IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hamza Salman
July 2024

Contents

Executive Summary

This capstone explores ways of predicting SpaceX's first stage Falcon 9 rocket launch outcomes based on historical data. The key stages for doing so include the following:

1. Data Collection

2. Data Wrangling

3. Exploratory Data Analysis (EDA)

4. Interactive Visual Analytics and Dashboard

5. Predictive Analysis (Classification)

It was found that recent missions have a substantially higher success rate than older ones (2015 and prior), and that missions with higher payload masses are more frequently successful. Additionally, decision tree classification appears to be the most accurate machine learning algorithm for predicting outcomes.

Introduction

Objectives:

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

Intended Outcomes:

- Use findings and relationships from data analysis to 1) determine the price of each launch prior to the operation, and 2) determine if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Summary of methodologies

1. **Data Collection** using SpaceX API and web scraping

2. **Data Wrangling** using NumPy and Pandas data frames

3. **Exploratory Data Analysis (EDA)** using SQL, Pandas, and Matplotlib

4. **Interactive Visual Analytics and Dashboard** using Folium and Plotly Dash

5. **Predictive Analysis (Classification)** using logistic regression, support vector machine (SVM), k-nearest neighbors (KNN), and decision tree.

Data Collection

SpaceX launch data was collected using the SpaceX REST API and web scraping methods for additional information on Wiki pages. Data collected includes rocket used, payload delivered, launch specifications, landing specifications, and landing outcome

Data Collection – SpaceX API

Define functions that identify data frames within the SpaceX API link by creating a response with added string elements to the link

Append the collected data to lists that will later be converted to data frame columns

Print the contents of the SpaceX launch API requests

Sample code:

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

https://github.com/Hamzsal/Final/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Request the Falcon9 Launch Wiki page as an HTTP response from its URL using an HTTP GET method

Create a BeautifulSoup object from the HTML response and extract all column/variable names from the HTML table header

Iterate through the `<th>` elements and apply the provided `extract_column_from_header()` function to extract column names.

Create a data frame by parsing the launch HTML tables

https://github.com/Hamzsal/Final/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling Stages

https://github.com/Hamzsal/Final/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

- Import the collected dataset from the data collection stage
- Identify and calculate the percentage of the missing values in each attribute
- Identify which columns are numerical and categorical
- Replace missing values with the mean
- Calculate the number of launches on each site and occurrence of each orbit
- Create a binary classification to categorize landings based on their success (one-hot encoding)

EDA with Data Visualization

Visualization Methods:

- Scatter plot – allows for clear identification of patterns using grouped individual events
- Bar chart – allows for clear comparison between different variables
- Line plot – displays the relationship between two variables

https://github.com/Hamzsal/Final/blob/main/edadataviz%20(copy).ipynb

EDA with SQL

Queries used:

- **Distinct query** → used to identify unique elements
- **Where query** → establish conditions for extracting data
- **Like query** → used to extract information where certain variables are involved
- **Sum, average, and minimum queries** → numerical operations
- **Between query** → extracting information within a certain range
- **Subquery** → combining multiple queries for embedded operations

https://github.com/Hamzsal/Final/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Objects used:

- Markers – used to identify distinct launches and their properties
- Circles – used to present the launch location of rockets
- Lines – used to visualize proximity of different areas to the launch locations

https://github.com/Hamzsal/Final/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

Add each site's location on a map using site's latitude and longitude coordinates

Add a circle for each launch site in data frame

Create markers for all launch records

Identify coordinates of key points and apply the PolyLine function to determine proximity

Build a Dashboard with Plotly Dash

Graphs and interactions used:

- Pie chart: used to show and compare ratios for different launch sites
- Scatter plot: used to display the relationship between payload mass and launch success
- Slider: used to modify the payload mass range for the scatter plot
- Dropdown menu: used to select different data categories (all graphs change accordingly)

https://github.com/Hamzsal/Final/blob/main/Final%20Project%20-%20Plotly%20Dash%20Code.py

Predictive Analysis (Classification)

Machine learning objects used:

- Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Decision Tree

https://github.com/Hamzsal/Final/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Load the data frame, standardize the data, then split it into training and testing data

Create machine learning objects then create a GridSearchCV object

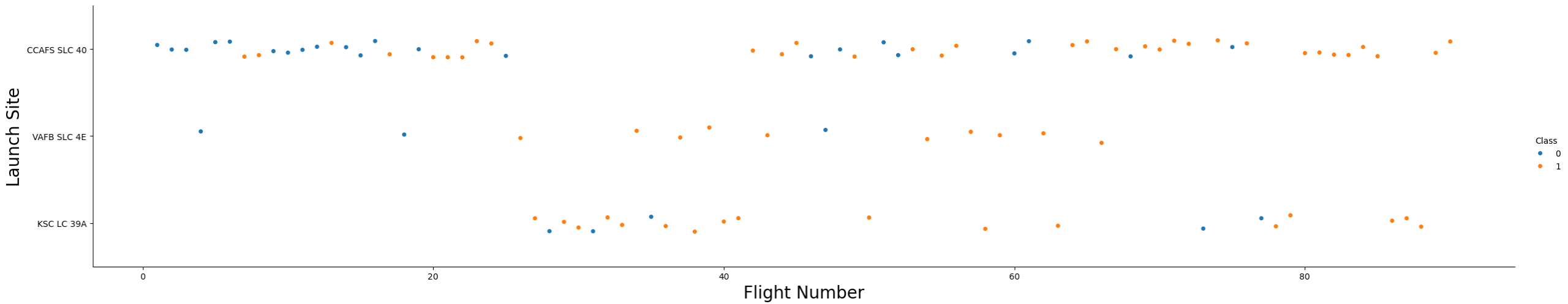
Fit the object to find the best parameters and calculate the model's accuracy on the test data

Create a confusion matrix

Section 2

Insights drawn from EDA

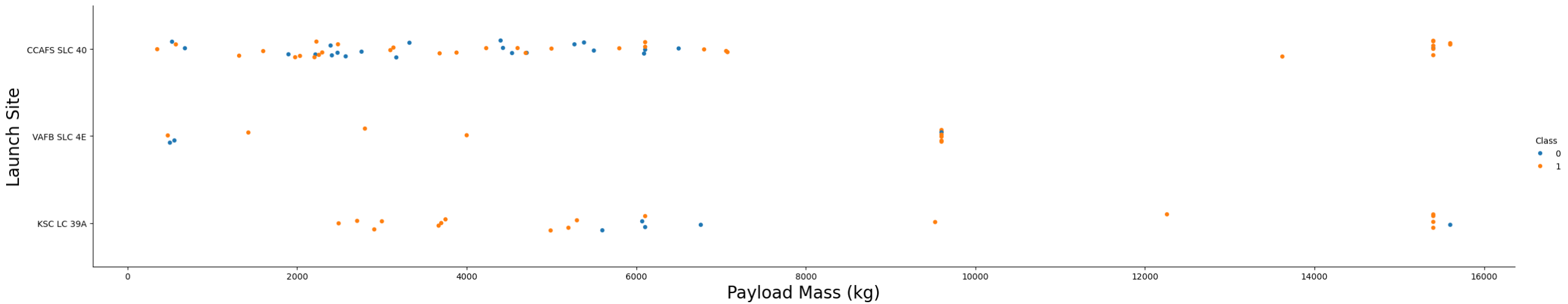
Flight Number vs. Launch Site



Key takeaways:

- CCAFS SLC 40 → greater rates of success with more flights (100% success rate on the last 8 flights compared to 25% in the first 8).
- VAFB has a 77% success rate but the fewest flights. As with CCAFS, the success rate increased with later flights
- Highest success rate for KSC (77.3%)
- Last 5 flights for all launch sites have been successful

Payload vs. Launch Site



Key takeaways:

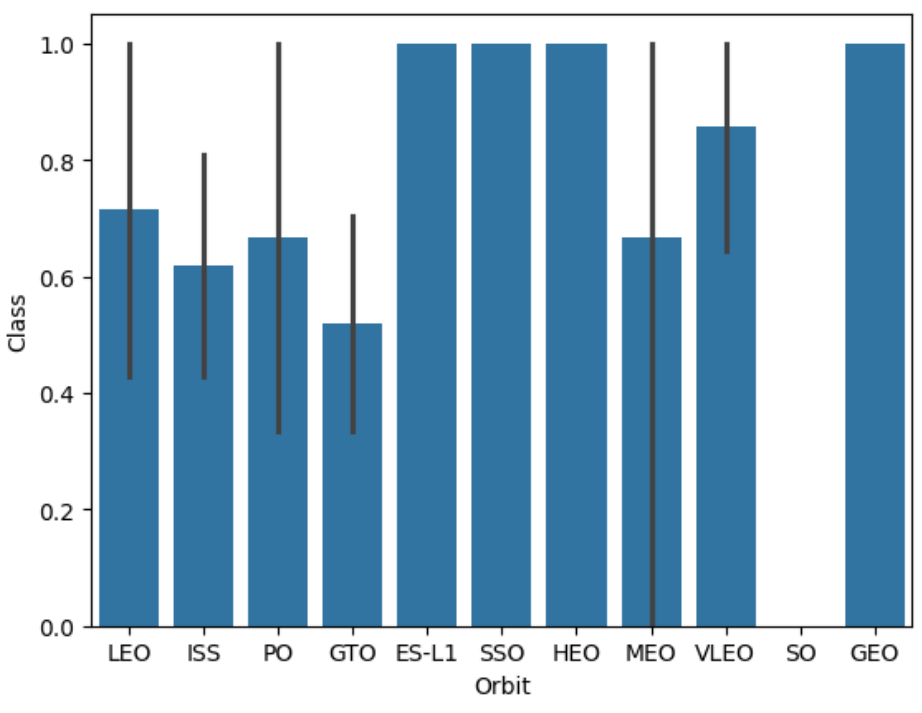
- CCAFS has a low success rate (56.5%) for low payload masses (below 8000kg) and a 100% success rate for payload masses above 8000kg
- High success rate for VAFB with the best performance in the 1500-4000kg range
- High success rate for KSC on low and high payload masses but not middle masses

Success Rate vs. Orbit Type

Lowest 3 success rates: GTO, ISS, SO **Highest 4 (tie):** ES-L1, SSO, HEO, GEO

Potential reasons:

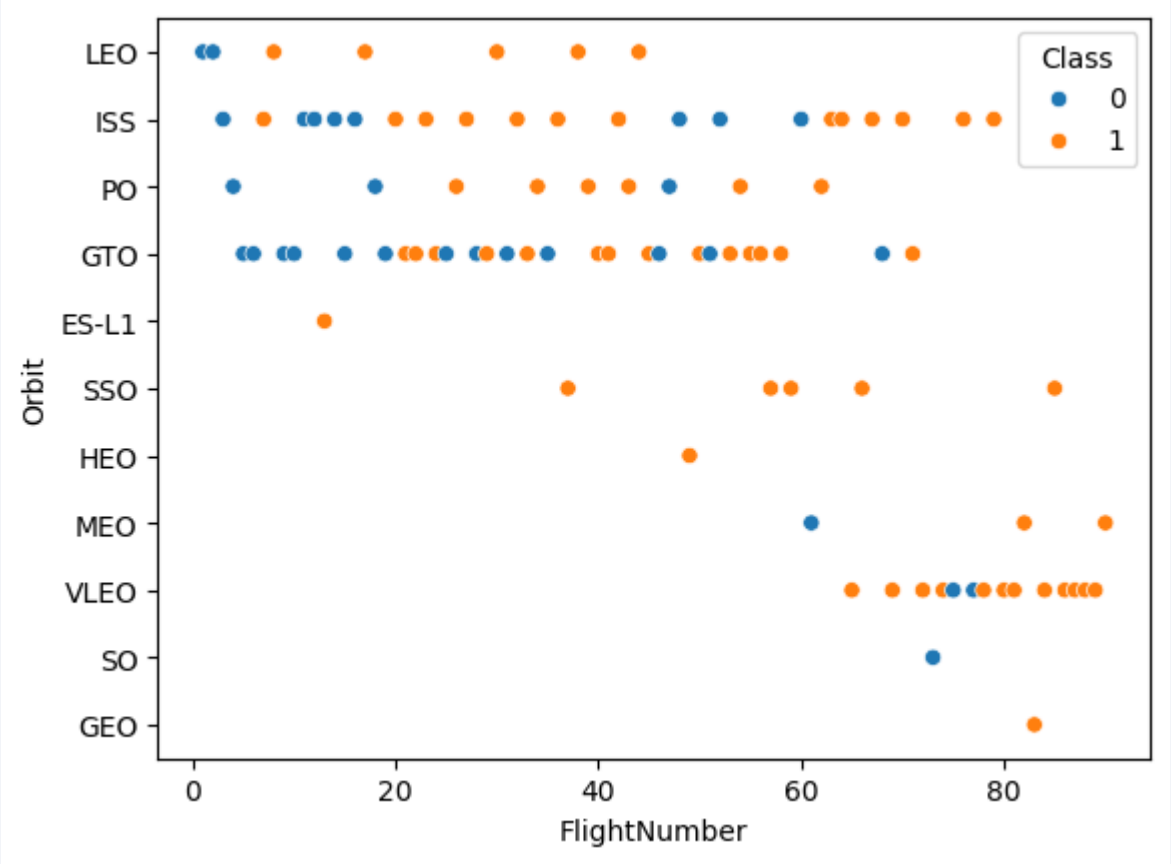
- GTOs require great precision in order to allow launched rockets to match the Earth's rotation and they carry high payloads.
- ISS orbit launches require great precision in accordance with the International Space Station's orbit and they carry cargo and crew, adding more complexity to the mission.
- SO operations require the rockets to undergo constant maneuvering in order to reach the desired synchronous orbit. This makes the operation very complex.
- Once a spacecraft reaches ES-L1, it remains relatively stable with minimal propulsion adjustments, reducing risks associated with maintaining the orbit.
- HEOs offer more flexibility in terms of orbital insertion points compared to geostationary orbits, reducing the risk associated with precise insertion.
- SSO's have a plethora of pre-existing experimental data, giving new operations reliable references.
- Caveats: HEO, GEO, and SO rockets were only launched once, so the success rate is not an accurate indicator of the performance.



