IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicholas Patrick
February 23, 2024

OUTLINE

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

EXECUTIVE SUMMARY

- Summary of methodologies
  - Data Collection with API and Web Scraping
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Interactive Folium Map
  - Plotly Dash Dashboard
  - Predictive Anaylsis

- Summary of all results
  - EDA results
  - Interactive Analytics
  - Predictive Analysis

INTRODUCTION

- Project background and context

This project aims to 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.

- Problems we are attempting to answer:
    - How do different variables affect the first stage landing success rate?
    - Does landing success rate increase over time?

Section 1

Methodology

METHODOLOGY

Executive Summary

Data collection methodology:

SpaceX Rest API

Wikipedia Web Scraping

Perform data wrangling

Filtered Data

Dealt with Missing Values

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, evaluating, adjusting models

DATA COLLECTION

Data sets were collected via a combination of SpaceX REST API requests and web scraping from a Wikipedia SpaceX table. The combination of the two resources allowed us to gather a more complete dataset for analysis.

REST API data included Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude, and Latitude.

Wikipedia Web Scraping data included Flight Number, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Version Booster, Booster Landing, Date, and Time.

Some of the redundant data allowed verification of data.

DATA COLLECTION – SPACEX API

1

Task One:

- Requested data from SpaceX API
- JSON file normalized with .json_normalize()
- File turned into dataframe with data = pd.json_normalize(response.json())
- Cleaned data by creating subset of variables
- Global variables declared
- BoosterVersion and calls were used to update list (example: getLaunchSite(data))
- Dataset constructed, and columns combined into a dictionary

2

Task Two:

- Falcon 1 launches were removed (to have dataframe containing only Falcon 9 info)
- Data Wrangling – missing values found for dataset using isnull().sum()

3

Task Three:

- Missing data for PayloadMass (np.nan) was replaced (.replace()) with the average (.mean())

Github - API Data Collection Reference

DATA COLLECTION - SCRAPING

Task One:

HTTP Get method used to request Falcon9 Launch HTML page

Beautiful Soup object created from HTML response

Verified BeautifulSoup object with soup.title

Task Two:

Tables were stored to a variable using soup.find_all('table')

A list of column names was created by iterating through the elements of table headers

Task Three:

Data added to dictionary through appending data from iterations.

Dataframe created from the dictionary

Github - Web Scraping Reference

9

Data Wrangling

Task 1	The number of launches per site found using df['LaunchSite'].value_counts()
Task 2	The number of each orbit found using df['Orbit'].value_counts()
Task 3	The quantity of each type of landing outcome stored as landing_outcomes Bad_outcomes created by iterating through landing_outcomes and adding landing failures
Task 4	Bad outcomes were converted to 0, while the remaining outcomes were converted to 1. This allowed storage by classification Success rate determined to be 67% based on df['Class'].mean()

- Github - Data Wrangling Reference

EDA WITH DATA VISUALIZATION

Charts utilized:

1. Scatter plots to show the relationship between variables
   - Payload Mass vs Flight Number
   - Launch Site vs Flight Number
   - Payload Mass vs Launch Site
   - Orbit Type vs Flight Number
   - Payload vs Orbit Type
2. Bar Charts to show comparisons of categories
   - Payload Mass vs Launch Site
   - Success Rate by Orbit Type
3. Line charts to show data trends
   - Yearly Launch Success Rate

Github - EDA with Data Visualization Reference

EDA WITH SQL

SQL Queries performed in this section:

- Display names of unique launch sites
- Display first 5 records of launch sites beginning with 'CCA'
- Display total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date where the successful landing outcome in drone ship was achieved
- List the names of the boosters which have success in ground pad and have payload greater than 4000, but less than 6000
- List the total number of successful failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass using a subquery
- List the records which will display the month names, successful landing_outcomes in ground pad, booster versions, launch_site for the months in year 2017.
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Github - EDA with SQL Reference

BUILD AN INTERACTIVE MAP WITH FOLIUM

Task One:

Mark all launch sites on a map

- NASA Johnson Space Center used as the starting point
- A circle, marker, popup label, and text labels were used for visibility
- Additional markers were added by iterating through launch sites

Task Two:

Mark successful/failed launches for each site on map

- Green was used for successful missions
- Red was used for unsuccessful missions

Task Three: Calculate distance between launch site to its proximities

Lines were added to show distances between launch site and proximities

Github - Visual Analytics with Folium Reference

BUILD A DASHBOARD WITH PLOTLY DASH

Interactions utilized:

- Launch Sites Dropdown
  - Used for easy selection of sites
- Successful Launch Pie Chart
  - Used to easily demonstrate success vs. failures by site
- Payload Mass Slide
  - Used to select payload range
- Payload vs. Success Rate Scatter Chart
  - Used to show correlation between payload and success rate

PREDICTIVE ANALYSIS (CLASSIFICATION)

