IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ivan Lopes Bicudo de Castro
April 3rd, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Collecting data using the SpaceX API and webscraping
  - Data wrangling
  - Exploratory Data Analysis (EDA) using various methods
  - Creation of a machine learning pipeline for prediction
- Summary of all results
  - EDA results, insights from interactive dashboard and predictive machine learning results

Introduction

- Project background and context
  - It is said that space is the final frontier. More and more companies are venturing into the business of space exploration, but as it is with all business, costs and results are key.
  - One of the main players in the new space exploration age is SpaceX, with its Falcon 9 launch vehicle, which is advertised as low as US$ 62 million per launch, while other companies charge upwards of US$ 165 million. The savings on SpaceX are due to the reusability of the Falcon 9's first stage.
  - If we are to compete with SpaceX, we need to predict if a rocket's first stage will land successfully.
- Problems you want to find answers
  - Is it possible to determine from data if a rocket's first stage will land successfully?
  - What are the factors that contribute to a successful or failed landing?
  - What conditions need to be achieved for a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected from SpaceX API
  - Webscraping from SpaceX's Wikipedia page
- Perform data wrangling
  - Convert various landing outcomes into training labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating machine learning models

Data Collection

Data collection was performed in two ways:

1. Using get requests from the SpaceX API. Response was decoded and converted to a Pandas dataframe using .json methods. Data were reviewed, cleaned and checked for missing values, which were filled using data averages.
2. Webscraping SpaceX's Wikipedia page for other relevant data points using Beautiful Soup, extracting and parsing HTML tables into dataframes.

The two processes yielded valuable complementary data to be further analysed.

Data Collection – SpaceX API

- Data is collected with get requests from SpaceX API, decoded and converted with .json methods
- Data is cleaned, assembled into a dataframe and filtered to include only Falcon 9 launches
- Missing values are replaced with data averages
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/jupyter-labs-spacex-data-collection-api.ipynb

1 – Get request from SpaceX API

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

[10]: response = requests.get(spacex_url)
```

2 – Decode and convert

```
[14]: # Use json_normalize meethod to convert the json result into a dataframe
      js = response.json()
      data = pd.json_normalize(js)
```

3 - Filter

```
[27]: # Hint data['BoosterVersion']!='Falcon 1'
      data_falcon9 = df.loc[df['BoosterVersion']!='Falcon 1']
```

4 – Calculate averages and replace missing values

```
[30]: # Calculate the mean value of PayloadMass column
      mean_pm = data_falcon9['PayloadMass'].mean()

      # Replace the np.nan values with its mean value
      data_falcon9["PayloadMass"].replace(np.nan,mean_pm,inplace=True)
      data_falcon9.isnull().sum()
```

Data Collection - Scraping

- Data is collected with get requests from Wikipedia and parsed to a BeautifulSoup object
- HTML launch tables are collected and assembled into a dataframe
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/jupyter-labs-webscraping.ipynb

1 – Get request from SpaceX API

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

```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```

2 – Parse to a BeautifulSoup Object

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

3 – Create a dataframe from parsed HTML launch tables

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Data Wrangling

- The launch data was processed by calculating the number of launches from each launch site, the number and ocurrences of each orbit and the mission outcome of the orbits
- A label for landing outcomes was created and added to our dataframe to help us assess when a first stage had successfully returned for reuse.
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The data was examined by plotting the following relationships vs. success:
  - Flight Number and Payload Mass
  - Flight Number and Launch Site
  - Payload Mass and Launch Site
  - Orbit type success rate
  - Flight Number and orbit type
  - Payload Mass and orbit type
  - Successful launches yearly trend
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/edadataviz.ipynb

EDA with SQL

- Data was loaded into an SQLite database
- The following SQL queries were performed:
  - Displaying the unique names of launch sites in space missions
  - Calculate the total payload mass carried by boosters launched by NASA (CRS)
  - Calculate the average payload mass carried by booster version F9 v1.1
  - Finding the date when the first successful landing outcome in ground pad was achieved
  - Finding out which boosters have landed successfully on a drone ship with a payload mass greater than 4 and less than 6 metric tons.
  - Which boosters carried the maximum payload mass
  - Showing failed landings in drone ships in the year 2015
  - Ranking the outcomes between June 4th, 2010 and March 20th, 2017
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites were marked in the Folium map, with added circles, markers and lines showing the outcomes of launches on those sites
- The locations allow us to visualize the relative proximity of launch sites to other relevant features, such as coastlines, highways, railroads and cities
- Markers of different colors allow us to quickly assess the success rate of each site
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

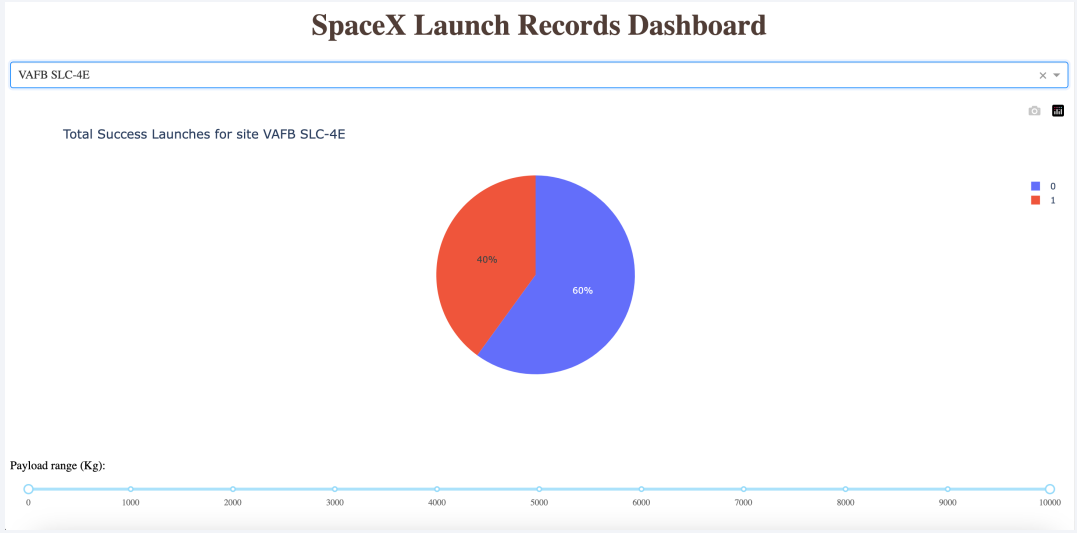
- An interactive dashboard with the following elements was built:
- Dropdown menu with launch site options
- A pie chart showing total successful launch count in all sites or the success launch count for each launch site, depending on dropdown menu choice
- A payload mass range slider
- A scatterplot of payload and success for all sites or for a specific site, depending on dropdown choice and payload range selection
- The elements on the dashboard allow the user to easily visualize the success rates on launch sites according to the payload mass range
- App URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/spacex_dash_app.py

