IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sonya Lawrence
07/31/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using an API
  - Data Collection through Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis Using SQL
  - Exploratory Data Analysis using Data Visualization
  - Interactive Visual Analytics using Folium
  - Machine Learning Model Building and Predictions
- Summary of all results
  - Exploratory data analysis results presented using visual aids
  - The optimal machine learning model was acquired
  - Interesting insights gained from data

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.

- Problems you want to find answers

1. What factors determine if the rocket will land successfully?
2. What is the likelihood that SpaceX will reuse the first stage of the rocket?

Section 1

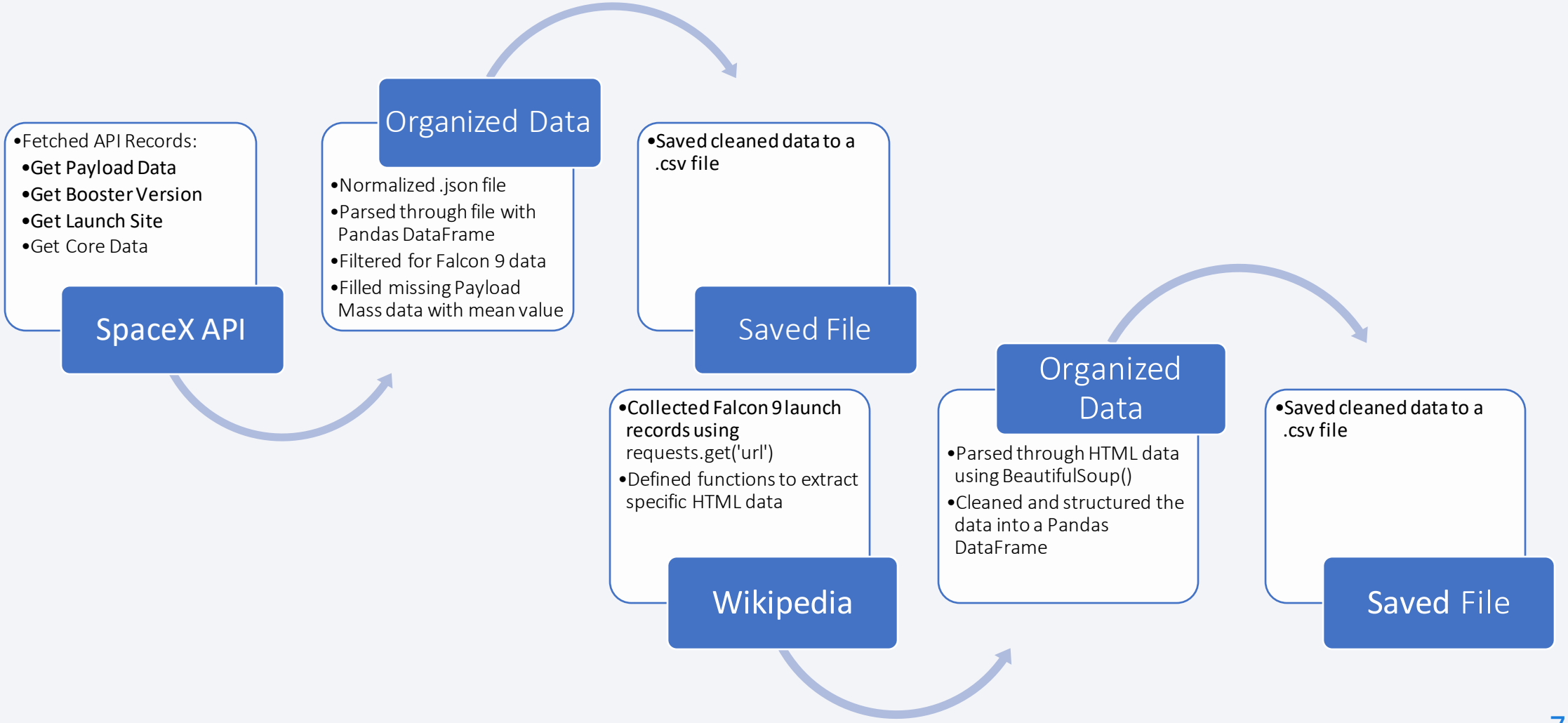
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API and Web scraping from Wikipedia
- Perform data wrangling
  - Cleaned data accounting for missing values
  - One-hot encoding applied to categorical data for transformation
- 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 – SpaceX API

- Several get requests were used to extract the data needed from the API obtained from: https://api.spacexdata.com/v4
- This data was then cleaned and wrangled for purposes of this project.
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/SpaceX%20Assessment%20(Data%20Cleaning).ipynb

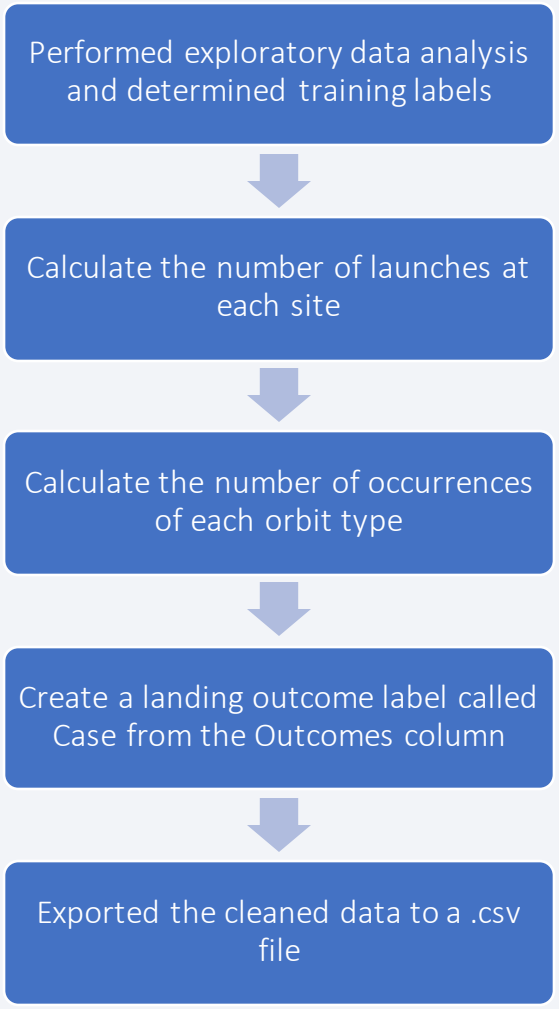
```
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])

def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])

def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])

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

Data Collection - Scraping

- A Python BeautifulSoup object was created from the SpaceX Falcon9 data scraped from: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- This data was further wrangled and structured into a Pandas DataFrame for the purposes of this project.
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/SpaceX%20Assessment%20(Web%20Scraping).ipynb