Github - Predictive Analysis Reference

RESULTS

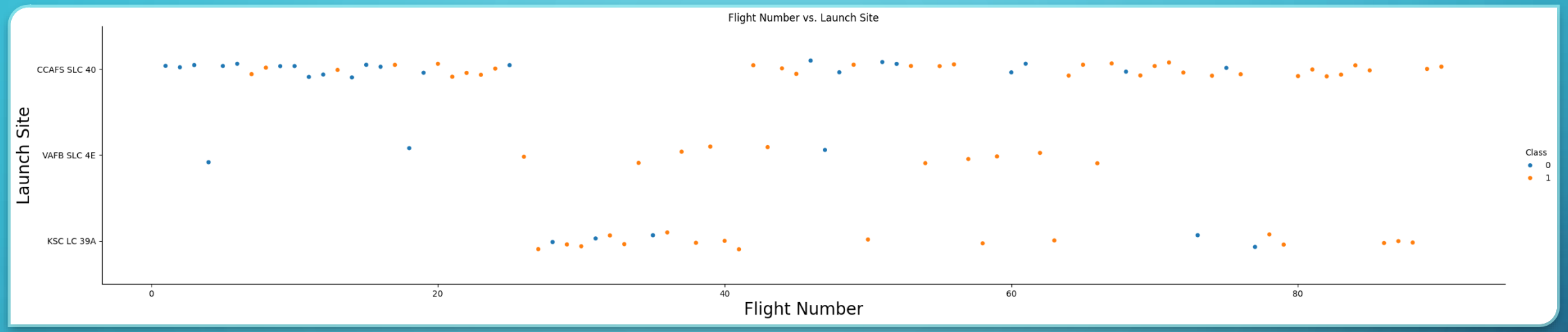
EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

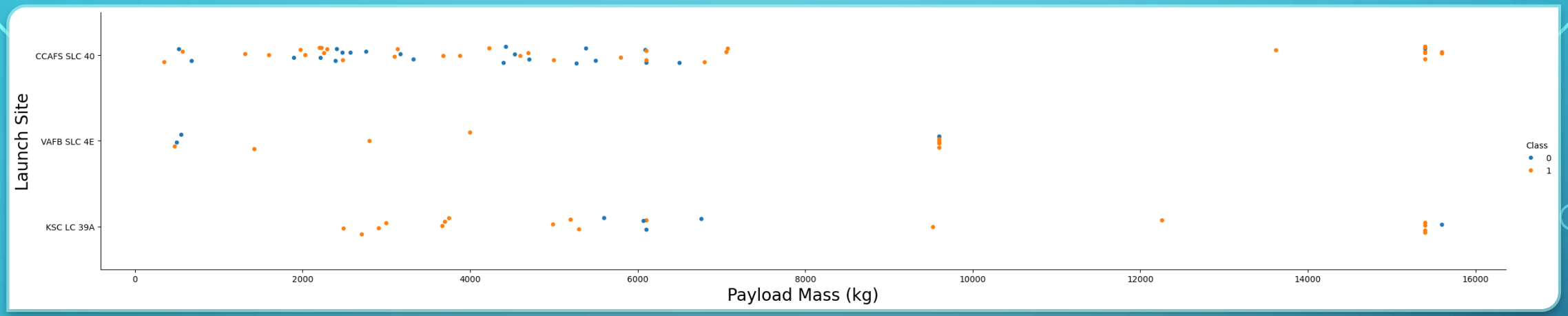
Section 2

Insights drawn from EDA



FLIGHT NUMBER VS. LAUNCH SITE

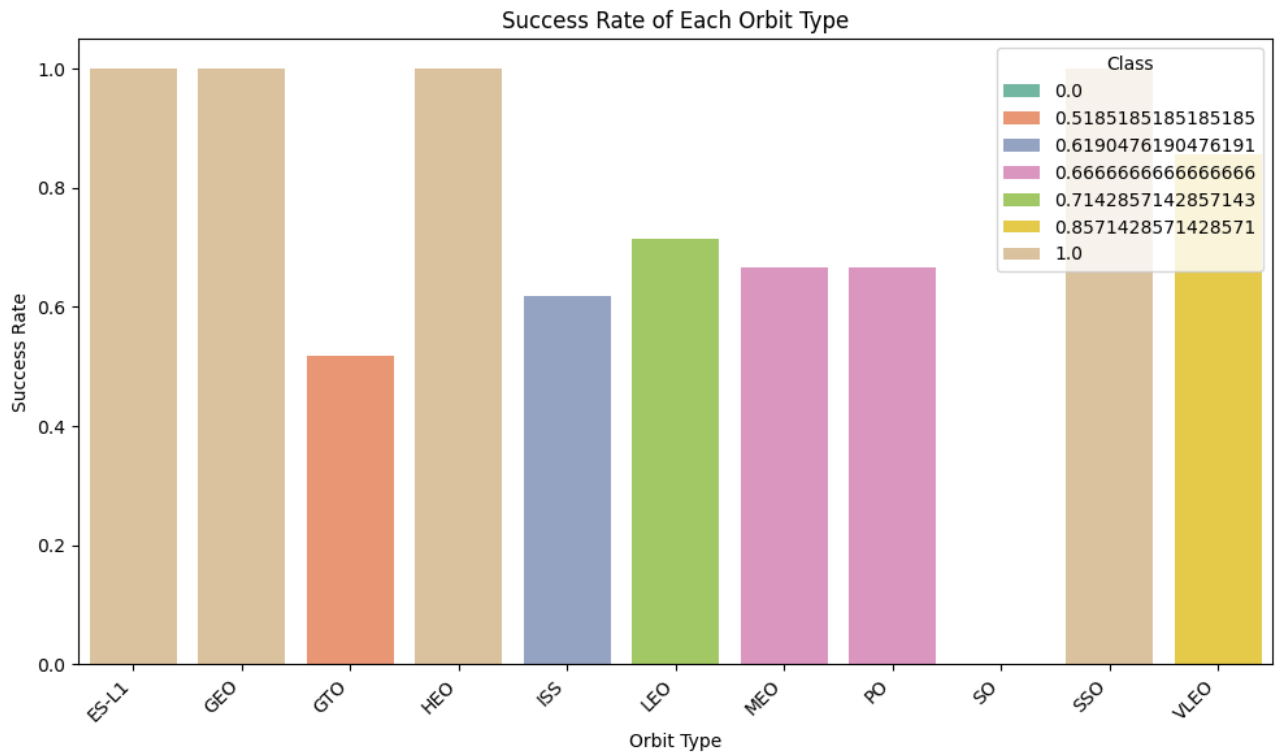
- CCAFS SLC 40 has the most flights
  - Most of the early flights of SLC 40 failed, but most of the recent flights have been successful
- VAFB SLC 34 has the least flights
  - The first 2 flights failed, but most flights have succeeded since
- Overall, newer launches have a higher success rate



