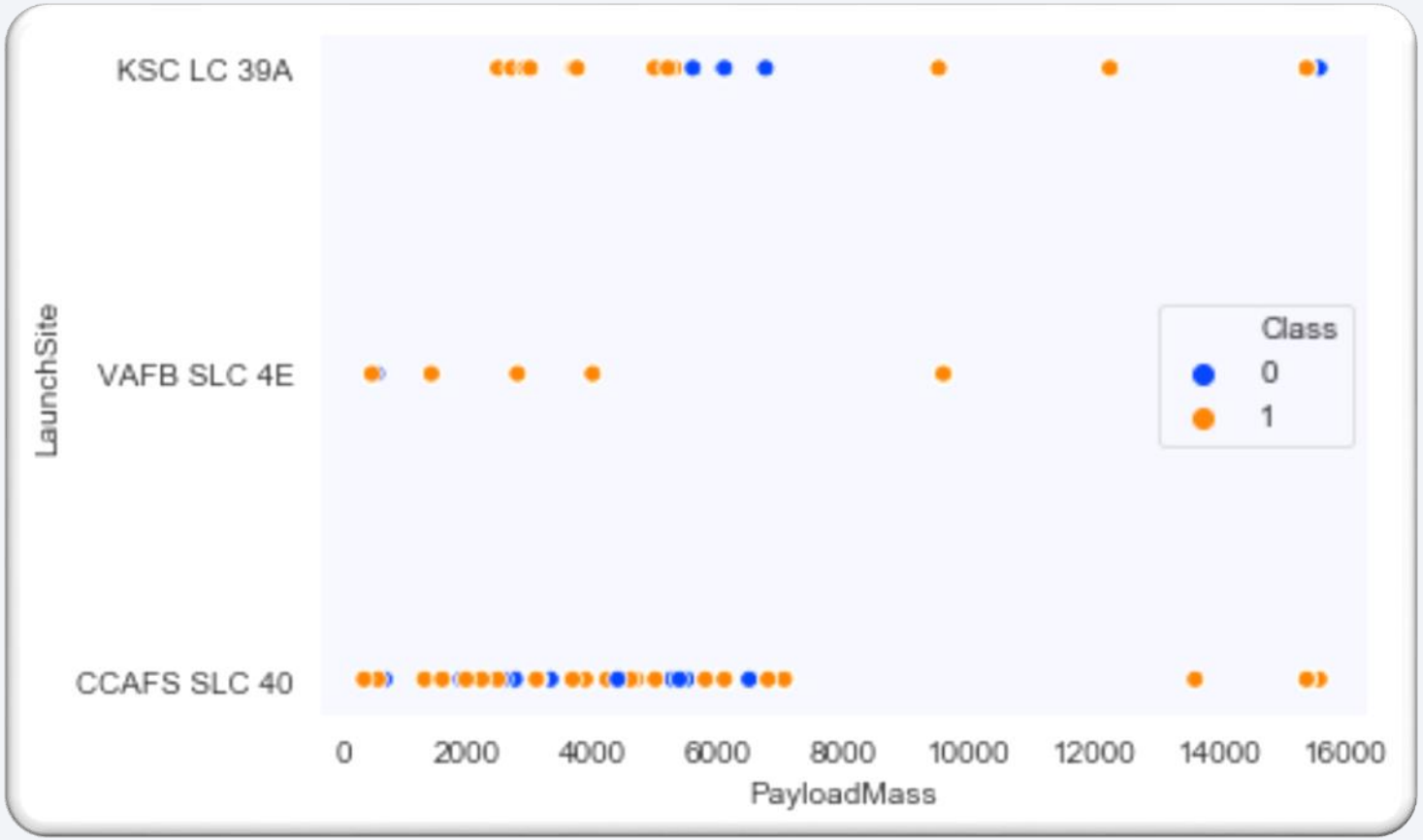
- Exploratory data analysis results
- Interactive analytics
- Predictive analysis results


Section 2
Insights drawn from EDA

Payload vs. Launch Site

The scatter plot of Launch Site vs. Payload Mass shows that:

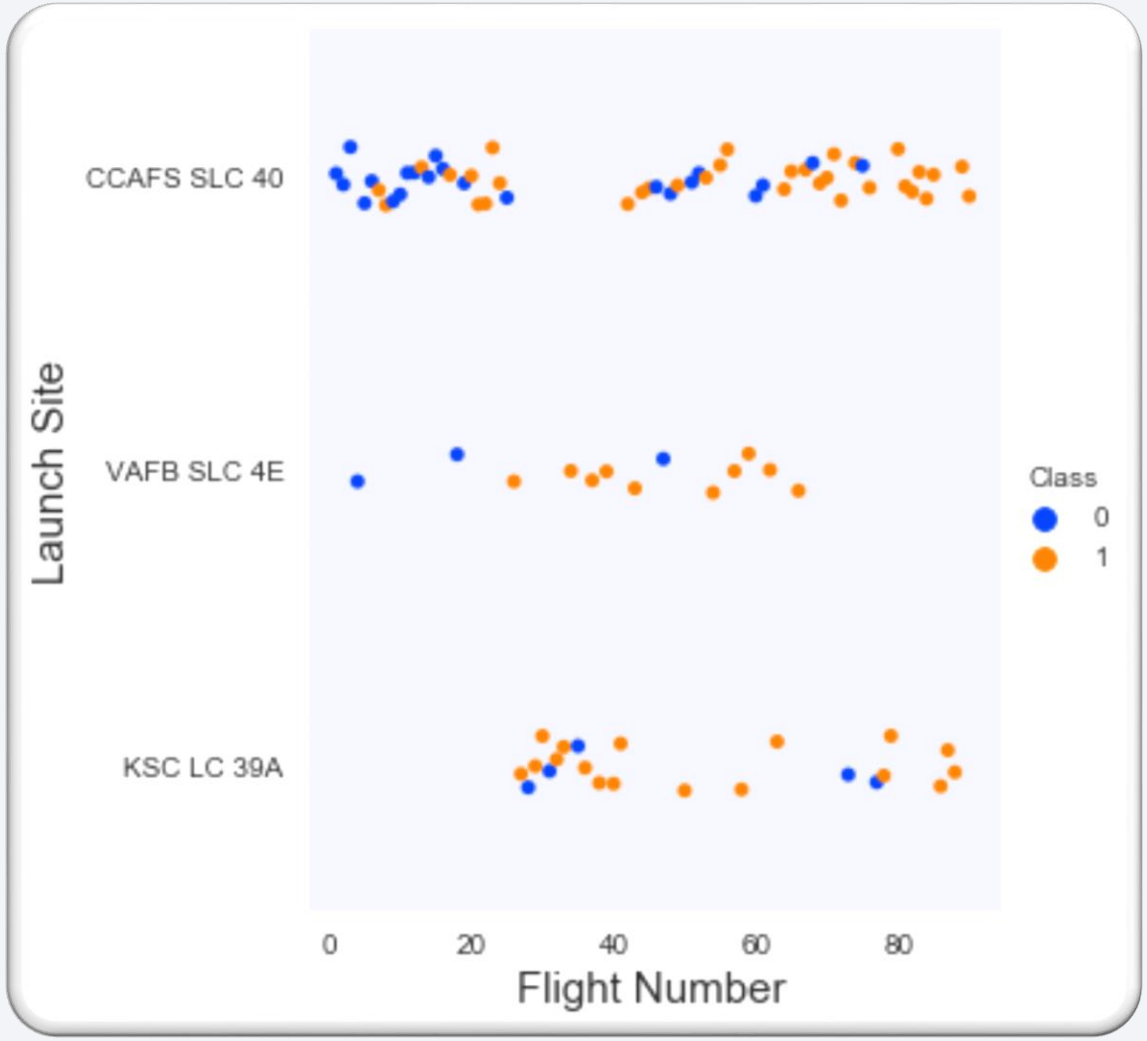
- Above a payload mass of around 7000 kg, there are very few unsuccessful landings, but there is also far less data for these heavier launches.
- There is no clear correlation between payload mass and success rate for a given launch site.
- All sites launched a variety of payload masses, with most of the launches from CCAFS SLC 40 being comparatively lighter payloads (with some outliers).



