IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Taylor Mason
December 1st, 2022

Table of Contents

Executive Summary

- Summary of methodologies
  - Data Collection with API's
  - Data Collection with Web Scraping
  - Data Cleaning/Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Map Analytics with Folium
  - Machine Learning Prediction with SciKit-Learn
- Summary of all results
  - Exploratory Data Analysis Results
  - Screenshots from Interactive Dashboards & Maps
  - Predictive Analytics Results

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - What factors impact the rocket landing success rates?
  - Are there interactions between various features that impact the rate of a successful landing?
  - What are the optimal conditions for a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

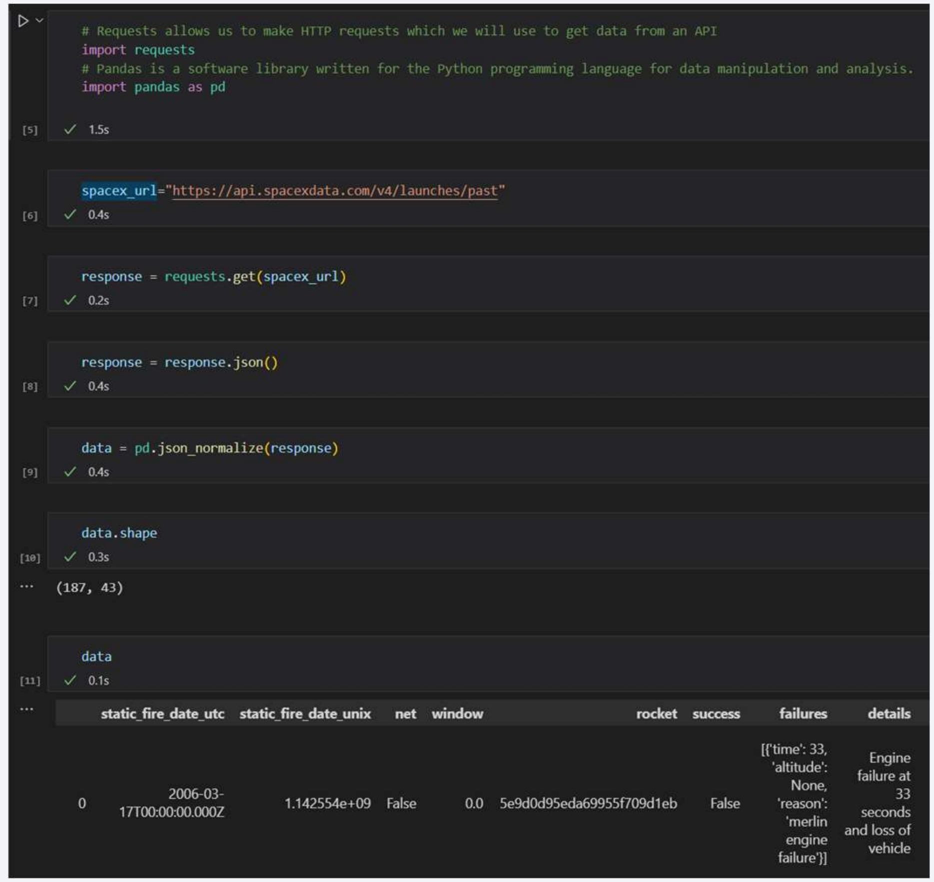
- Data collection methodology:
  - Data was collected with the SpaceX API as well as webscraping from Wikipedia
- Perform data wrangling
  - Data was processed and cleaned with Pandas to prepare data for analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data collection methods/process
  - First, SpaceX Launch data was pulled from the SpaceX public API using the Requests library and the Get function
  - Next, the response from the first step was decoded using the .json() function
  - After that, the .json_normalize() function was used to turn the json file into a Pandas dataframe
  - Once the data was in a dataframe it was reviewed and cleaned for further analysis
  - In addition to using the Requests library, the Beautiful Soup Library was also used to extract data from SpaceX Launch Data from Wikipedia

Data Collection – SpaceX API

- The general steps included using the Requests library and the Get function to extract the SpaceX Launch data.
- Once we have the data we use the .json() and .json_normalize functions to transform the data into a dataframe for further analysis.
- [GitHub URL]()



```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
```
[5] ✓ 1.5s

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```
[6] ✓ 0.4s

```
response = requests.get(spacex_url)
```
[7] ✓ 0.2s

```
response = response.json()
```
[8] ✓ 0.4s

```
data = pd.json_normalize(response)
```
[9] ✓ 0.4s

```
data.shape
```
[10] ✓ 0.3s

```
(187, 43)
```

```
data
```
[11] ✓ 0.1s

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle

Data Collection - Scraping

- BeauftifulSoup is a common python library that is used for webscraping.
- In this project we utilized BeatutifulSoup to extract HTML tables from Wikipedia to extract additional SpaceX Launch data.
- GitHub URL

```python
[1]
import requests
from bs4 import BeautifulSoup
import pandas as pd

[3]
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

[4]
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to? a object
type(response)

requests.models.Response

[5]
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')

[6]
# Use soup.title attribute
soup.title

<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>

[7]
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

html_tables = soup.find_all('table')
html_tables

Outputs are collapsed ...

[8]
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)

Outputs are collapsed ...
```

Data Wrangling

- Data Wrangling Included:
  - Better understanding the data with functions like:
    - .dtypes()
    - .describe()
    - .shape()
    - .is_null()
    - .value_counts()
  - We also added a column called “class” which simplified the landing_outcome column to a “good” or “bad” launch outcome
- GitHub URL

EDA with Data Visualization

- Charts Included
  - flight number and launch Site
  - payload and launch site
  - success rate of each orbit type
  - flight number and orbit type
  - the launch success yearly trend.
- GitHub URL

EDA with SQL

- In an effort to understand the SpaceX Launch data further, we loaded the dataset into an IBM Db2 database so we could apply EDA with SQL queries
- SQL Queries Included:
  - The names of the unique launch sites
  - The total payload mass carried by boosters launched at various sites
  - The average payload mass carried by various booster types
  - The total number of successful and failed launches
  - The failed landing outcomes on drone ships, their booster versions, and the site names
- GitHub URL

Build an Interactive Map with Folium

- Using Folium an interactive map displaying the SpaceX Launch site locations, colored green if the mission was successful, and red if it was not.
- Maps included map objects such as:
  - Markers – markers are used to display information about a location
  - Circles – these are identifiers for a location and can be highly customized
  - Lines – these are a powerful tool to show distances from certain locations or objects like rail roads
- Adding these map objects gives the user more information to better understand the various launch sites.
- GitHub URL

Build a Dashboard with Plotly Dash

- Using Plotly Dash an interactive dashboard was created to analyze mission success/failure
- Charts included
  - Pie Chart
  - Scatter Chart
- Key features analyzed
  - Launch Site
  - Payload Mass
- GitHub URL

Predictive Analysis (Classification)

- SciKit-Learn was used to perform predictive analysis on launch outcomes using various classification methods.
  - First the model is split into X and Y, and then the data is further split into training and testing data
  - Once the data was ready various machine learning models were built
  - Results were improved by testing various hyperparameters using GridsearchCV
  - Once the optimal hyperparameters were in place we were able to determine which model had the highest accuracy
- GitHub URL