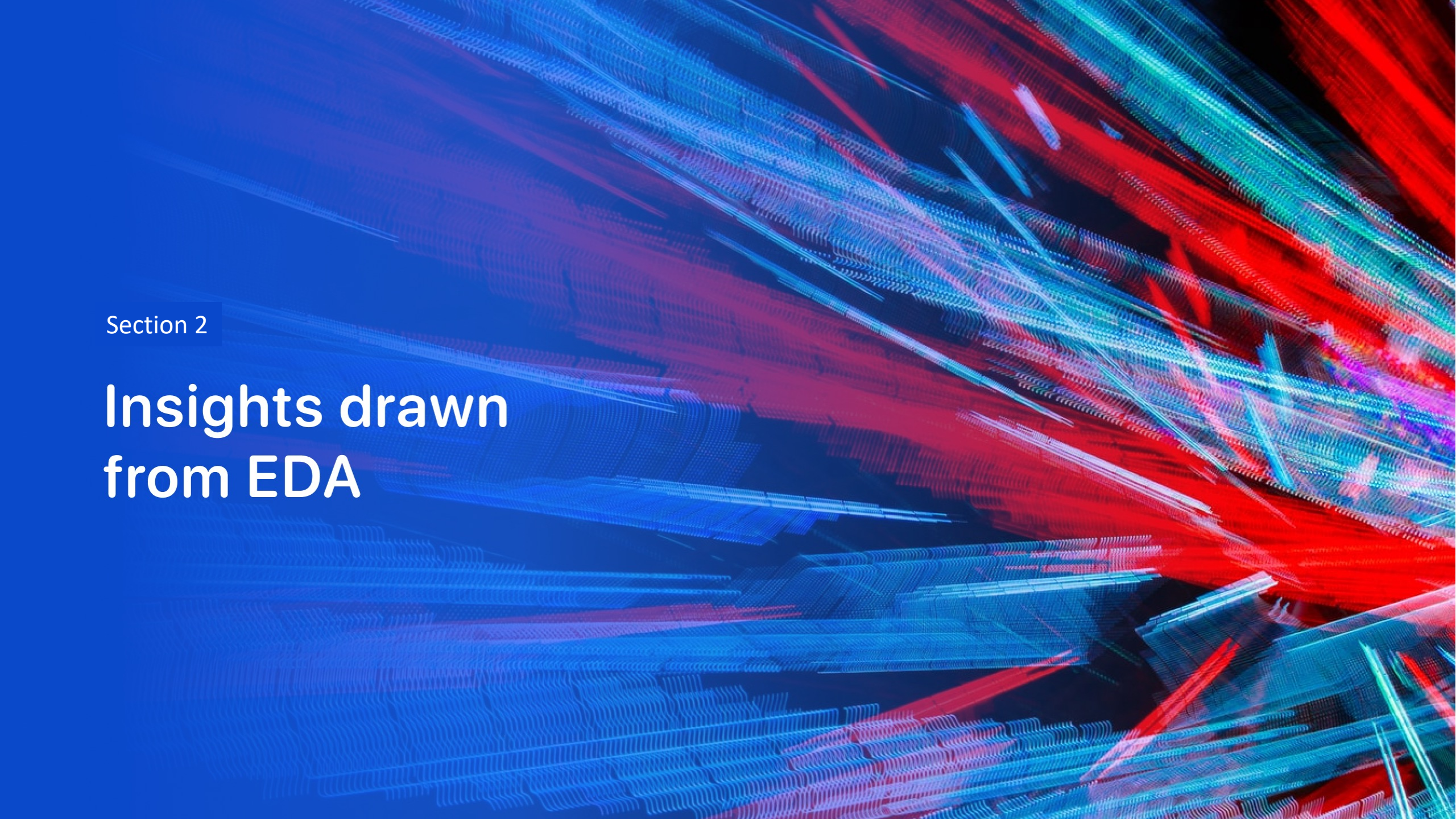
Predictive Analysis (Classification)

- For the machine learning pipeline, data was loaded with Numpy and Pandas, transformed and split into training and test sets
- Several distinct machine learning models were built and refined with GridSearchCV hyperparameters
- Using accuracy as our model estimation, it was possible to determine the best model for the task
- Notebook URL: https://github.com/ivanbicudo/loom/blob/DS-Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

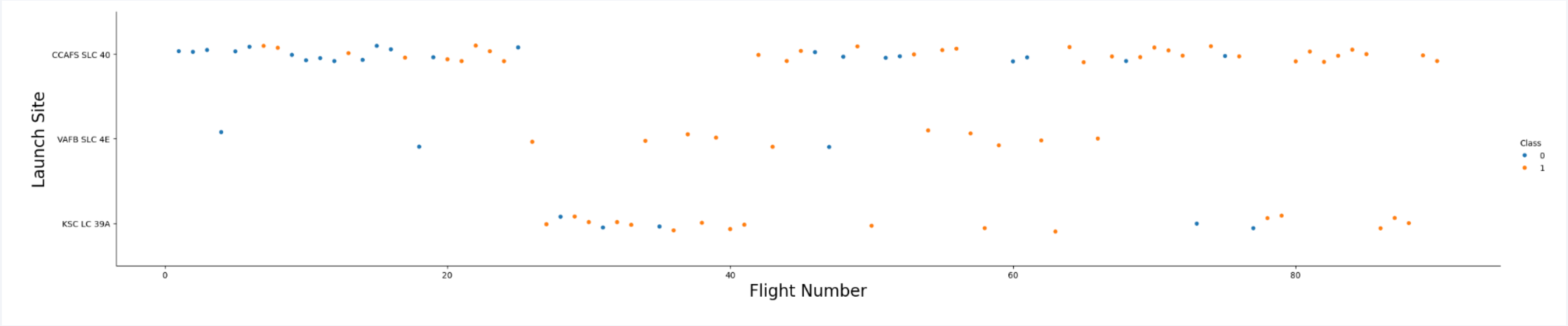
Results

- Our EDA was able to provide valuable insights, such as the relationship between payload mass, launch sites, booster version and orbit types, as well as launch site locations characteristics
- The interactive dashboard shows the relationships between launch sites, payload mass and successful missions
- Our machine learning models were evaluated, and the best classification model was found



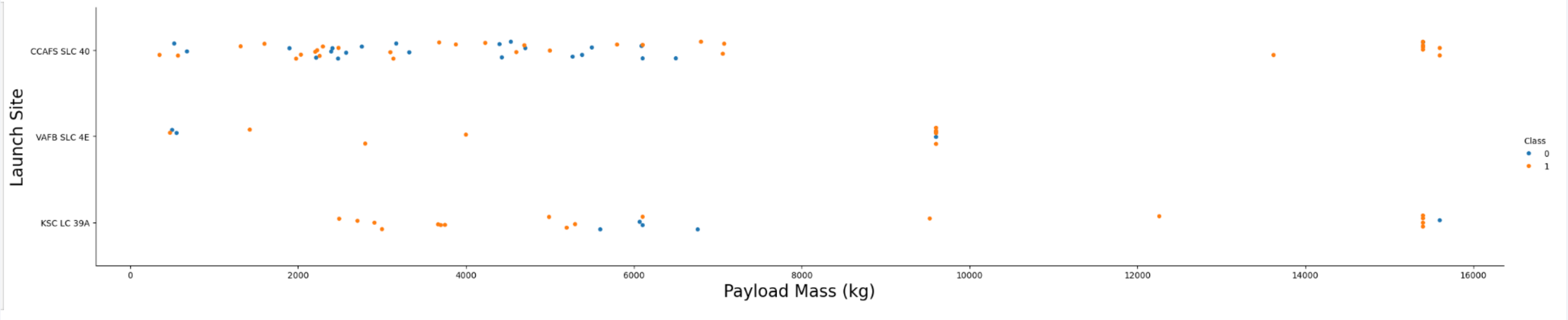

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- From this plot we can see how the first stage landing capabilities improved with each flight. We can also notice that SpaceX tends to perform consecutive launches from the same site, especially on earlier Flight numbers.