Flight Number vs. Launch Site

The scatter plot of Launch Site vs. Flight Number shows that:

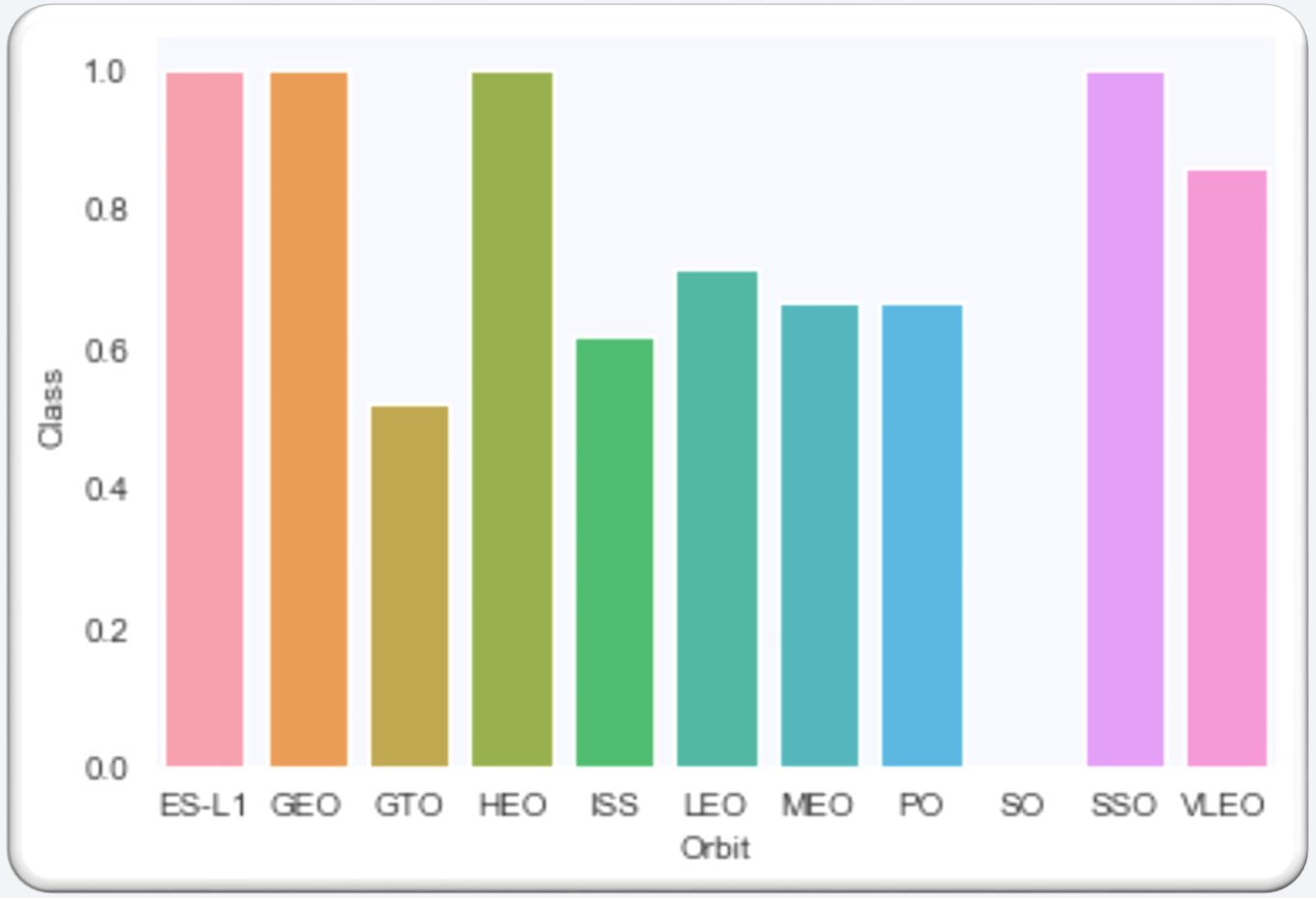
- As the number of flights increases, the rate of success at a launch site increases.
- Most of the early flights (flight numbers < 30) were launched from CCAFS SLC 40, and were generally unsuccessful.
- The flights from VAFB SLC 4E also show this trend, that earlier flights were less successful.
- No early flights were launched from KSC LC 39A, so the launches from this site are more successful.
- Above a flight number of around 30, there are significantly more successful landings (Class = 1).