```
def date_time(table_cells):
    return [data_time.strip()
            for data_time in list(table_cells.strings)][0:2]

def booster_version(table_cells):
    out=''.join([booster_version
                 for i,booster_version in enumerate( table_cells.strings)
                 if i%2==0][0:-1])
    return out

def landing_status(table_cells):
    out=[i for i in table_cells.strings][0]
    return out

def get_mass(table_cells):
    mass=unicodedata.normalize("NFKD", table_cells.text).strip()
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find("kg")+2]
    else:
        new_mass=0
    return new_mass

def extract_column_from_header(row):
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.contents)
    if not(colunm_name.strip().isdigit()):
        colunm_name = colunm_name.strip()
        return colunm_name
```

Data Wrangling



- 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.
- We converted those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/SpaceX%20Assessment%20(Data%20Wrangling).ipynb

EDA with Data Visualization

- Charts plotted:
  - Flight number vs. Payload mass, Flight number vs. Launch site, Payload mass vs. Launch site, Success rate of each original, Flight number vs. Orbit type, Payload mass vs. Orbit type, and Average launch success trend.
- Chart types used:
  - Scatter plots show the relationship between variables.
  - Bar charts comparison among discrete categories.
  - Line graphs show trends in data over time
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/Spacex%20Assessment(Data%20Visualization).ipynb

EDA with SQL

- The SpaceX dataset was loaded into Jupiter notebook the built-in SQLITE3 module
- Exploratory data analysis:
  - Unique launch sites and Launch sites beginning with 'KSC'