Results

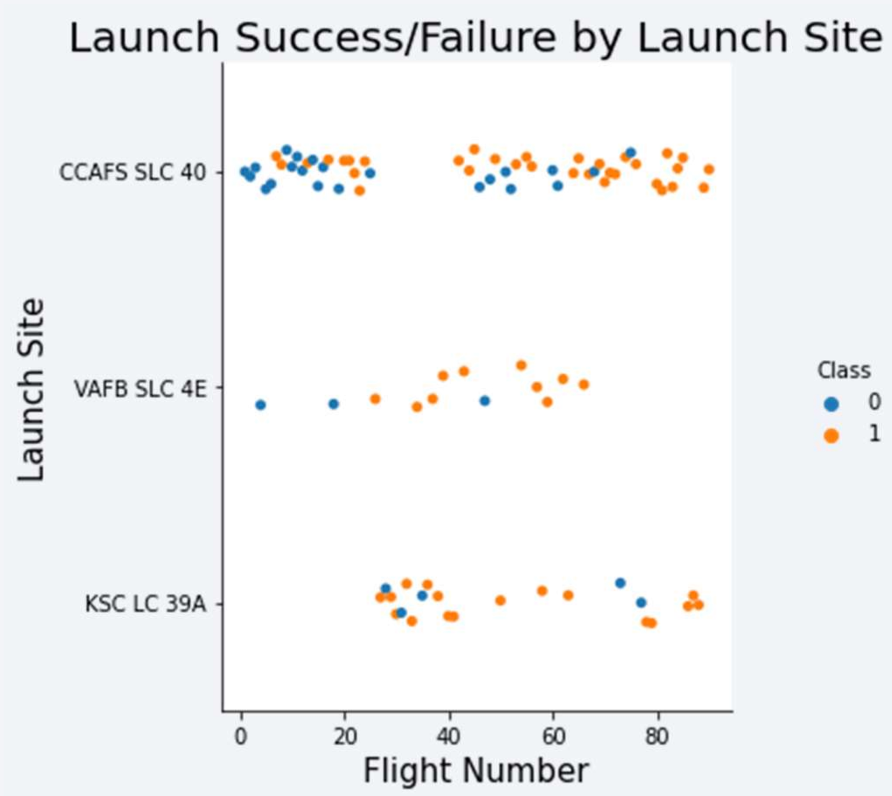
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

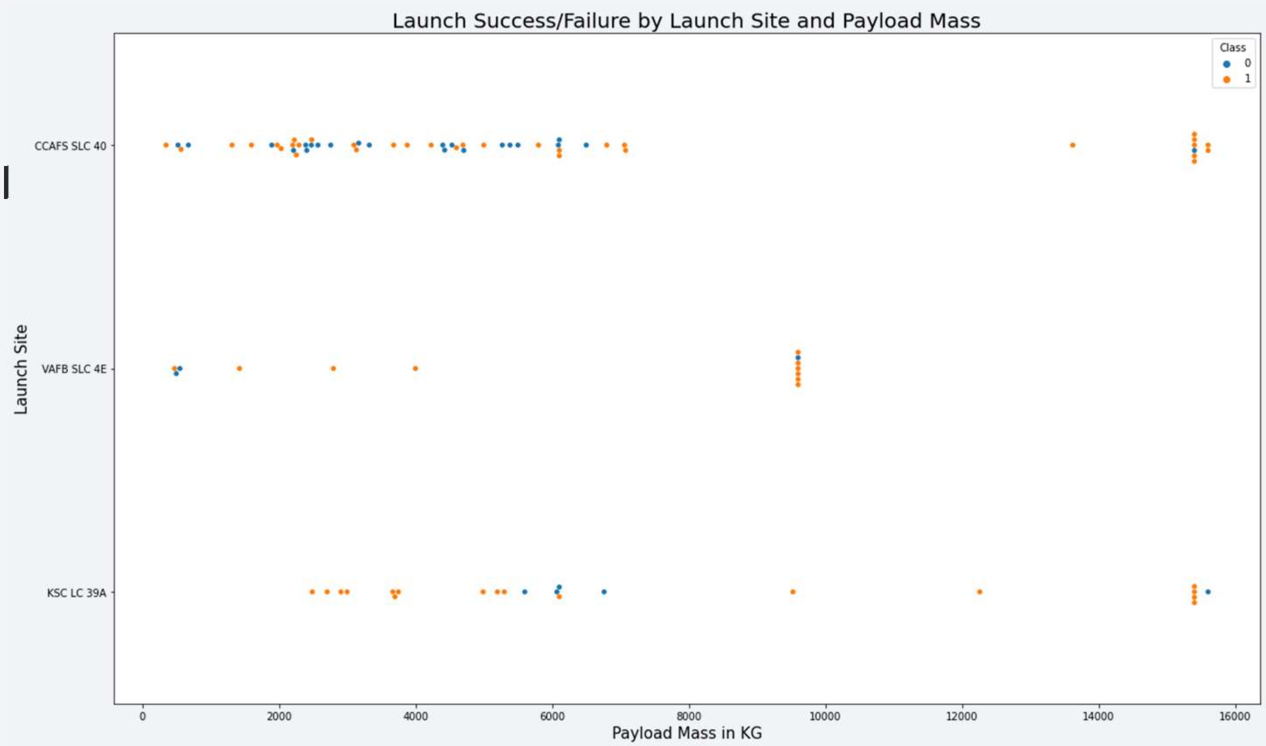
Flight Number vs. Launch Site

- From the plot, we notice that the success rate (1) increases as the flight number increases
- We also note CCAFS SLC 40 appears to have significantly more launches than the other sites