Success Rate vs. Orbit Type

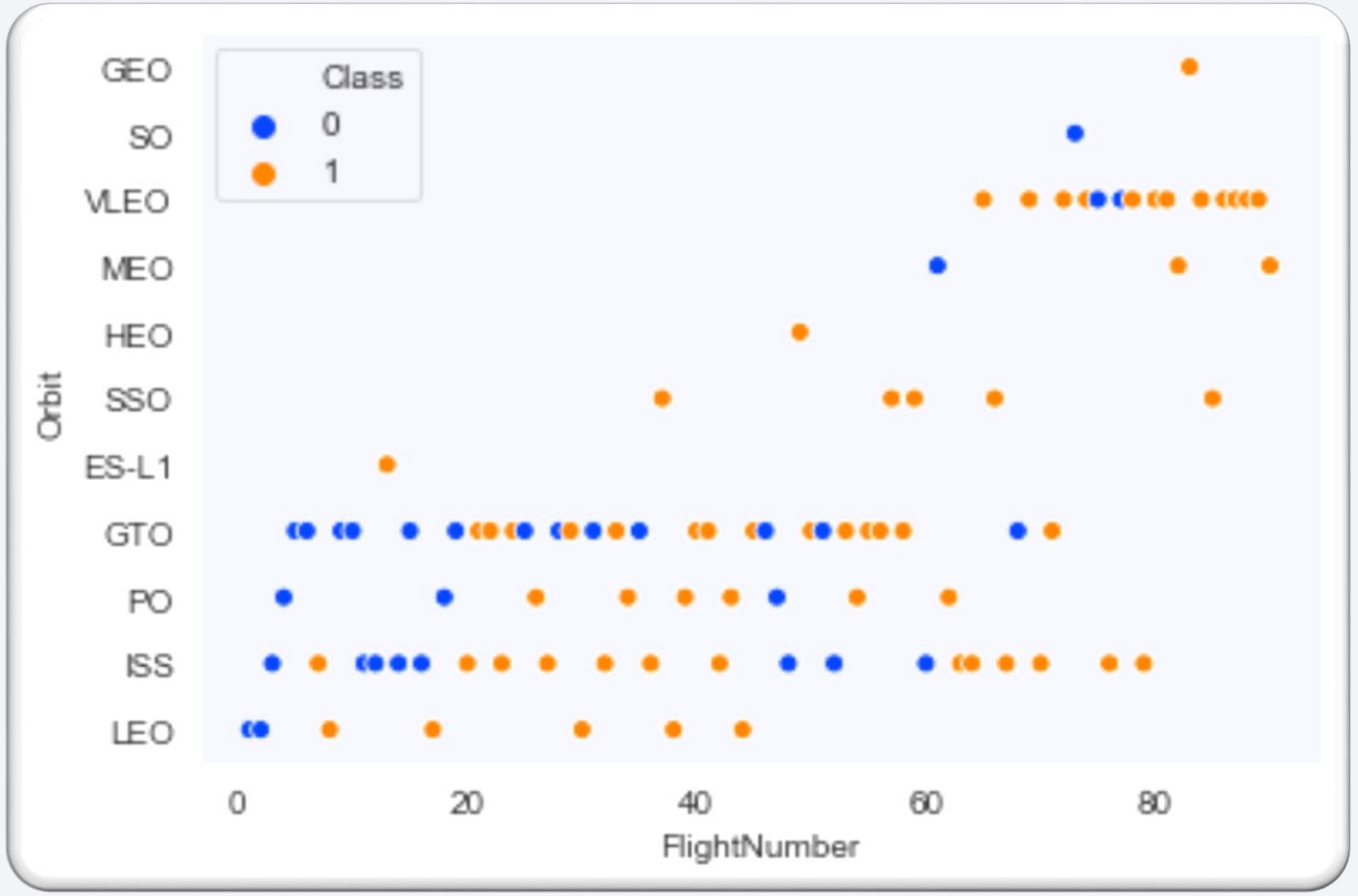
- Orbits with 100% success rate
- ES-L1 (Earth-Sun First Lagrangian Point),
- GEO (Geostationary Orbit),
- HEO (High Earth Orbit),
- SSO (Sun-synchronous Orbit)

- Orbits with 0% success rate
- SO (Heliocentric Orbit



Flight Number vs. Orbit Type

- The scatter plot of Orbit Type vs Flight Number shows that:
- The 100% success rate of GEO, HEO, and ES-L1 orbits can be explained by only having 1 flight into the respective orbits.
- Success rate in SSO is more impressive, with 5 successful flights.
- Relationship between Flight Number and Success Rate for GTO is little.



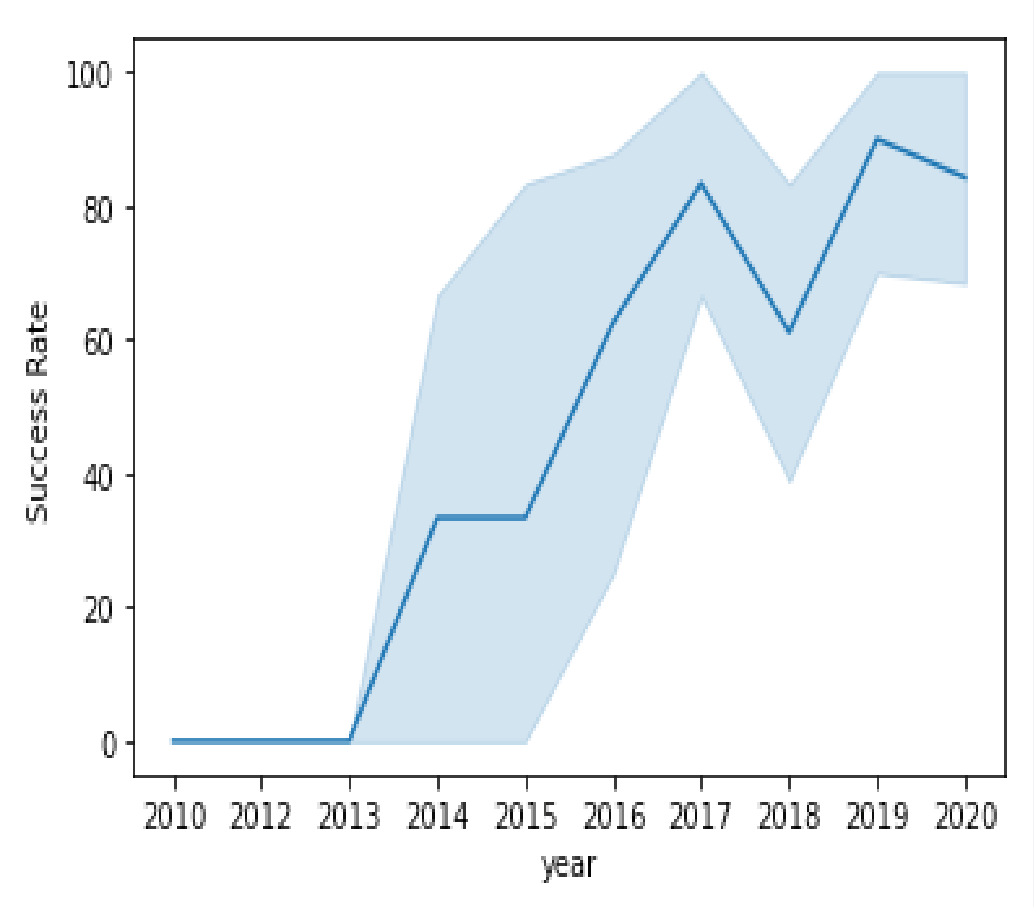
Payload vs. Orbit Type

- The scatter plot of Orbit Type vs Payload Mass shows that:

- Orbits types (PO, ISS and LEO) have more success with heavy payloads
- Relationship between payload mass and success rate in GTO is unclear.
- VLEO (Very Low Earth Orbit) launches are associated with heavier payloads.

Launch Success Yearly Trend

- The line chart of yearly average success rate shows that:
- Between 2010 and 2013, all landings were unsuccessful (as the success rate is 0).
- After 2013, the success rate generally increased, despite small dips in 2018 and 2020.
- After 2016, there was always a greater than 50% chance of success.