Flight Number vs. Orbit Type

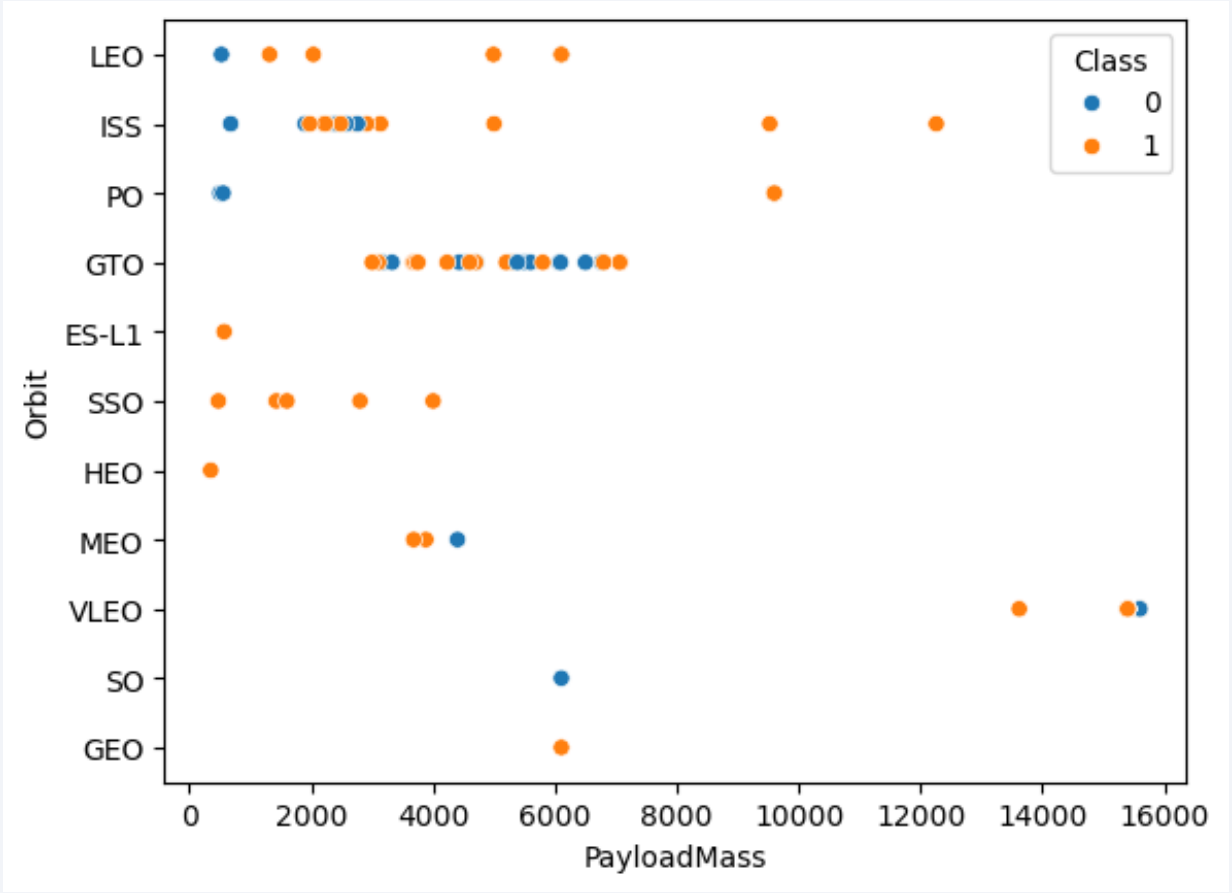
Key takeaways:

- ES-L1, HEO, GEO, and SSO orbits have the highest success rate (100%) but some of the lowest number of flights
- ISS, GTO, and VLEO have the greatest number of attempts
- LEO, ISS, PO, and GTO missions have seen improvements in success rates with more flights



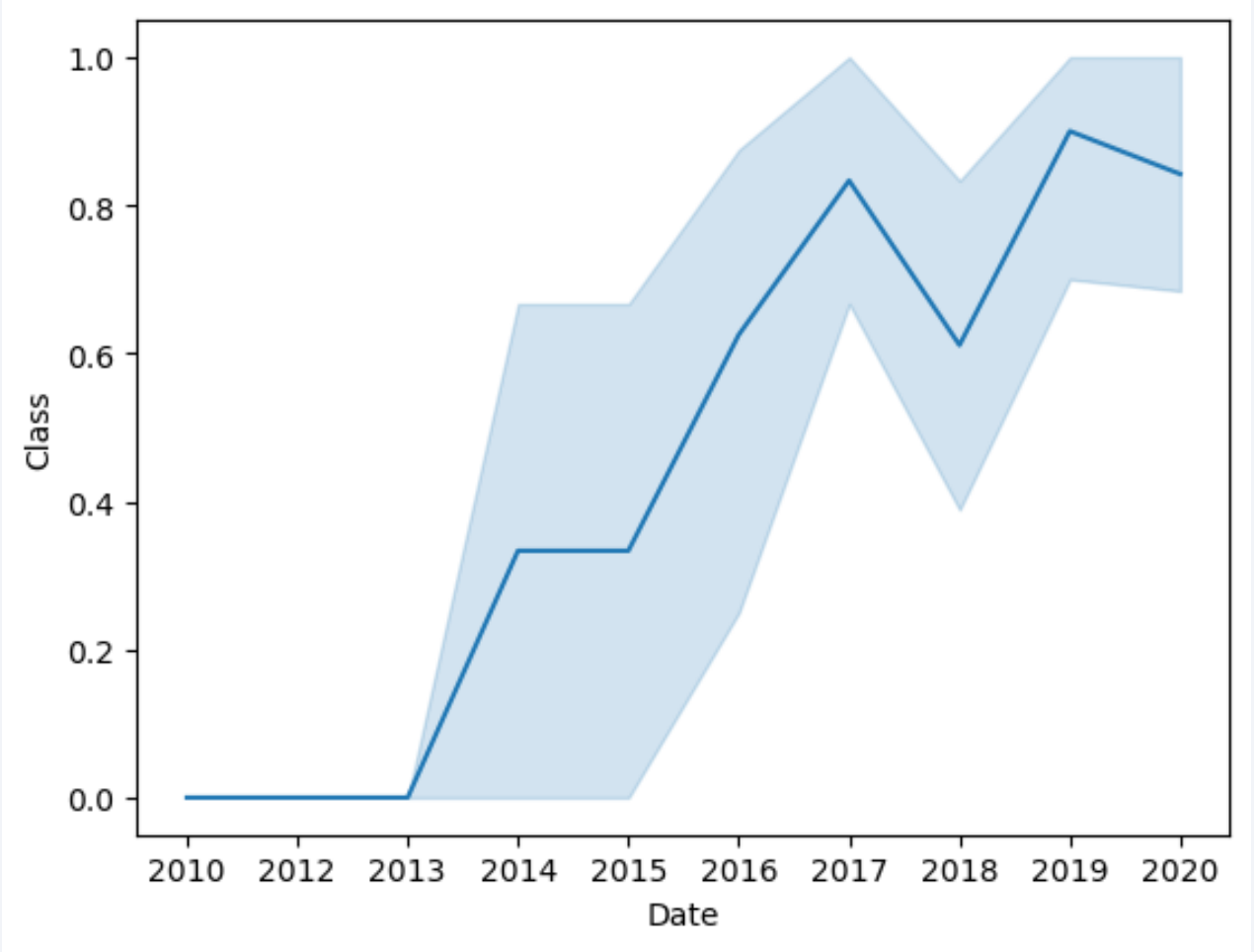
Payload vs. Orbit Type

- Most missions carry low payload masses. This can be attributed to lower costs and energy requirements.
- ISS has the greatest variety of tests; it is more successful with higher payload masses.
- No particular trends for other orbits.