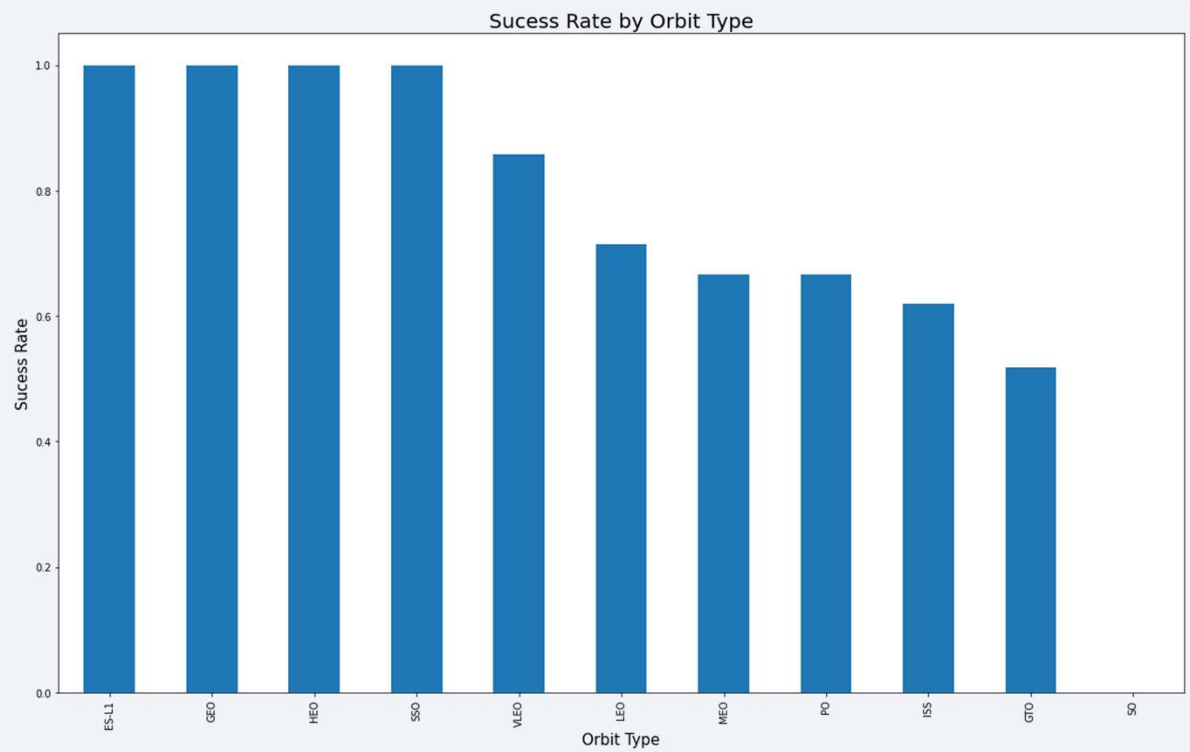
Payload vs. Launch Site

- Payloads larger than 8,000 KG tend to be more successful
- Many of the failed launches result in the occur between payload mass of 4,000kg and 6,000kg at CCAFS SLC 40.



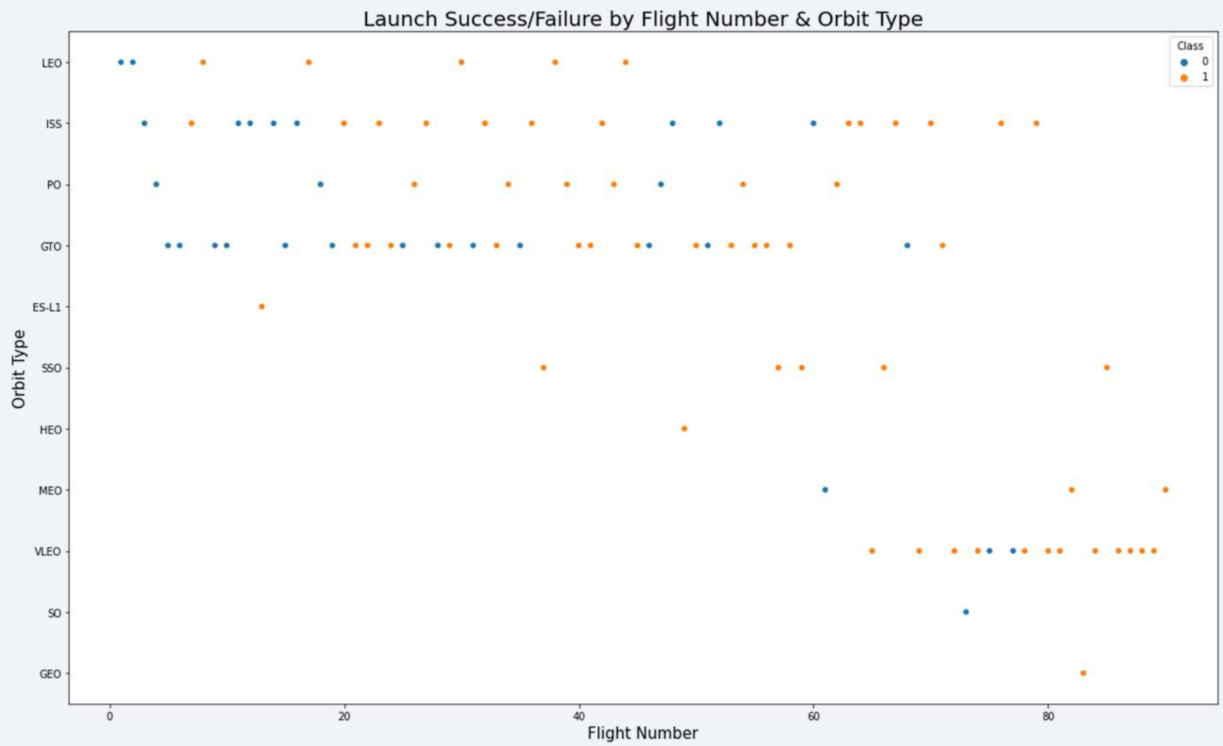
Success Rate vs. Orbit Type

- Half of the orbit types have 80-100% success rate
  - ES-L1, GEO, HEO, SSO, VLEO
- SO, GTO and ISS are the least successful launch types



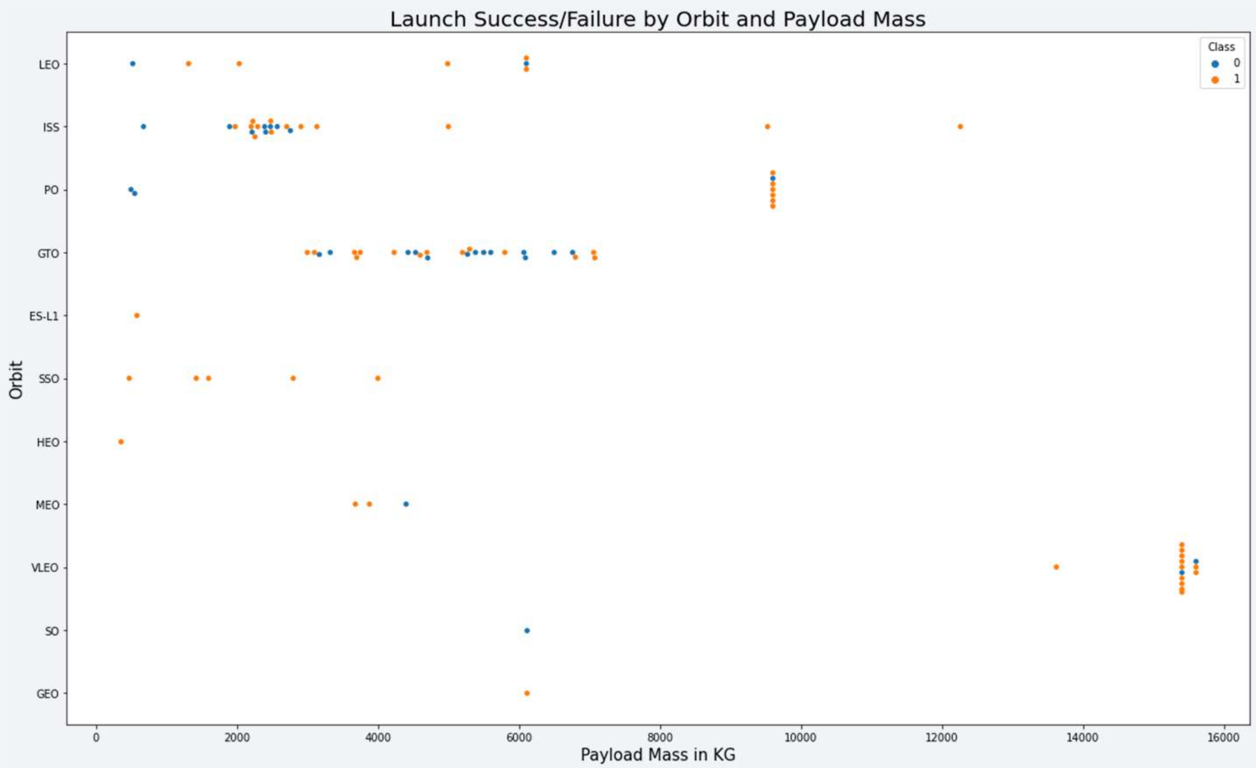
Flight Number vs. Orbit Type

- Following up on the last slide, SO had a 0% success rate; however, there has only been 1 attempt so more information is likely needed
- GTO and ISS had some of the lower success rates, but here we notice many of them occurred towards the beginning indicating improvement over time