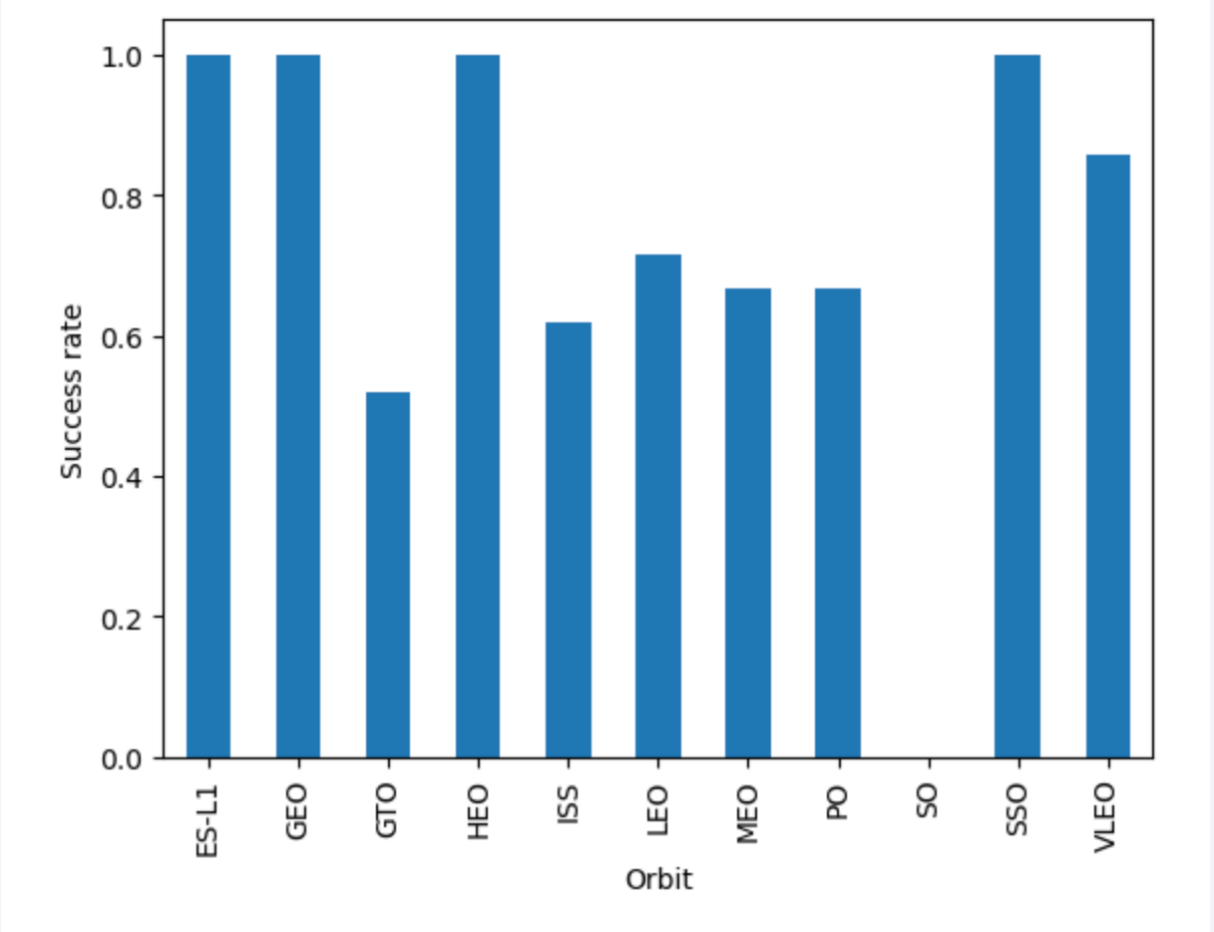
Payload vs. Launch Site



- We can see that no flights carrying more than 10000 kg are launched from VAFB SLC 4E

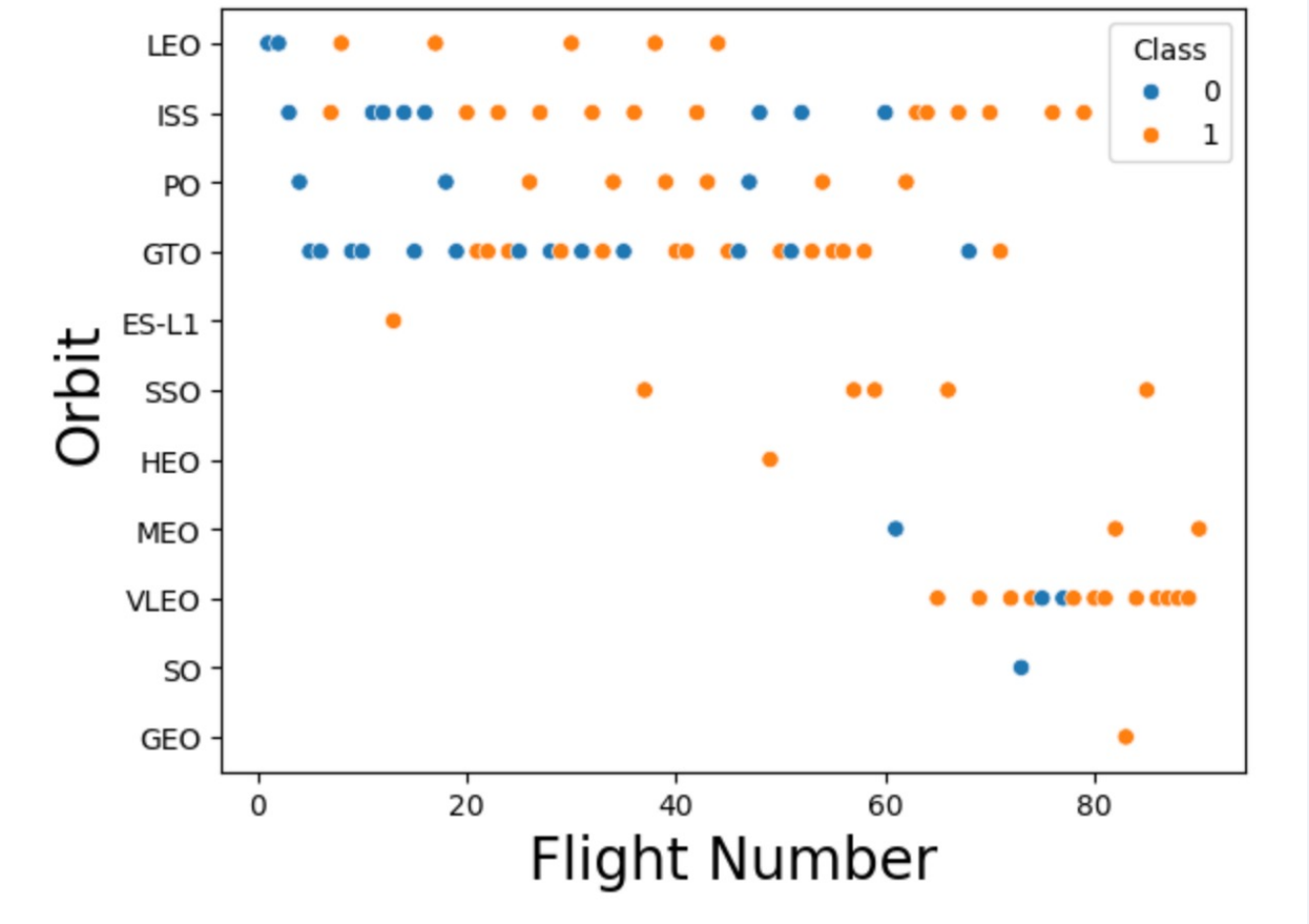
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO orbit types have a very high rate of success



Flight Number vs. Orbit Type

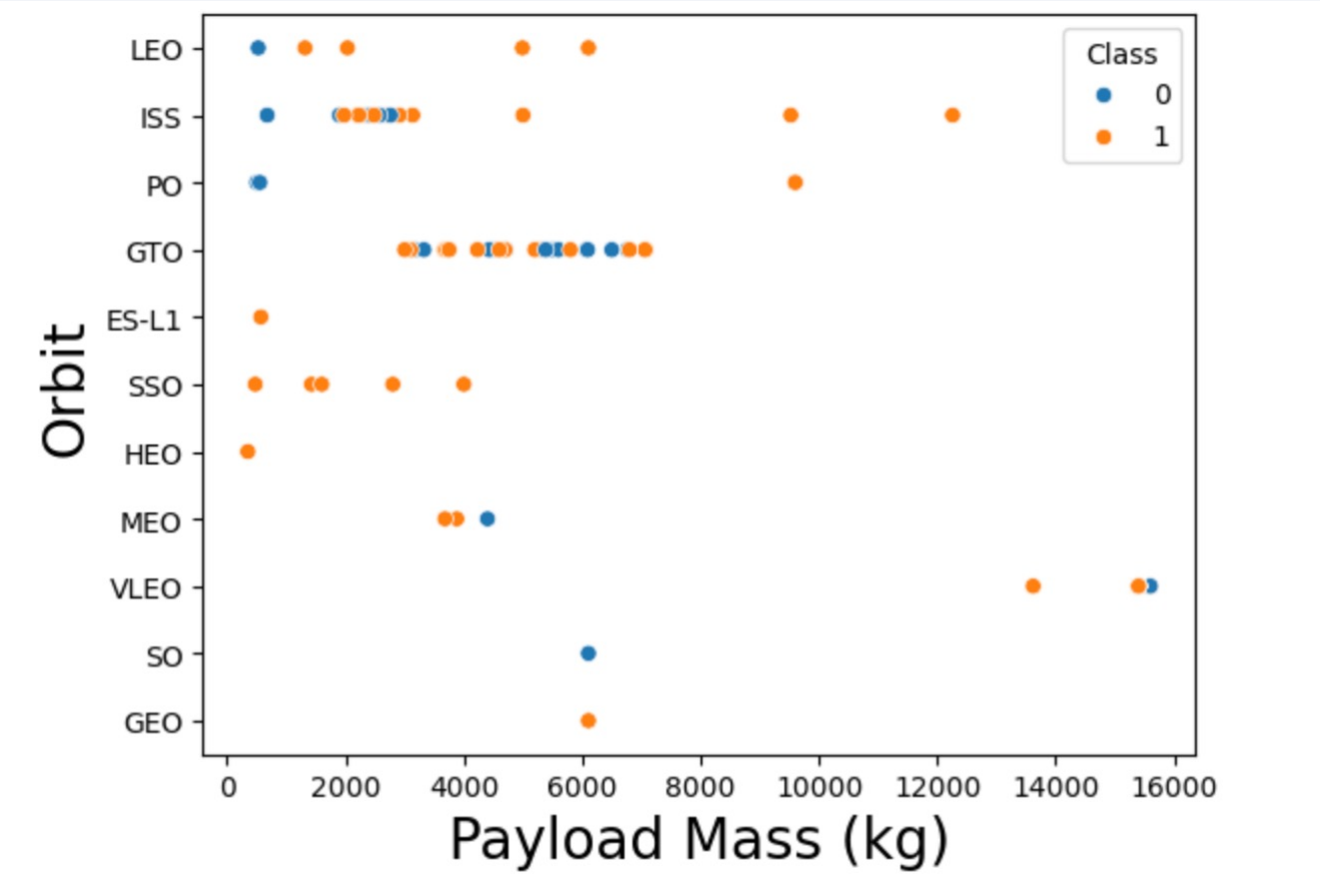
- In LEO orbit the Success appears related to the number of flights
- There seems to be no relationship between flight number when in GTO orbit



