IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rudi Gallegos
January 2nd, 2023

OUTLINE

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

EXECUTIVE SUMMARY

Summary of Methodologies

- Data collection via web scraping and Rest API queries
- Data wrangling to classify launches by success.
- Exploratory Data Analysis with SQL
- Data Visualization with python packages, and interactive maps with Folium
- Predictive analysis for classification of rocket landing success

Summary of all Results:

- Exploratory data analysis results
- Predictive analysis results

INTRODUCTION

Project Background and Context

In this project, we will predict the success of the Falcon 9 first stage launch. On the SpaceX website, the Falcon 9 rocket is advertised to cost $62 million, while other providers cost upward of $165 million. Much of the savings seen with SpaceX is due to their reuse of the first stage. Thus, if we can determine whether the first stage will land, we can determine the cost of the launch. This information is particularly useful when another company wants to bid against SpaceX.

Problems to be Addressed

- What factors of a mission with affect the success of the Falcon 9 launch?
- What conditions must be met in order for SpaceX to ensure the highest probability of success for a given mission?

Methodology

METHODOLOGY

- Data collection methodology:
  - Web scraped tabular data on SpaceX launches from Wikipedia.
  - Requested past launch data from SpaceX REST API
- Perform data wrangling
  - Removed data for non Falcon 9 launches
  - Transformed categorical data into integrals
  - Replaced missing payload mass data with the sample mean
  - Classified successful landings as 1 and unsuccessful landings as 0
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Used bar graphs and scatterplots to visually represent relationships between variables.
  - Used SQL to explore relationships between variables
- Perform interactive visual analytics using Folium and Plotly Dash
  - Explored launch sites with interactive Folium maps
  - Used plotly to visualize and interact with payload and success launch data
- Perform predictive analysis using classification models

DATA COLLECTION

01

We used SpaceX REST API and web scraped with BeautifulSoup to gather data of Falcon 9 launches

02

Our goal is to predict Falcon 9 landing success based on mulitple variables

*GitHub Notebook: API

Data Collection – SpaceX API

1. Make GET Request from SpaceX API. Then, decode content as a .Json file to convert into a Pandas DataFrame.

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

response = requests.get(spacex_url)
```

2. Clean data to extract desired information.

```
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also ext
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype a
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)]
```

3. Create a dictionary form the cleaned data, then convert that dictionary into a new dataframe.

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

4. Filter the data to only include that of Falcon 9 launches. FInally, replace missing PayLoadMass data with the PayLoadMass average.

```
# Hint data['BoosterVersion']!='Falcon 1'
indexnames = df.loc[df['BoosterVersion']!='Falcon 9'].index
df.drop(indexnames, inplace = True)
data_falcon9 = df
data_falcon9.head()
```

```
# Calculate the mean value of PayloadMass column
pm_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
temp = data_falcon9['PayloadMass'].replace(np.nan, pm_mean)
data_falcon9['PayloadMass'] = temp
data_falcon9
```

*GitHub Notebook: Web Scraping

Data Collection – Web Scraping

1. Request HTML page from static URL, and create a BeautifulSoup object from the response.

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

```
soup = BeautifulSoup(response.text, 'html.parser')
print(soup.prettify())
```

2. Find all tables from the HTML page.

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

3. Extract column names.

```
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))
```

4. Create an empty dictionary with the column names as the keys.

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

5. Finally, fill in parsed launch records into dictionary, and create a Pandas DataFrame.

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

Data Wrangling

*GitHub Notebook: Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

Here, we will mainly convert those outcomes into Training Labels with 1 meaning the booster successfully landed 0 meaning it was unsuccessful.

1. **First we load the data.**

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-
df.head(10)
```

2. **We then identify outcomes as "good" or "bad."**

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

3. **Next, we create a list of training labels.**

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
def bad(item):
    if item in bad_outcomes:
        return 0
    else:
        return 1
landing_class = df['Outcome'].apply(bad)
landing_class
```

4. **Finally, we add the list to the dataframe under the new column "Class"**