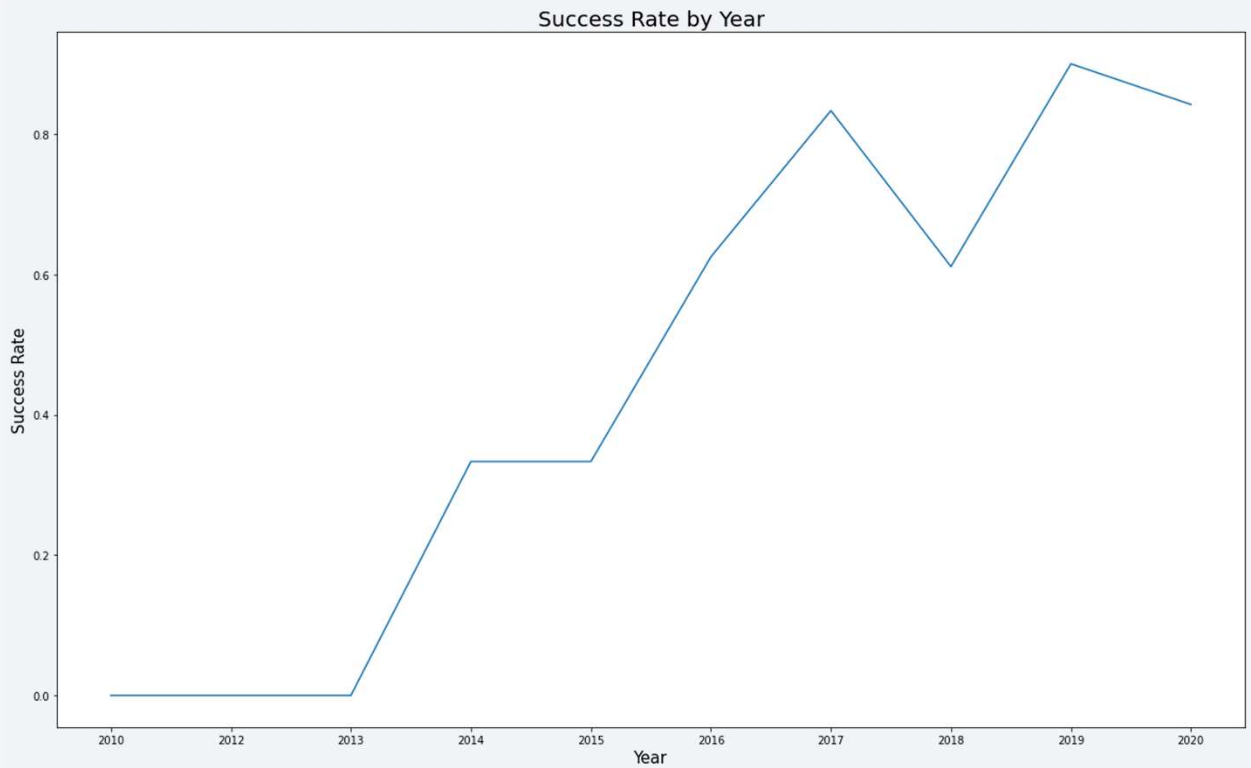
Payload vs. Orbit Type

- Here we continue to notice that the launches carrying heavier payload mass are more successful
- Despite the lower payload mass ES-L1, SSO, and HEO have all been successful with lower a lower payload mass



Launch Success Yearly Trend

- From 2010 to 2012 there were no successful launches
- In 2013, successful launches began to trend up significantly
- From 2016 to 2020 success rates hovered around 80%, with 2018 being the exception at 60%



All Launch Site Names

- SQL was also utilized for exploratory data analysis
- For the SQL queries we use “SPACEX” as the source table
- Here you can see the various launch sites

```
%sql SELECT DISTINCT launch_site FROM SPACEX
```

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

Launch Site Names Begin with 'CCA'

- Below, we utilized SQL to see Launches at Launch Site location beginning with "CCA"
- This then allowed us to do further analysis on a specific site

```
%sql SELECT * FROM SPACEX WHERE launch_site LIKE 'CCA%' limit 5
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- SQL can also be used to total values
- In this example we were able to find out the total payload mass sent by NASA on SpaceX Launches

```
%sql SELECT SUM(payload_mass__kg_) AS Total_NASA_CRS_Payload_Mass_in_KG FROM SPACEX WHERE customer LIKE('NASA (CRS)')
```

total_nasa_crs_payload_mass_in_kg
45596

Average Payload Mass by F9 v1.1

- In this example we totaled the payload mass for a specific booster type, the F9 V1.1.

```
%%sql SELECT AVG(payload_mass__kg_) AS Avg_Booster_F9v1_1_Payload_Mass_in_KG
    FROM SPACEX
    where booster_version like('F9 v1.1')
```

avg_booster_f9v1_1_payload_mass_in_kg
2928

First Successful Ground Landing Date

- Dates are also easy to work with in SQL
- Here we were able to determine the first ground landing date was December 22nd, 2015.

```
%%sql SELECT MIN(date) AS first_sucessful_landing_outcome
    FROM SPACEX
    WHERE landing__outcome LIKE('%ground pad%')
```

first_sucessful_landing_outcome
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL has a variety of methods to filter tables to extract the exact data needed.
- In this example we were able to find the boosters that had a payload mass greater than 4,000 KG but less than 6,000 KG

```
%%sql SELECT DISTINCT(booster_version) AS boosters_with_sucess_on_drone_ship_greaterthan_4000kg_lessthan_6000kg
    FROM SPACEX
    where 4000<payload_mass__kg_<6000
```

boosters_with_sucess_on_drone_ship_greaterthan_4000kg_lessthan_6000kg
F9 B4 B1039.2
F9 B4 B1039.1
F9 B4 B1042.1
F9 B4 B1045.1
F9 B4 B1045.2
F9 B5 B1046.1
F9 B5 B1046.3
F9 B5 B1048.2
F9 B5 B1056.2
F9 B5 B1058.4
F9 B5 B1059.2
F9 B5 B1059.4
F9 B5B1050
F9 B5B1056.1
F9 B5B1059.1
F9 B5B1063.1
F9 FT B1029.2
F9 FT B1035.2
F9 FT B1038.2
F9 FT B1019
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1024
F9 FT B1025.1
F9 FT B1031.1
F9 FT B1035.1
F9 FT B1038.1
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1
F9 v1.1 B1003
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1015
F9 v1.1 B1017
F9 v1.1 B1018

Total Number of Successful and Failure Mission Outcomes

- SQL also allows users to group values for further analysis
- In this example we grouped all of the mission outcomes and can see only one mission failed in flight.