Launch Success Yearly Trend

The rocket launch success rate has seen a consistent increase from 2013 to 2020 and one significant drop from 2017 to 2018, which was quickly offset in 2019.



All Launch Site Names

Query

```
%sql select distinct Launch_Site from SPACEXTBL
```

Output

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

*Distinct query used to avoid duplicates

Launch Site Names Begin with 'CCA'

Query

```
%sql select * from SPACEXTBL where Launch_Site like '%CCA%' limit 5
```

Output

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

Key Observations:

- Either failure or no landing
- All LEO orbits
- Low payload masses
- All launched in 2010-2013

Total Payload Mass

Query

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL
```

Output

Total payload mass = 619967kg

Average Payload Mass by F9 v1.1

Query

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like '%F9 v1.1%'
```

Output

Average payload mass = 2534.7kg

First Successful Ground Landing Date

Query

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome like '%Success%'
```

Output

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Query

```
%sql select Booster_Version from SPACEXTBL where Landing_Outcome like '%Success (drone ship)%' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

Output

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

Total Number of Successful and Failure Mission Outcomes

Query

List the total number of successful and failure mission outcomes

```
%sql select count(Mission_Outcome) as Successful from SPACEXTBL where Mission_Outcome like '%Success%'
```

Output

```
 * sqlite:///my_data1.db
Done.
```

Successful
100

Query

```
%sql select count(Mission_Outcome) as Failure from SPACEXTBL where Mission_Outcome like '%Failure%'
```

Output

```
 * sqlite:///my_data1.db
Done.
```

Failure
1

99% of mission outcomes are successful
(not to be confused with landing outcomes)

Boosters Carried Maximum Payload

Query

```
%%sql

WITH summed_payloads AS (
    SELECT Booster_Version, SUM(PAYLOAD_MASS__KG_) AS total_payload_mass
    FROM SPACEXTBL
    GROUP BY Booster_Version
),
max_payload AS (
    SELECT MAX(total_payload_mass) AS max_total_payload_mass
    FROM summed_payloads
)

SELECT Booster_Version, total_payload_mass
FROM summed_payloads
WHERE total_payload_mass = (SELECT max_total_payload_mass FROM max_payload);
```

Output

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

12 boosters share the maximum payload mass at 15600kg

2015 Launch Records

Data on failed landing outcomes for launches in 2015

Query

```
%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 like '%Failure%'
```

Output

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

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

The number of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Query

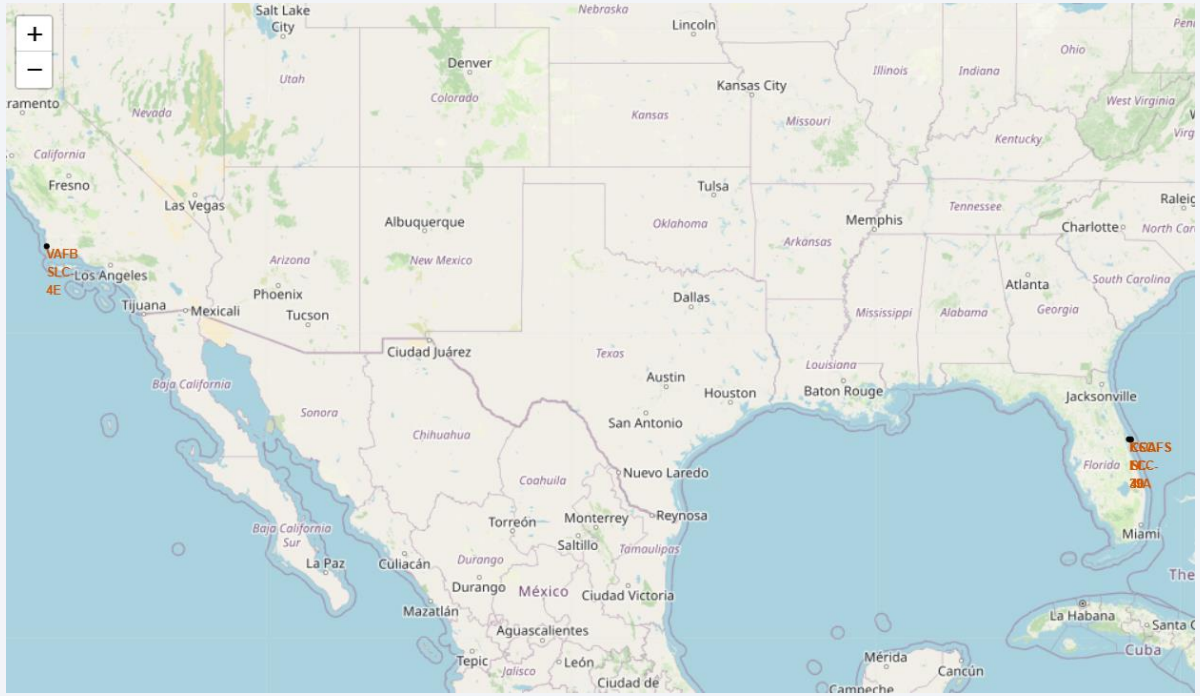
```
%sql select Landing_Outcome, count(Landing_Outcome) from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome
order by count(Landing_Outcome) desc
```