  - Total payload mass for NASA (CRS)
  - Average payload mass by f9 v1.1
  - Date of first successful drone ship landing
  - Boosters with payload mass between 4000 and 6000
  - Total number of successful and failed missions
  - Booster versions carrying the max payload
  - Successful landing outcomes by unique categories
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/Spacex%20Assessment(data%20analysis-sqllite).ipynb[12]

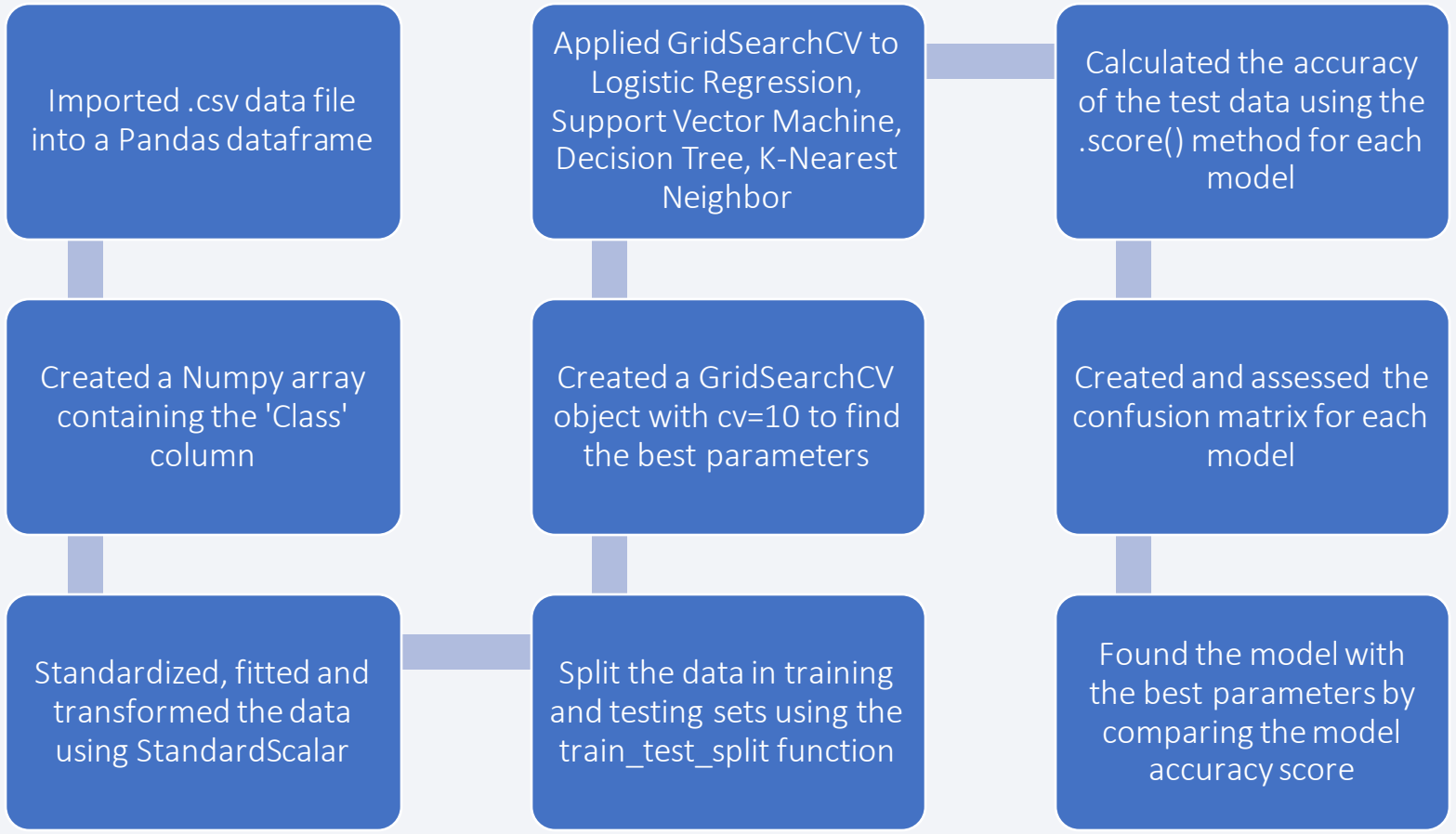
Build an Interactive Map with Folium

- All launch sites were marked using circles and labeled
- Failed launch sites were colored red and successful launch sites were colored green
- Using the color labeled marker clusters, the launch sites having relatively high success rates were identified
- The distance from several landmarks to the launch site was calculated and marked using lines. The distance calculated is shown on the map.
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/Spacex%20Assessment(Visual%20Analytics).ipynb

Build a Dashboard with Plotly Dash

- Launch site drop-down list:
  - Added a drop down list to enable launch site selection
- Pie chart showing successful launches:
  - Added an interactive pie chart giving users the ability to view the count of successful and failed launches from each site and overall
- Slider of payload mass range
  - Added a slider to select payload mass range to view
- Scatter plot of payload mass and success rate per booster version:
  - Added a scatter plot showing the correlation between payload mass and anchor success
- GitHub Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)



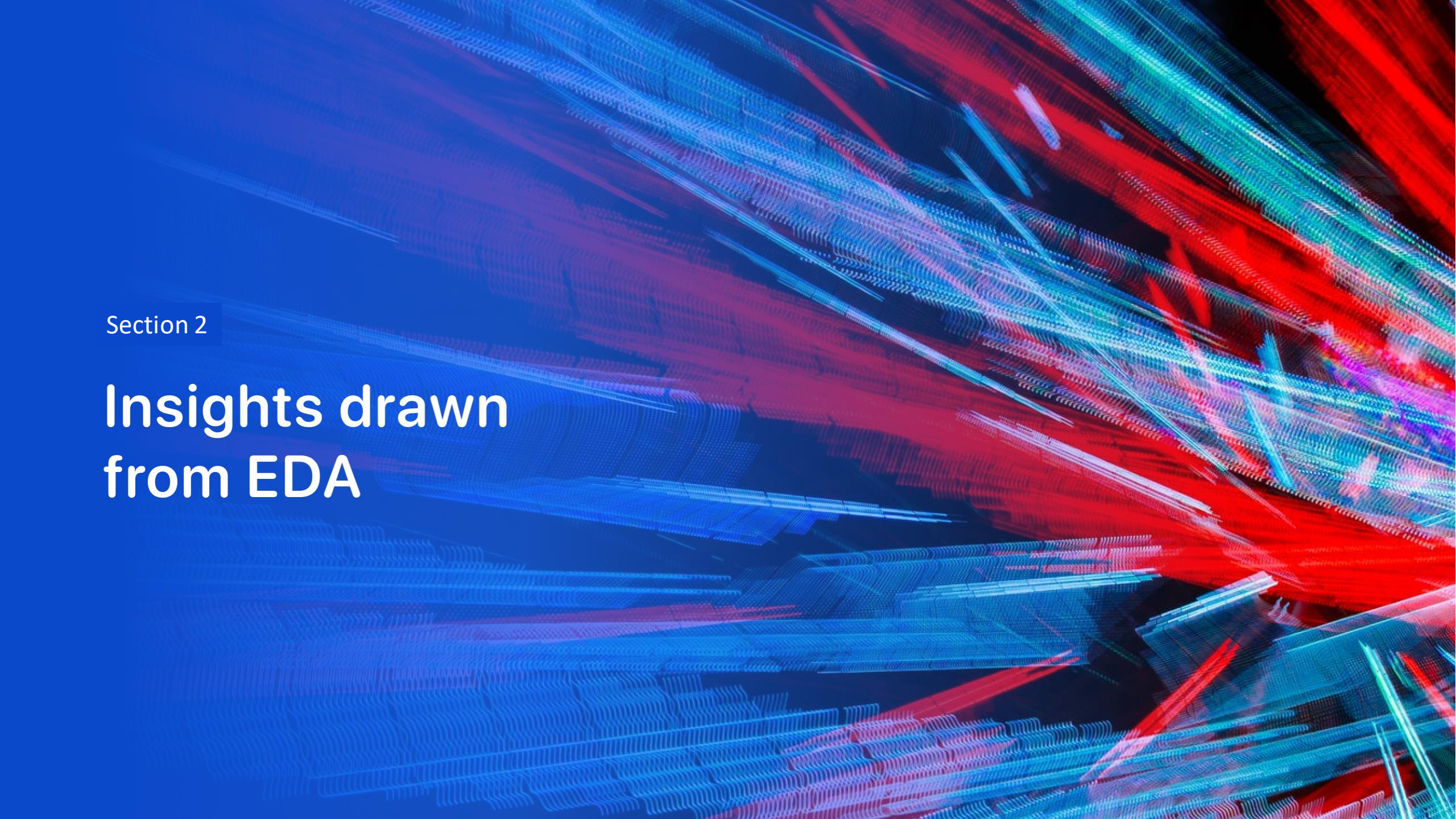
Github Repo Link: https://github.com/Sonya-7/DS_IBM_Capstone/blob/master/Spacex%20Assessment(Machine%20Learning%20Models).ipynb