```
%%sql SELECT mission_outcome, COUNT(mission_outcome) AS mission_outcome_totals
        FROM SPACEX
        GROUP BY mission_outcome
```

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

Boosters Carried Maximum Payload

- In this example, we were able to determine the maximum payload mass that has been carried by a booster, and then use that to determine which boosters carried the maximum payload mass.

```
%%sql SELECT booster_version, payload_mass__kg_
    FROM SPACEX
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_)FROM SpaceX)
    ORDER BY booster_version
```

booster_version	payload_mass__kg_
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

2015 Launch Records

- In this example we were able to find the failed drone ship missions for a specific year

```
%%sql SELECT landing__outcome, booster_version, launch_site
        FROM SPACEX
        WHERE landing__outcome
            LIKE 'Failure (drone ship)'
            AND
            DATE BETWEEN '01/01/2015' AND '01/01/2016'
```

landing_outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Lastly, we grouped and sorted the launches by the various Landing Outcomes

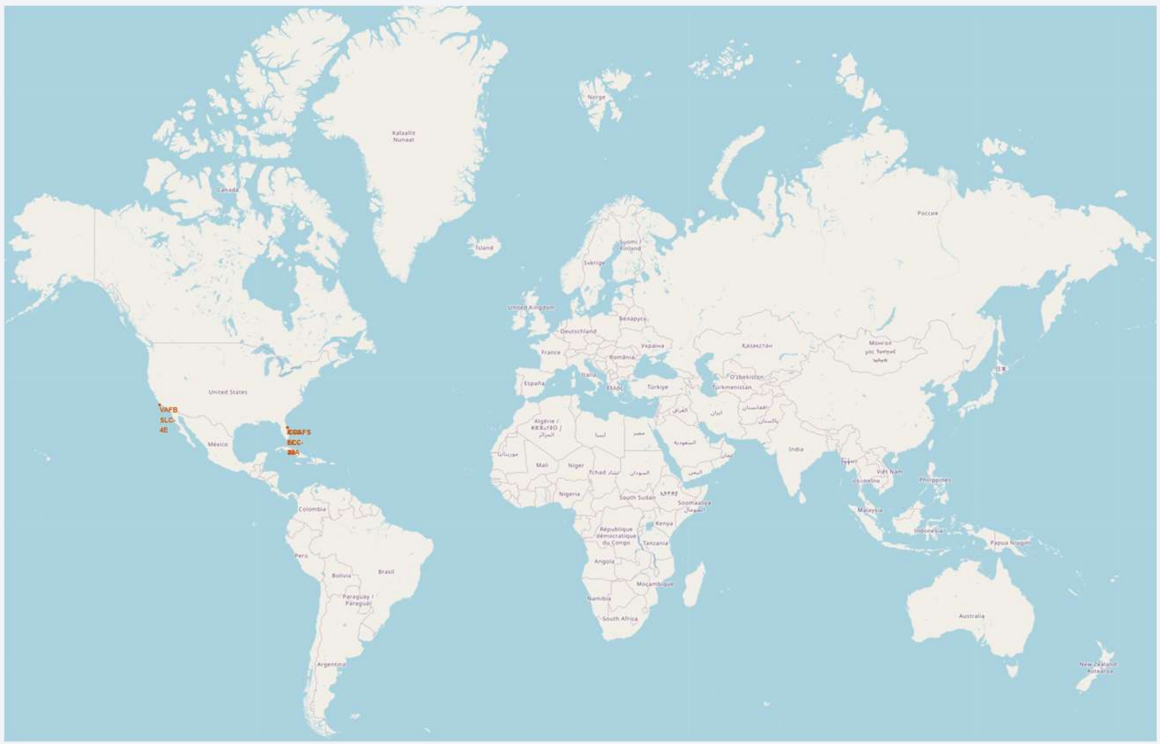
```
%%sql
    SELECT landing__outcome, COUNT(landing__outcome)
        FROM SPACEX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY landing__outcome
        ORDER BY COUNT(landing__outcome) DESC
```

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


Section 3
Launch Sites
Proximities Analysis

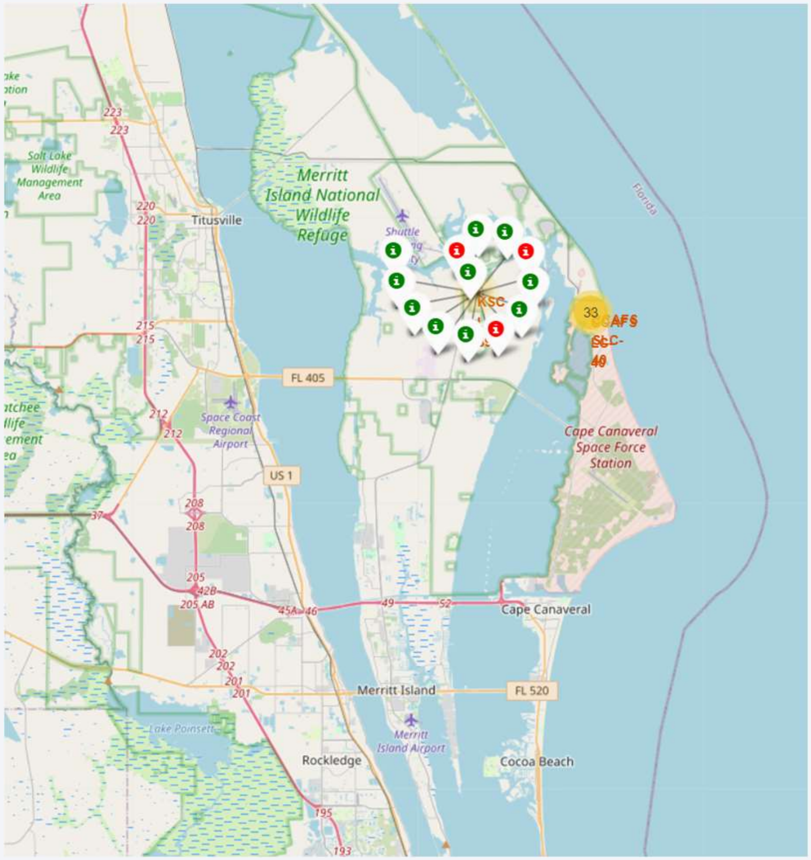
SpaceX Launch Sites on the World Map

- So far SpaceX Launch sites are all in the US on coastlines



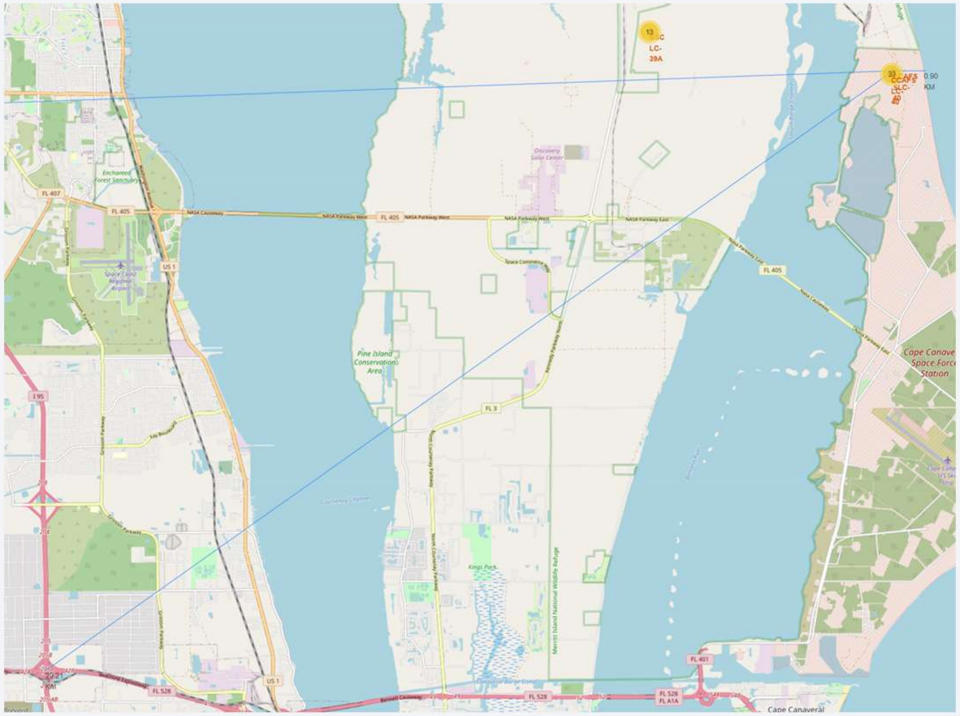
Map of SpaceX Launch Outcomes

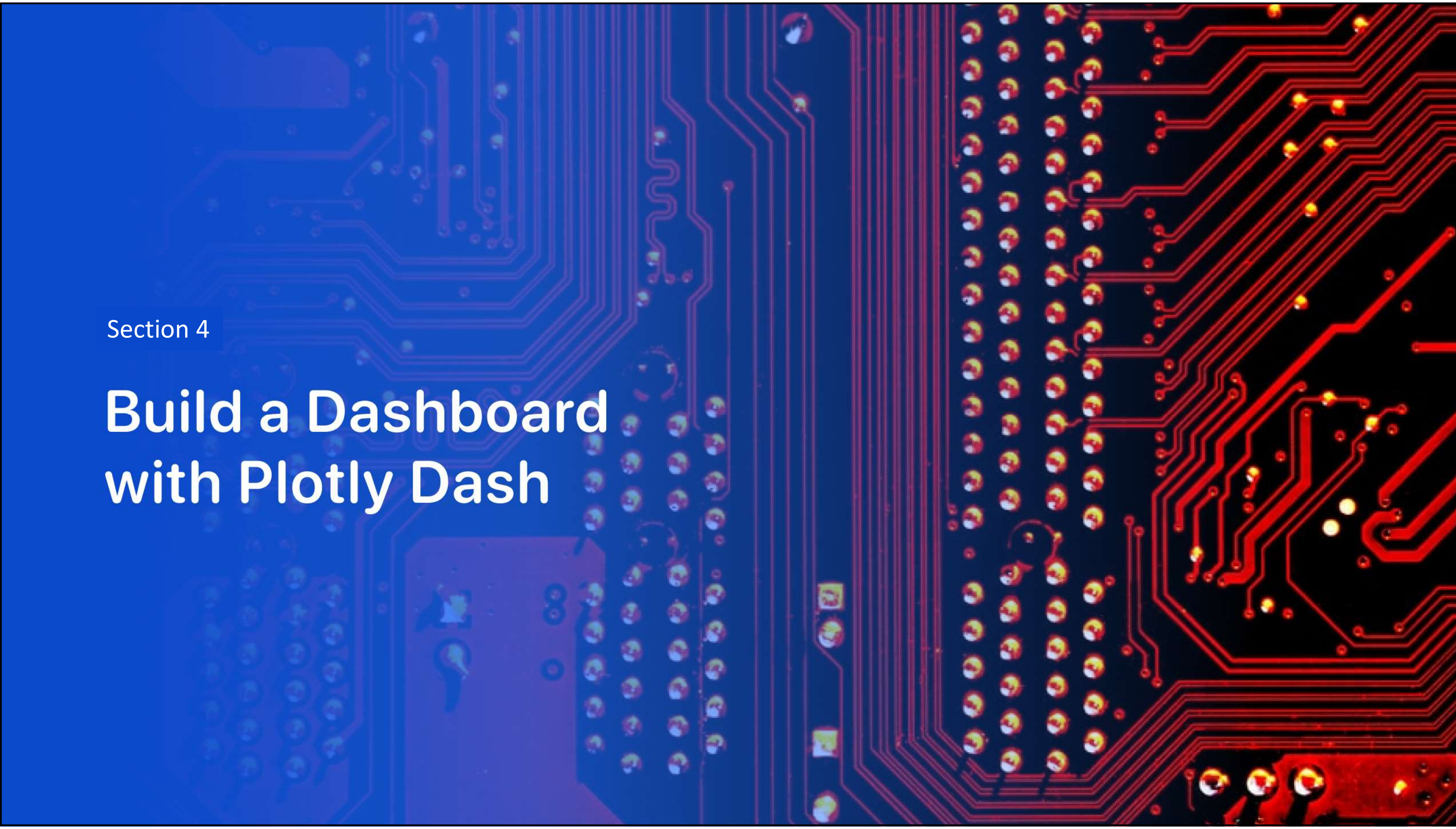
- The Folium library allows us to add markers for each SpaceX launch.
- Here we were able to color code them green for success and red for failed mission.
- When zoomed out, the total number of launches is displayed as seen in yellow here.



Distances from Key Locations

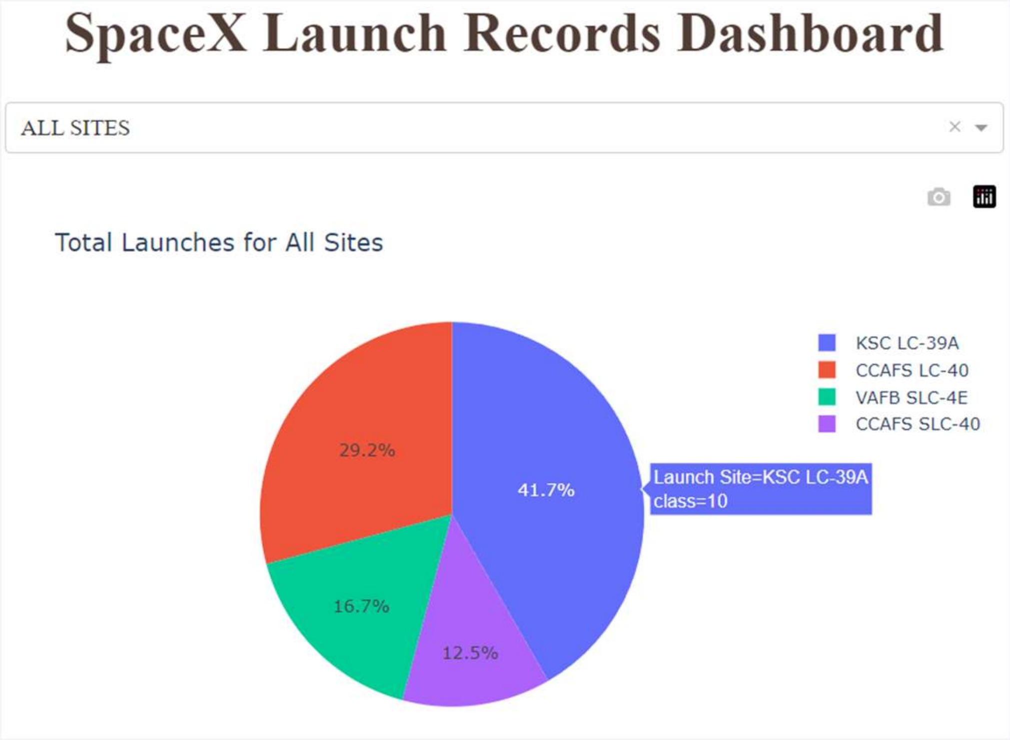
- The Folium library also allows us to calculate and display distances between various locations.
- In this example we have blue lines to the closest ocean, highway, and the closest city.




Section 4
Build a Dashboard
with Plotly Dash