```
df['Class']=landing_class
df[['Class']].head(8)
```

EDA with Data Visualization

*GitHub Notebook: EDA with Data Visualization

- We used a few different types of visuals to explore the data, and relationships between certain variables:

Bar Graphs

By grouping our success rate based on orbit, we can visualize the relationship between landing success and orbit distance.

Scatter Plots

Here we can visualize the relationship between different variables:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload Mass vs. Launch Site
- Orbit vs. Flight Number
- Orbit vs. Payload Mass

Line Graph

Here we can visualize the success rate over the years. We expect SpaceX to improve over time, but how fast are they improving?

*GitHub Notebook: EDA with SQL

EDA with SQL

We used SQL queries to gain certain insights on the dataset.

- Displayed 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
- 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. Use a subquery
- List the failed landing outcomes in drone ship, their booster versions, and launch site names for 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 2017-03-20, in descending order

Build an Interactive Map with Folium

We created an interactive map with Folium.

- Using the latitude and longitude coordinates, we created circle markers and labels for each launch site.
- Next, we added color coded markers so illustrate the landing success for each launch; green for a successful landing, and red for an unsuccessful landing.
- Finally, we calculated the minimum distance from launch site to highways, railroads, and coastlines. These distances are represented with lines on our Folium map.

Using these calculations and visuals, we can conclude that launch sites are close in proximity to highways, railroads, and coastlines, and that they average about 50 km away from cities.

Ne

*GitHub Notebook: Data Visualization with Folium

Build a Dashboard with Plotly Dash

We built a Plotly Dashboard to make an interactive web application to visualize the data.

Pie Chart to visualize launch landing success, broken down according to launch site.

- If all sites are selected, we get the proportion of successful launches each launch site contributed.
- If we select one launch site, we see the proportion of all launches that site landed successfully.

Scatter Plot of Payload (kg) vs Landing Success Rate (0 or 1), color coded according to booster version.

- Can select a single site to remove all other site data, or can select all sites to visualize all data
- Payload Mass can be selected via a slider from min to max values.

Predictive Analysis (Classification)

Build

- Transform data to scale columns
- Split data into testing and training sets
- Select machine learning algorithms to use for classification (KNN, logistic regression, SVM, and decision tree)

Evaluate

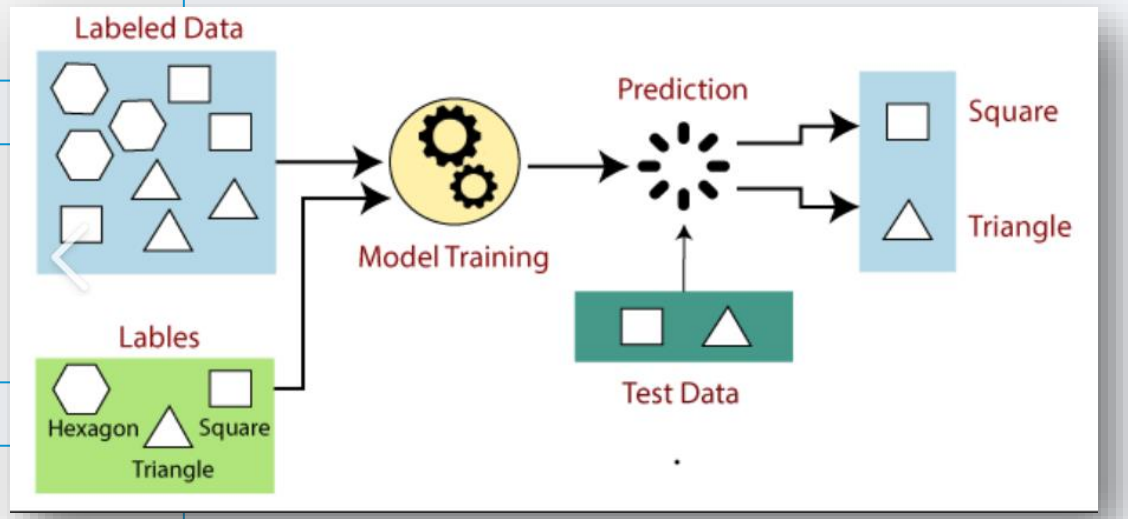
- Evaluate the accuracy of each model on training and testing sets
- Plot confusion matrix for each model
- Use GridSearchCV to determine best parameters for each model

Improve

- Tuning algorithms and parameters

Select

- The model with the highest accuracy score is the best to use for prediction
- If there are multiple models with the same high score, we will check the accuracy of the training set as well.



*GitHub Notebook: Machine Learning and Predictive Analysis

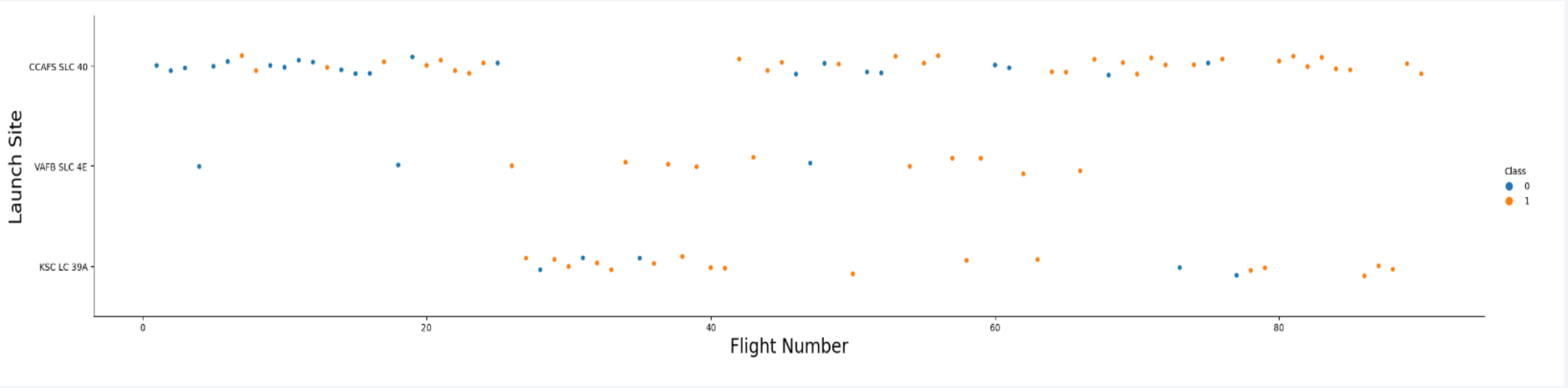
Results

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

Section 2

Insights drawn from EDA

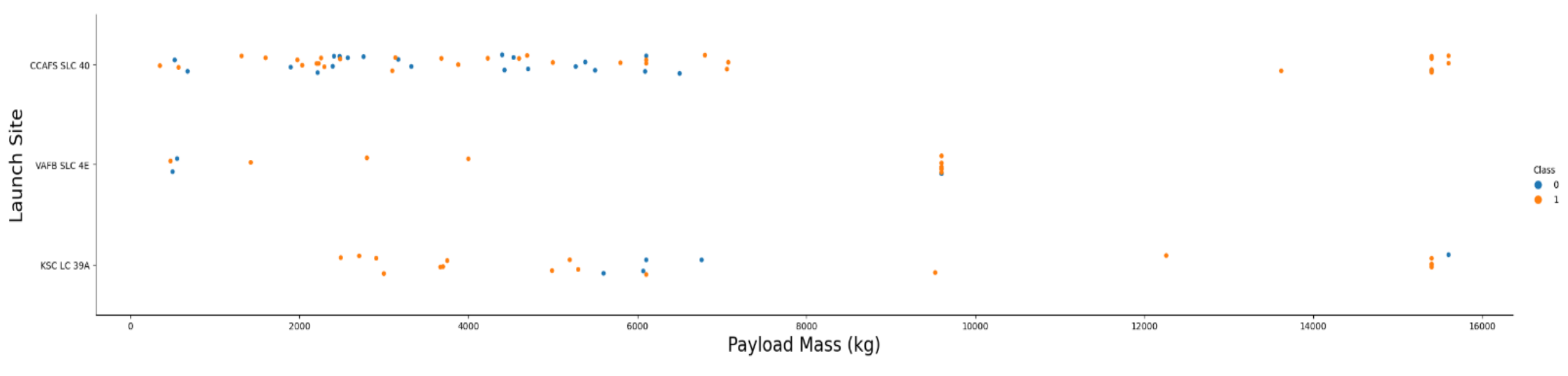
Flight Number vs. Launch Site



From this plot we can see:

- Landing success seemed to increase with flight number. This is true overall and for each individual launch site as well.
- Flight numbers greater than 40 have a much higher success rate than those less than 40
- Sites VAFB SLC 4E and KSC LC 39 A have higher landing success rates than CCAFS SLC 40, though they have launched less flights.

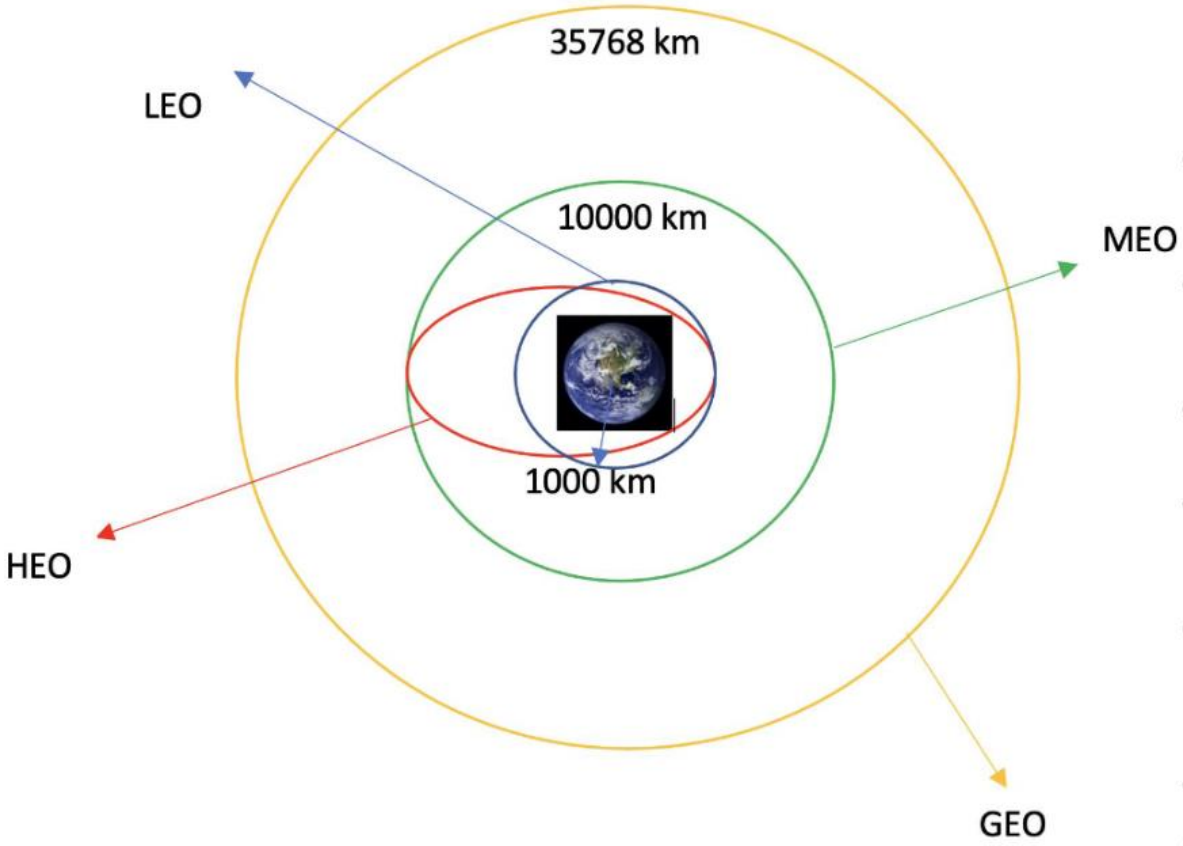
Payload vs. Launch Site



From this plot we can see:

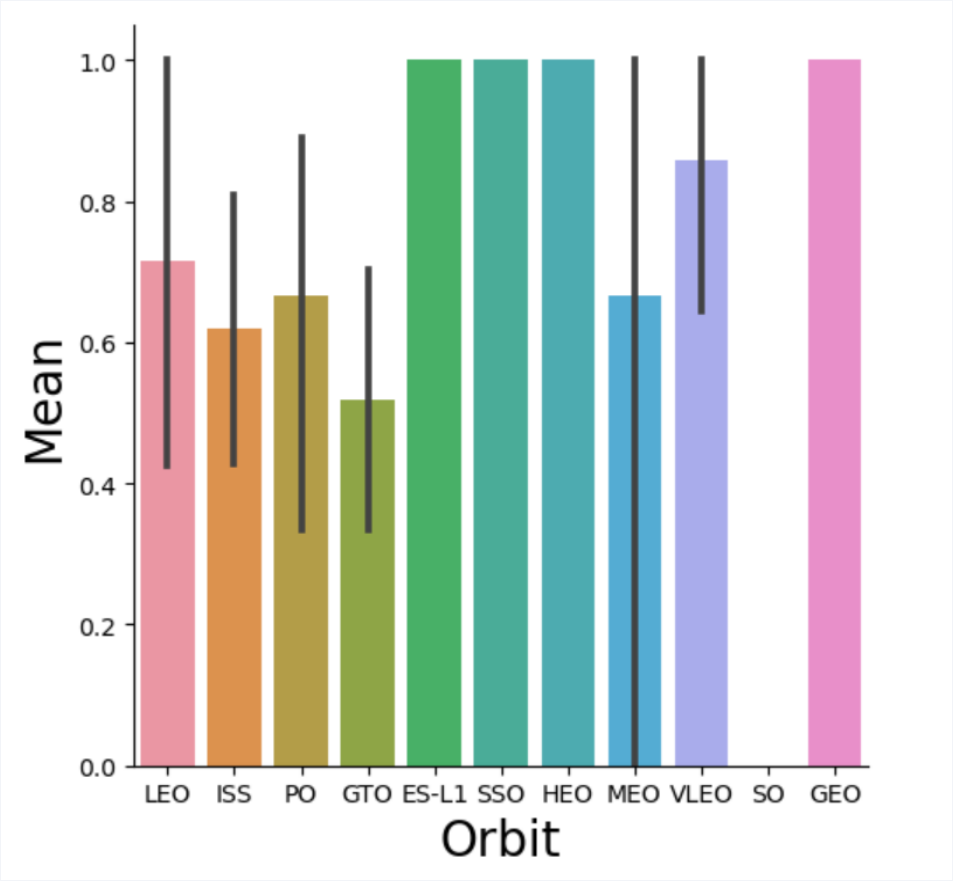
- For site CCAFS SLC 40, as payload mass increases, the likelihood of a successful landing increase.
- There is no obvious correlation between payload mass and landing success for the other two launch sites.