Payload vs. Orbit Type

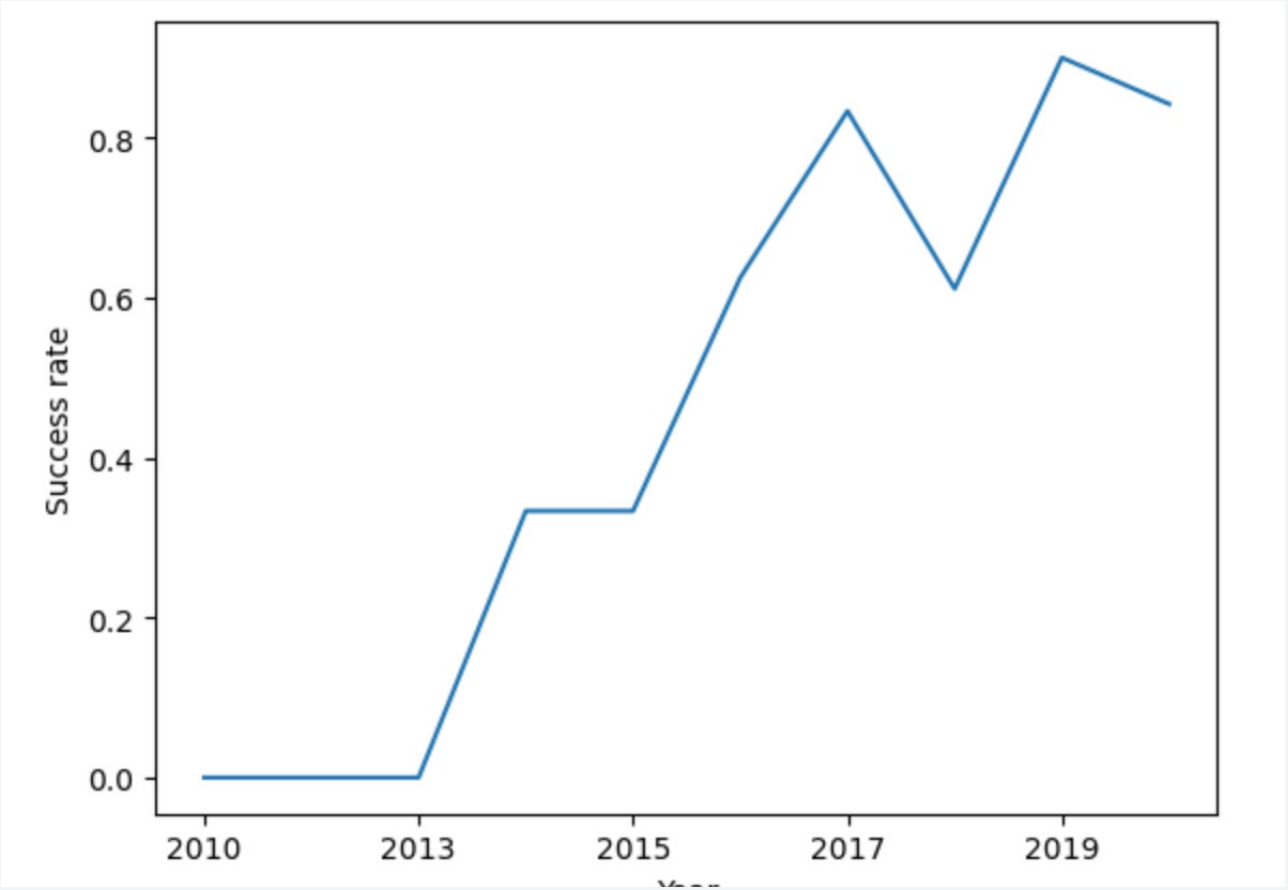
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

- In GTO it cannot be so easily distinguished



Launch Success Yearly Trend

- Launch success rates rise from 2013 until 2020



All Launch Site Names

- In SQL, the DISTINCT method was used to locate unique launch site names

```
%sql select distinct launch_site from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- We queried the database for data in the LAUNCH_SITE column where the string %CCA appeared, limiting our results to 5 items

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

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

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

Total Payload Mass

- We can calculate the total payload mass carried by booster for NASA (CRS) using the SUM operator. The total is 45596 kg.

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = "NASA (CRS)"
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 can be calculated using AVG. It's 2928.4 kg.

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_version = "F9 v1.1"
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- We can find the date first successful landing on a grounf pad by combining the function MIN in the Date column where the landing outcome is a success on ground pad. It happened on December 22nd, 2015.

```
%sql select MIN(Date) from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We can the list names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 just by combining those requirements with AND in our query.

```
%%sql select Booster_version from SPACEXTBL
where ("PAYLOAD_MASS__KG_" > 4000)
and ("PAYLOAD_MASS__KG_" < 6000)
and ("Landing_Outcome" = 'Success (drone ship)')
```

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

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 will calculate the total number of successful and failure mission outcomes using COUNT on as TOTAL and grouping with GROUP BY mission outcomes.

```
%sql select Mission_Outcome, count(*) as TOTAL from SPACEXTBL group by MISSION_OUTCOME order by MISSION_OUTCOME
```

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

Mission_Outcome	TOTAL
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We can also list the names of the booster which have carried the maximum payload mass using a subquery.

```
%%sql select distinct Booster_Version from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select max("PAYLOAD_MASS__KG_")
                           from SPACEXTBL) order by Booster_version
```

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

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

2015 Launch Records

- These are the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015. We select the attributes from the database where the outcome is failure to land on a drone ship and the year is 2015.

```
%%sql
SELECT substr(Date,6,2) as month,booster_version,"Landing_Outcome","Launch_Site"
from SPACEXTBL where "Landing_Outcome"
= 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

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

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

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

- Finally, we can 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 using functions such as COUNT, GROUP BY, ORDER BY and DESC