All Launch Site Names

- Find the names of the unique launch sites

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- Present your query result with a short explanation here
- The word Distinct returns only unique values from the LAUNCH_SITE column of the SPACEXTBL table

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

- Present your query result with a short explanation here

- **LIMIT 5 fetches only 5 records, and the LIKE keyword is used with the wild card ‘CCA%’ to retrieve string values beginning with ‘CCA’.

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

- %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';

Total Payload Mass by NASA (CRS)

22007

- Present your query result with a short explanation here

- **The SUM keyword is used to calculate the total of the LAUNCH column, and the SUM keyword (and the associated condition) filters the results to only boosters from NASA (CRS).

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

- `%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \WHERE BOOSTER_VERSION = 'F9 v1.1';`

Average Payload Mass by Booster Version F9 v1.1

3676

- Present your query result with a short explanation here

- `**The AVG keyword is used to calculate the average of the PAYLOAD_MASS__KG_ column, and the WHERE keyword (and the associated condition) filters the results to only the F9 v1.1 booster version.`

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

- `%sql SELECT MIN(DATE) AS FIRST_SUCCESSFUL_GROUND_LANDING FROM SPACEX \WHERE LANDING__OUTCOME = 'Success (ground pad)';`
-

first_successful_ground_landing

2017-01-05

- Present your query result with a short explanation here
- `**The MIN keyword is used to calculate the minimum of the DATE column, i.e. the first date, and the WHERE keyword (and the associated condition) filters the results to only the successful ground pad landings.`

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone
ship)' \ AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

booster_version
F9 FT B1022
F9 FT B1031.2

- Present your query result with a short explanation here

```
**The WHERE keyword is used to filter the results to include only those
that satisfy both conditions in the brackets (as the AND keyword is also
used). The BETWEEN keyword allows for 4000 < x < 6000 values to be
selected.
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM SPACEX GROUP
BY MISSION_OUTCOME;
```

mission_outcome	total_number
Success	44
Success (payload status unclear)	1

- Present your query result with a short explanation here
- **The COUNT keyword is used to calculate the total number of mission outcomes, and the GROUPBY keyword is also used to group these results by the type of mission outcome.