- VAFS SLC 4E doesn't seem to launch masses over 10,000 kg, unlike the other two sites.

Orbit Types and Radius Visualization



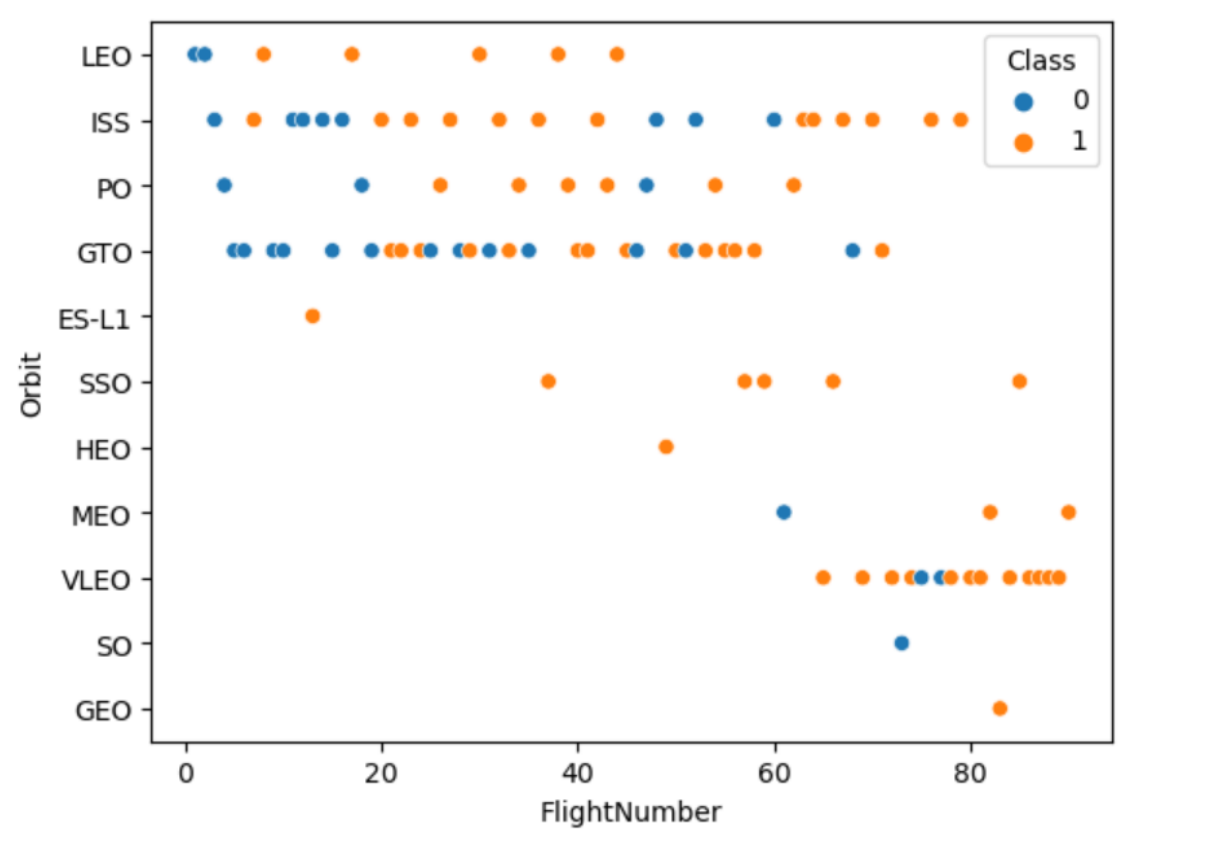
- **LEO**: Low Earth orbit (LEO)is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less (approximately one-third of the radius of Earth),[1] or with at least 11.25 periods per day (an orbital period of 128 minutes or less) and an eccentricity less than 0.25.[2] Most of the manmade objects in outer space are in LEO \[1].
- **VLEO**: Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 450 km. Operating in these orbits can provide a number of benefits to Earth observation spacecraft as the spacecraft operates closer to the observation\[2].
- **GTO** A geosynchronous orbit is a high Earth orbit that allows satellites to match Earth's rotation. Located at 22,236 miles (35,786 kilometers) above Earth's equator, this position is a valuable spot for monitoring weather, communications and surveillance. Because the satellite orbits at the same speed that the Earth is turning, the satellite seems to stay in place over a single longitude, though it may drift north to south," NASA wrote on its Earth Observatory website \[3] .
- **SSO (or SO)**: It is a Sun-synchronous orbit also called a heliosynchronous orbit is a nearly polar orbit around a planet, in which the satellite passes over any given point of the planet's surface at the same local mean solar time [4] . (800 – 1000 km)
- **ES-L1** :At the Lagrange points the gravitational forces of the two large bodies cancel out in such a way that a small object placed in orbit there is in equilibrium relative to the center of mass of the large bodies. L1 is one such point between the sun and the earth \[5] .
- **HEO** A highly elliptical orbit, is an elliptic orbit with high eccentricity, usually referring to one around Earth \[6].
- **ISS** A modular space station (habitable artificial satellite) in low Earth orbit. It is a multinational collaborative project between five participating space agencies: NASA (United States), Roscosmos (Russia), JAXA (Japan), ESA (Europe), and CSA (Canada) \[7]
- **MEO** Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 kilometers (22,236 mi). Also known as an intermediate circular orbit. These are "most commonly at 20,200 kilometers (12,600 mi), or 20,650 kilometers (12,830 mi), with an orbital period of 12 hours \[8]
- **HEO** Geocentric orbits above the altitude of geosynchronous orbit (35,786 km or 22,236 mi) \[9]
- **GEO** It is a circular geosynchronous orbit 35,786 kilometres (22,236 miles) above Earth's equator and following the direction of Earth's rotation \[10]
- **PO** It is one type of satellites in which a satellite passes above or nearly above both poles of the body being orbited (usually a planet such as the Earth

Success Rate vs. Orbit Type



- ES-L1, SSO, HEO, and GEO have a 100% landing success rate.
- SO has a 0% landing success rate.
- Flights with the highest landing success seem to have orbits between 800-10,000 km (see previous slide).
- VLEO has the second highest landing success rate, and is the lowest orbit, below 450 km (see previous slide).

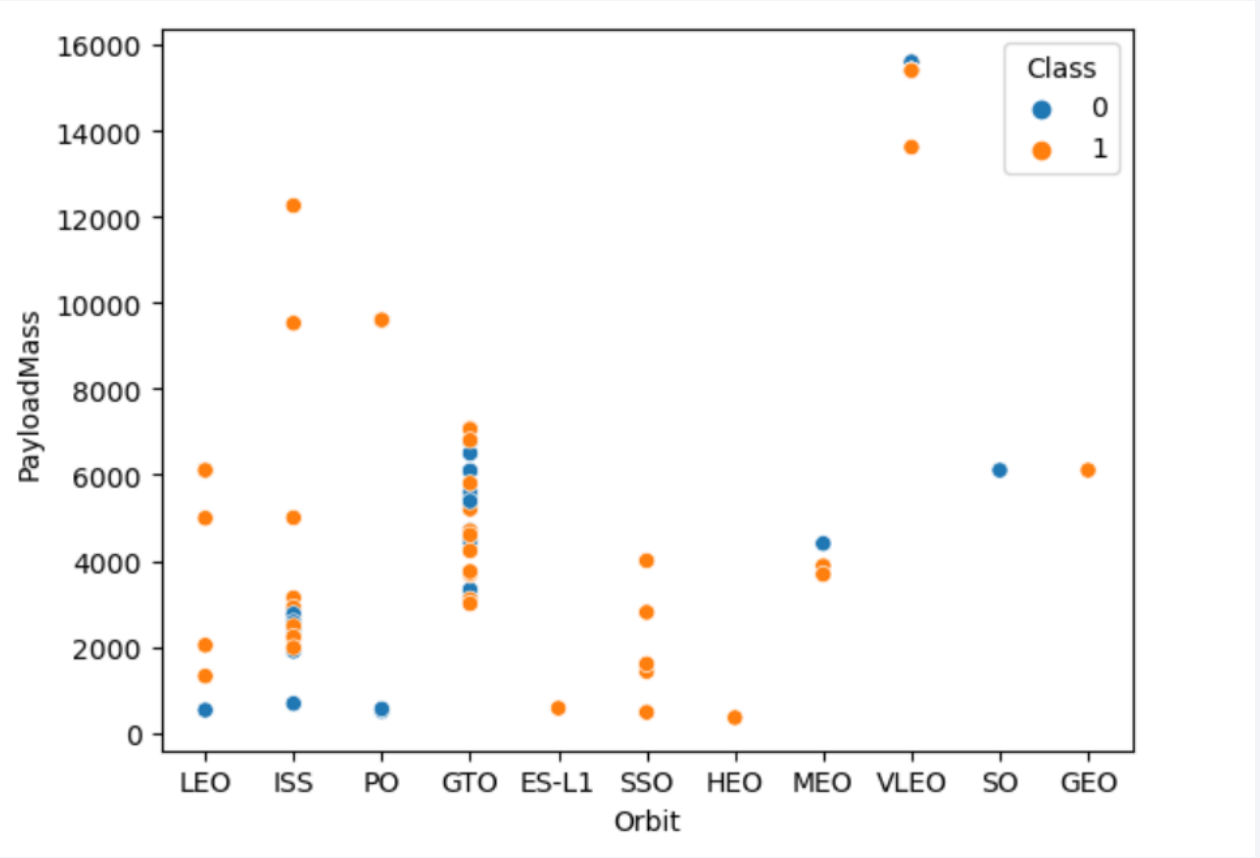
Flight Number vs. Orbit Type



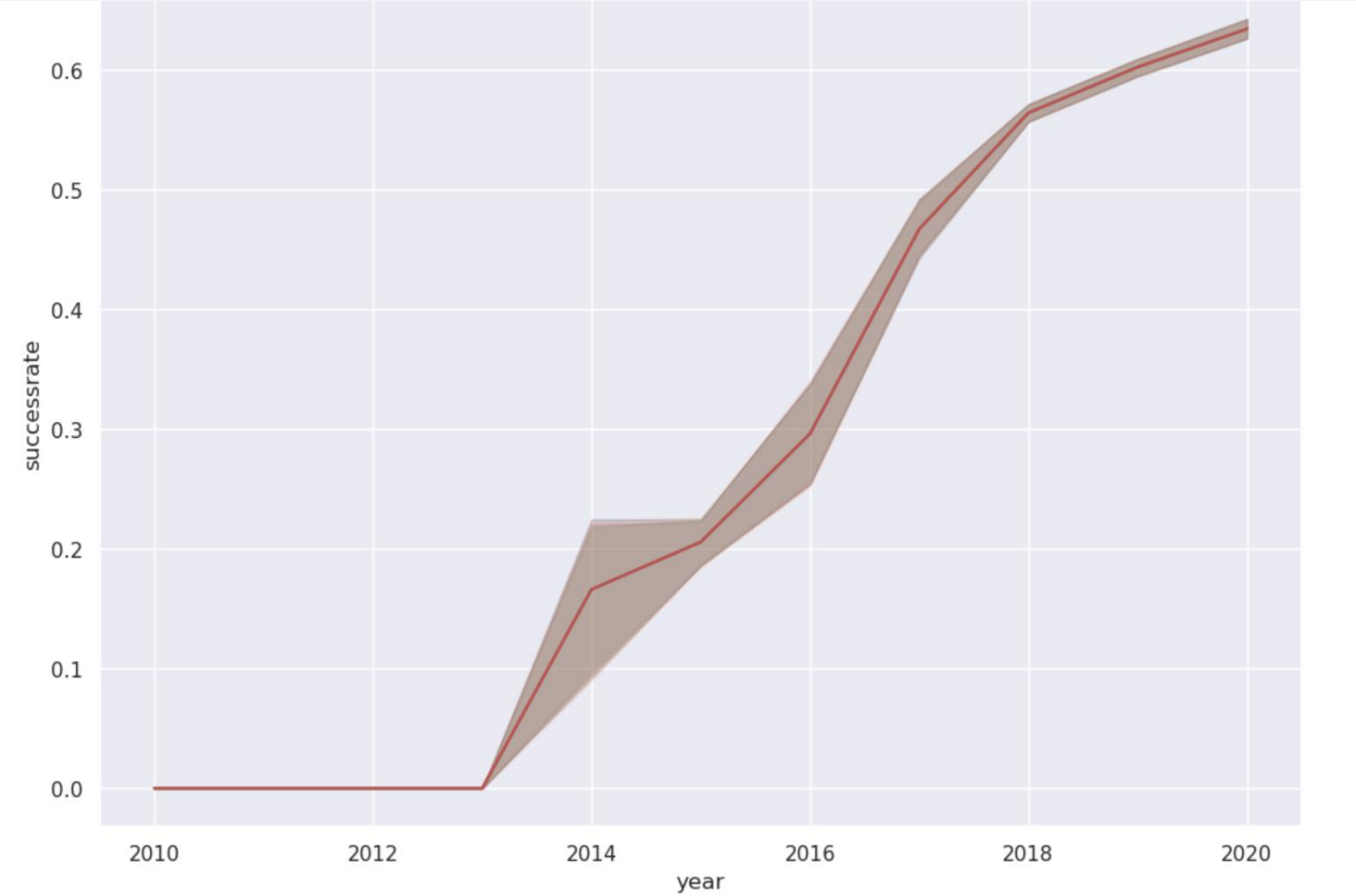
- The 100% landing success rate in ES-L1, GEO, and HEO can be explained by each orbit only having a single launch.
- SSO launch rate is more impressive with 5 launches in its orbit.
- For LEO and PO, success rate appears to be related to number of flights.
- There seems to be no correlation between flight number and orbit for GTO and ISS.

Payload vs. Orbit Type

- ISS and LEO appear to have higher success rates as payload mass increases.
- GTO does not seem to have a clear relationship between success and payload mass.
- All other orbits do not have enough data to conclude any relationship between payload mass and success.



Launch Success Yearly Trend



- All flights prior to 2013 have been unsuccessful.
- Since 2013, there has been a clear increase in landing success rates. However, after 2018 the rate of success has begun to increase at a slower rate than previous years.
- The final success rate reached in 2020 appears to be approximately 65%.
- Launch landing rate has increased approximately 65% in 7 years.

ALL LAUNCH SITE NAMES

Using SQL we can query the database as follows:

```
%sql SELECT UNIQUE(LAUNCH_SITE) FROM SPACEXDATASET
```

By specifying "Unique" the database will only return distinct entries from the column launch_names, from the table SpaceX.

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'. We do this using an SQL magic query:

```
%sql SELECT LAUNCH_SITE FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

We select the entries from the column launch_site that contain the string 'CCA'. LIMIT 5 restricts the results to the first 5 entries with this string.

Total Payload Mass

We use a SQL magic query to calculate the total mass. The SUM calculates the total. The AS clause renames the total value TOTAL_PAYLOAD_MASS. The WHERE clause ensures we only add values where the customer is NASA (CRS).

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXDATASET WHERE CUSTOMER = 'NASA (CRS)'
```

sum_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

Using SQL we query the dataset to calculate the average Payload Mass. The AVG clause calculates the average payload mass. The AS clause renames the value as AVG_PAYLOAD_MASS. The WHERE clause ensures that we are calculating the average mass of F9 v1.1 booster versions.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXDATASET WHERE BOOSTER_VERSION = 'F9 v1.1'
```

avg_payload_mass_kg
2928

First Successful Ground Landing Date

Using SQL we can determine the first successful landing date. We use the MIN function to retrieve the minimum date. The WHERE clause limits the MIN function to only successful landings.

```
%sql SELECT MIN(DATE) AS DATE FROM SPACEXDATASET WHERE LANDING__OUTCOME LIKE 'Success'
```

DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

In the following SQL query, the WHERE clause restricts the results to only successful mission outcomes, and payload masses between 4,000 and 6,000 kg.

```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success')
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)
')
```

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

Total Number of Successful and Failure Mission Outcomes

We use SQL to query the data to find the total number of successful and failed mission outcomes. The COUNT function is used to count the total number of outcomes, and the GROUP BY clause is used to group these results by the type of outcome.

```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS COUNT FROM SPACEXDATASET GROUP BY MISSION_OUTCOME
```

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

Boosters Carried Maximum Payload

Here we use SQL to query the dataset to show boosters that carried the maximum payload. We use the MAX function to select the max payload mass value. We return the booster version associated with this value using the WHERE clause.

```
%sql SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET)
```

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

2015 Launch Records

Using SQL magic, we query the dataset to show the failed landings that occurred in 2015. We use the WHERE clause to restrict the dates selected to only failed landing outcomes within the year 2015.

```
%sql Select landing__outcome, booster_version, launch_site, DATE from SPACEXDATASET
```

```
where landing__outcome = 'Failure (drone ship)' and Year(DATE) =  2015
```

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

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

We use SQL we can rank the landing outcomes between the desired dates. We use the COUNT function to obtain a count of landing outcomes. The WHERE clause limits the landing outcomes we retrieve to a specific window of time. Finally, we use GROUP BY clause to group the landings by type.

```
%sql select landing__outcome, count(landing__outcome) as "Total" from SPACEXDATASET
```

```
where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by "Total" desc
```

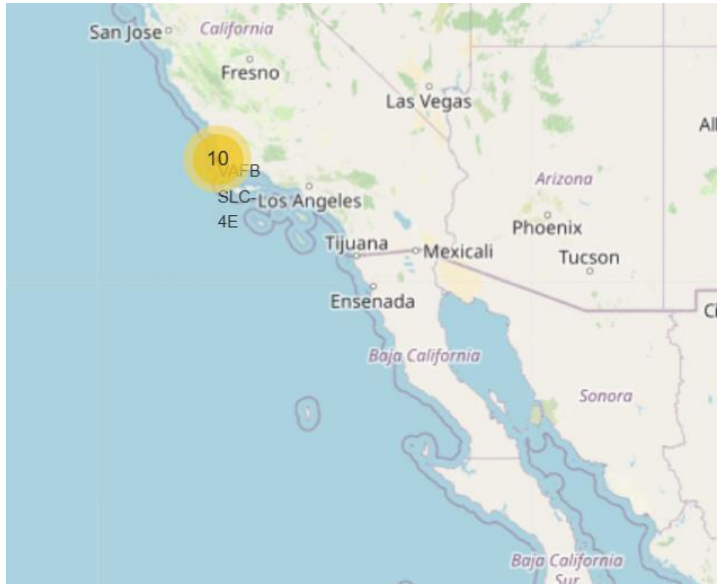
landing__outcome	Total
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

Section 3

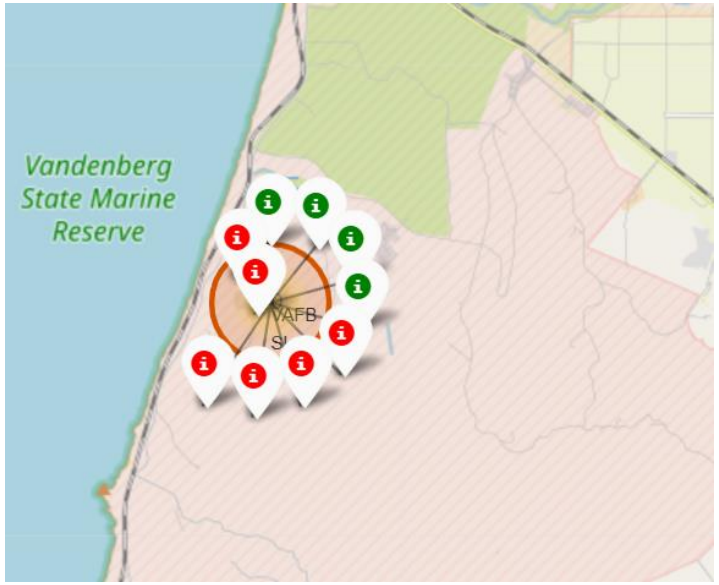
Launch Sites Proximities Analysis

GLOBAL MAP OF SPACEX LAUNCH SITES

- SpaceX has launch sites exclusively in the United States. There is one site located on the west coast in California, and three sites on the east coast in Florida.