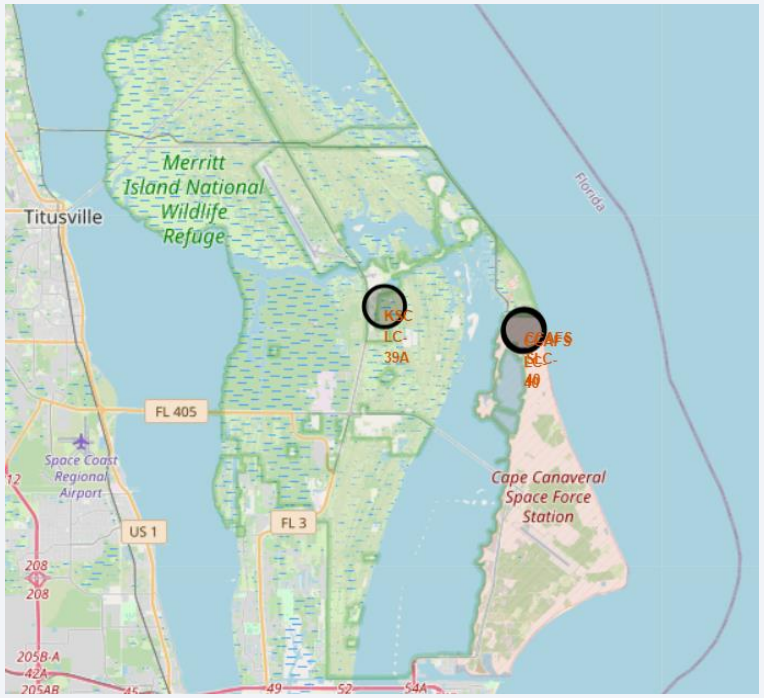
Output

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


Section 3
Launch Sites
Proximities Analysis

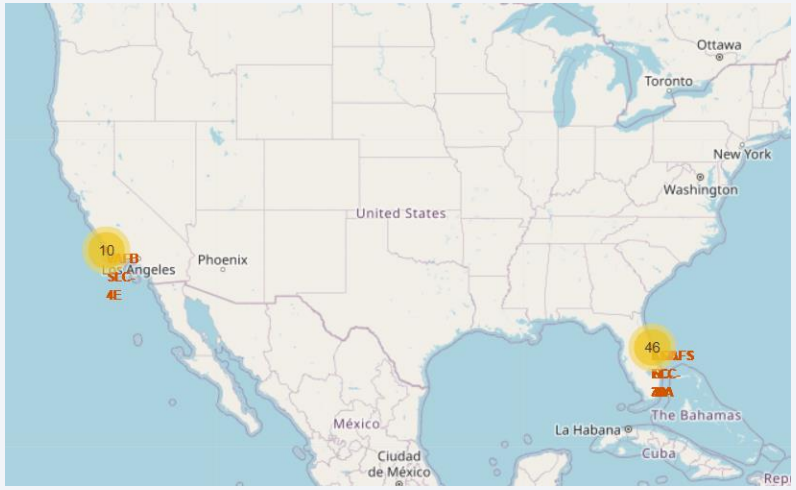
Launch site locations

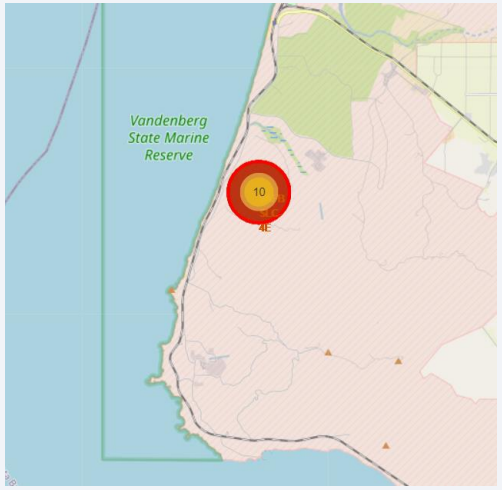


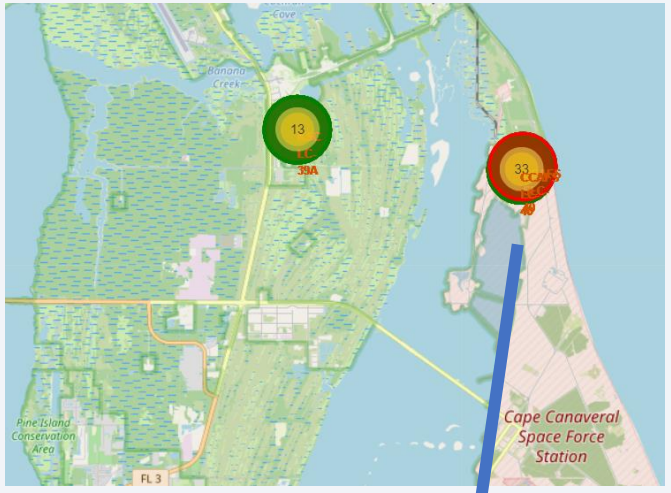


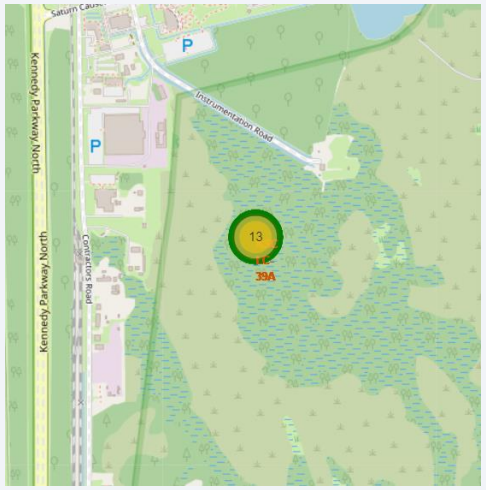
Notice how all site locations are on the far east and west coasts and how close they are to the equator line. Reasons for coast proximity are discussed in slide 37. As for the equator proximity, launching rockets near the equator line allows SpaceX to maximally utilize Earth's rotational speed.

Number of launches and success rates









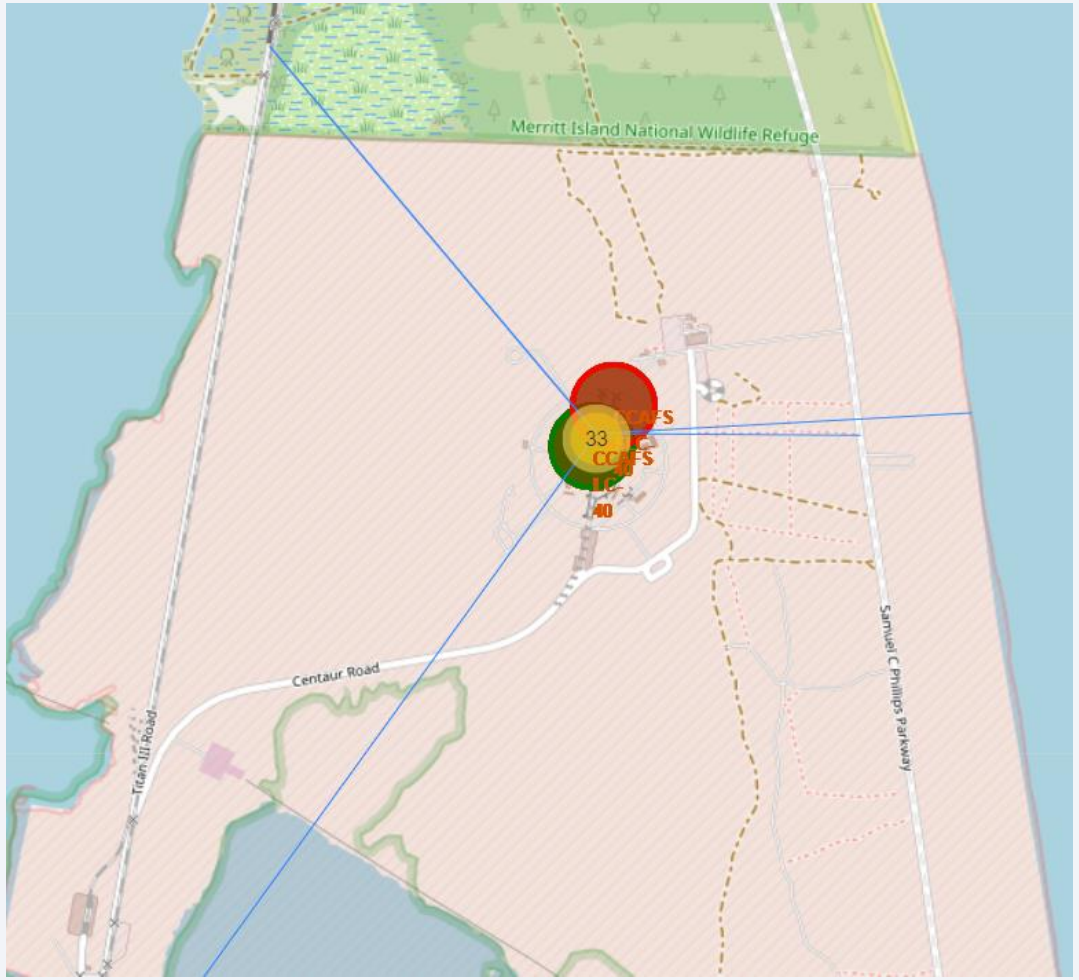
46 launches on the east coast and 10 on the west coast