Boosters Carried Maximum Payload

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

```
%sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEX \ WHERE PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1058.3
F9 B5 B1060.2

- Present your query result with a short explanation here

- **The SELECT statement within the brackets finds the maximum payload, and this value is used in the WHERE condition. The DISTINCT keyword is then used to retrieve only distinct /unique booster versions

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX \ WHERE (LANDING__OUTCOME =
'Failure (drone ship)') AND (EXTRACT(YEAR FROM DATE) = '2015');
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40

- Present your query result with a short explanation here
- **The WHERE keyword is used to filter the results for only failed landing outcomes, AND only for the year of 2015.

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

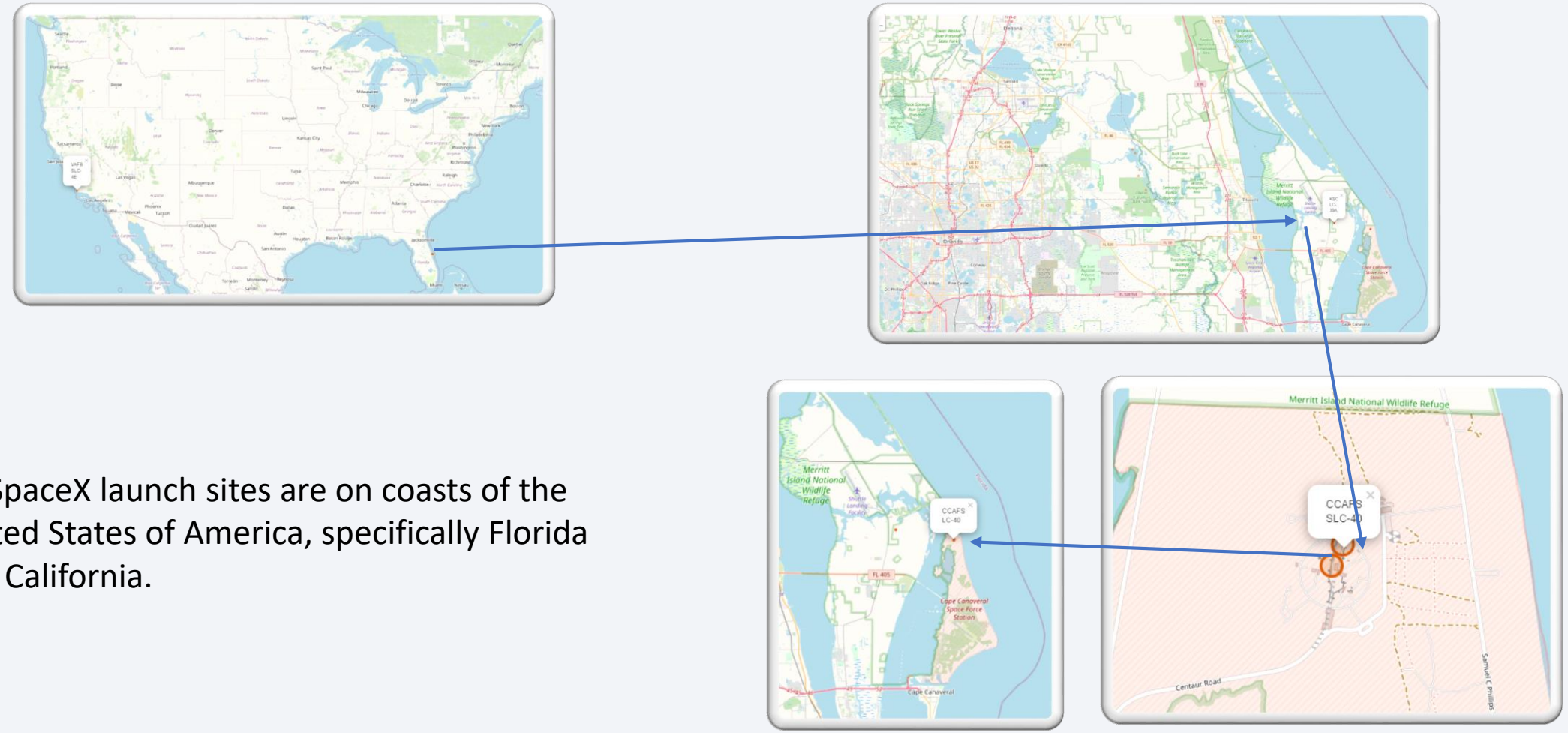
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- `%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \ WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \ GROUP BY LANDING__OUTCOME \ ORDER BY COUNT(LANDING__OUTCOME) DESC ;`
- Present your query result with a short explanation here