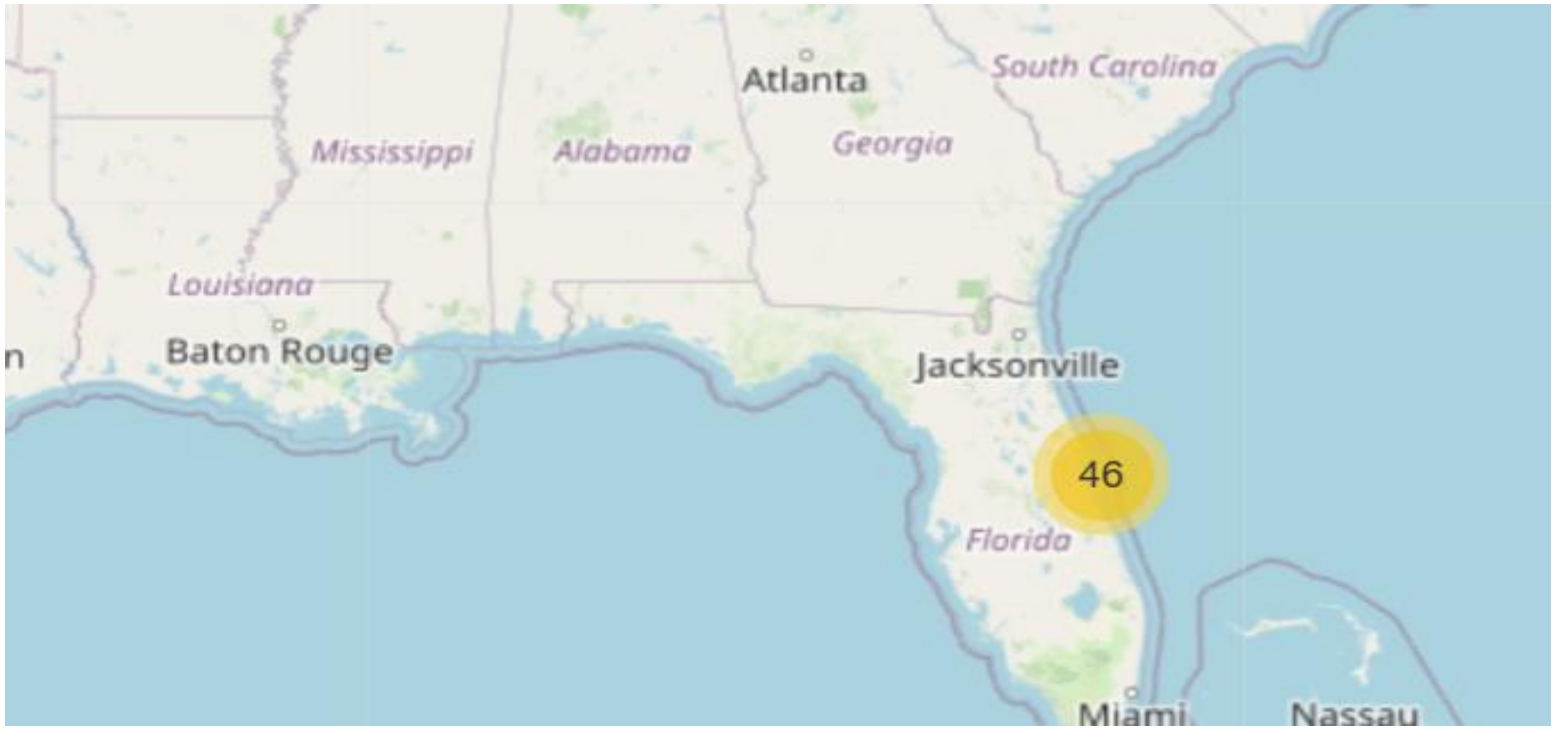


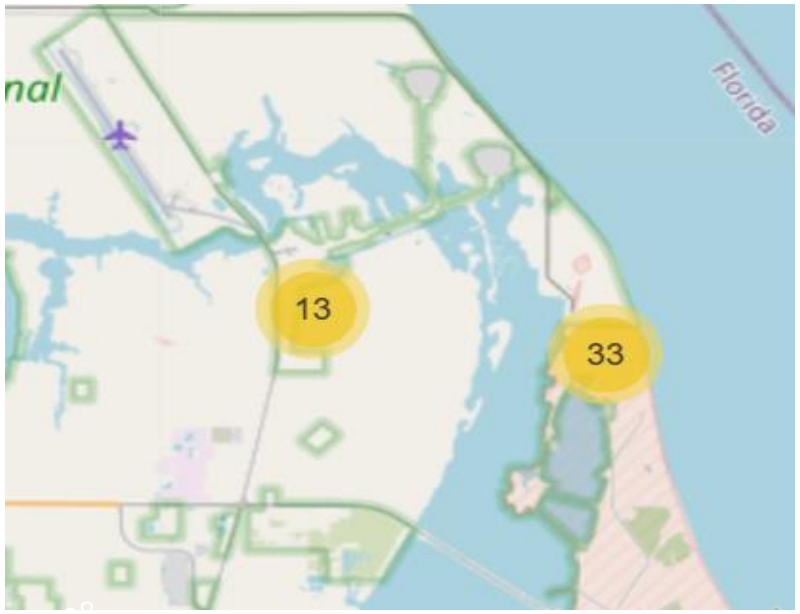
SPACEX CALIFORNIA LAUNCH SITE



SpaceX has one launch site on the coast of Santa Maria, California. Our cluster marker indicates that there have been 10 launches at this site.

When we zoom in, we can see that 4 launches landed successfully, and 6 launches had unsuccessful landings.





SPACEX LAUNCH SITES IN FLORIDA

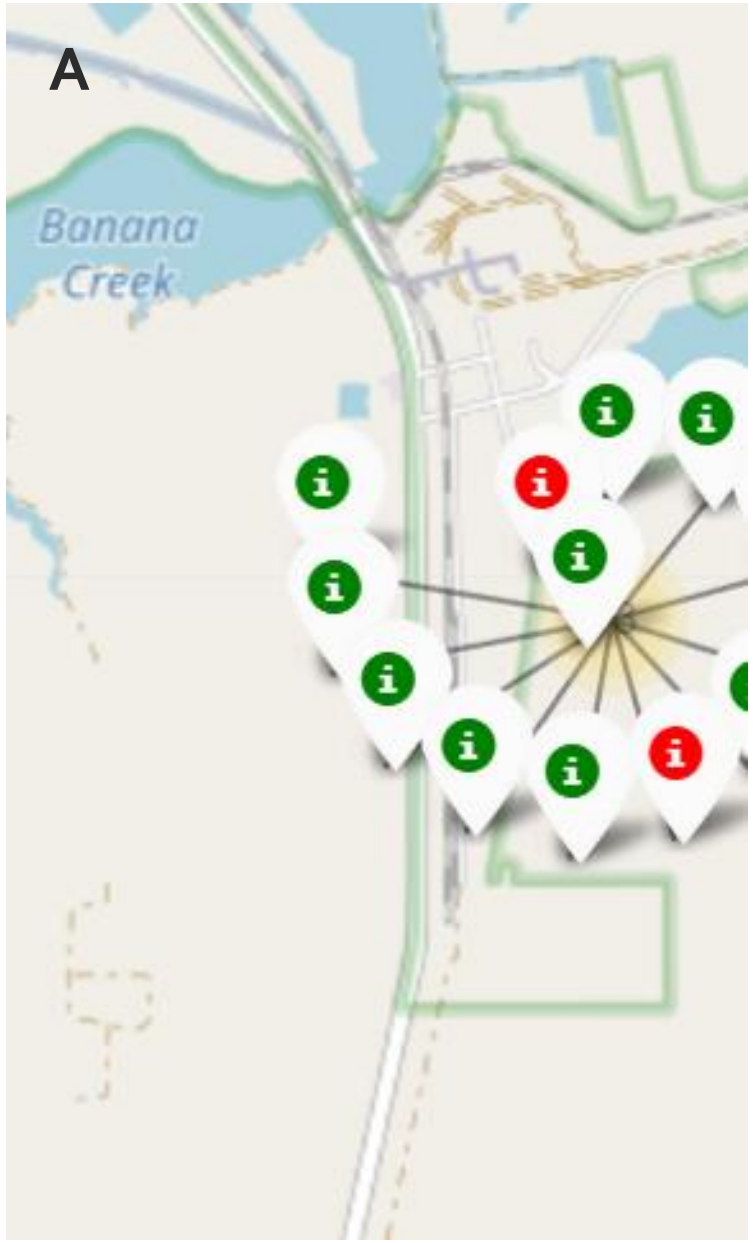
We can see that SpaceX has three separate launch sites in Florida:

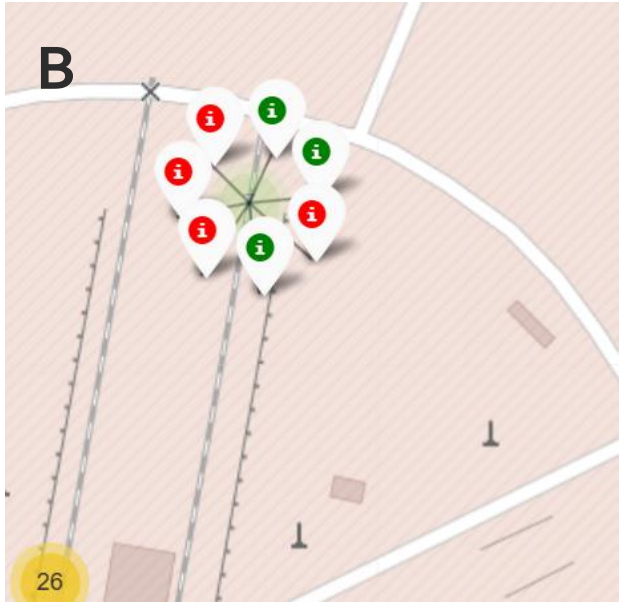
- Merritt Island
  - KSC LC-39A
- Cape Canaveral
  - CCAFS LC-40
  - CCAFS SLC-40

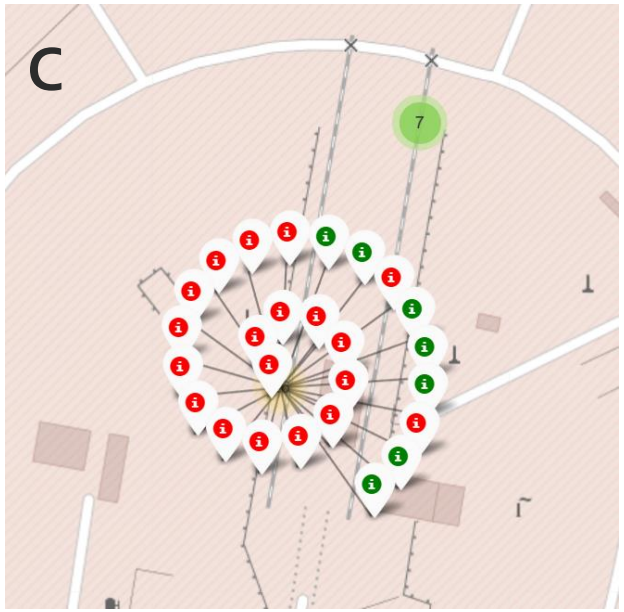
Cluster markers show that there have been 46 total launches in Florida: 13 at Merritt Island, and 33 at Cape Canaveral. CCAFS LC-40 at Cape Canaveral has seen 26 launches, while CCAFS SLC-40 has seen 7.