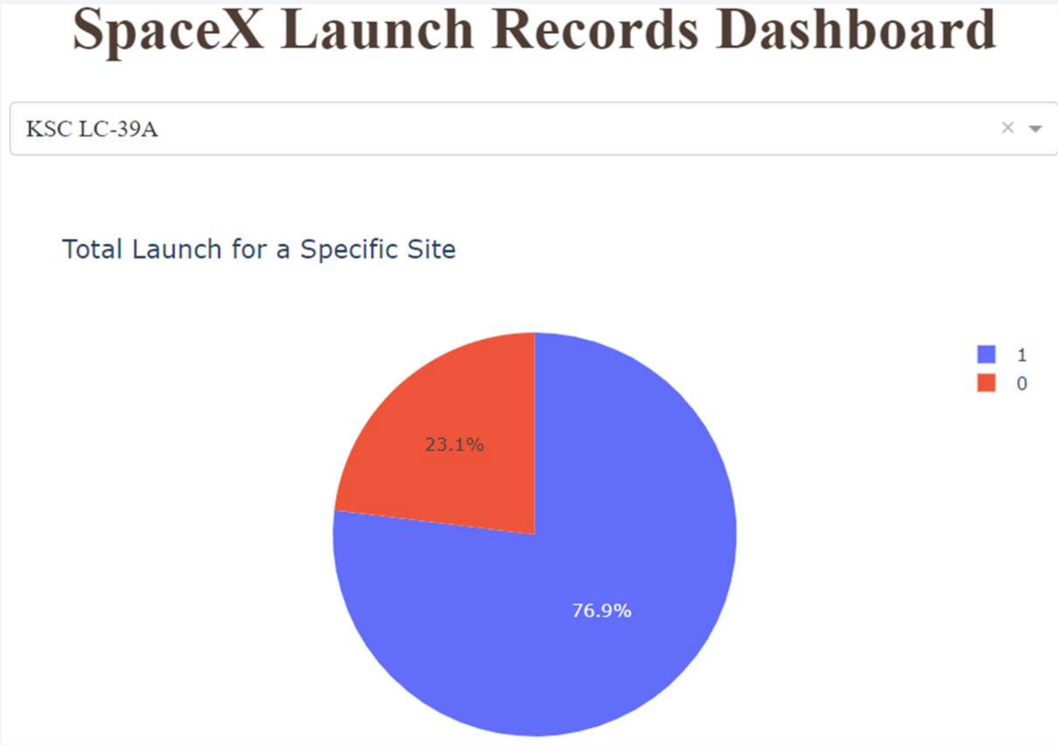
Piechart of Total Successful Records by Site

- With Pandas, Dash, and Plotly a dashboard was created to display launch outcomes
- In this pie chart we have totaled the successful launches and sorted them by launch site
- Here we can see KSC LC-39A has the most successful missions



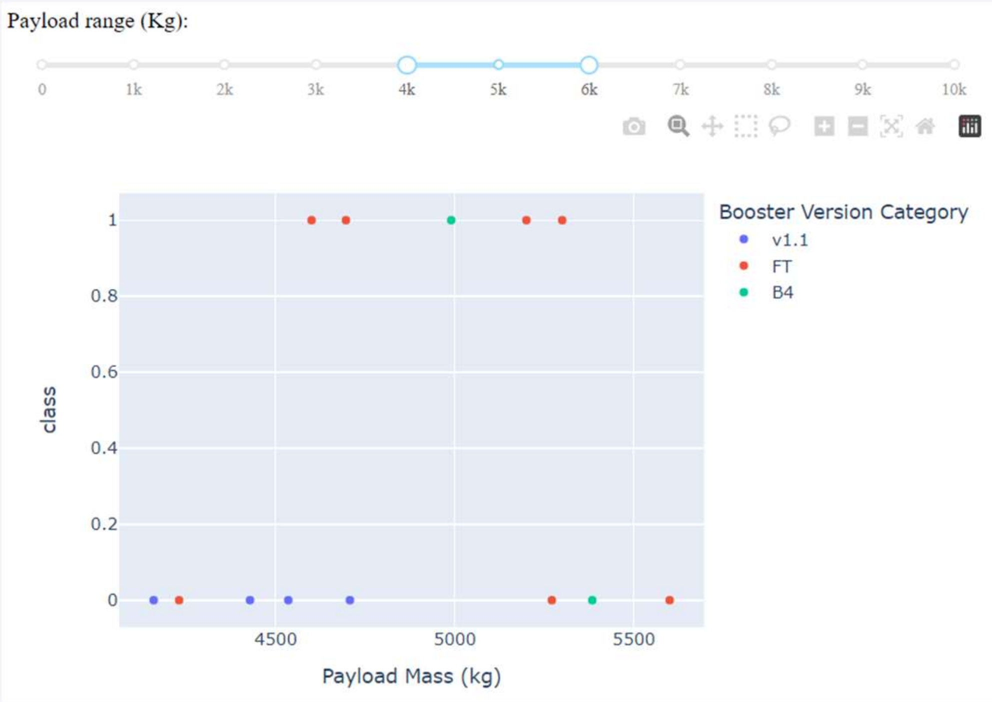
Total Lanches for a Specific Site

- When we use the dropdown filter and change the selection to a specific launch site we are able to see the breakdown of successful and failed launches



Payload Mass by Booster Type

- In the same dashboard we were also able to plot the various booster types by their payload mass.
- We are even able to add a sliding selection bar to further analyze certain payload ranges that might be of interest.

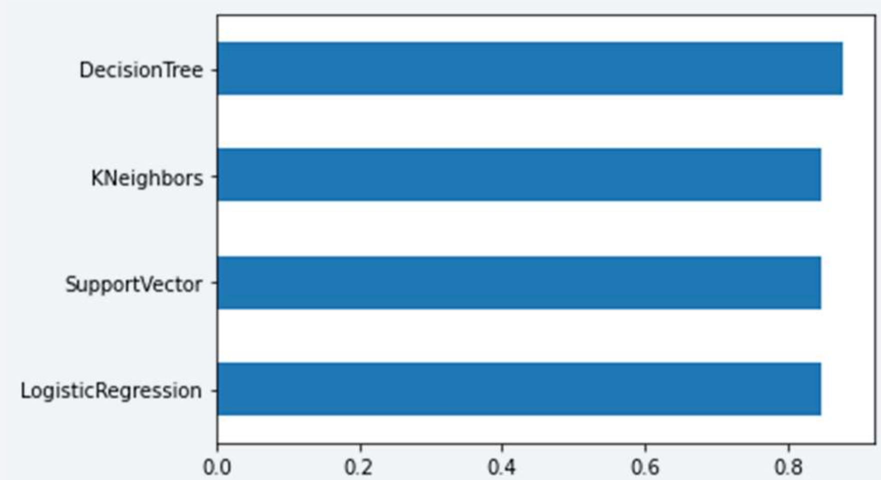


Section 5

Predictive Analysis (Classification)

Classification Accuracy

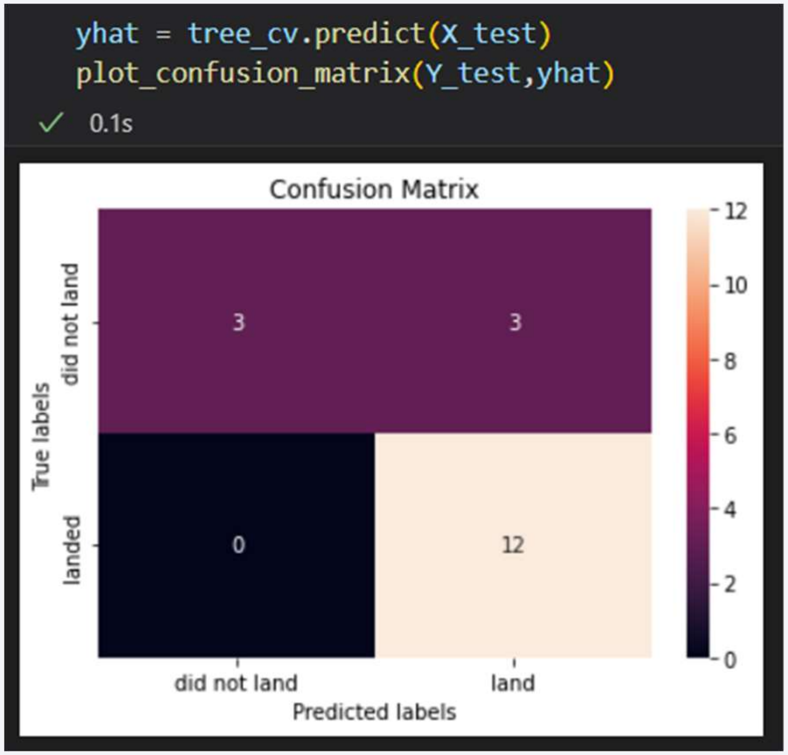
- Using Sci-Kit Learn we are able to create machine learning models to try and predict the outcome of launches before they occur.
- Most models had similar accuracy for this project, but the Decision Tree model performed the best.



