IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Keely Harris
2022-05-01

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API
  - Data Collection using Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Predictions
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Analytics Screenshots
  - Predictive Analysis Results

Introduction

- Project background and context
  - Space X launches the Falcon 9 rocket for with a savings of over 100 million dollars against their competitors. These savings are because Space X reuses the first stage of their rockets. Being able to determine if the first stage will land will allow Space X to know the cost of a launch. The goal of this project is to create a Machine Learning Pipeline to determine if the first stage will land.
- Problems you want to find answers
  - What are determining factors in successful landings?
  - How do these factors interact to create successful landings?
  - What are the operating procedures that need to be in place for successful landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using the Space X API and web scraping from Wikipedia
- Perform data wrangling
  - Applied one-hot encoding to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Looked at Logistical Regression, SVM, Decision Trees, and K Nearest Neighbors

Data Collection

- Describe how data sets were collected.
  - Data was collected by sending a get request to the Space X API
  - The response data was decoded and normalized using the .json() and .json_normalize()
  - The dataframe was cleaned, checked for missing values, and had missing values filled in where needed.
  - Additionally, data was collected by scraping Wikipedia for Falcon 9 launch records using Beautiful Soup.
  - Data was extracted from the HTML tables and converted into a dataframe for future use.

Data Collection – SpaceX API

- Data was collected using the get request, then decoded, normalized, cleaned, and missing values were taken care of.

- The GitHub URL for this is:

https://github.com/keeharr/courseracaprepo/blob/master/Coursera%20Capstone%20project.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

In [7]: response = requests.get(spacex_url)
```

Check the content of the response

Now we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

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

Using the dataframe data print the first 5 rows

```
mean_payload = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(to_replace = np.nan, value = mean_payload)

data_falcon9.isnull().sum()
```

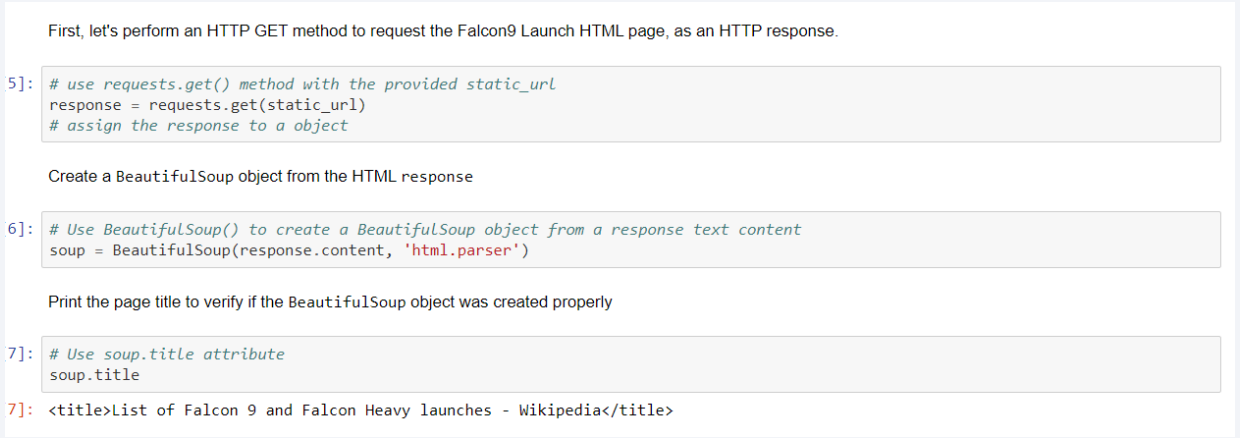
```
Out[44]: FlightNumber      0
         Date              0
         BoosterVersion    0
         PayloadMass       0
         Orbit             0
         LaunchSite        0
         Outcome           0
         Flights           0
         GridFins          0
```

Data Collection - Scraping

- Scraped the information from the Wikipedia Page, parsed through the information, and placed the needed data in a pandas database

- The GitHub URL is:

https://github.com/keeharr/courseracaprepo/blob/master/Capstone%20Data%20Collection.ipynb



First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

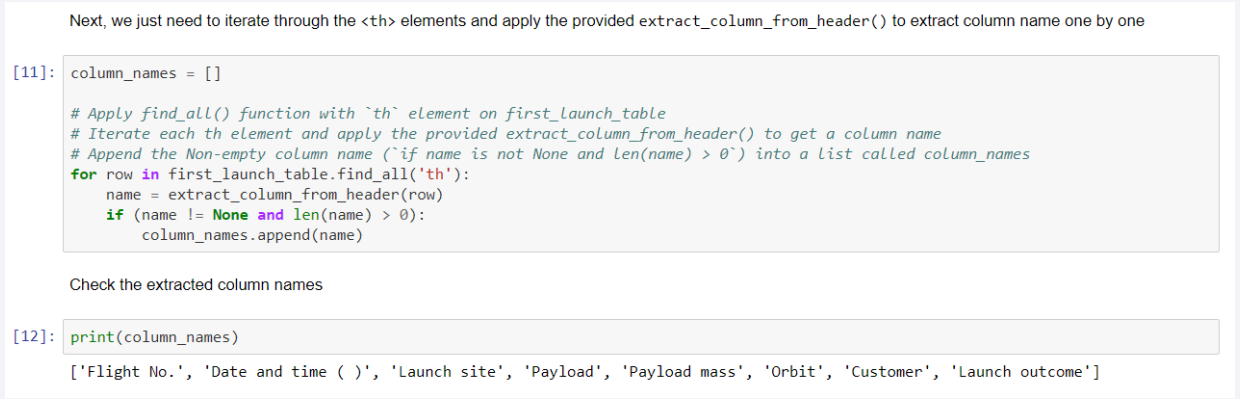
```
5]: # use requests.get() method with the provided static_url
    response = requests.get(static_url)
    # assign the response to a object
```

Create a BeautifulSoup object from the HTML response

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

Print the page title to verify if the BeautifulSoup object was created properly