Yellow circles around numbers indicate a near even split of successful and failed attempts

Zooming in creates more circles for better distinguishing

East coast attempts are generally more successful, but the greater number of attempts must be acknowledged

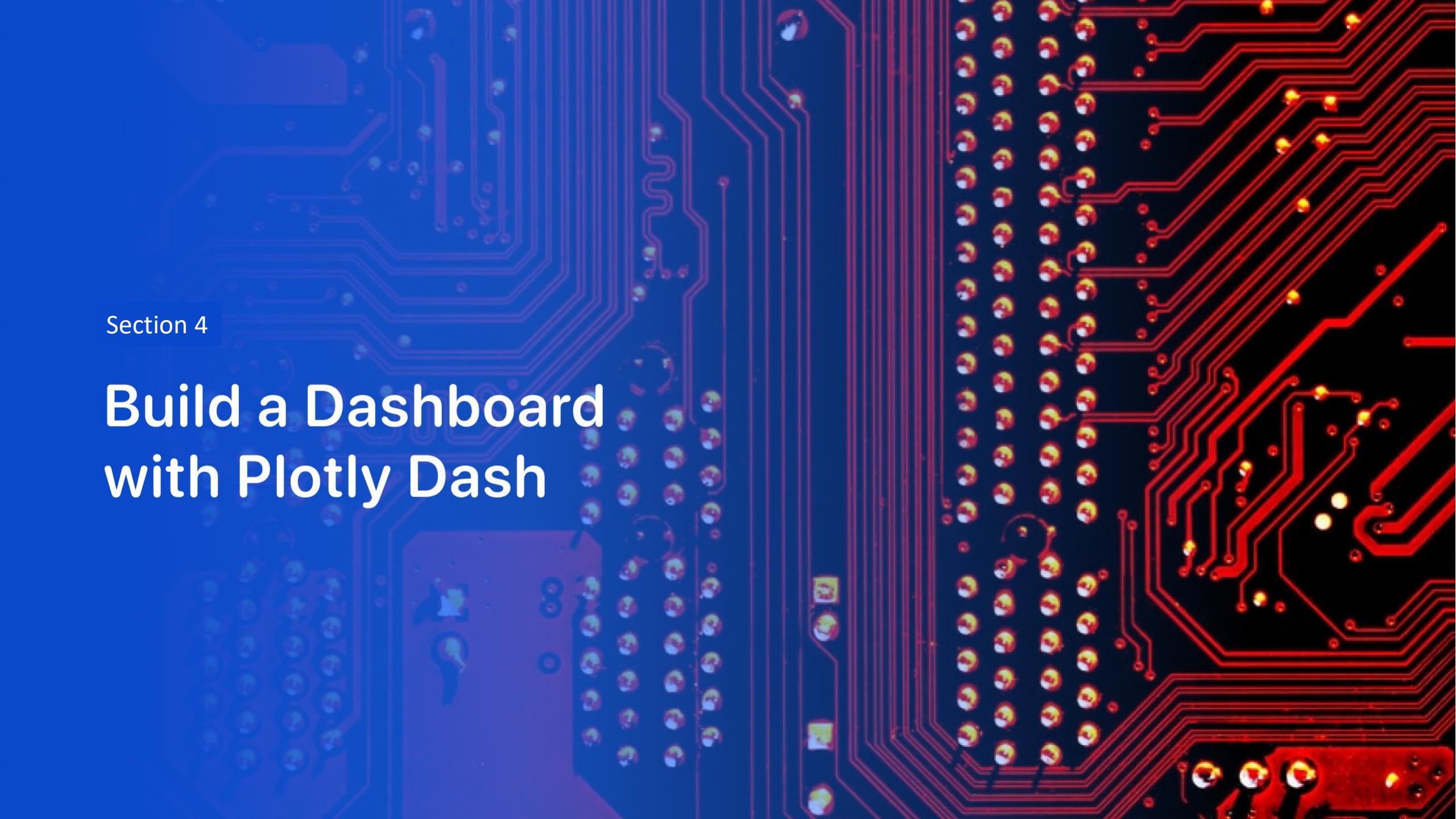
Launch site proximities



Values given in kilometers (km)

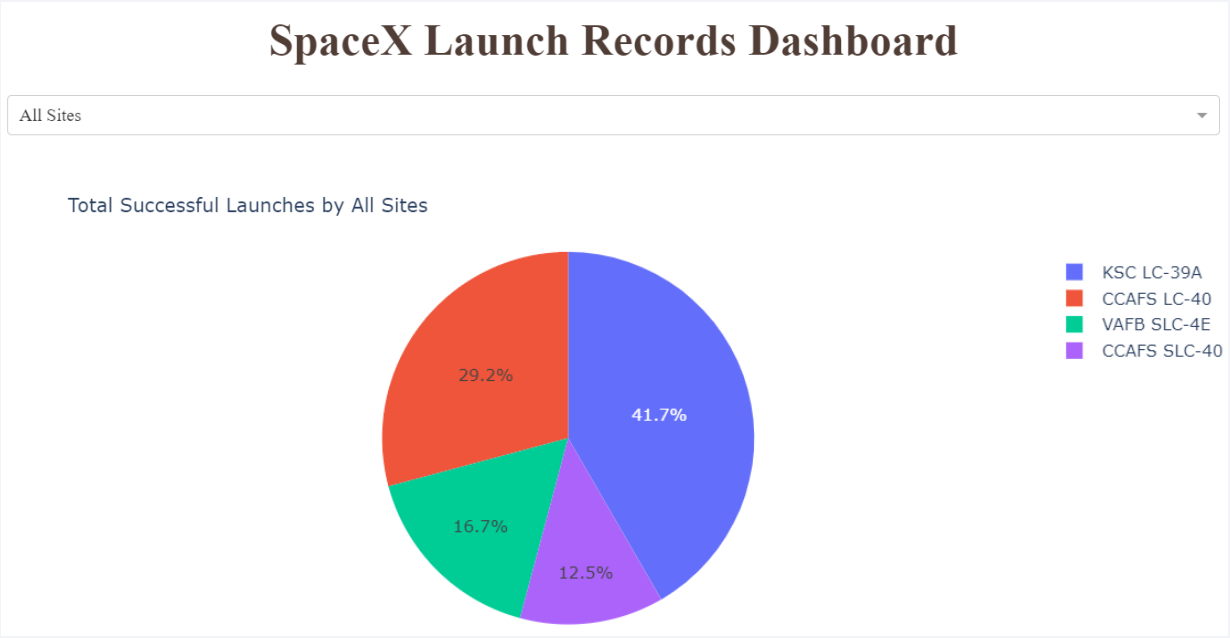
```
Distance to railway: 1.2334744707220642
Distance to highway: 0.641864857782944
Distance to coastline: 0.9156512111639094
Distance to city (Rockledge): 31.40987575231474
```