```
%%sql select "Landing_Outcome", count (*) as 'count' from SPACEXTBL
where substr(Date,1,4) || substr(Date,6,2) || substr(Date,9,2)
between '20100604' and '20170320' GROUP BY "Landing_Outcome" ORDER BY "COUNT" DESC
```

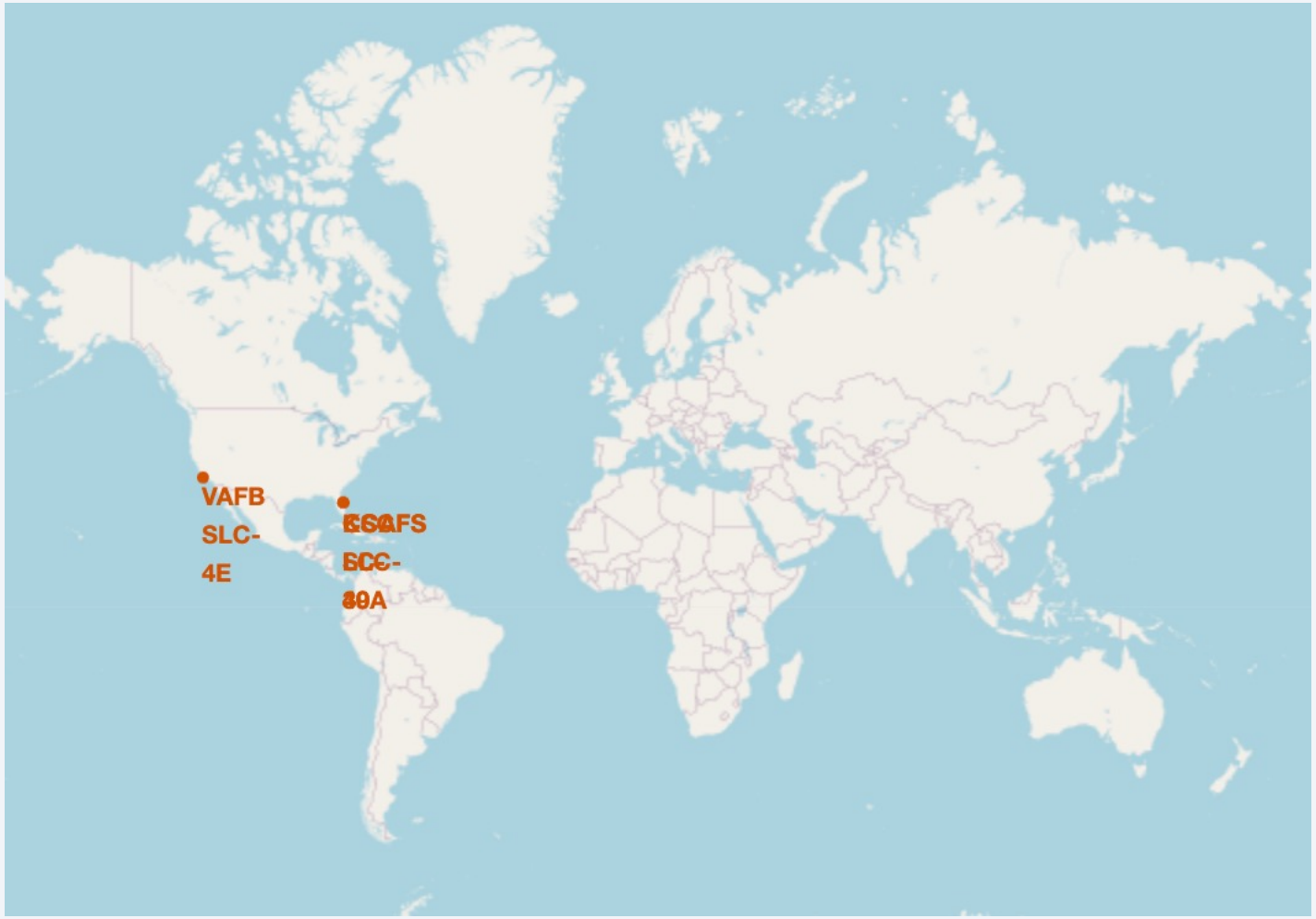
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count
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

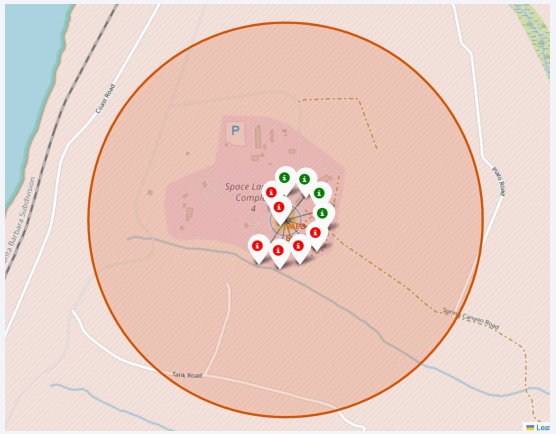

Section 3
Launch Sites
Proximities Analysis

Space X Launch Sites Global Map

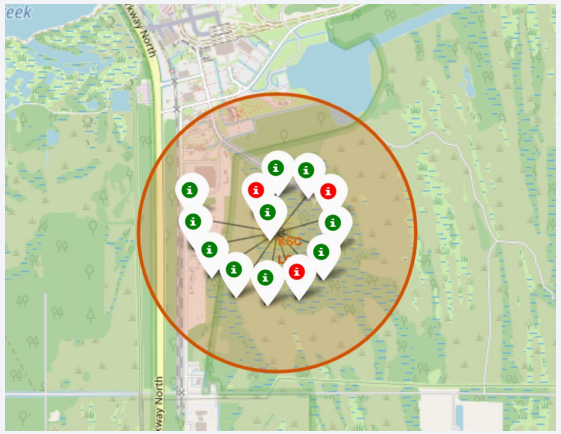
- SpaceX uses launch sites with the following characteristics:
  - On the east and west US coasts
  - Close to coastlines
  - As close to the equator as possible



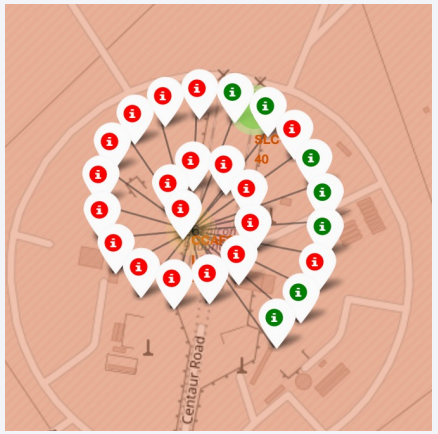
Launch site outcomes cluster map



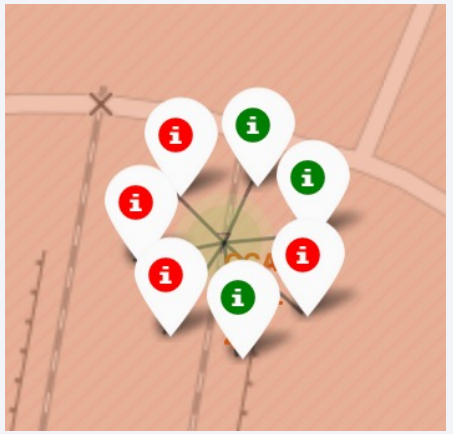
VAFB SLC



KSC – 39A



CCAFS LC 40

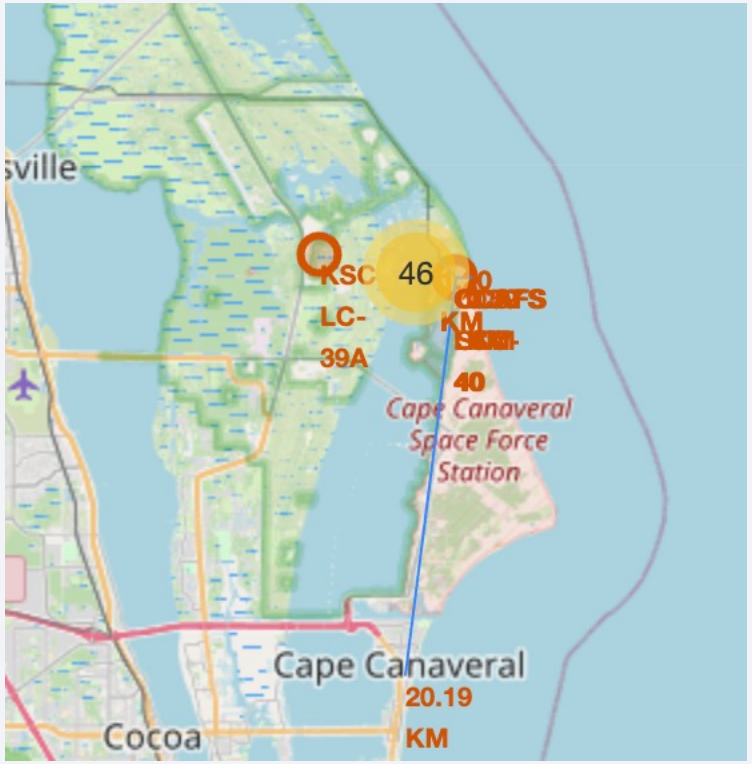