- **The WHERE keyword is used with the BETWEEN keyword to filter the results to dates only within those specified. The results are then grouped and ordered, using the keywords GROUP BY and ORDER BY, respectively, where DESC is used to specify the descending order.

Landing Outcome	Total Count
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

Section 3

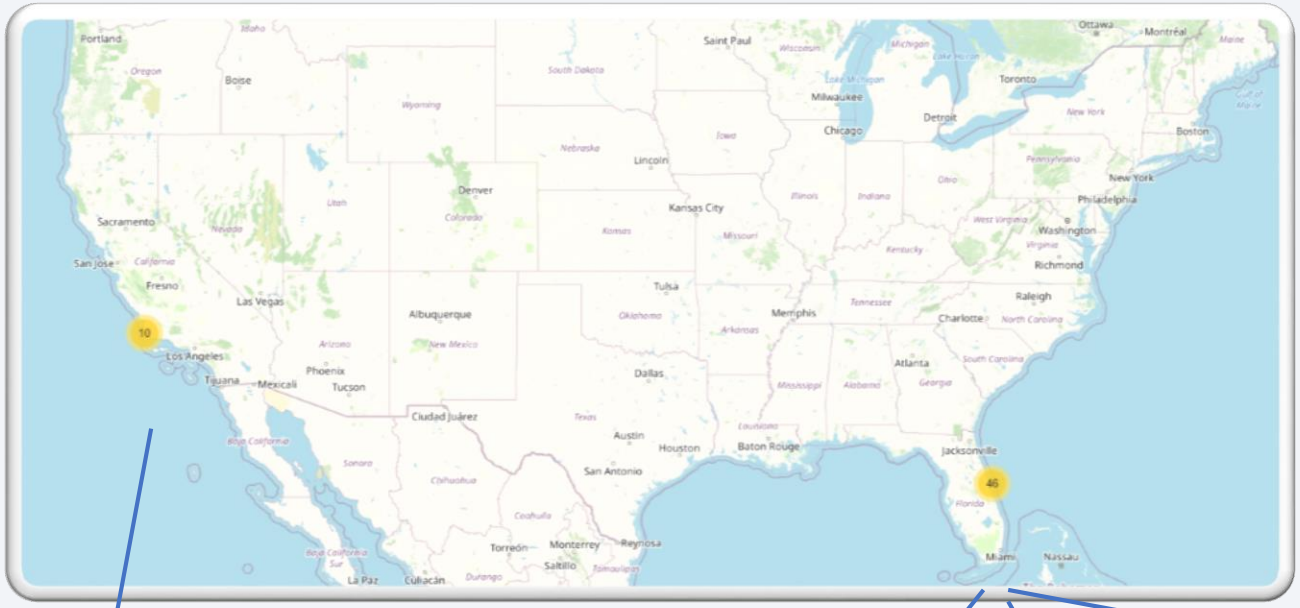
Launch Sites Proximities Analysis

ALL LAUNCH SITES ON A MAP



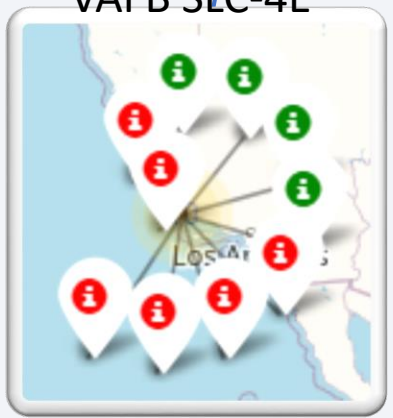
All SpaceX launch sites are on coasts of the United States of America, specifically Florida and California.

Success and Failed Launches For Each Site

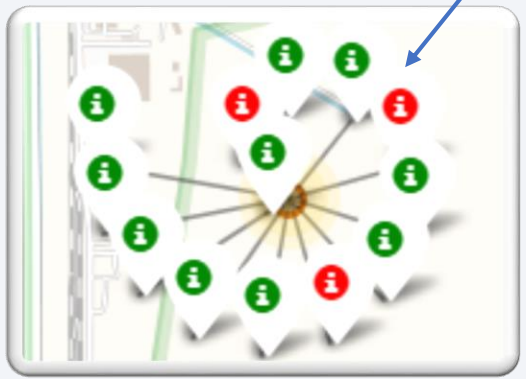


Launches have been grouped into clusters, green icons for successful launches, and red icons for failed launches.

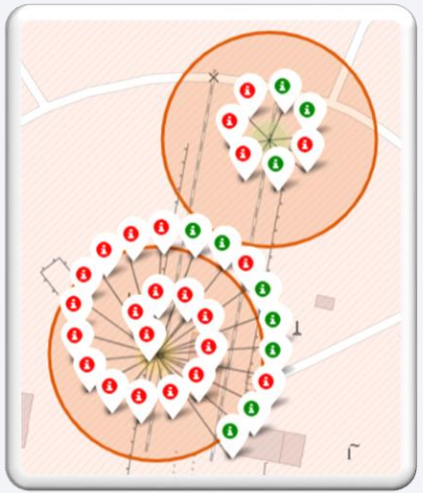
VAFB SLC-4E



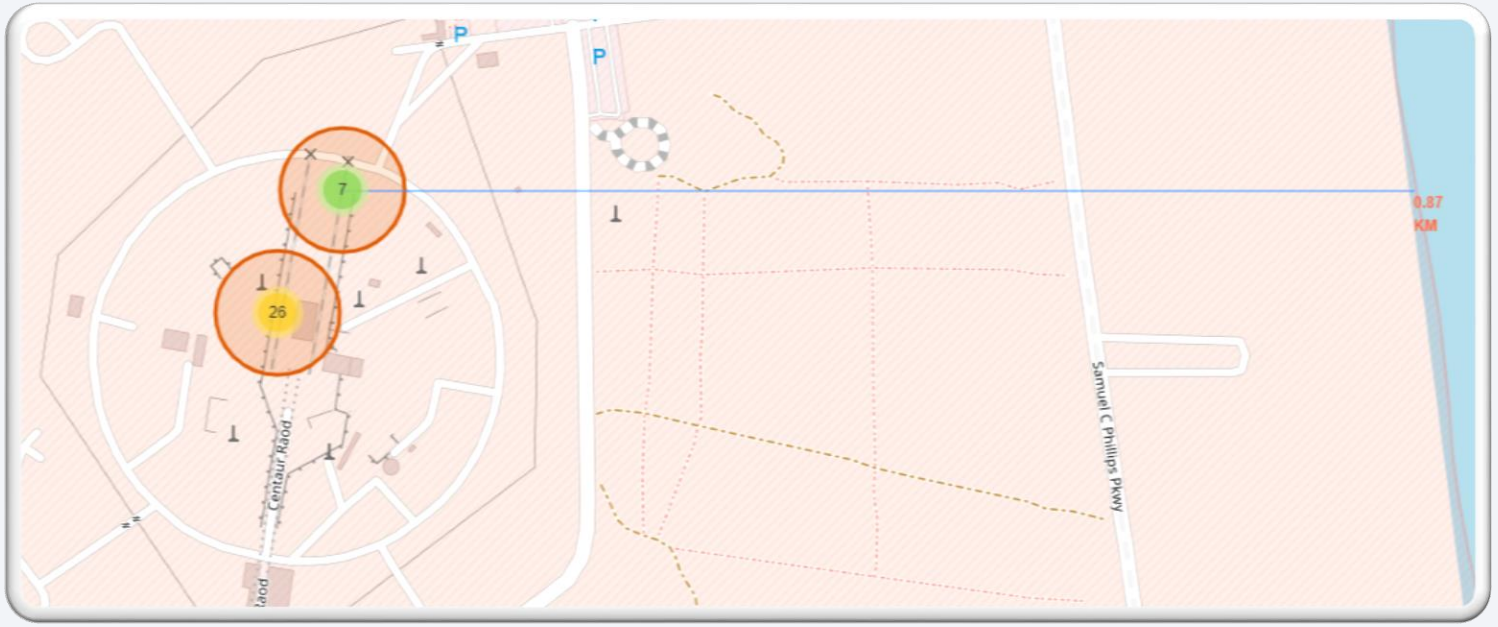
KSC LC-39A

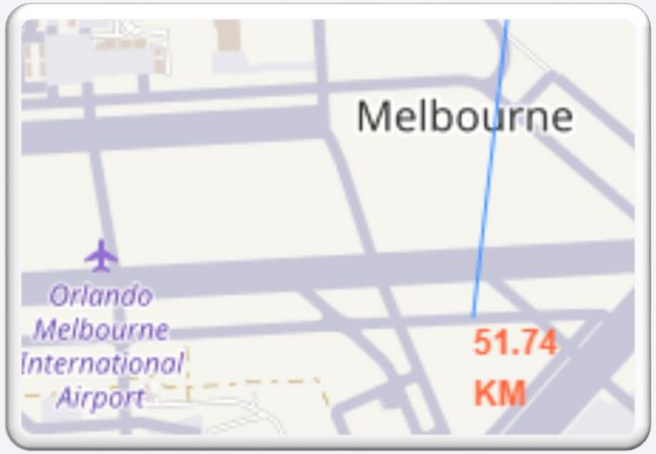


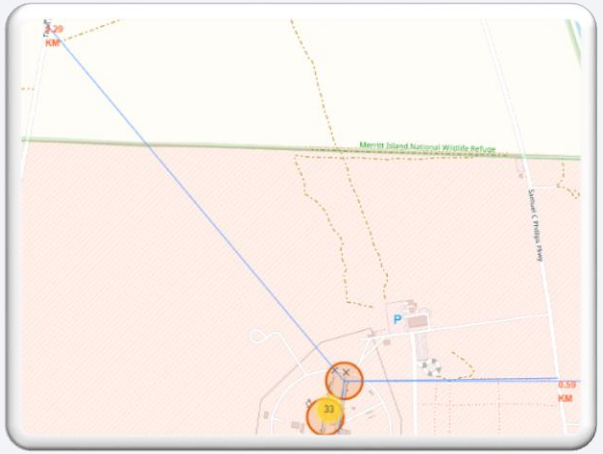
CCAFS SLC-40 and CCAFS LC-40



Location Proximities of Launch Sites to important locations





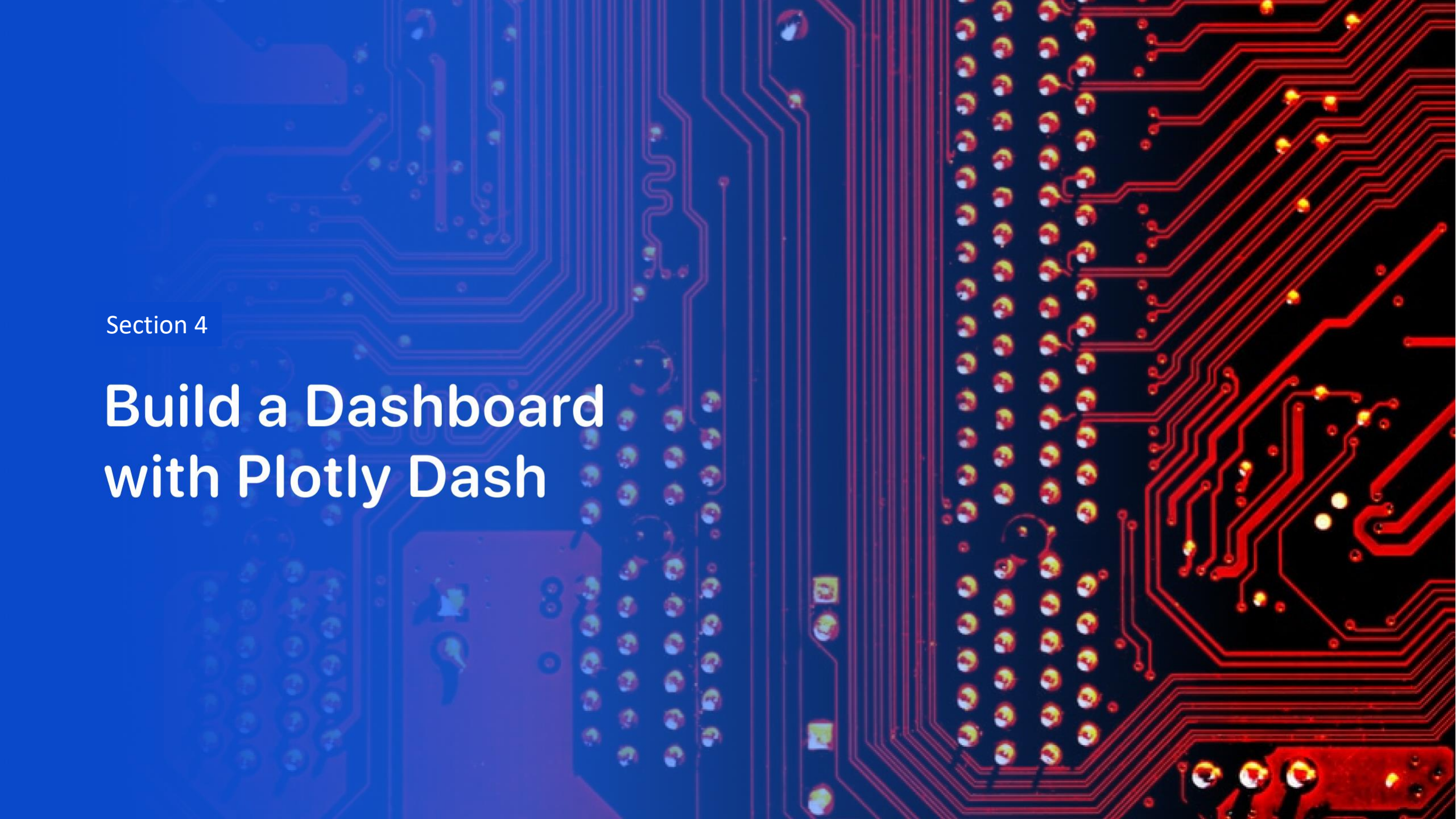


- Launch sites in close proximity to railways? YES.
- Launch sites in close proximity to highways? YES.