FLORIDA LAUNCH SITES (CONT.)

A. Of the 13 launches at Merritt Island, 10 have landed successfully, while 3 have not landed successfully.
B. CCAFS SLC-40 at Cape Canaveral has had 3 successful landings, and 4 unsuccessful landings.
C. CCAFS LC-40 at Cape Canaveral has landed 7 rockets successfully, while 19 have had failed landings.





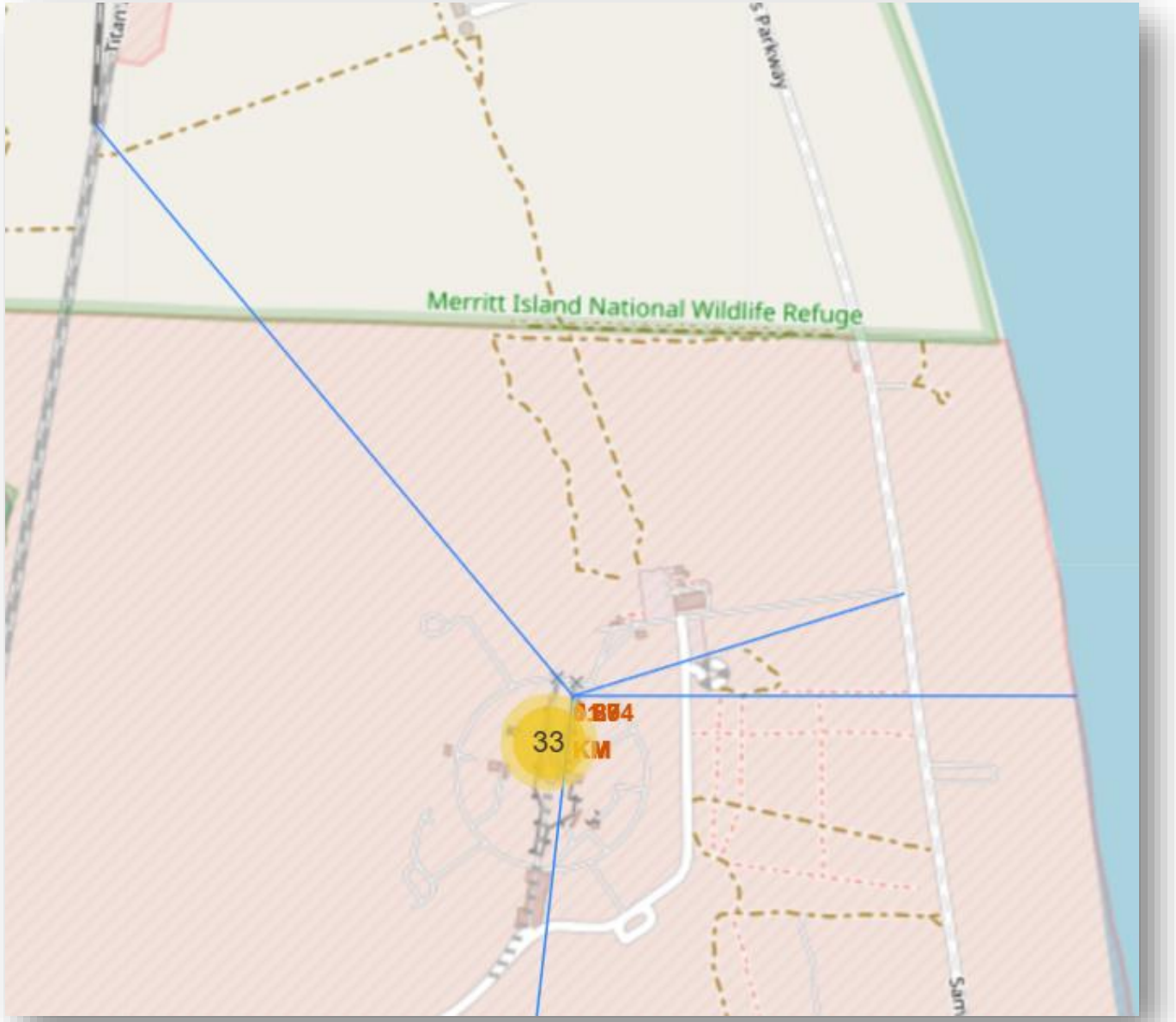


PROXIMITY OF INFRASTRUCTURE TO CAPE CANAVERAL SITE

- We can see that both the coastline and the nearest highway are within 1 kilometer of the launch site.

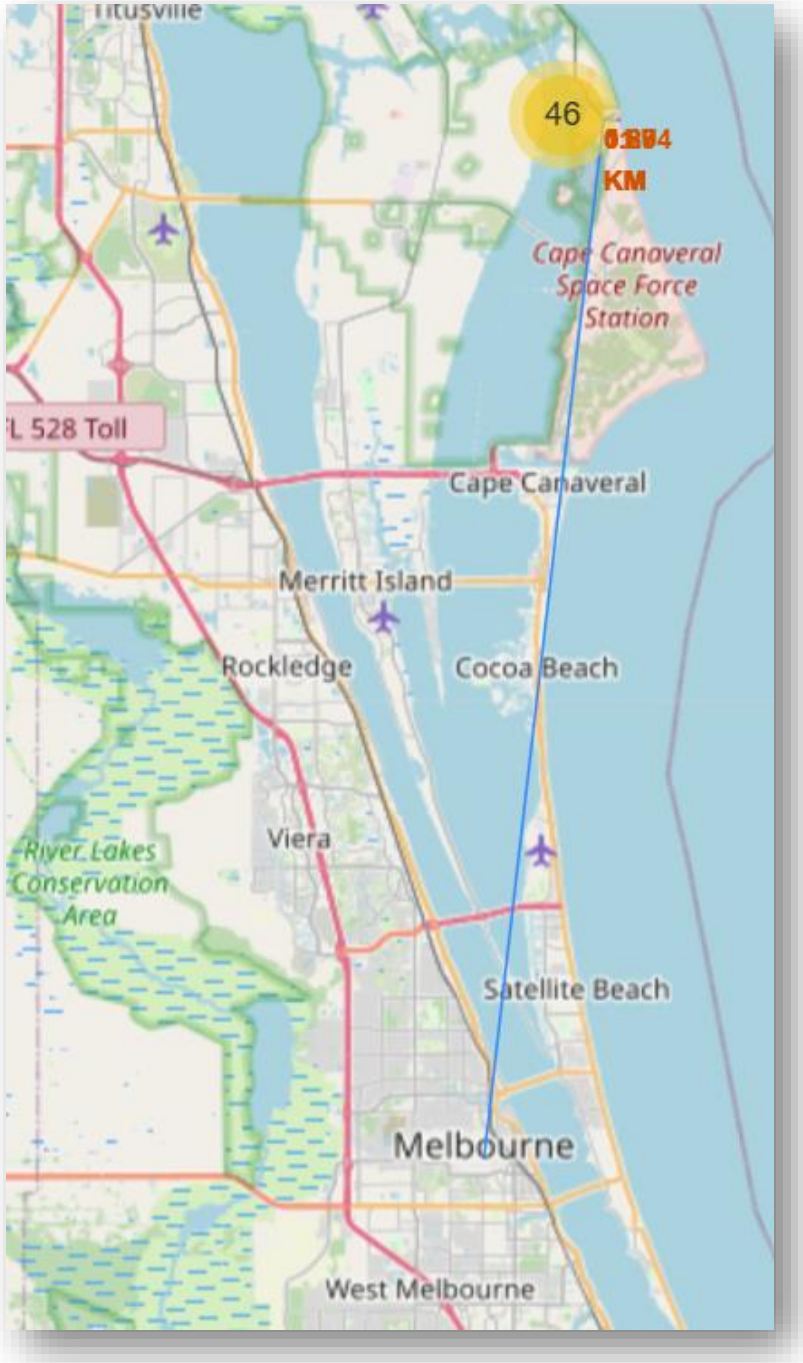
- We can also see that the nearest railway is within 1.5 kilometers of the launch site.

- From this we can deduce that close proximity to infrastructure is important for a launch site, likely for easy access to mechanical parts.