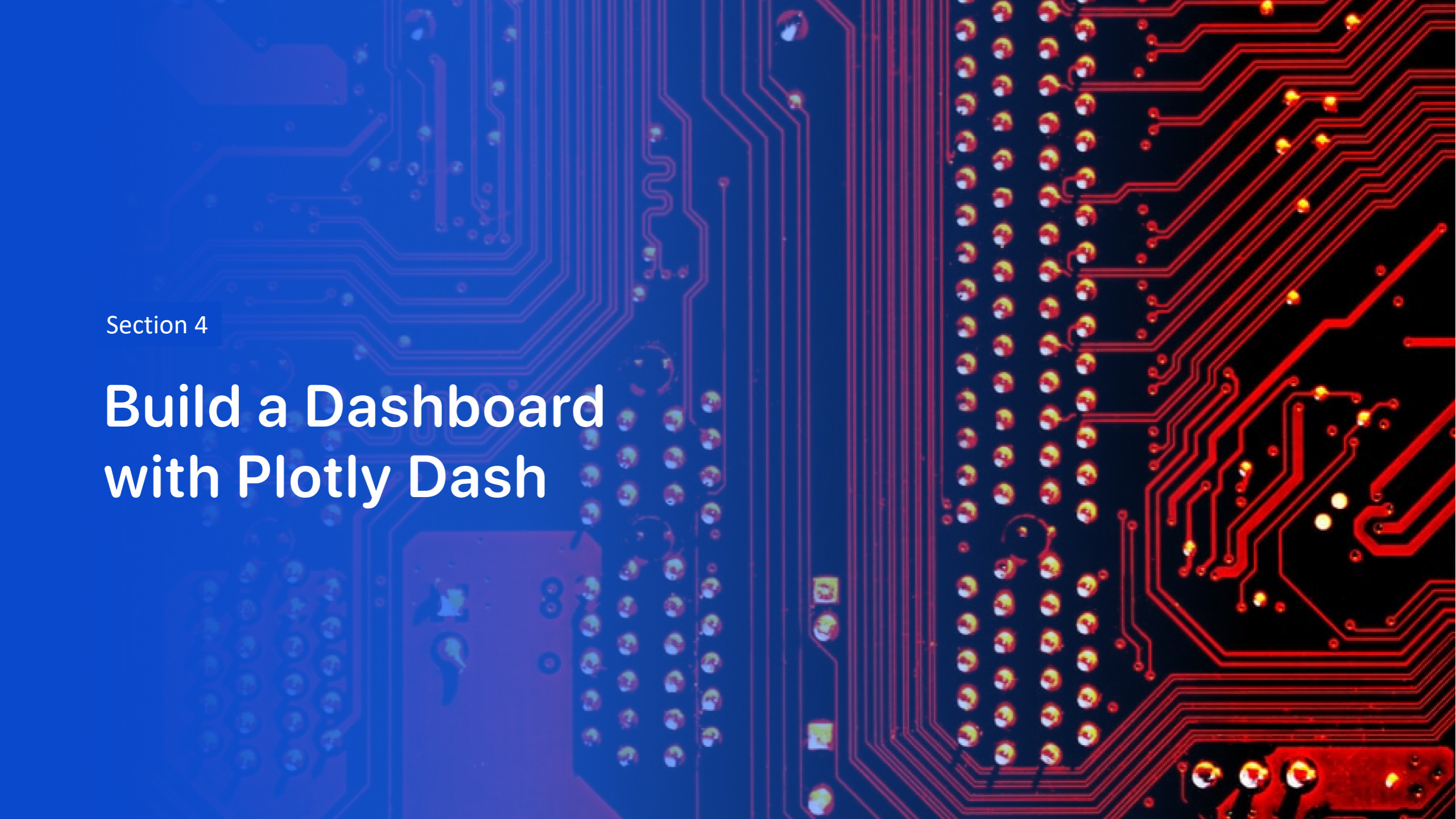
CCAFS SLC 40

- Green markers represent successful missions and red markers represent failed missions

Launch sites distance from landmarks

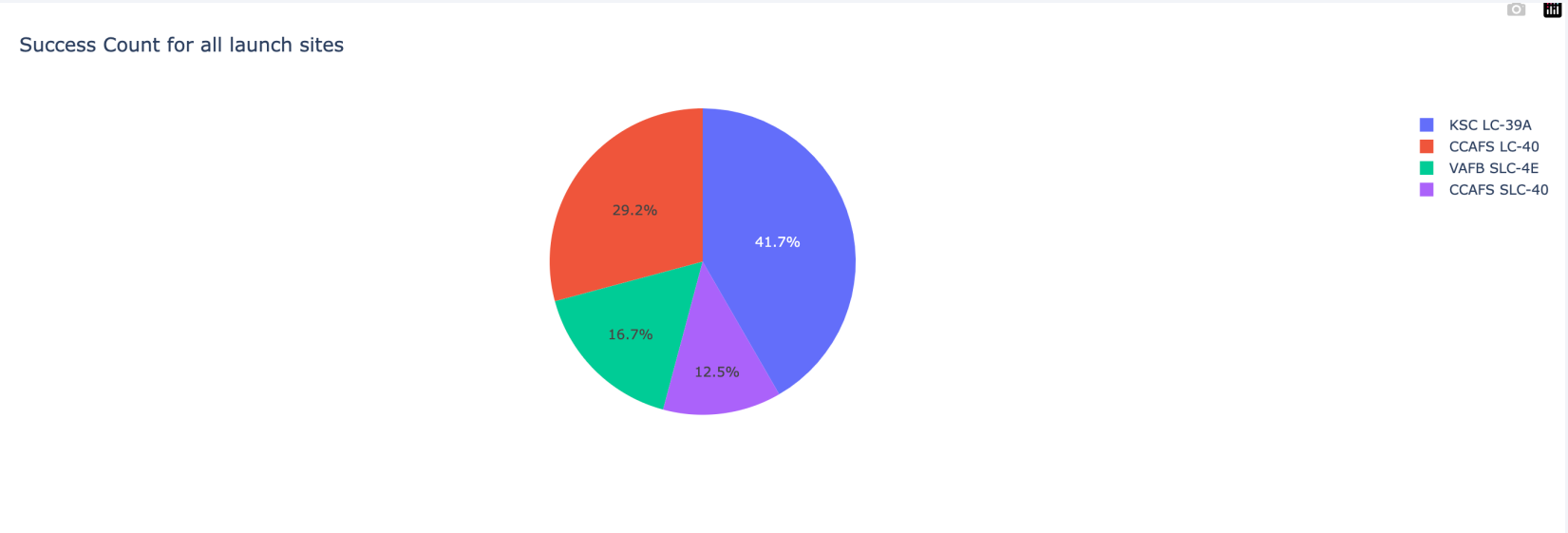
- Launch sites are close to railroads and highways to facilitate access of resources and personnel
- They are as close as possible to the coastline to minimize rocket flight path over land in case of accidents
- For the same reason, they are distant from cities and urban centers



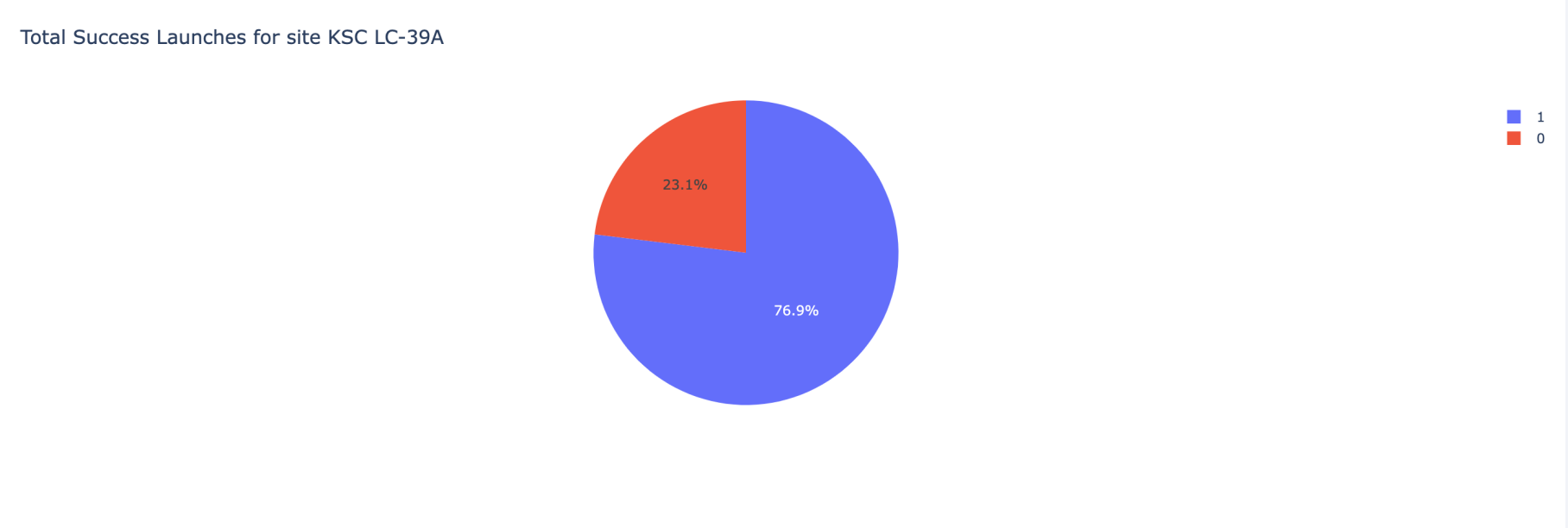

Section 4
Build a Dashboard with Plotly Dash

Launch site success pie chart

- On the graph we can see that KSC LC-39A has the most successful launches of all launch sites

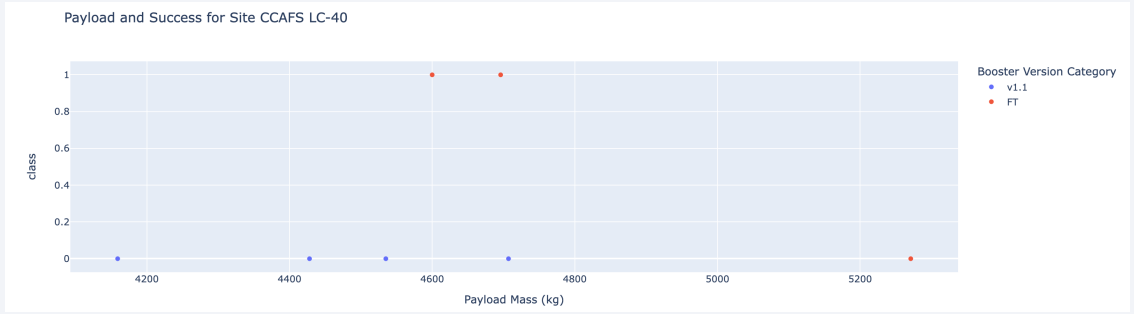


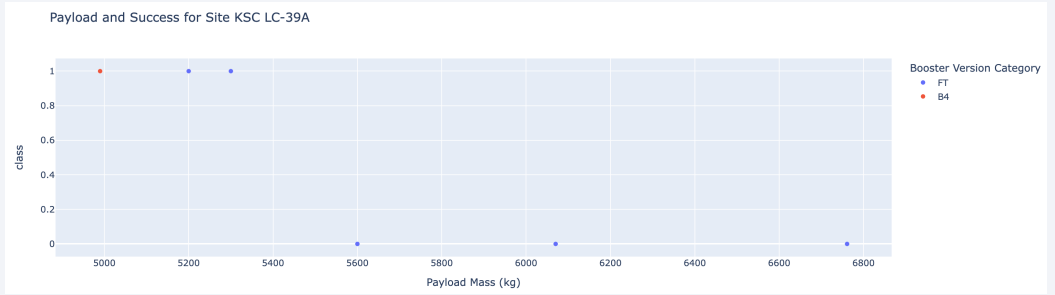
Highest success ratio launch site pie chart