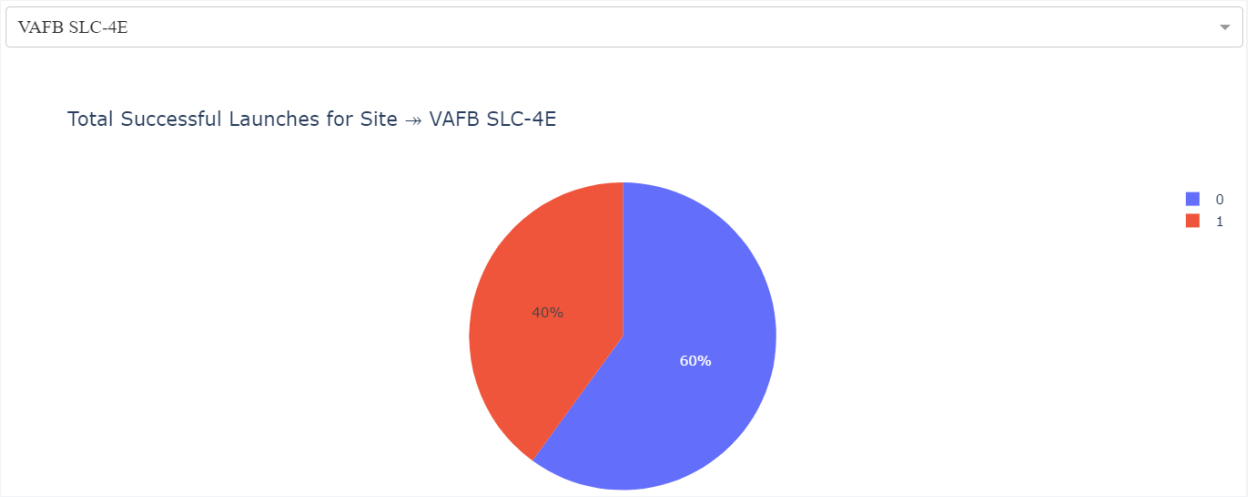
Launch sites are positioned near railways and highways for convenient transportation of required components when setting up missions. They are also near coastlines and away from cities for safety purposes as well as for easy recovery of components that detach from the rockets.


Section 4
Build a Dashboard with Plotly Dash

Successful Launches Filtered by Site

Multiple pie charts can be generated on the dashboard based on the launch site (see drop down menu with the label ‘All Sites’)

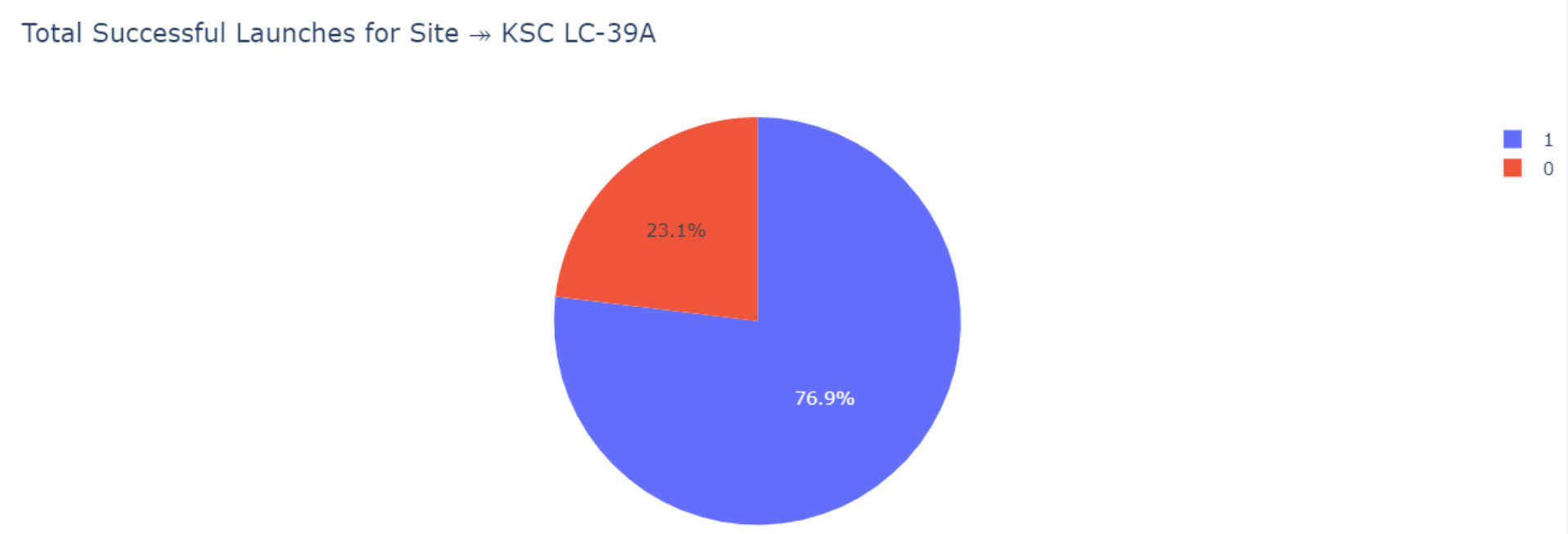




Selecting individual locations displays the percentage of successful/unsuccessful attempts where 0 represents failure and 1 represents success

Launch Site with the Highest Success Rate

- KSC LC-39A has the highest successful launch rate at 76.9%
- This is significantly greater than the second most successful launch site (CCAFS LC-40), which has a 26.9% success rate