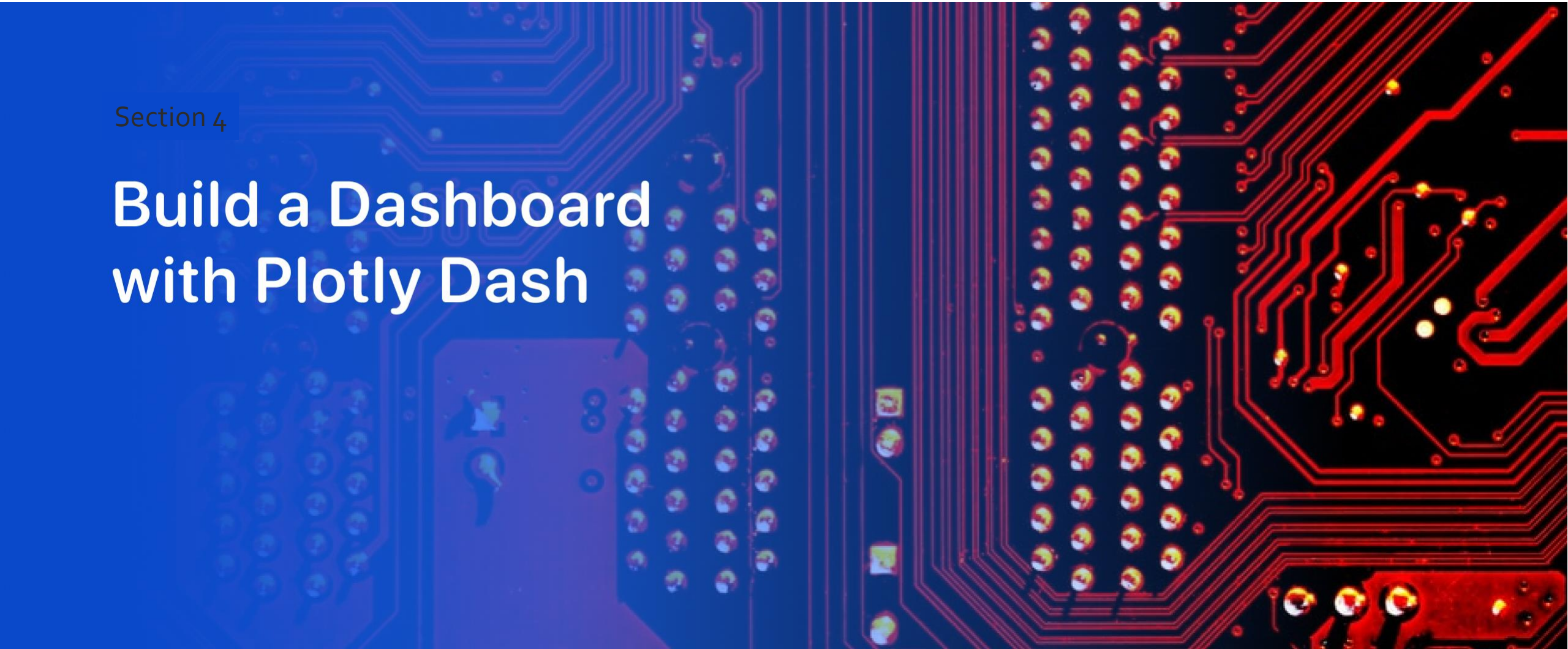
PROXIMITY TO NEAREST CITY FROM CAPE CANAVERAL LAUNCH SITE

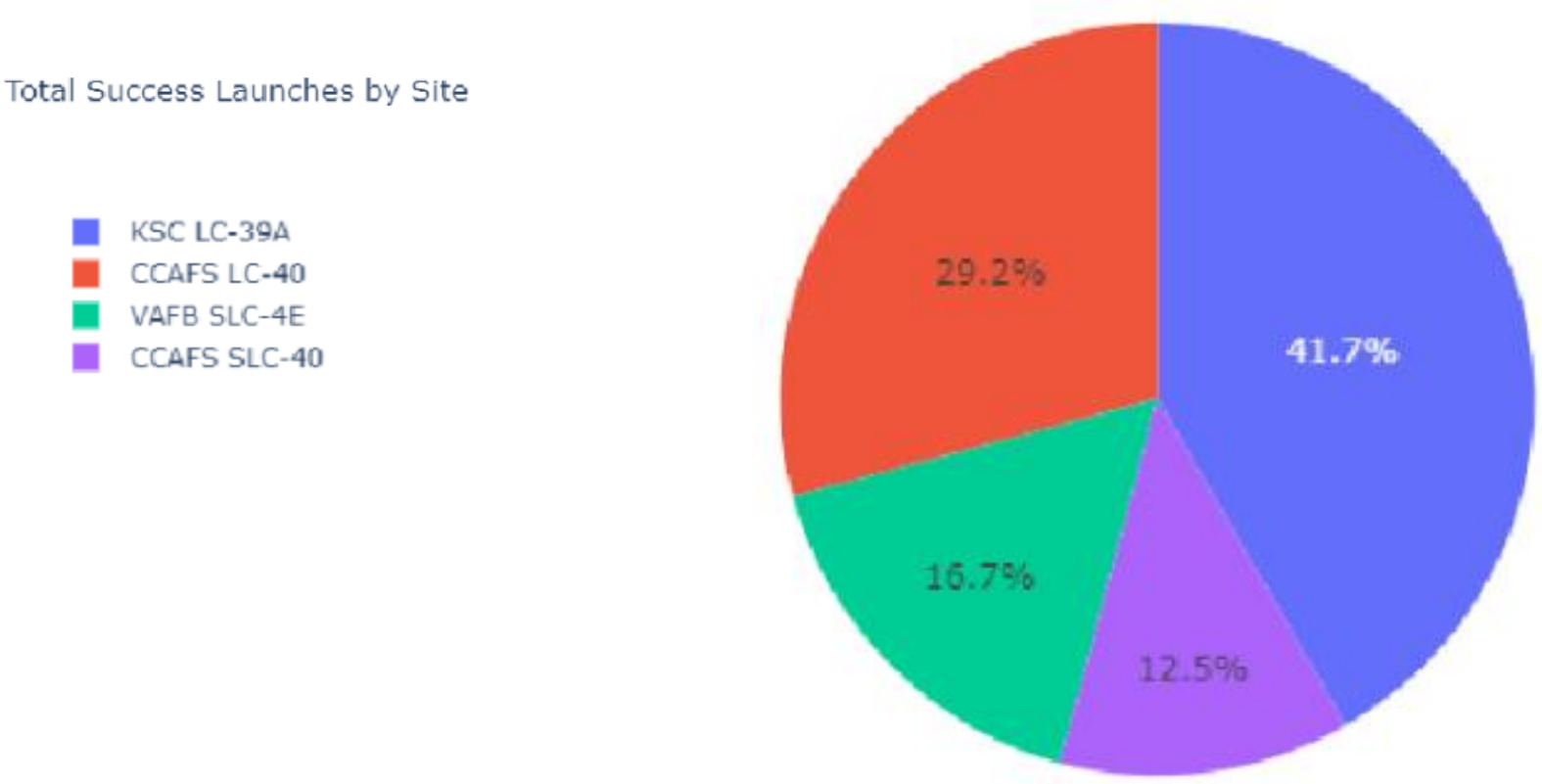
- Here we can see that the nearest city is over 50 km away from the launch site.
- From this we can deduce that proximity to a city is not a priority for the launch site, likely because they have infrastructure set up nearby, and to keep away from the general population.



Section 4

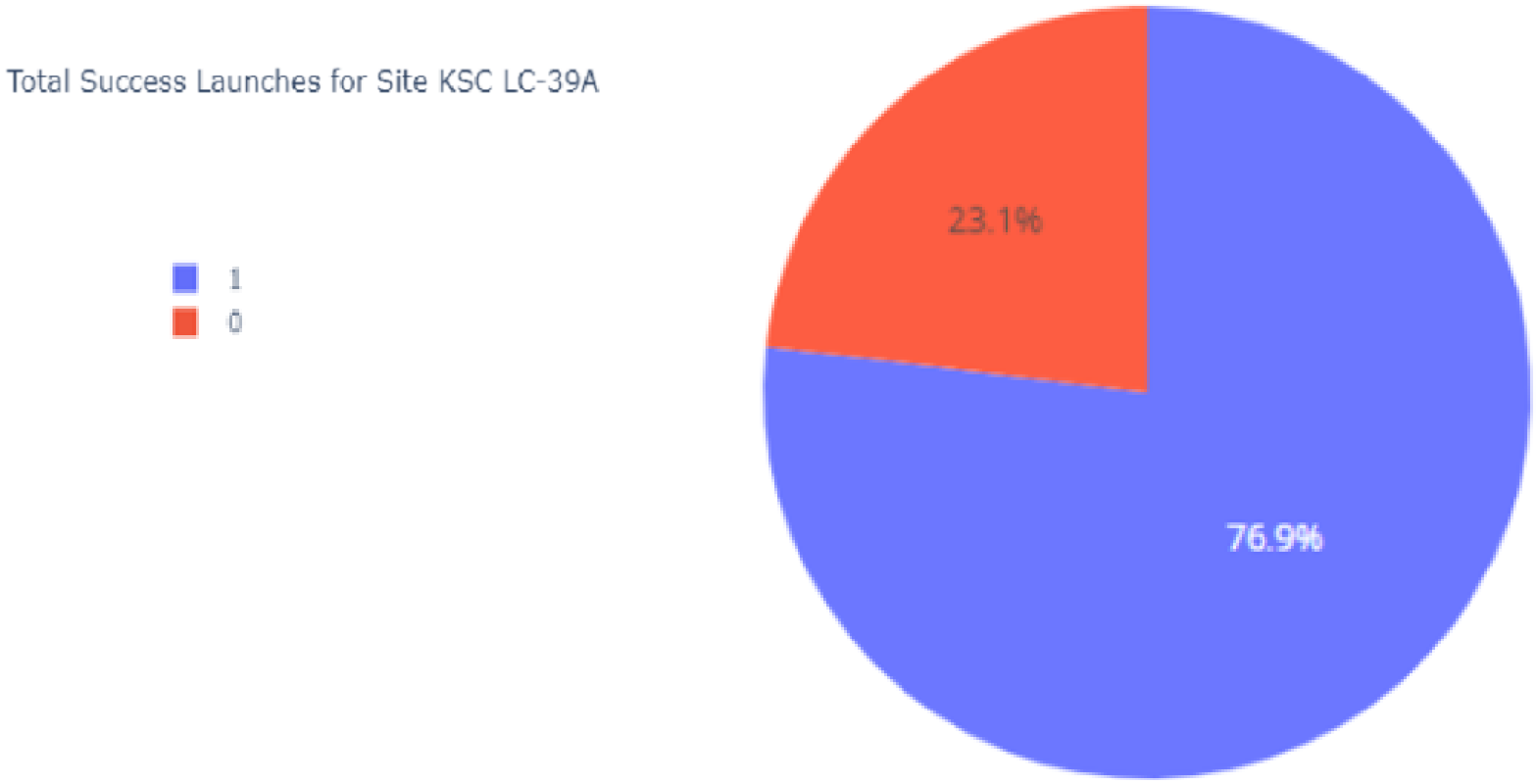
Build a Dashboard with Plotly Dash





From the Plotly pie chart, we can see that site KSC LC-39A contributes the largest number of successful landings, 41.7%

PROBABILITY OF LANDING SUCCESS BY LAUNCH SITE



Here we can see the success rate of launches from site KSC LC-39A. The site has a 76.9% launch landing success rate.