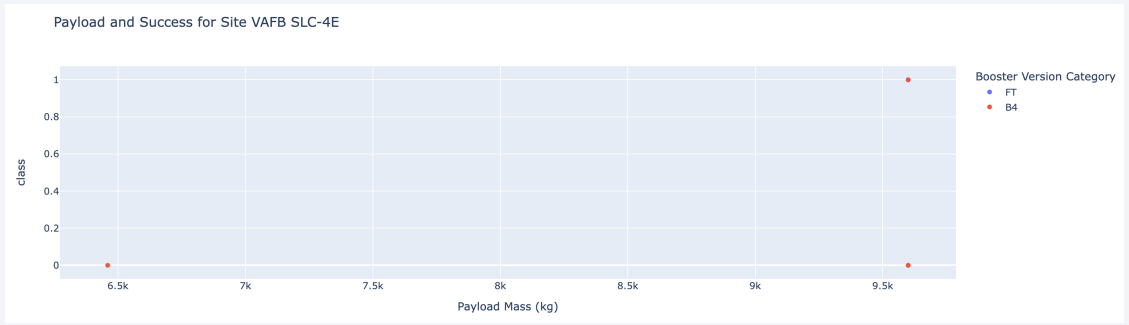


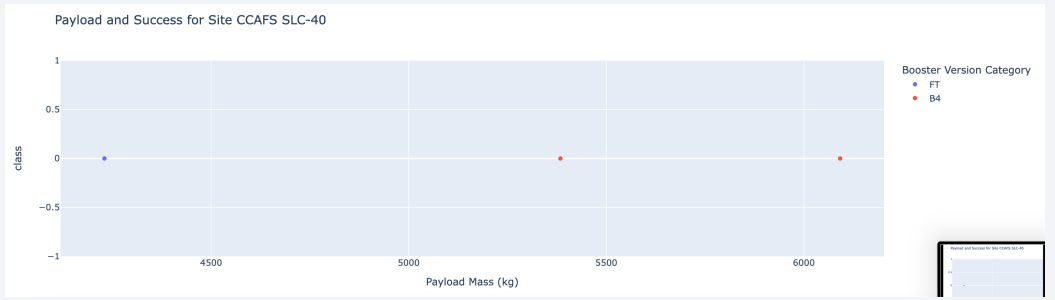
- KSC LC-39A has a success ratio of 76.9%

Payload vs. Launch Outcome (more than 4000 kg)



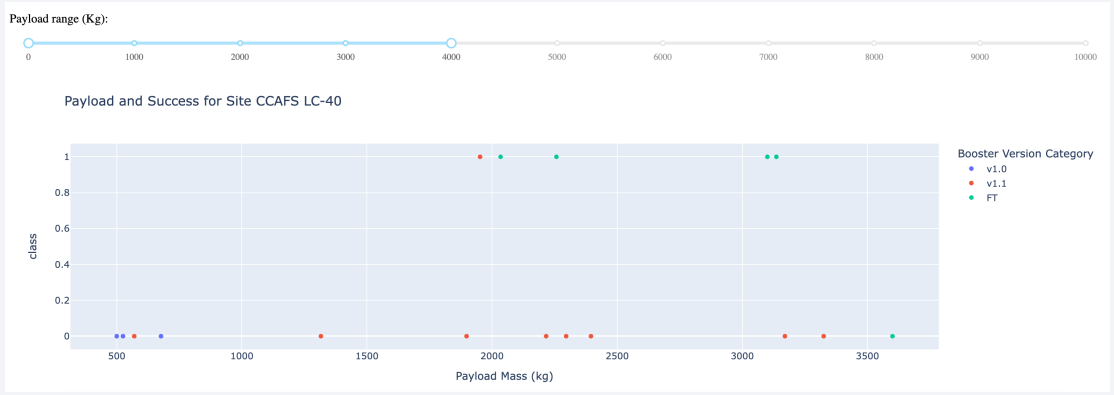


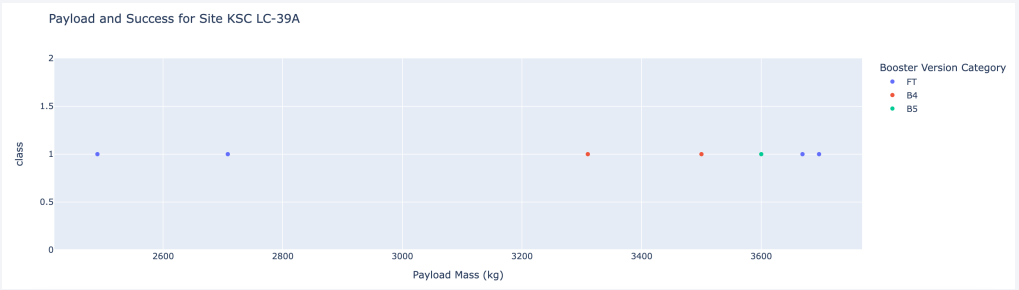


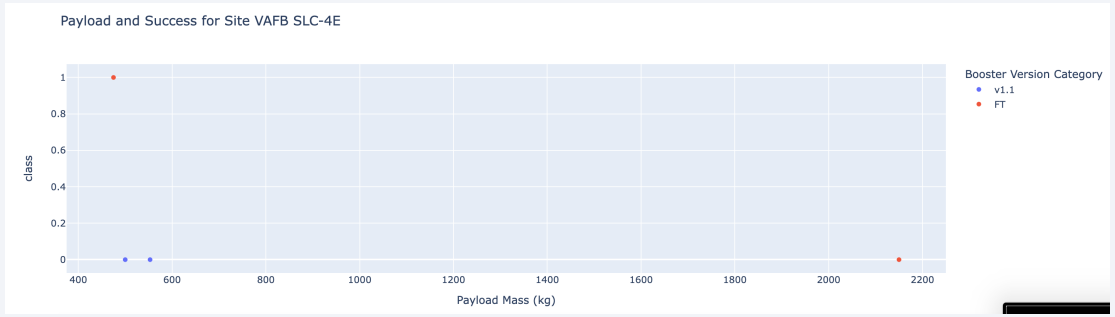


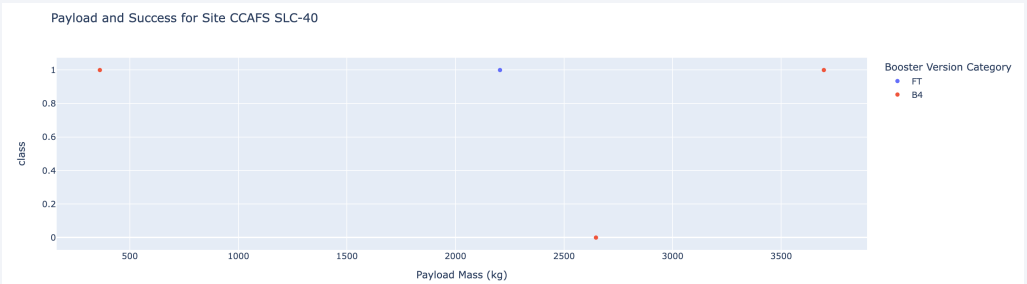
- With heavier payloads, B4 boosters perform better than FT versions

Payload vs. Launch Outcome (up to 4000 kg)









- On payloads up to 4000 kg, booster versions FT, B4 and B5 have very high success rates, while boosters v1.0 and v1.1 have low success rates

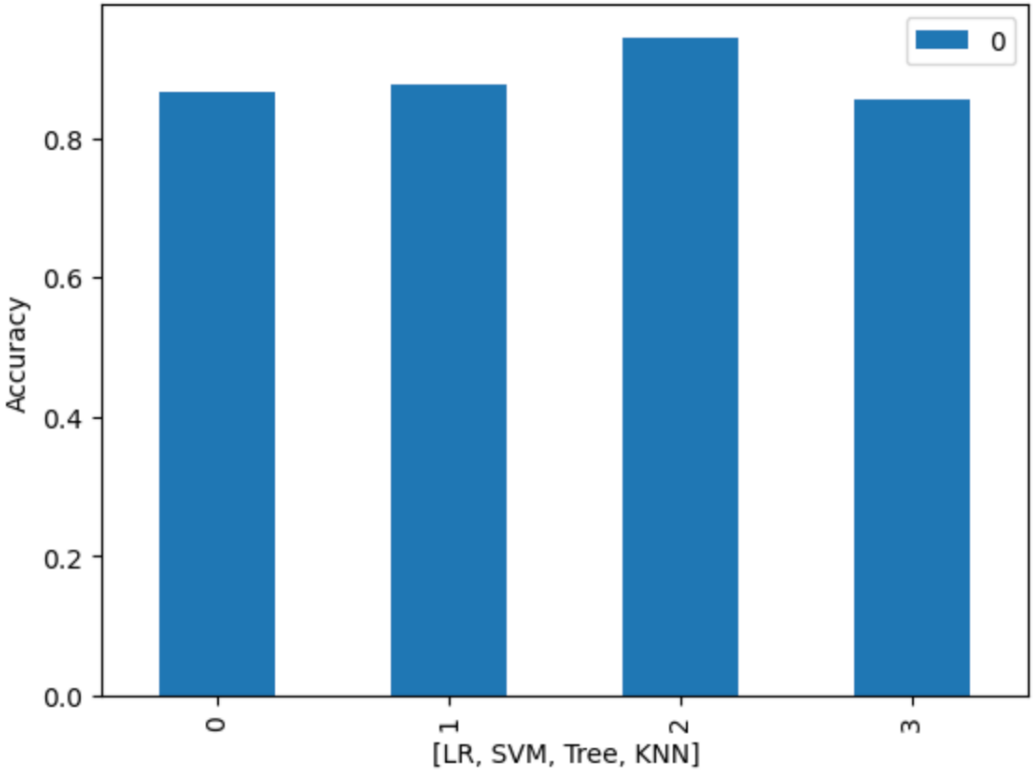
Section 5

Predictive Analysis (Classification)

Classification Accuracy

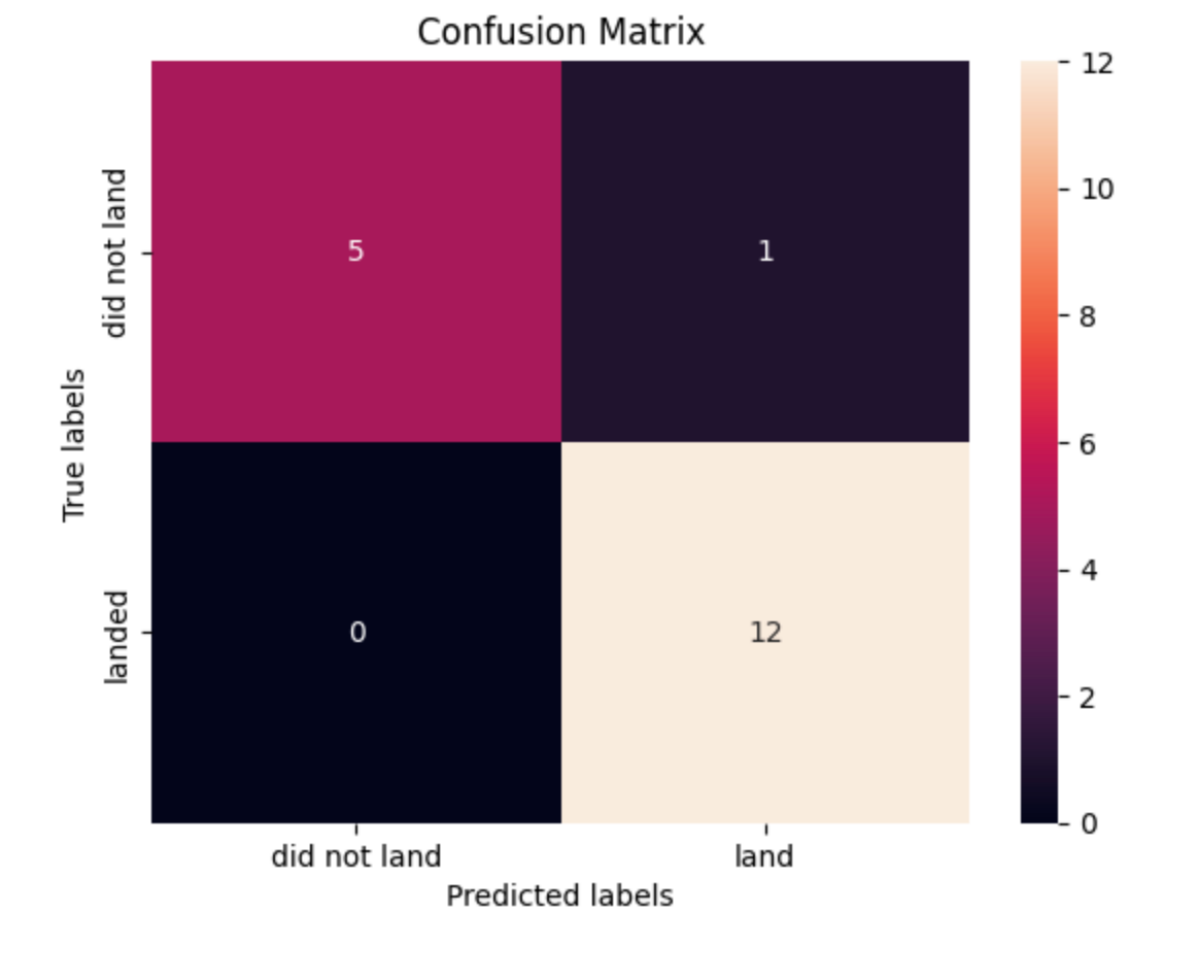
- Decision tree model has the highest classification accuracy

```
Accuracy for LR Model: 0.8666666666666667
Accuracy for SVM Model: 0.8777777777777778
Accuracy for Tree Model: 0.9444444444444444
Accuracy for KNN Model: 0.8555555555555555
```



Confusion Matrix

- The confusion matrix for the best performing model shows only one false positive and no false negative



Conclusions

- Development and testing of improved booster versions increases success rate of booster landings, as launch success rates rose from 2013 to 2020 and later booster versions performed better
- More modern booster versions perform very well with lower payloads
- Launch sites should be near the equator, on the coast, close to transportation infrastructure and away from urban centers
- SpaceX uses four different launch sites, with KSC LC-39A being the most successful
- Decision Tree is the best machine learning algorithm for the task of predicting whether a booster will land

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!