PAYLOAD VS. LAUNCH SITE

- KSC LC 39A has 100% success rate for payload masses below 5000 kg
- Almost all launches over 8000 kg were successful
- Higher success rates correlate with higher payload masses across all sites

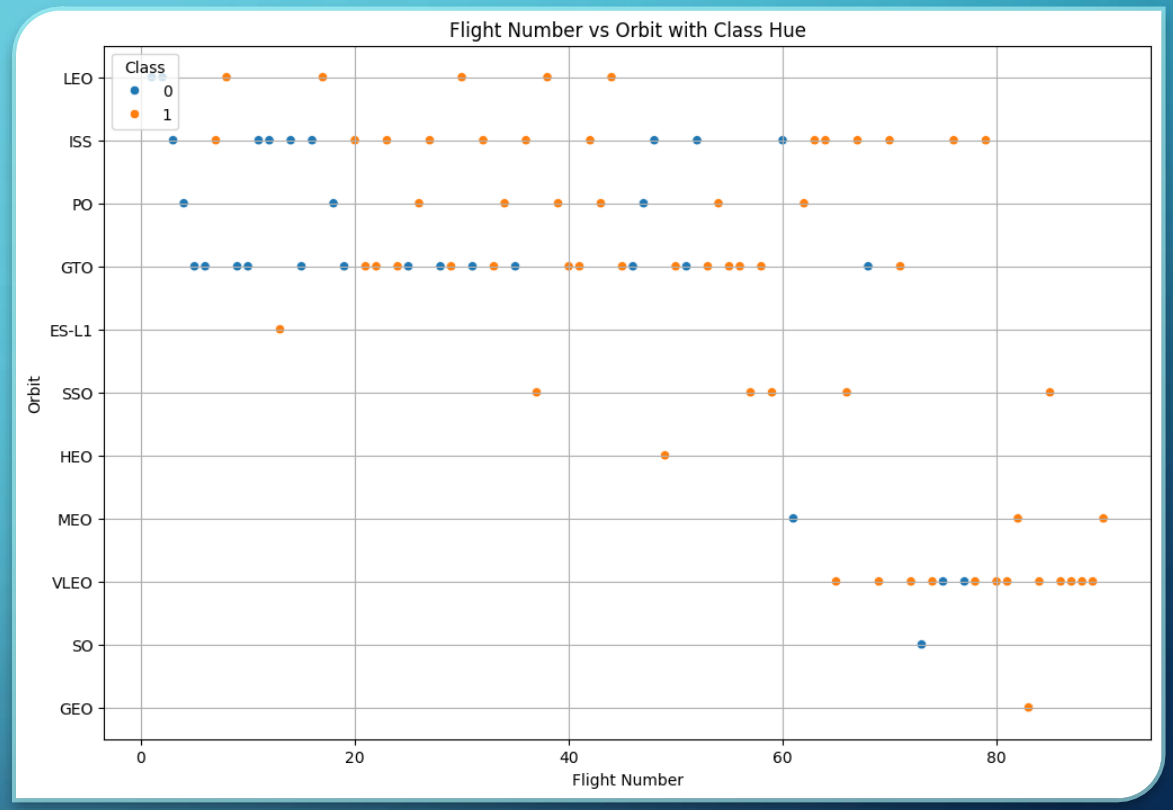
SUCCESS RATE VS. ORBIT TYPE



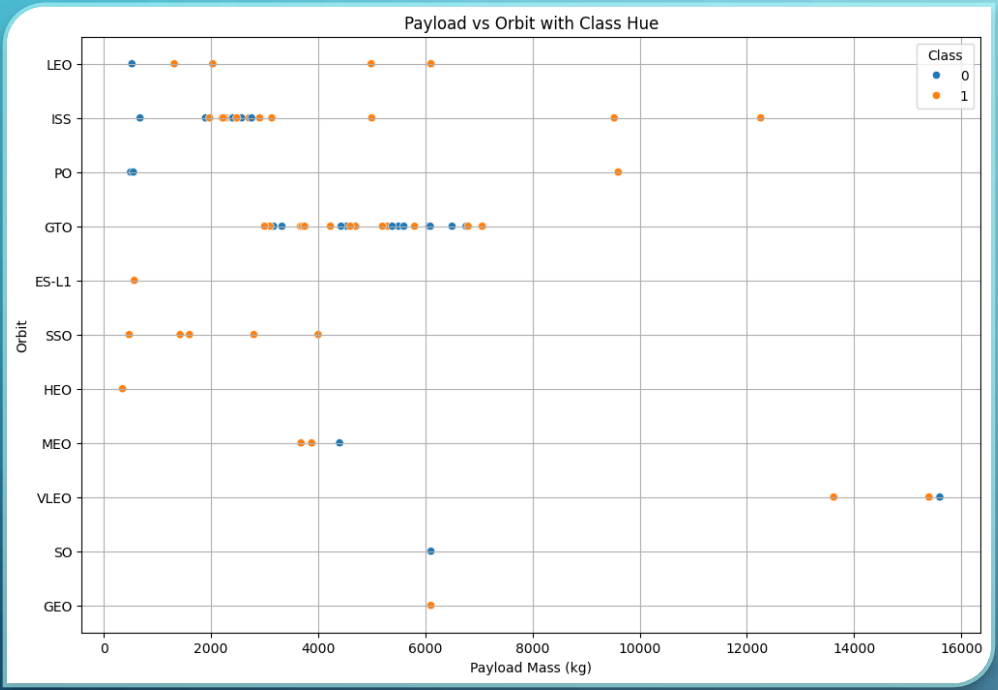
- ES-L1, GEO, HEO, and SSO orbits have 100% success rate
- SO orbits have 0% success rate