HIGHEST LAUNCH SUCCESS RATIO

Here we can see relationships between payload mass, booster version, and landing success.

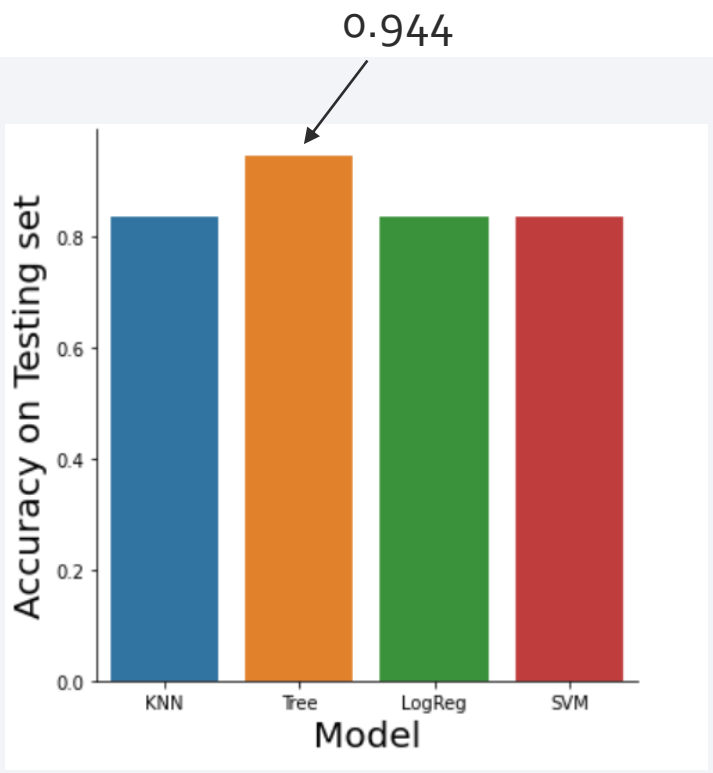
PAYLOAD VS. LAUNCH OUTCOME

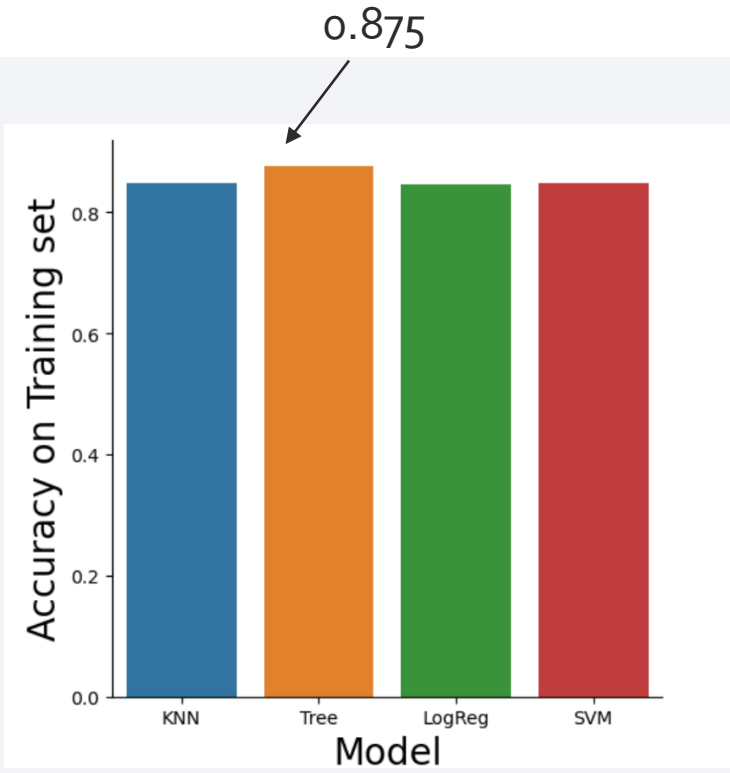
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Here we can see that the decision tree model performed best on both the testing set and the training set. Clearly the decision tree is the best predictive model.



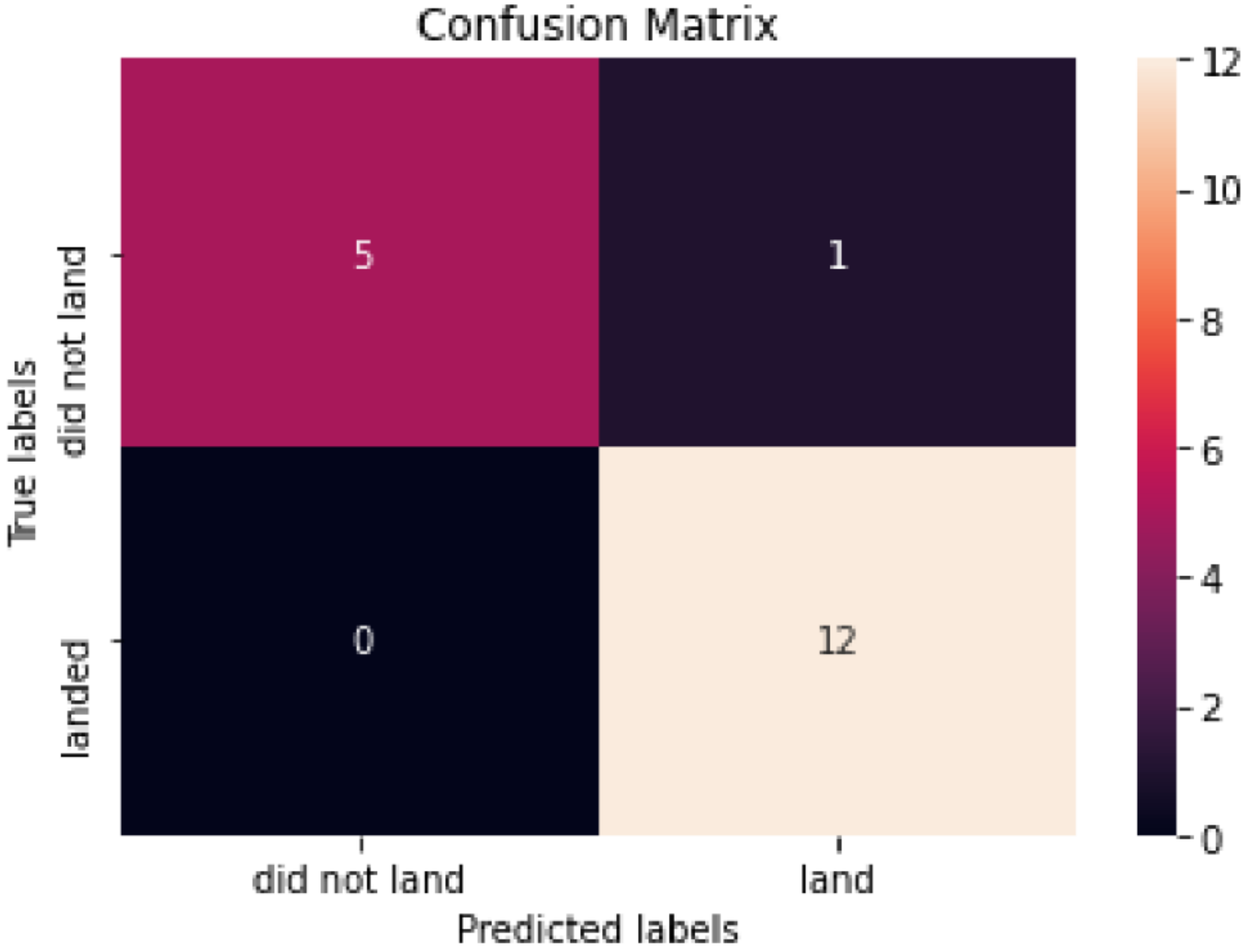



```
Best Model is Tree with a score of 0.875
Best Params is : {'criterion': 'entropy', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'spli
tter': 'random'}
```

Confusion Matrix

The confusion matrix for the decision tree shows how accurately it predicts landing success.

- Our model is able to predict 5 out of the 6 testing points that failed to land. Thus sensitivity is equal to 83.3%
- The model is able to predict 12 out of 12 testing points that had successful landings. Thus, specificity is 100%
- Accuracy = (5+12)/(5+1+12+0) = 0.944



CONCLUSIONS

- Most early SpaceX launches resulted in unsuccessful landings
- Since 2013, SpaceX has improves landing success rate every year, resulting in a 63% success rate in 2020
- Orbit type has an impact on landing success. Orbits ES-L1, SSO, HEO, and GEO have a 100% landing success rate.
- Launch site KCS LC-39A has the highest probability of success per launch at 76.9%
- We built decision tree model to predict launch landing success, with 87.5% accuracy on in sample data, and 94.4% accuracy on out of sample data

Appendix

- My GitHub Repository: https://github.com/rudigallegos/Capstone-Project.git
- The data for the SQL Exploratory Data Analysis portion of this project is stored in an IBM DB2 cloud storage environment.

Thank you!