Results

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

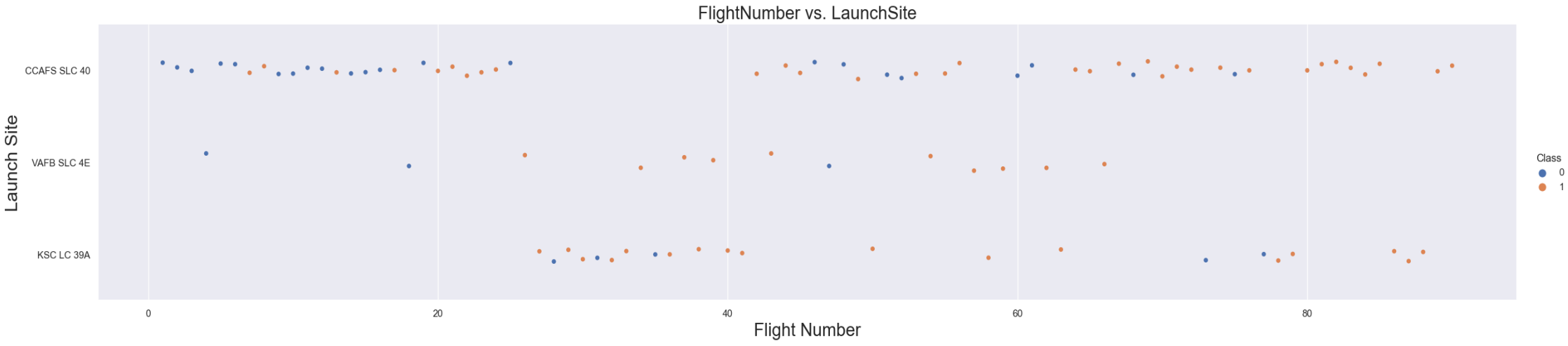
Section 2

Insights drawn from EDA



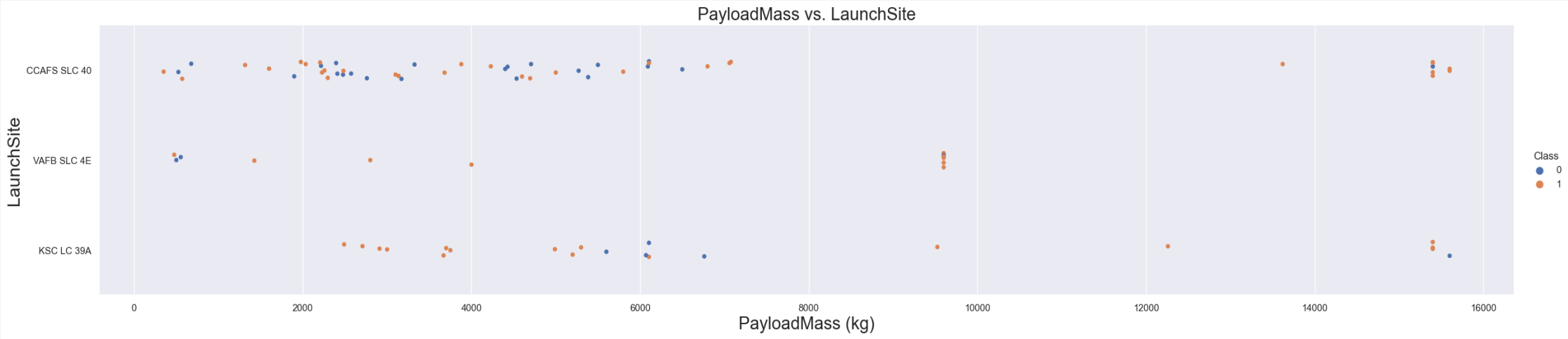
Flight Number vs. Launch Site

- The chart below shows that as the flight number increases, the first stage is more likely to land successfully.
- CCAFS LC-40 has a 60% success rate.
- Both KSC LC-39A and VAFB SLC 4E have a 77% success rate.



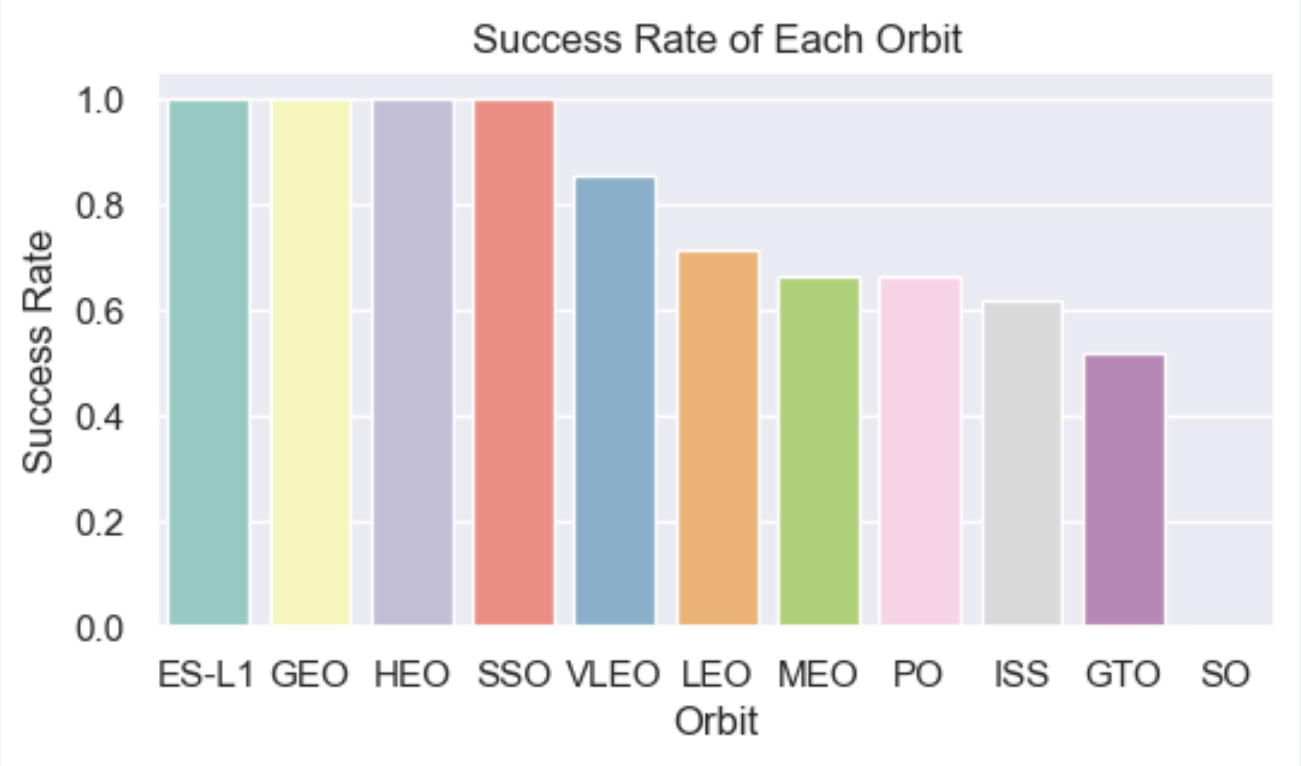
Payload vs. Launch Site

- The below chart shows that the higher the Payload mass, the higher the success rate for each launch site.
- For VAFB-SLC 4E, there are no rockets launched for payload mass over 10,000kg.
- KSC LC 39A has a 100% success rate for Payload mass under 5,500kg.
- More than half the flights were from the CCAFS SLC 40 launch site.



Success Rate vs. Orbit Type

- Orbits with 100% success rate:
  - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate:
  - SO
- Orbits with a 50% siccess rate:
  - GTO

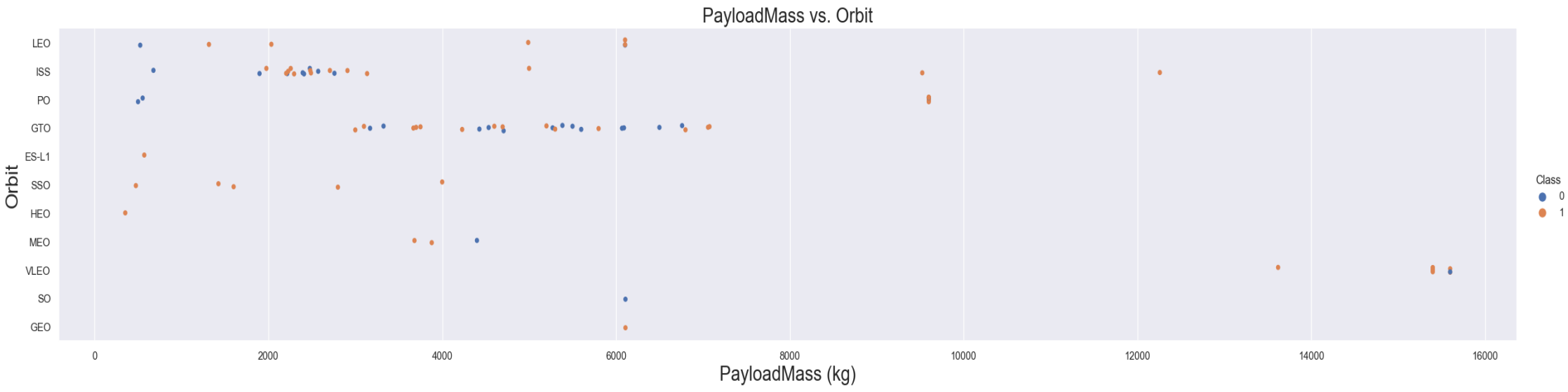


Flight Number vs. Orbit Type

- The chart below shows that:
  - the LEO orbit's success rate is directly related to the flight number
  - there is no relation between GTO orbit and flight number

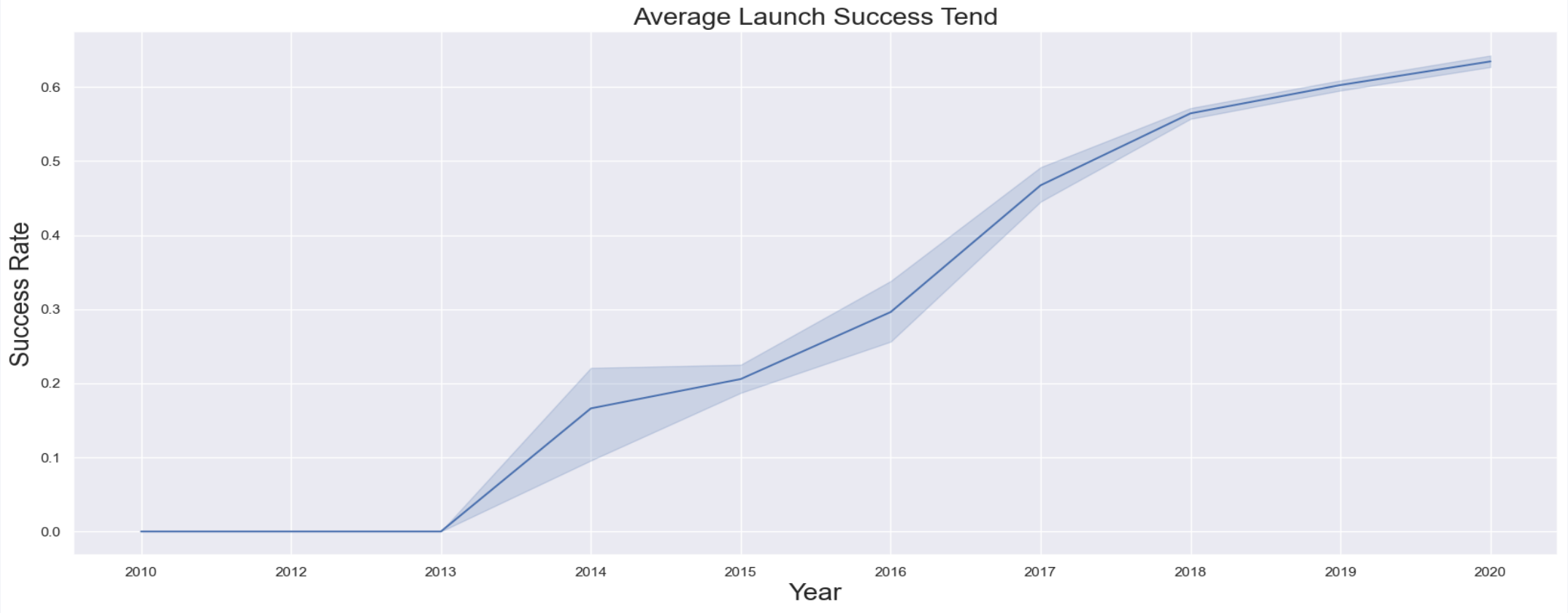
Payload vs. Orbit Type

- The below chart shows that with heavy payloads, the successful landing rates are more for LEO, ISS, and PO.



Launch Success Yearly Trend

- From the chart, it is evident that the success rate since 2013 has kept increasing until 2020.



All Launch Site Names

- Display the names of the unique launch sites in the space mission.
- The **DISTINCT()** key word was used to show the list of unique launch sites.