FLIGHT NUMBER VS. ORBIT TYPE

- There appears to be a correlation between number of flights and success rate for LEO orbits.
- There does not appear to be a strong correlation between the number of flights and success rate for GTO orbits.

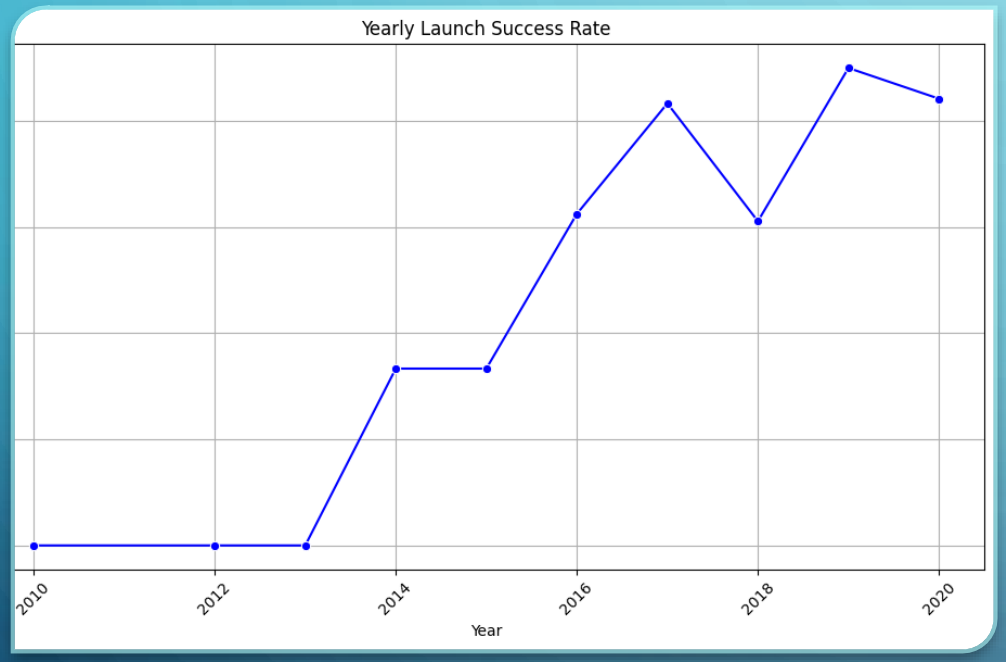


PAYLOAD VS. ORBIT TYPE



- ES-L1 and HEO only have data for payload masses under 2000 kg
- SSO has 100% success rate, and all of the payloads are 4000 kg or less
- SO has no successful orbits in the dataset
- LEO, ISS, and PO appear to be positive correlated with payload mass
- GTO orbits appear to be negatively correlated with payload mass

LAUNCH SUCCESS YEARLY TREND



- The success rate of launches has a positive overall trend 2013-2020
- The highest success rate was in 2019.
- 3 of the last 4 years have success rates over 80%

ALL LAUNCH SITE NAMES

A list of unique names of launch sites was pulled from the spacex table with the query.

Task 1

Display the names of the unique launch sites in the space mission

```
%sql select distinct launch_site from SPACEXTABLE;
```

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

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

LAUNCH SITE NAMES BEGIN WITH 'KSC'

Displays the first 5 records where launch sites begin with the string 'KSC'

Task 2

Display 5 records where launch sites begin with the string 'KSC'

```
%sql select * from SPACEXTABLE 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

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTABLE where customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

Displays the total payload mass carried by boosters launched by NASA (CRS)

AVERAGE PAYLOAD MASS BY F9 V1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTABLE where booster_version like '%F9 v1.1%';
```

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

average_payload_mass
2534.6666666666665

Displays average payload mass carried by booster version F9 v1.1

FIRST SUCCESSFUL GROUND LANDING DATE

Task 5

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

```
%sql select min(date) as first_successful_landing from SPACEXTABLE where landing_outcome = 'Success (ground pad)';
```

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

first_successful_landing
2015-12-22

Finds the dates of the first successful landing outcome on drone ship. Present your query result with a short explanation here

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

Task 6

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
%sql select booster_version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000
```

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

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

- Lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- %sql select booster_version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

Task 7

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(*) as total_number from SPACEXTABLE group by mission_outcome;
```

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

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

Lists the total number of successful and failure mission outcomes

BOOSTERS CARRIED MAXIMUM PAYLOAD

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql select booster_version from SPACEXTABLE where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTABLE)
```

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

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

Lists the names of the booster_versions which have carried the maximum payload mass using a subquery

2015 LAUNCH RECORDS

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.