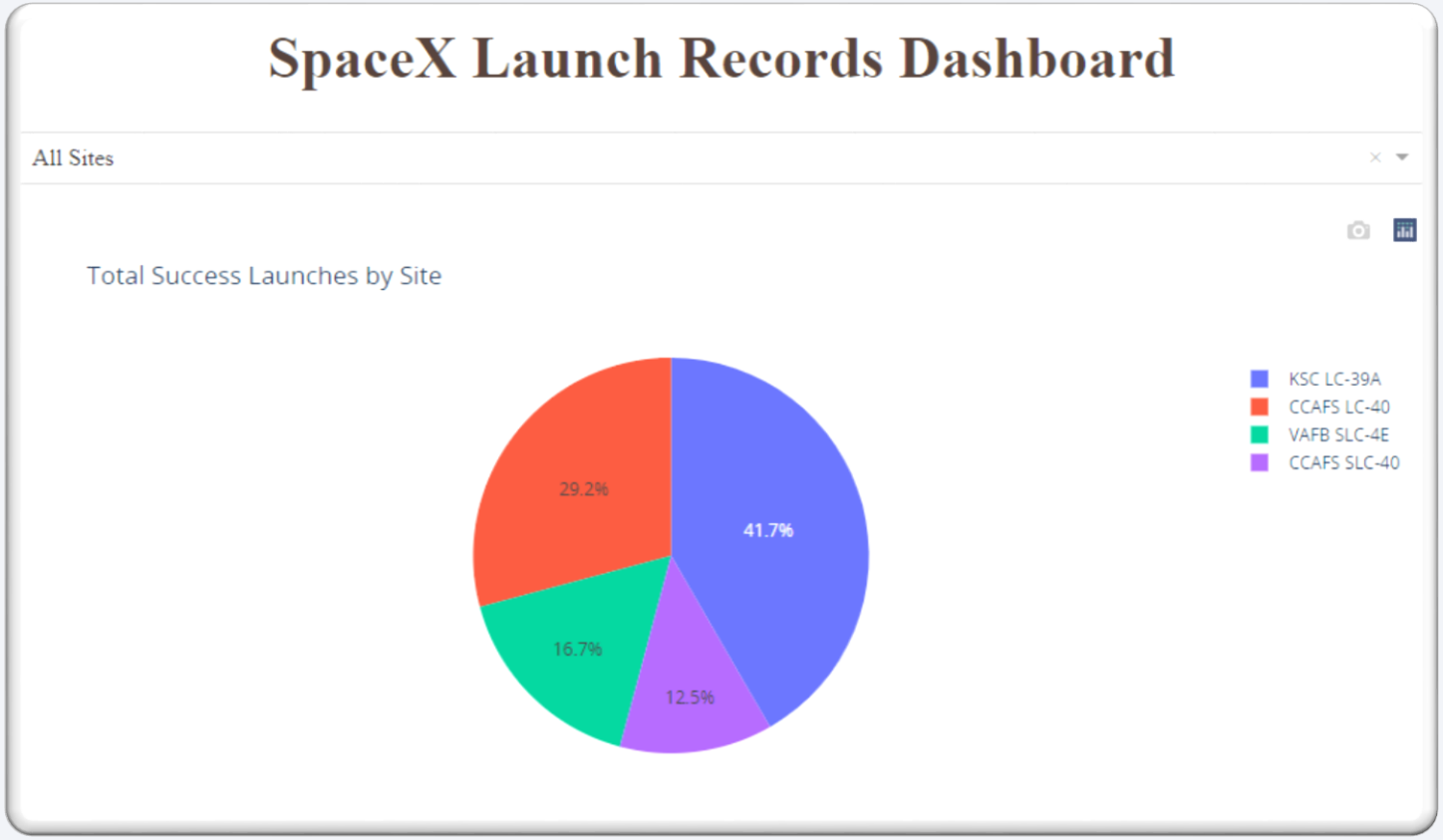
Nearest highway = 0.59km away.

- Launch sites in close proximity to railways? YES.

Nearest railway = 1.29 km away.

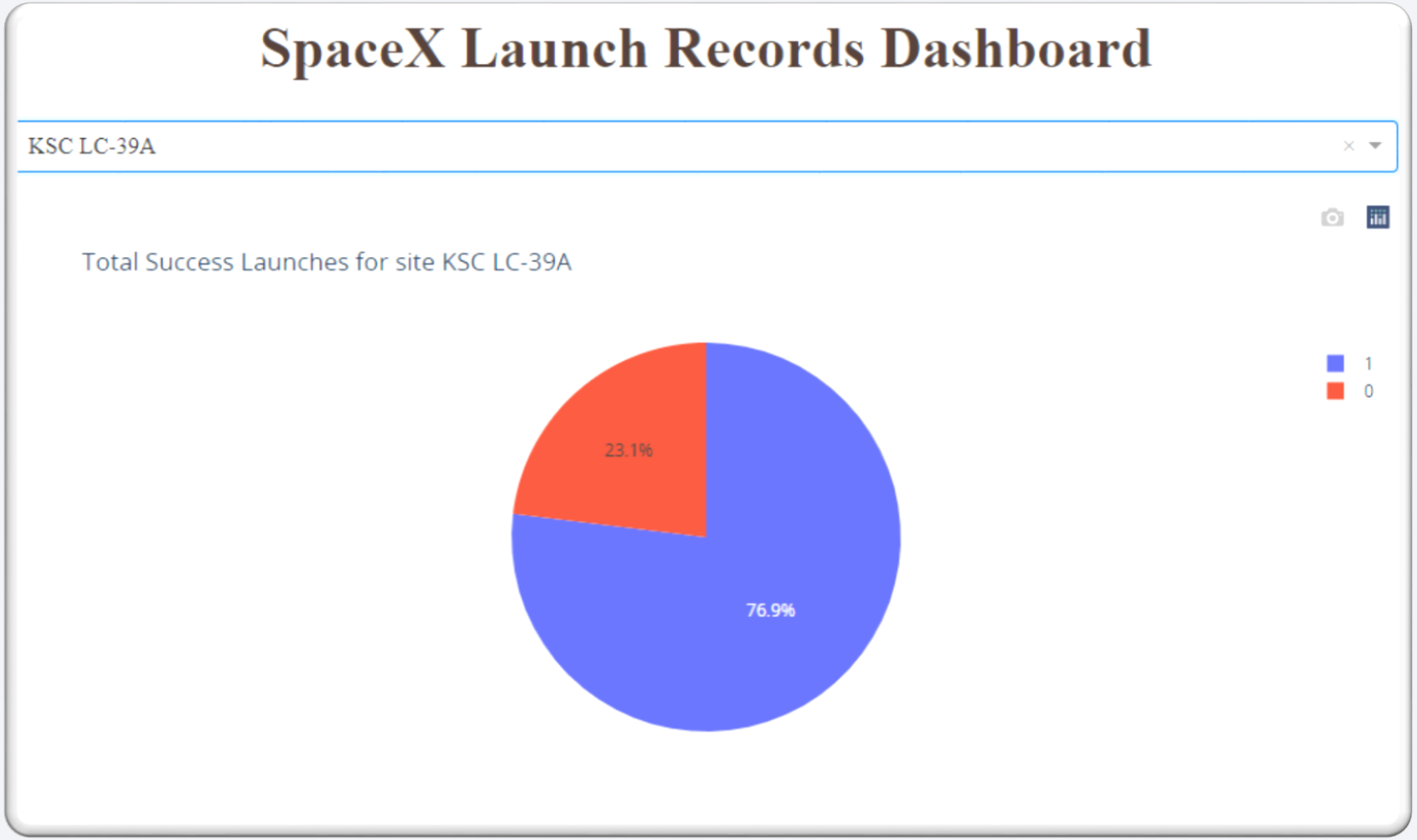

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



- The launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches.

launch site with highest launch success ratio



The launch site KSC LC-39 A also had the highest rate of successful launches, with a 76.9% success rate.

Launch Outcome VS. Payload scatter plot for all sites

- Plotting the launch outcome vs. payload for all sites shows a gap around 4000 kg, so it makes sense to split the data into 2 ranges:
  - 0 – 4000 kg (low payloads)
  - 4000 – 10000 kg (massive payloads)
- From these 2 plots, it can be shown that the success for massive payloads is lower than that for low payloads.
- It is also worth noting that some booster types (v1.0 and B5) have not been launched with massive payloads.

Note: $class \begin{cases} 0, Failure \\ 1, Success \end{cases}$

Section 5

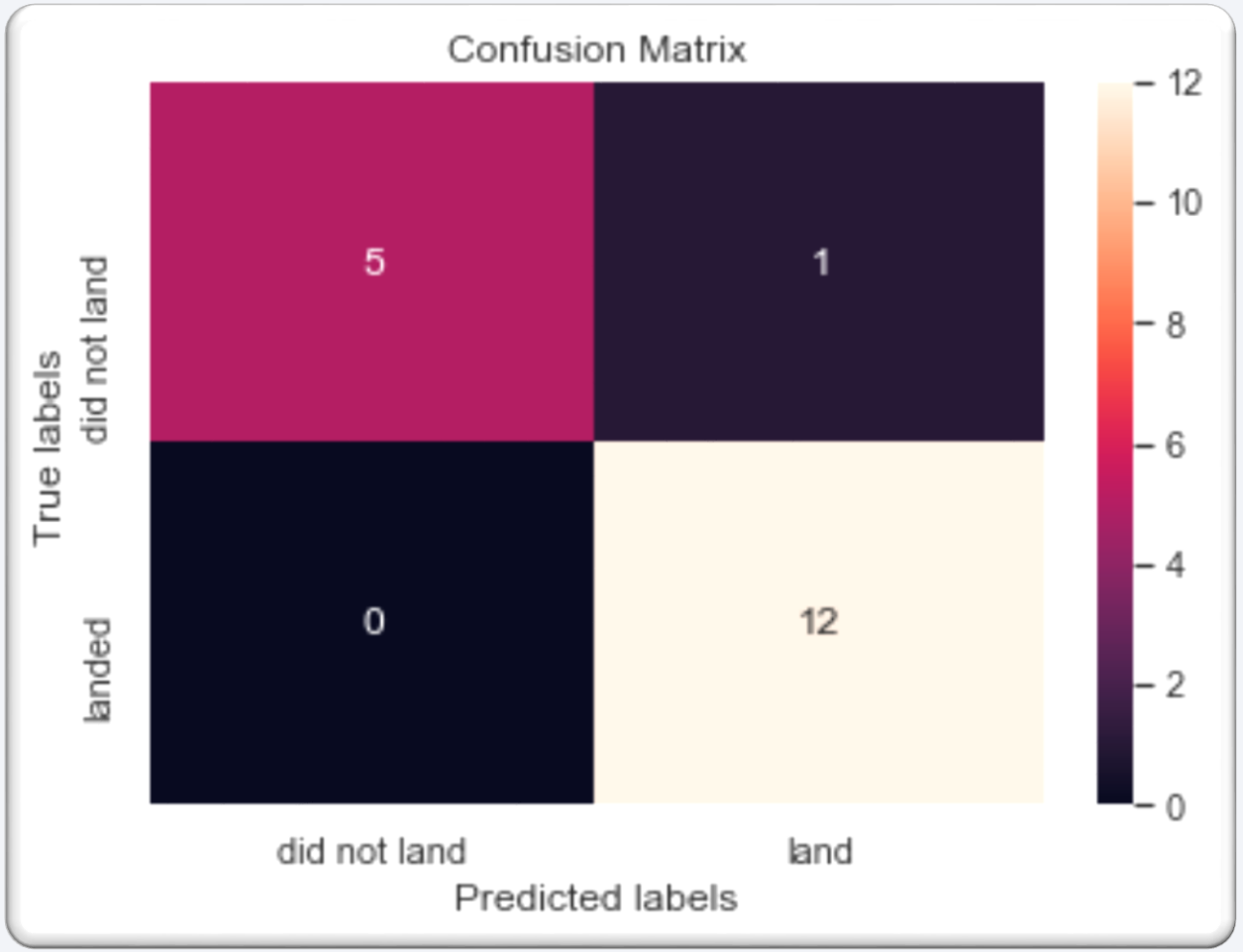
Predictive Analysis (Classification)

Classification Accuracy

- Plotting the Accuracy Score and Best Score for each classification algorithm produces the following result:
- The Decision Tree model has the highest classification accuracy
  - The Accuracy Score is 88.88%
  - The Best Score is 88.88%

	Algorithm	Accuracy Score	Best Score
0	KNN	0.833333	0.846429
1	Decision Tree	0.833333	0.848214
2	Logistic Regression	0.888889	0.885714
3	SVM	0.833333	0.848214

Confusion Matrix



- As shown previously, best performing classification model is the Decision Tree model, with an accuracy of 94.44%.
- This is explained by the confusion matrix, which shows only 1 out of 18 total results classified incorrectly (a false positive, shown in the top-right corner).
- The other 17 results are correctly classified (5 did not land, 12 did land).

Conclusions

- As the number of flights increased, the rate of success at a launch site increased.
- Most of the early flights were unsuccessful.
- Between 2010 and 2013, all landings were unsuccessful
- After 2013, the success rate generally increased, despite small dips in 2018 and 2020.
- Orbit types ES-L1, GEO, HEO, and SSO, have the highest success rate of 100%.
- Launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches, and also the highest rate of successful launches, with a 76.9% success rate.

Appendix

- https://github.com/kkamalk/ibm_final.git

Thank you!