```
[8]: %%sql
     select distinct(Launch_Site)
     from spacextbl;

      * sqlite:///my_data1.db
     Done.
[8]:  Launch_Site
     ------------
      CCAFS LC-40
      VAFB SLC-4E
      KSC LC-39A
      CCAFS SLC-40
```

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- The **WHERE()** function was used to find unique launch site names

```
%%sql
select *
from spacextbl
where Launch_Site like 'KSC%'
limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- The **SUM()** and **WHERE()** functions were used to calculate this value

```
%%sql
select SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA(CRS)"
from spacextbl
where customer like 'NASA (CRS)';
```

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

Total Payload Mass by NASA(CRS)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The **AVG()** and **WHERE()** functions were used to calculate this value

```
%%sql
select AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by F9 v1.1"
from spacextbl
where Booster_Version like 'F9 v1.1';
```

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

Average Payload Mass by F9 v1.1
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.
- The **like** function was used to filter for this unique criterion

```
%%sql
select min(Date) as 'First Successful Drone Ship Landing'
from spacextbl
where "Landing _Outcome" like 'Success (drone ship)';
```

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

First Successful Drone Ship Landing
06-05-2016

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- The **like** and **and** functions were used to filter for these unique criteria

```
%%sql
select Booster_Version as 'Boosters with Payload Mass between 4000 and 6000'
from spacextbl
where "Landing _Outcome" like 'Success (ground pad)' and PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Boosters with Payload Mass between 4000 and 6000
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- A **JOIN** clause was used to combine the rows of the nested SQL query

```
%%sql
select "Successful Missions", "Failed Missions"
from(
(select count(*) as 'Successful Missions' from spacextbl where  Mission_Outcome like 'Success')
join
(select count(*) as 'Failed Missions' from spacextbl where  Mission_Outcome like 'Failure%')
);
```

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

Successful Missions	Failed Missions
98	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- A subquery was used to filter for the specific criteria in the result

```
%%sql
select Booster_Version AS 'BOOSTERS CARRYING THE MAX PAYLOAD'
from spacextbl
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_)
                           from spacextbl)
ORDER BY Booster_Version ASC;
```

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

BOOSTERS CARRYING THE MAX PAYLOAD
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

2017 Launch Records

- 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
- A **WHERE()** clause containing **like** and **and** filters was used to parse through the database to solve this problem