```
%%sql SELECT
    strftime('%m', date) AS month,
    date,
    booster_version,
    launch_site,
    landing_outcome
FROM
    SPACEXTABLE
WHERE
    landing_outcome = 'Failure (drone ship)'
    AND strftime('%Y', date) = '2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Lists the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017.

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-0
in descending order

```
%%sql select landing_outcome, count(*) as count_outcomes from SPACEXTABLE
      where date between '2010-06-04' and '2017-03-20'
      group by landing_outcome
      order by count_outcomes desc;
```

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

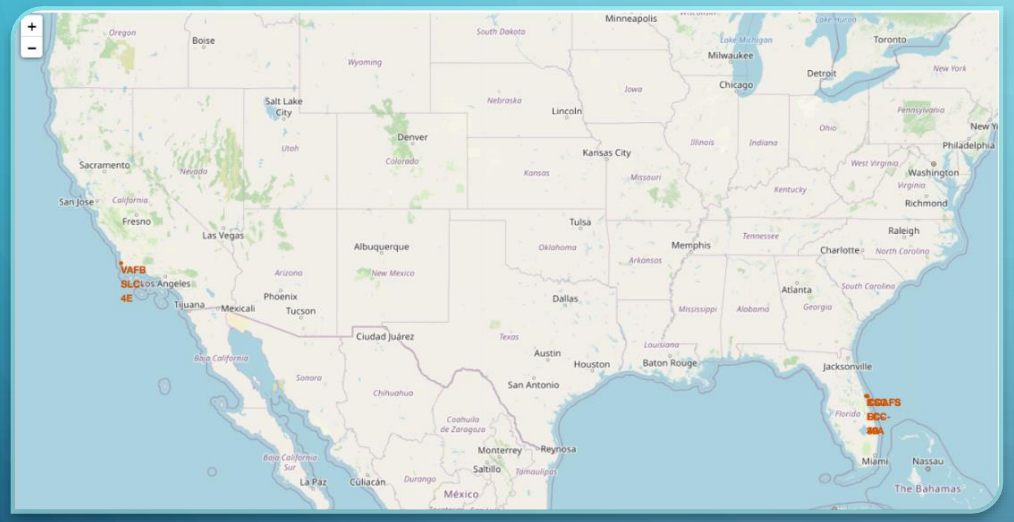
Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Ranks 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