	Accuracy
DecisionTree	0.876786
KNeighbors	0.848214
SupportVector	0.848214
LogisticRegression	0.846429

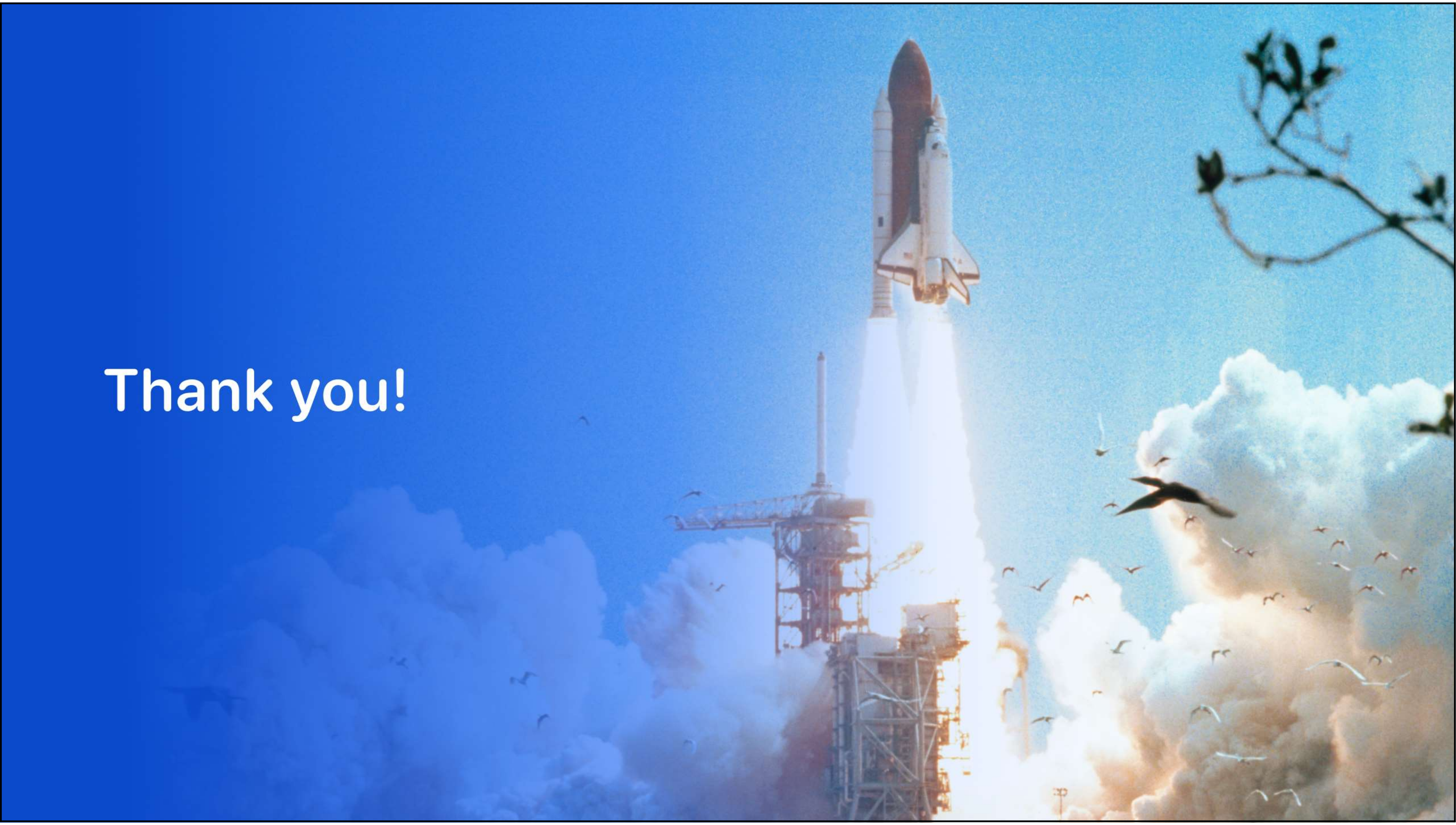
Confusion Matrix

- We can further analyze the Decision Tree model with a confusion matrix.
- While the model predicted 15 of the 18 correctly, it still was not able to identify 3 of the 6 missions that did not land



Conclusions

- Launches with larger payload mass have a higher rate of success.
- Orbit types can also be a useful indicator of how successful a mission will be.
  - Orbit types ES-L1, GEO, HEO, SSO, VLEO were all 80-100% successful
- KSC LC-39A have the most successful launches of any sites; 76.9%.
- Classification algorithms were accurate predictors of launch outcomes.
  - Accuracy ranged from 84-87% with the Decision Tree Algorithm performing the best.
- Since 2013 the trend of successful missions has increased drastically.
  - Missions from 2016 to 2020 had an average success rate of 76% and 2019 to 2020 was 87%.
  - This is an impressive improvement that will forever change the approach to future space missions.


Thank you!