```
%%sql

select substr(Date, 4, 2) as Month, "Landing _Outcome", Booster_Version, Launch_Site
from spacextbl
where Date like '%2017' and "Landing _Outcome" like 'Success (ground pad)';
```

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

Month	Landing _Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order
- The landing outcomes and count of landing outcomes was used in conjunction with a **WHERE()** clause to filter the landing outcomes by this date range
- The **GROUP BY()** and **ORDER BY()** functions were used to order the results of this query.

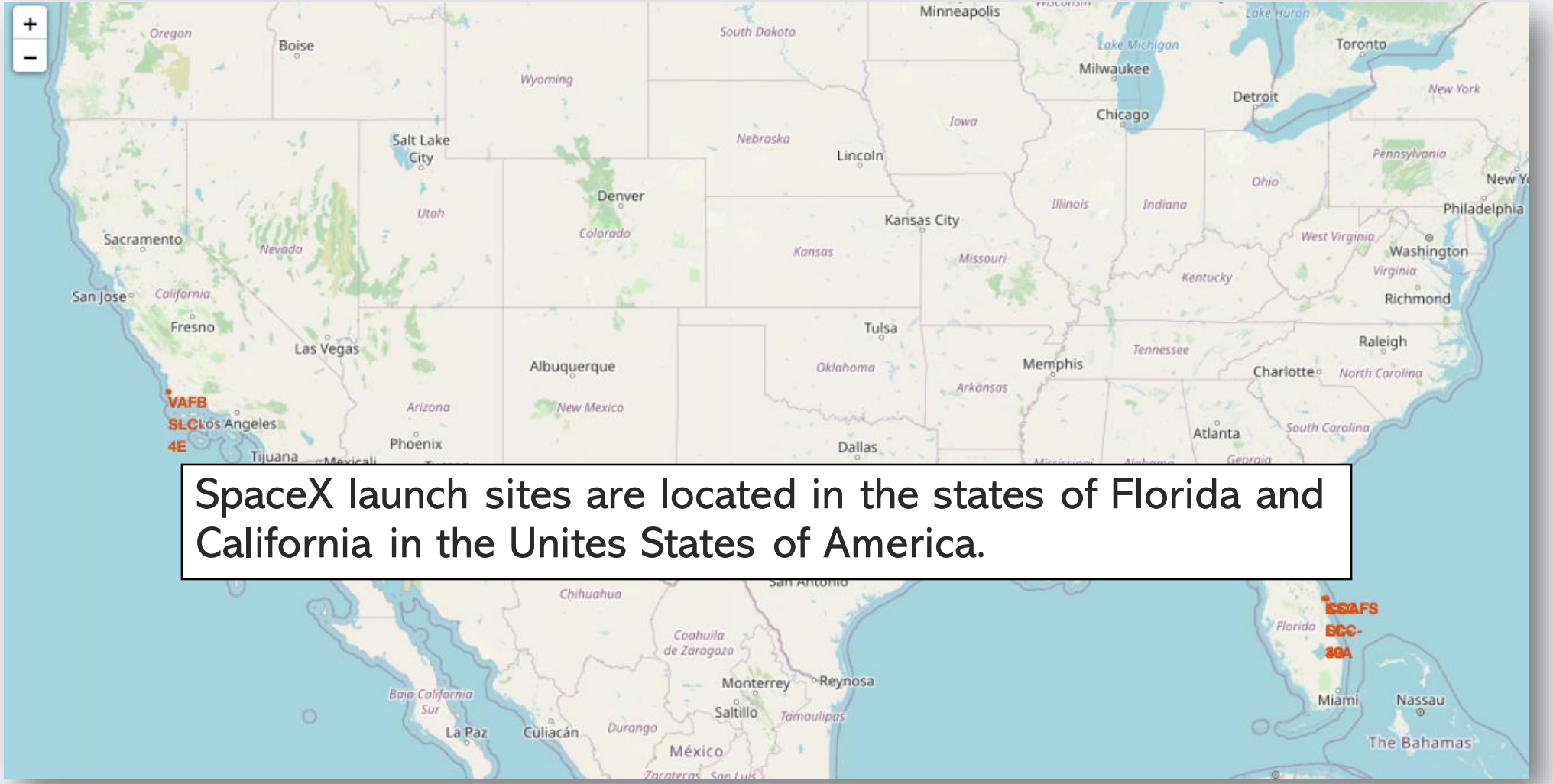
```
%%sql
select "Landing _Outcome" as 'Landing Outcomes between 04-06-2010 and 20-03-2017', count(*) as count
from spacextbl
where  Date between '04-06-2010' and '20-03-2017' and "Landing _Outcome" like 'Success%'
group by "Landing _Outcome"
order by  "Landing _Outcome" desc;
```

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

Landing Outcomes between 04-06-2010 and 20-03-2017	count
Success (ground pad)	6
Success (drone ship)	8
Success	20

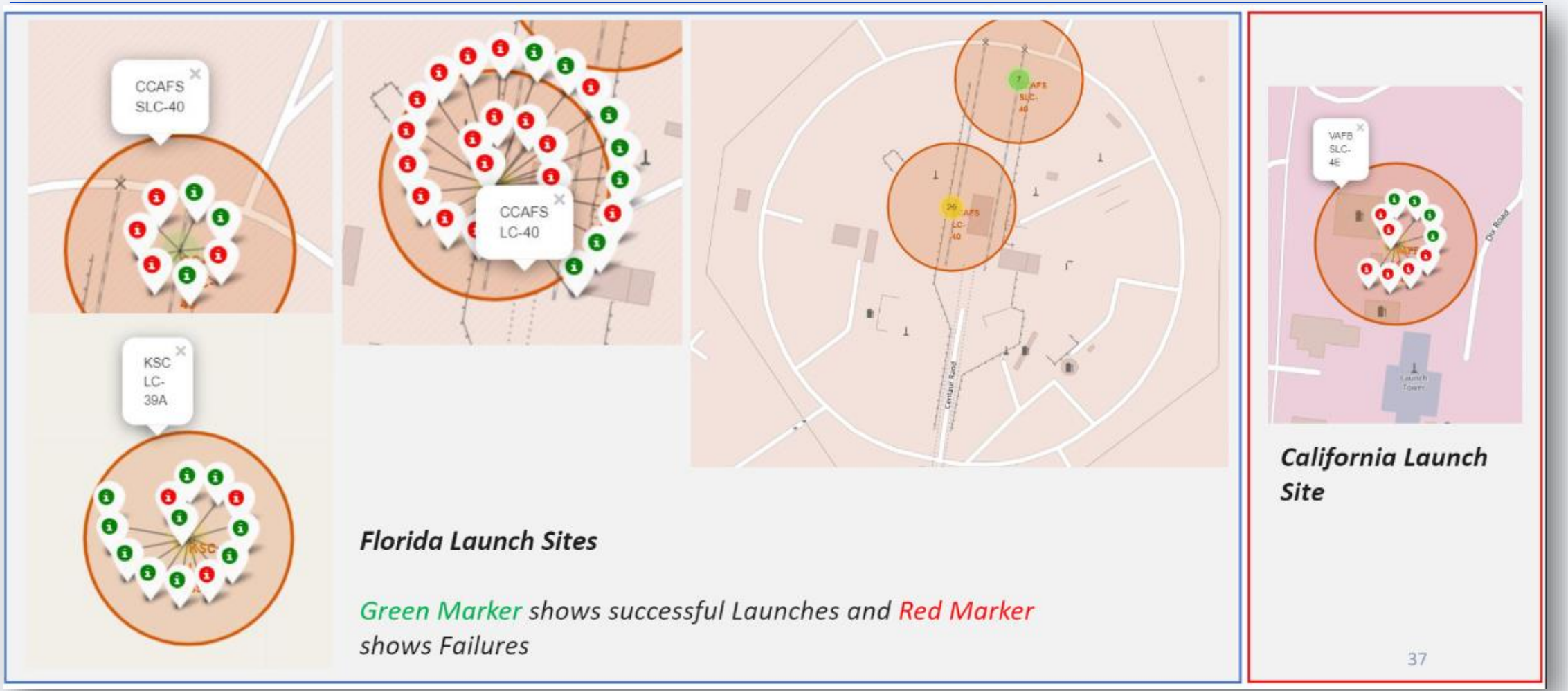