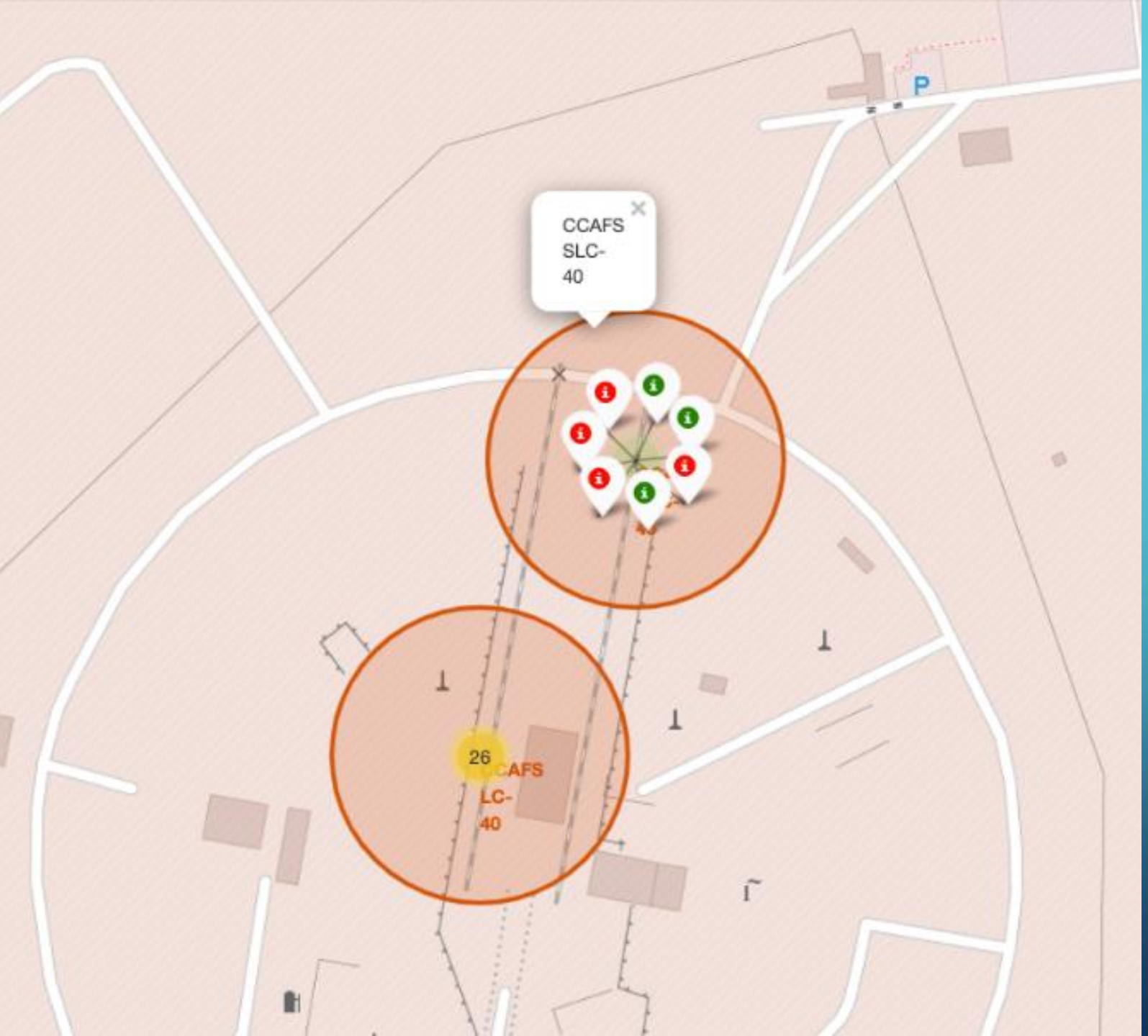
Section 3

Launch Sites Proximities Analysis

SPACEX LAUNCH SITES



- The launch sites are shown below with red markers and labels.
- Launch sites are in close to the equator to take advantage of the Earth's rotational speed. Since the Earth 's movement is faster near the equator, this speed assists with orbit.
- Launch sites are near the coasts to minimize debris in proximity to populated regions. This is also part of why some landings are attempted on the ocean, rather than on land.