Relationship Between Payload Mass and Launch Success

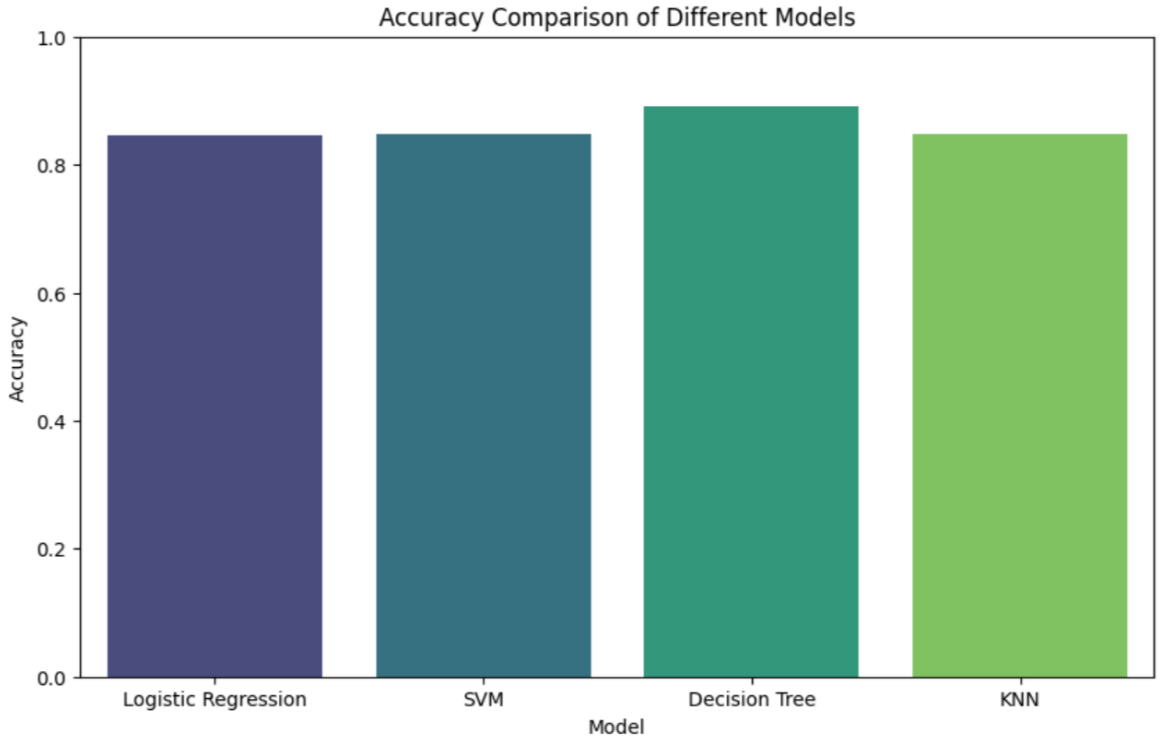
- The figure below displays the success rate for different sites based on payload mass.
- In the selected range below (4000-7000kg), we see that most launch attempts failed and the FT category has the most successful launches (although with a 44.4% success rate)
- Moving the slider displays the success rate for different mass ranges

Slider to select desired payload mass range

Section 5

Predictive Analysis (Classification)

Classification Accuracy



Bar chart displaying the classification accuracy for different machine learning models

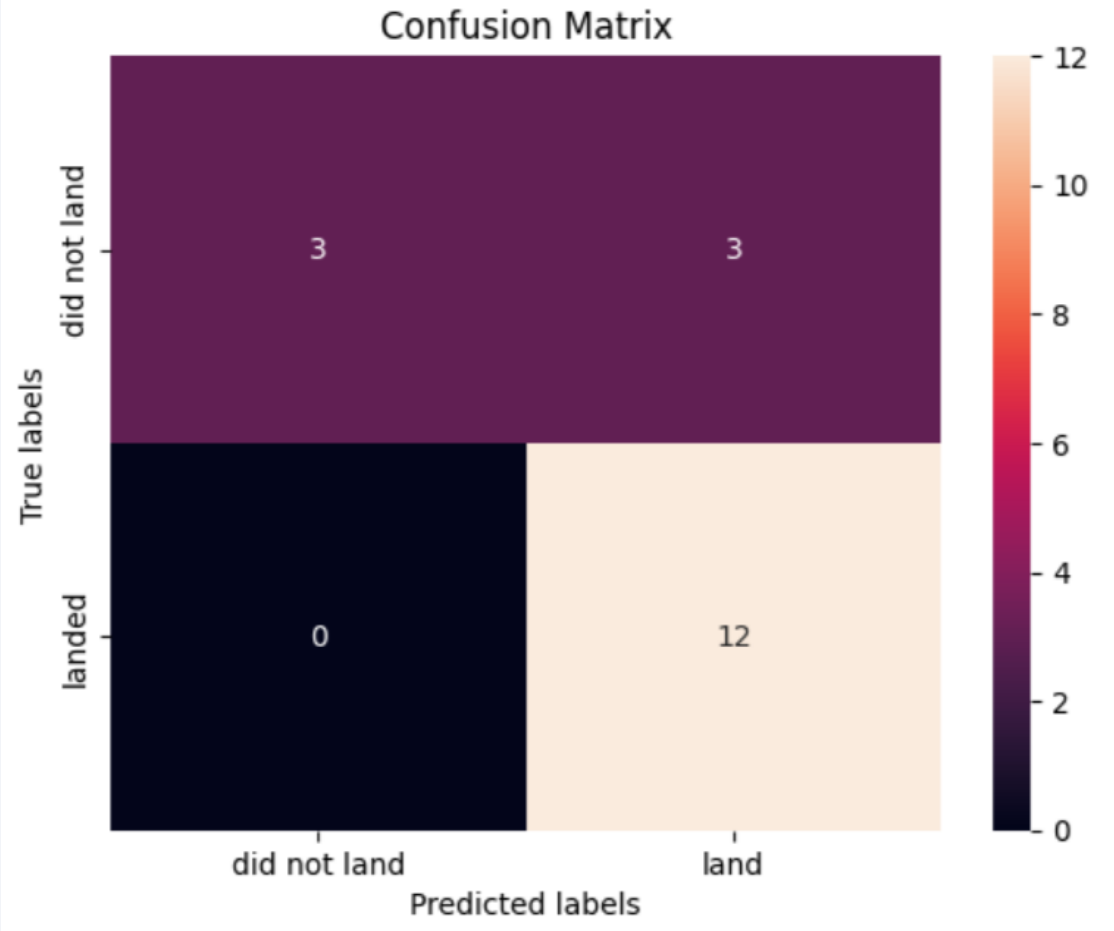
Classification accuracy ranking:

1. Decision Tree (89.1%)
2. SVM (84.8%) – larger 16th sig. fig.
3. KNN (84.8%) – smaller 16th sig. fig.
4. Logistic Regression (84.6%)

Confusion Matrix of the Best Performing Model

Key takeaways:

- The best performing model was decision tree with a prediction accuracy of 89.1%
- Most predictions (12) were true positives
- No predictions were false negatives
- 3 predictions were true negatives and false positives each
- 15 predictions were true and 3 were false overall



Conclusions

- KSC LC-39A is the most successful launch location
- Decision tree classification is the most accurate machine learning algorithm for predicting launch outcomes
- Launch success has seen a significant increase over the last decade
- F9 boosters carry the greatest payload masses
- ES-L1, HEO, GEO, and SSO orbits have the highest success rate (100%) but some of the lowest number of flights
- Greater launch attempts on the East Coast
- Launch sites are near oceans and at a safe distance away from cities
- Higher payload masses have greater success rates but fewer attempts

Appendix

See all Python code, SQL queries, charts, Notebook outputs, and data sets in the following GitHub repository: https://github.com/Hamzsal/Final

Thank you!