```
7]: # Use soup.title attribute
    soup.title
7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```



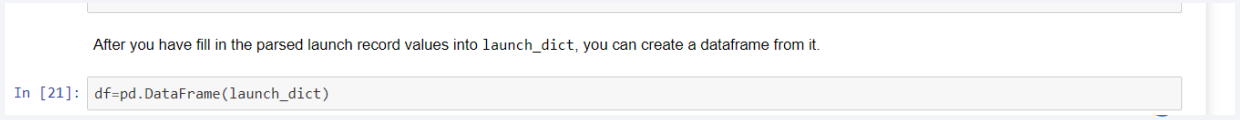
Next, we just need to iterate through the <th> elements and apply the provided extract_column_from_header() to extract column name one by one

```
[11]: 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 row in first_launch_table.find_all('th'):
          name = extract_column_from_header(row)
          if (name != None and len(name) > 0):
              column_names.append(name)
```

Check the extracted column names

```
[12]: print(column_names)
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```



After you have fill in the parsed launch record values into launch_dict, you can create a dataframe from it.

```
In [21]: df=pd.DataFrame(launch_dict)
```

Data Wrangling

- Performed Exploratory Data Analysis, and labeled training data.
- The GitHub URL is: https://github.com/keeharr/courseracaprepo/blob/master/Capstone%20EDA.ipynb

In [12]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

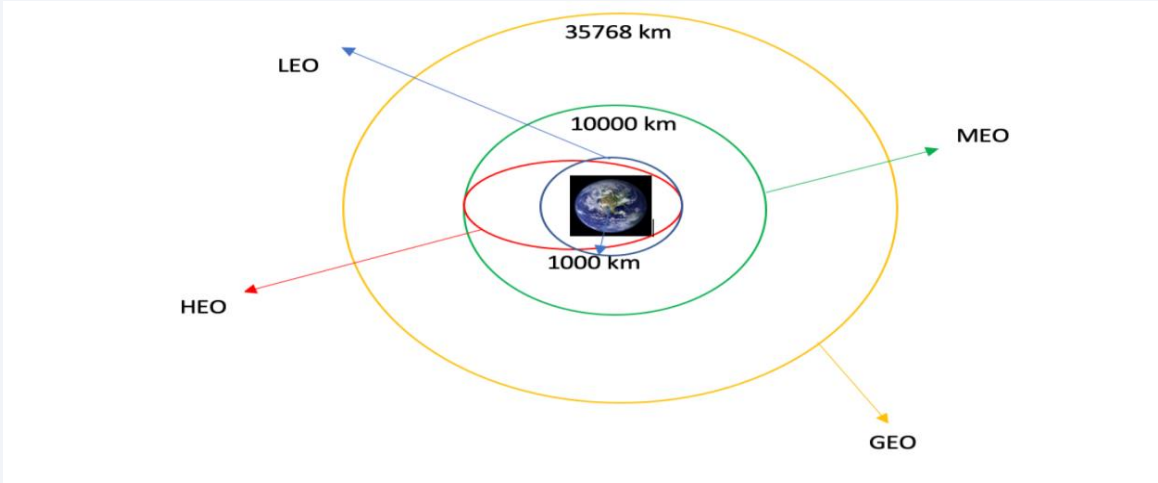
This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

In [13]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[13]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

In [14]:
```
df.head(5)
```



EDA with Data Visualization

- Flight Number and Payload Mass were plotted, overlaid with outcome of launch, we saw that as both Number and Mass increased, so did success
- Flight Number and Launch Site were plotted, overlaid with outcome of launch, we saw that the success increased on all Sites as Number increased
- Payload Mass and Launch Site were plotted, overlaid with outcome of launch, we saw that Success increased as Mass increased on all sites, but site VFAB SLC 4E did not launch Heavy Mass payloads
- Multiple plots were made to see success rate and orbit type, and saw that the above statements stayed true for all orbits, with one noted exception.
- The GitHub URL is: https://github.com/keeharr/courseracaprepo/blob/master/Capstone%20EDA%20with%20Visualization.ipynb

EDA with SQL

- SQL Queries
  - Found Distinct Launch Sites
  - Found 5 Records where Launch Sites began with “CCA”
  - Found total Mass caried by boosters launched by NASA
  - Found average Payload Mass caried by Booster Version F9 v1.1
  - Found Date of First successful landing
  - Found Total Success and Failures
  - Ranked the count of Landing Outcomes
- The GitHub URL is: https://github.com/keeharr/courseracaprepo/blob/master/Capstone%20EDA%20with%20SQL.ipynb