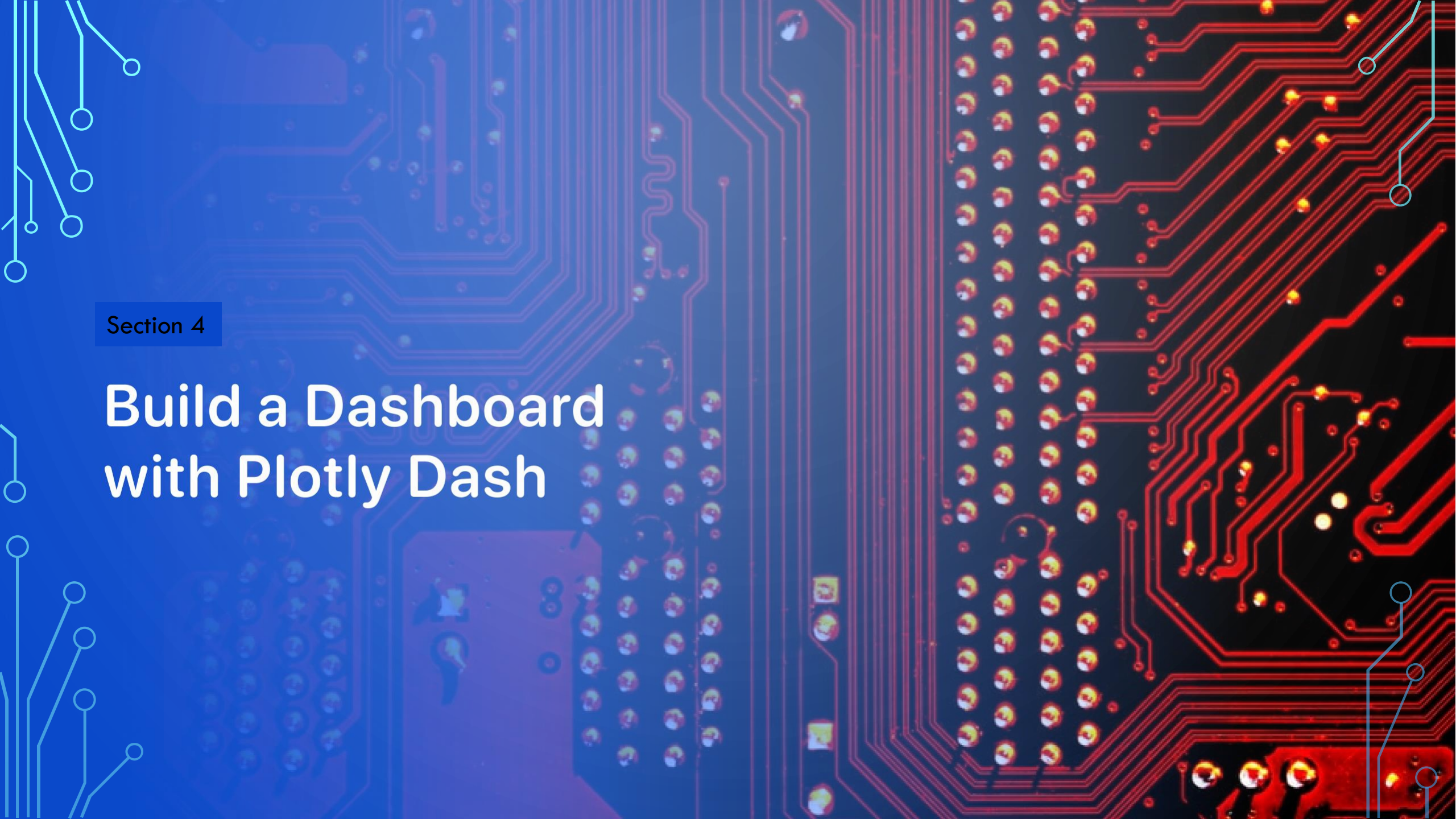


COLOR CODED LAUNCHES

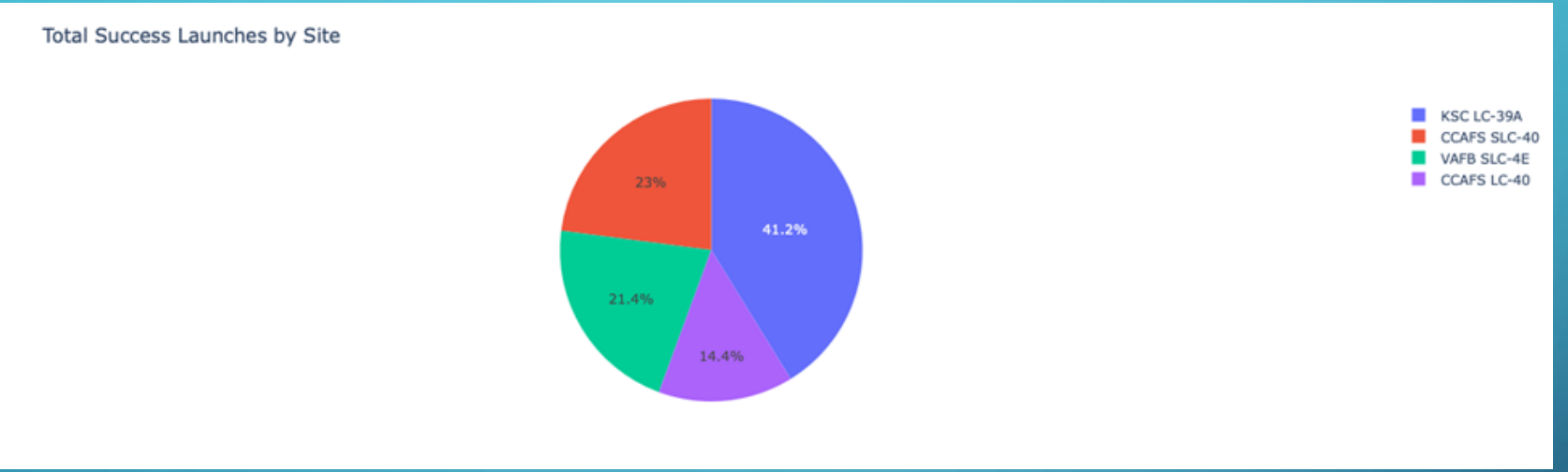
- Launches were color coded to show which areas are the most or least successful.
- Green markers indicate successful launches
- Red markers indicate unsuccessful launches
- Launch site KSC LC-39A has the highest success rate

PROXIMITY LINES

- Lines were added to show proximity to different points. In the example on the left, a proximity line was added to the coastline to show distance from the ocean.