Section 3
Launch Sites
Proximities Analysis

Lauch Site Locations – Global Map

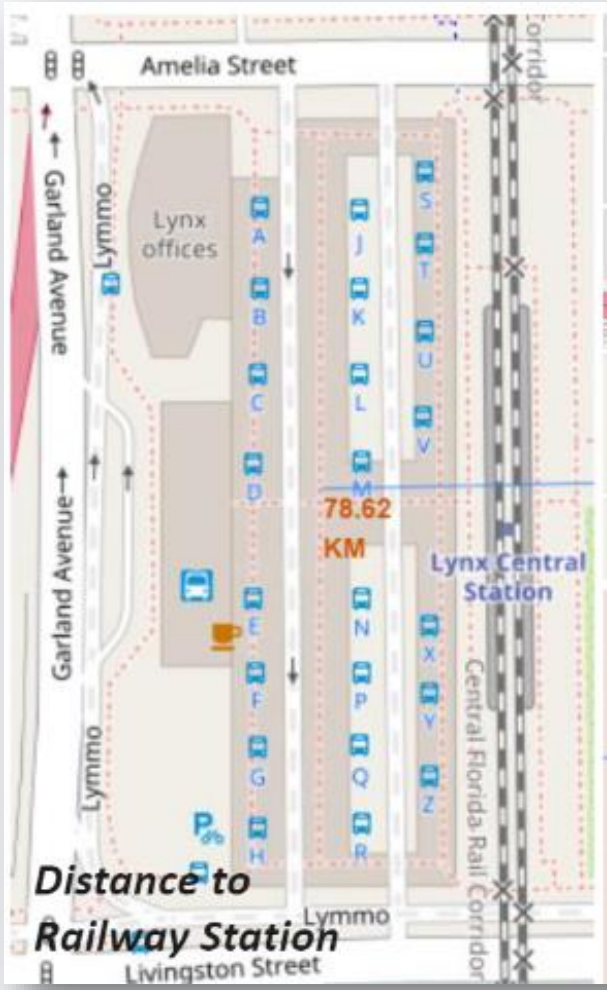


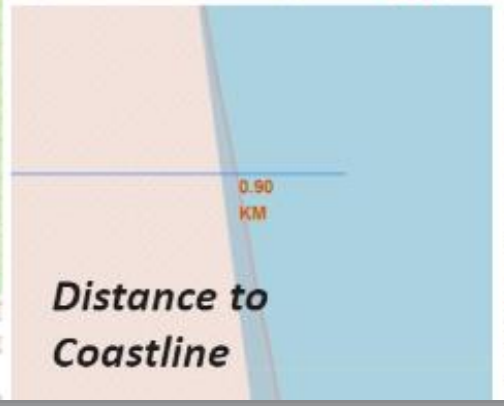
SpaceX launch sites are located in the states of Florida and California in the Unites States of America.

Color-Labeled Launch Sites



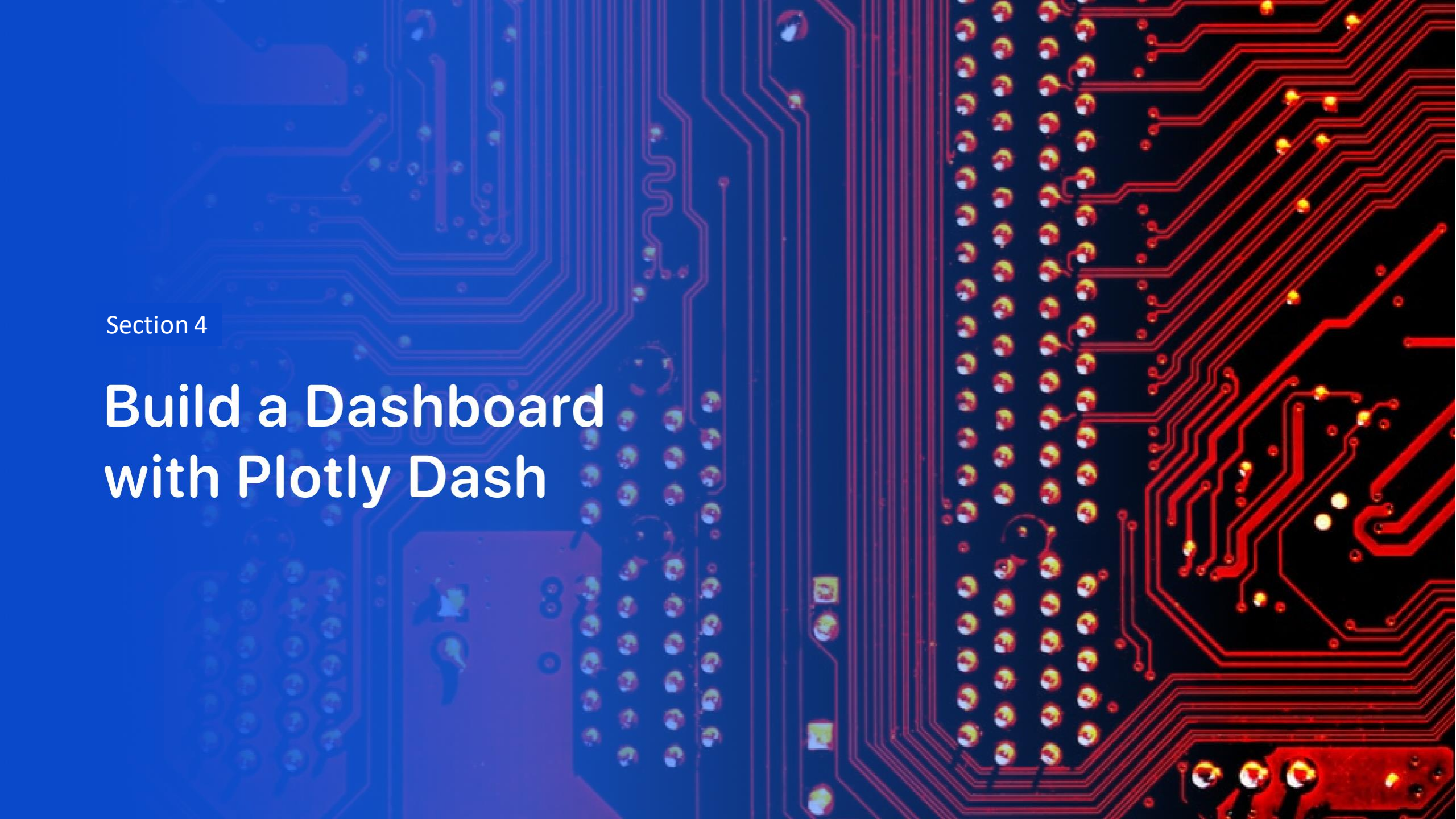
Distance from CCAFS SLC-40 to Landmarks



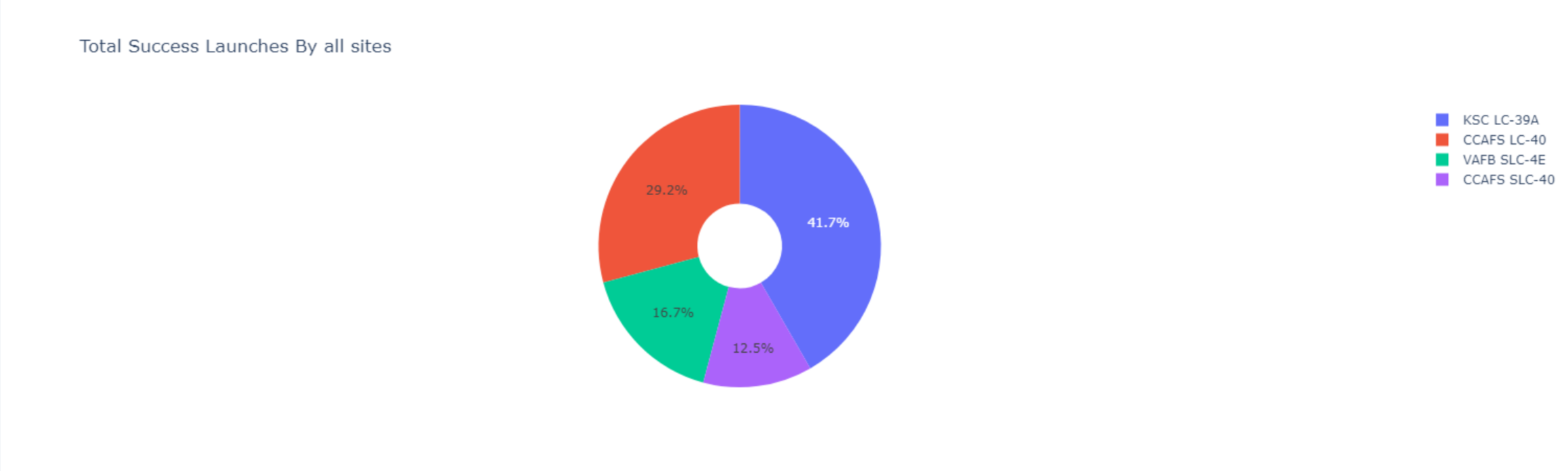


The distance between CCAFS SLC-40 and:

- the Coastline is 0.90 KM
- the Railway is 78.62 KM
- the Highway is 29.21 KM
- a major city(Orlando) is 78.45 KM

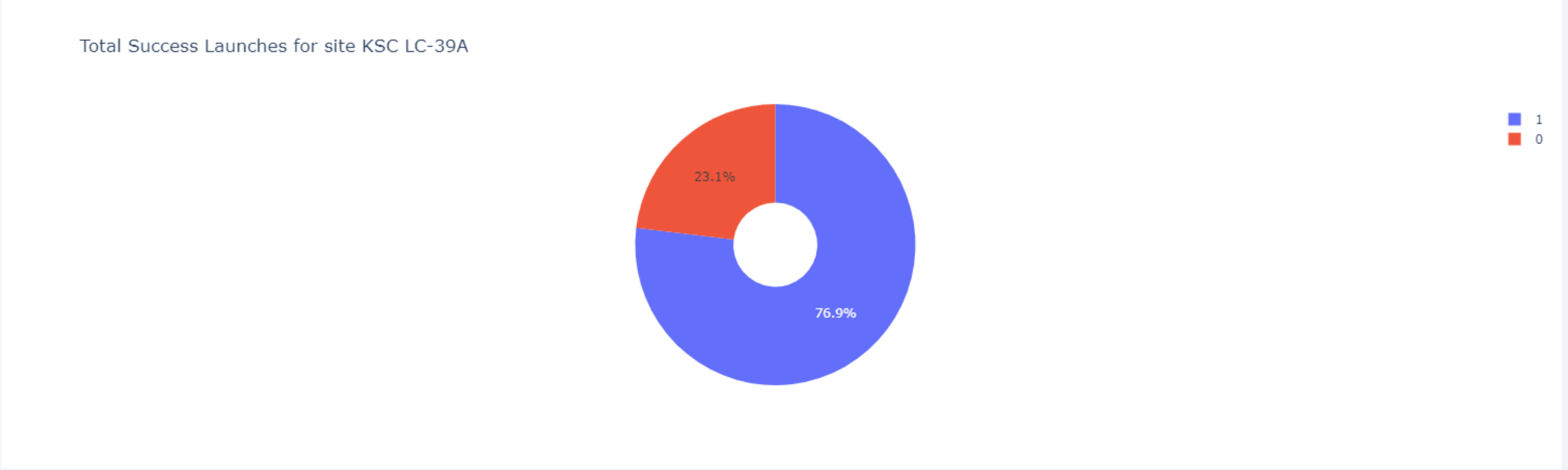

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for all Sites



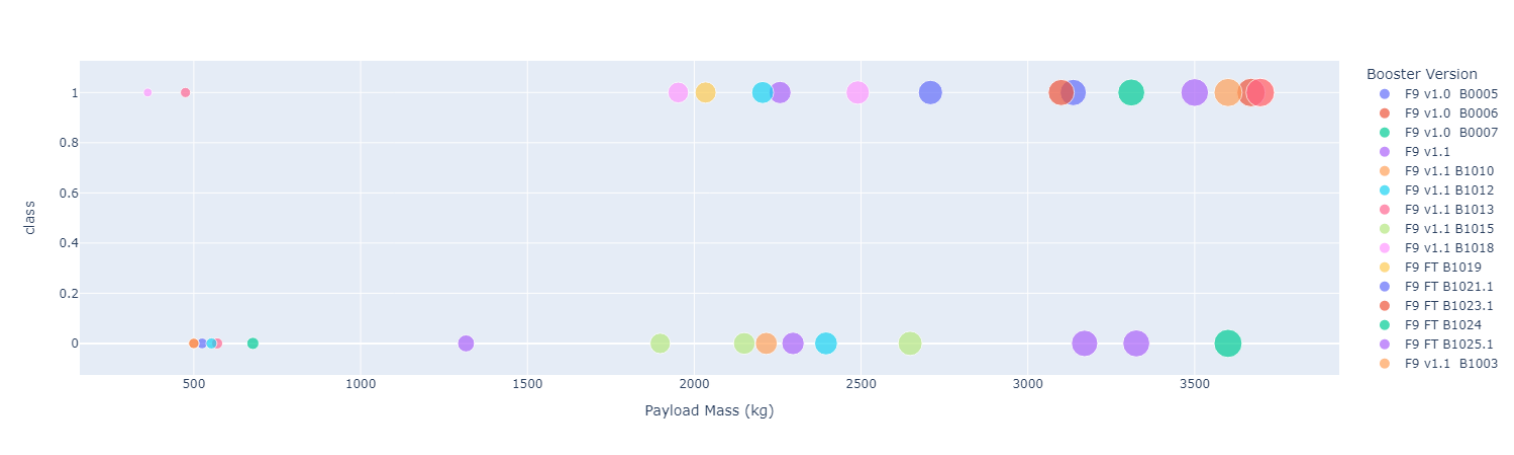
- The above pie chart shows that KSC LC 39A is the most successful launch site

Most Successful Launch Site



- KSC LC 39A has a launch success rate of 76.9% which is the highest of all the launch sites.

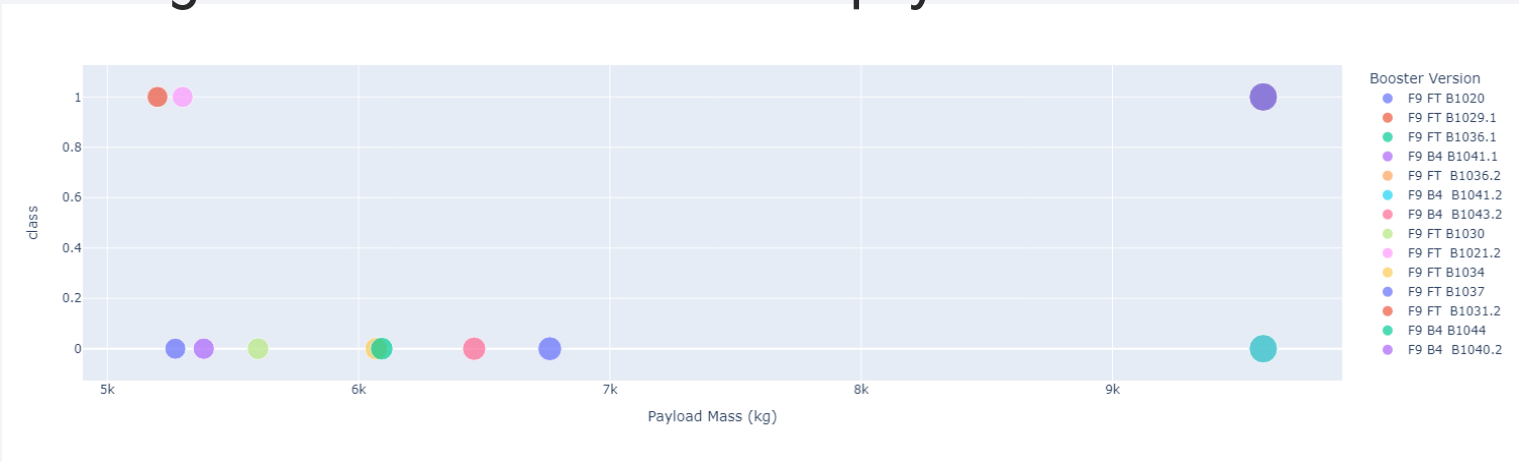
Payload Mass vs. Launch Outcomes for all Sites



- Low Payload Mass
  - Below 4000kg

❖The success rate is higher for launches with low payload mass

- High Payload Mass
  - Above 5000kg



Section 5

Predictive Analysis (Classification)

Classification Accuracy

The most accurate algorithm from the model is the Decision Tree Classifier

Confusion Matrix

		Predicted	
		Negative (**N**) -	Positive (**P**) +
Actual	Negative -	True Negatives (TN)	False Positives (FP) **Type I error**
	Positive +	False Negatives (FN) **Type II error**	True Positives (TP)

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landings marked as successful by the classifier.

Conclusions

- The finding from this assessment indicate:
  - The larger the flight amount at a launch site, the greater the success rate
  - Launches with a low payload mass have a higher success rate
  - Launch success rate of launches increases over the years
  - Orbits ES-L1, GEO, HEO, SSO, VLEO had the highest success rate
  - KSC LC-39A had the most successful launches of all sites
  - The Decision tree classifier is the best machine learning algorithm for this dataset

Appendix

- Special thanks to
  - My family
  - EdX Instructors
  - My Awesome peers

Thank you!