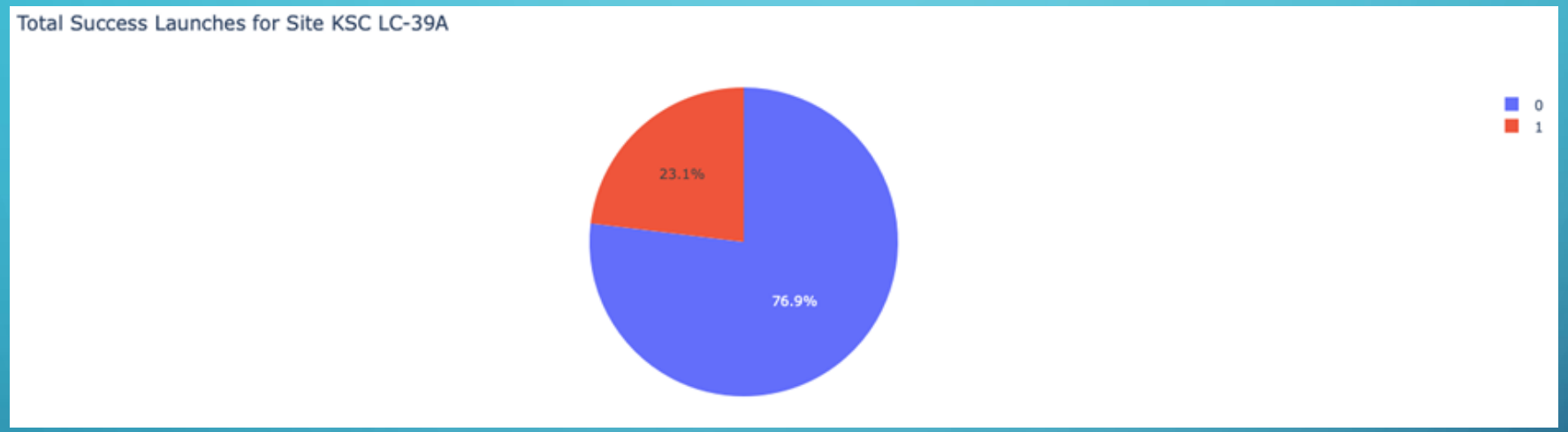

Section 4
Build a Dashboard with Plotly Dash

LAUNCH SUCCESS BY SITE



KSC LC-39A has the most successes based on the chart

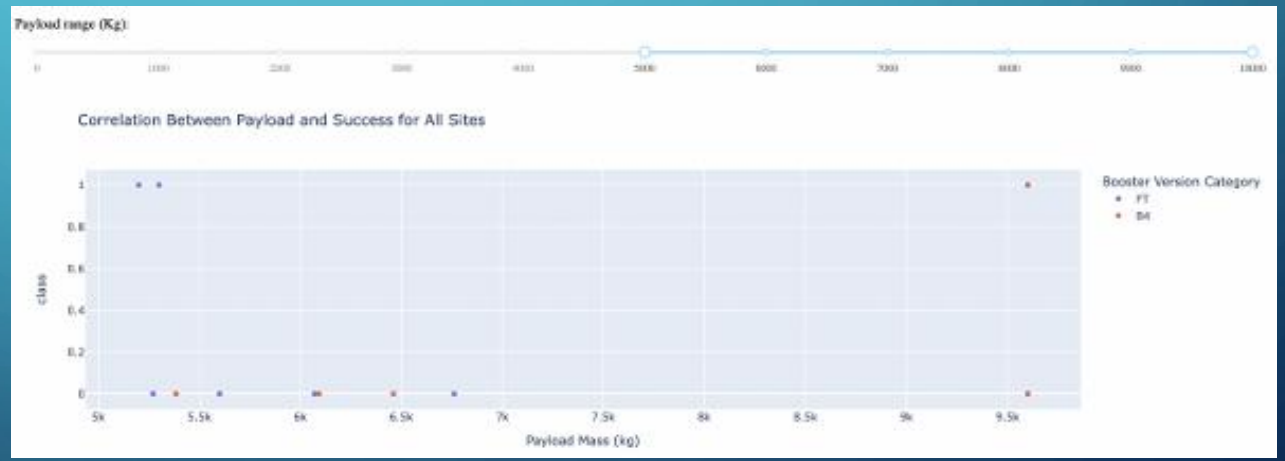
LAUNCH SITE WITH HIGHEST SUCCESS RATE



KSC LC-39A has the highest success rate (76.9%).

LAUNCH SUCCESS BY SITE

The highest success rate for launches is between 2000 kg and 4000 kg based on the graph.

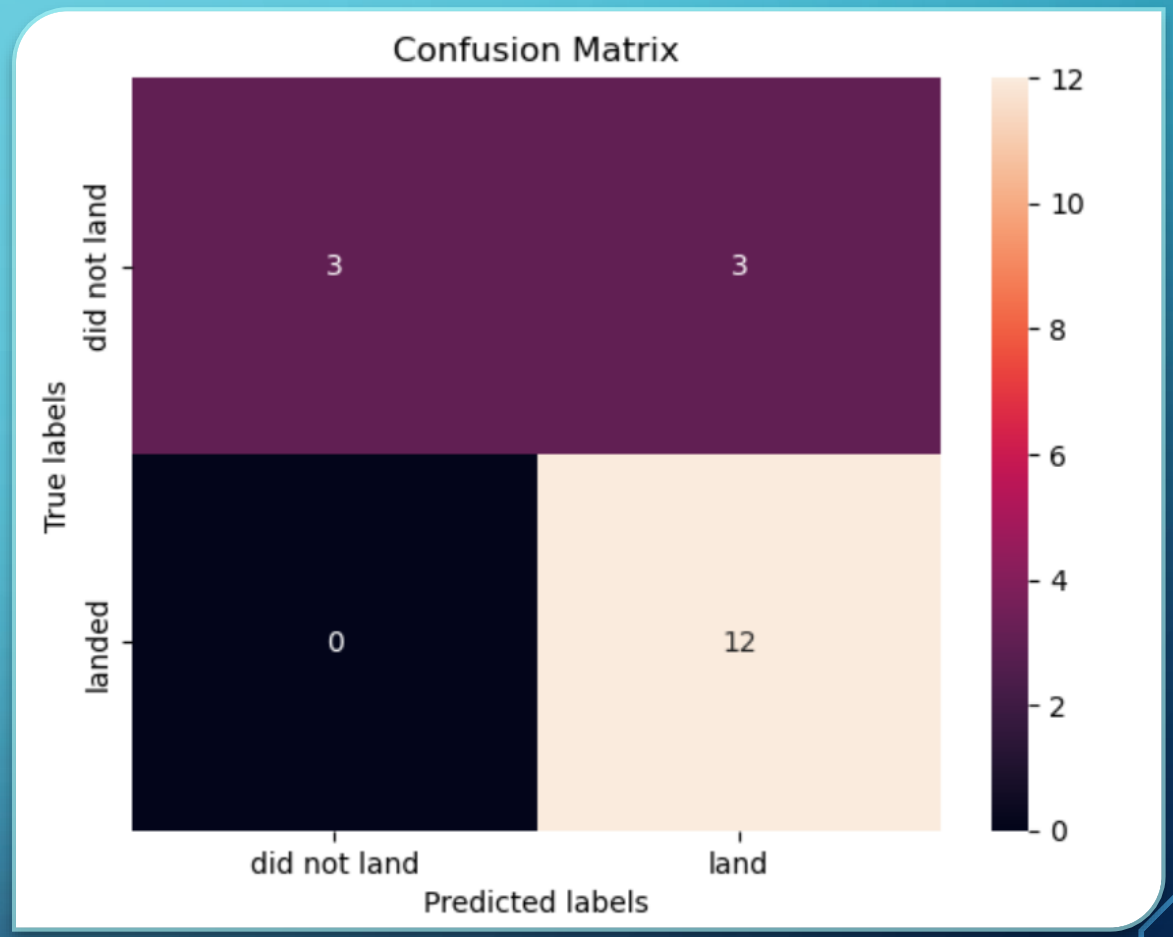


Section 5

Predictive Analysis (Classification)

CONFUSION MATRIX

The confusion matrix shows the logistics regression between classes. In the confusion matrix for this example, the biggest issue for the model is false positives.



CONCLUSIONS

- The orbits with 100% success rate are ES-L1, GEO, HEO, SSO
- KSC LC-39A has the highest success rate
- Launches with lower payload tend to have higher success rates
- Success rates increase over time
- Most launches are near the equator

APPENDIX

Labs were completed in IBM sponsored Jupyter lab, and course content was provided by IBM through EdX.

Thank you!