```
1/bludb
Done.
```

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

Build an Interactive Map with Folium

- Marked all Launch Sites, and added the circle map objects to signify successful and unsuccessful landings. These objects were used to see success rate at each Launch Site.
- Calculated Distances from Launch Sites to several places, including the coast, railroads, highways, and cities to determine proximity.
- The GitHub URL is: https://github.com/keeharr/courseracaprepo/blob/master/Capstone%20Interactive%20Visual%20Analysis.ipynb

Build a Dashboard with Plotly Dash

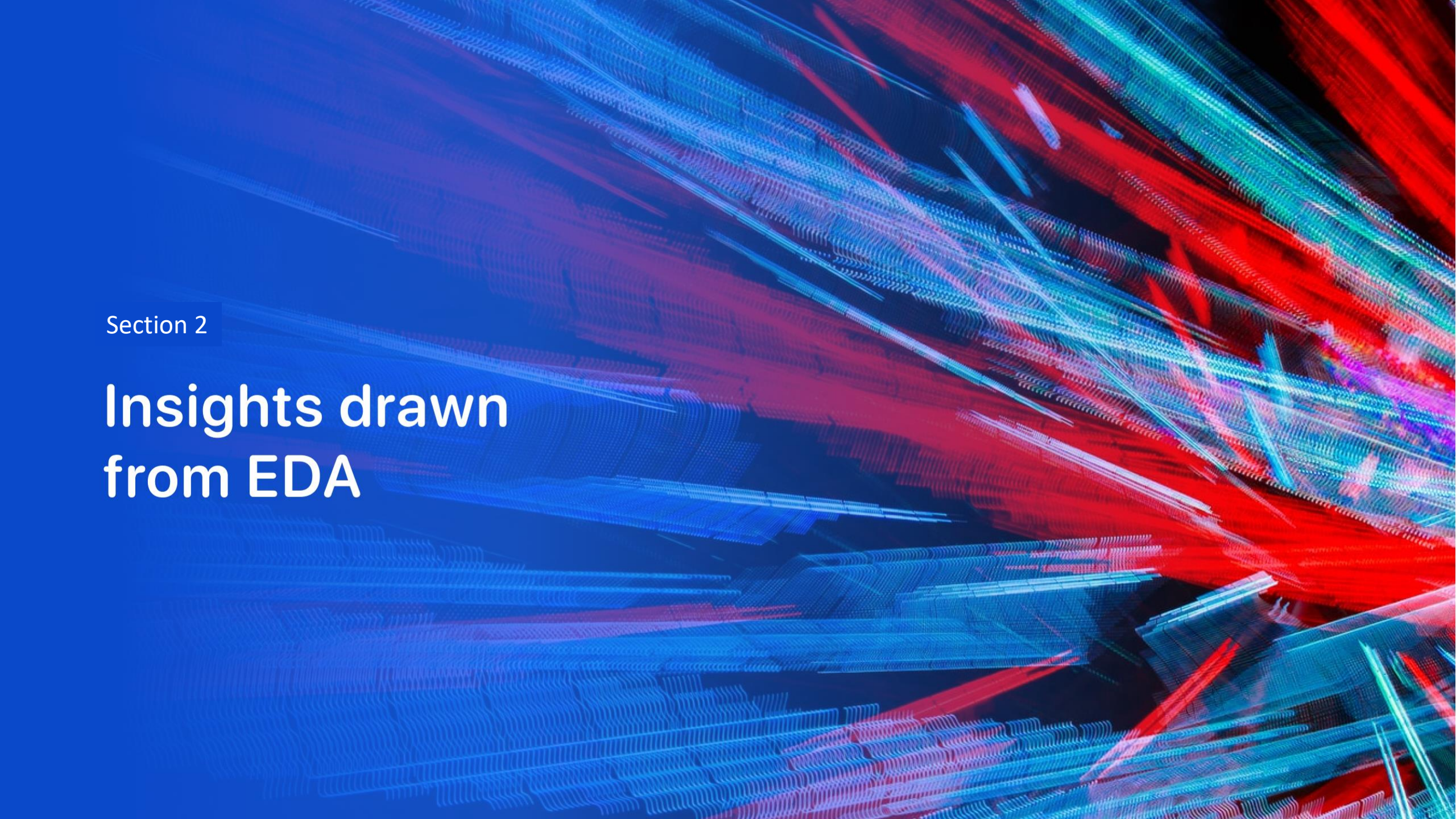
- Tried to create an interactive Dashboard
- Added Pie charts and scatterplots to determine successful launches by site and correlation between success and payload at each site.

Predictive Analysis (Classification)

- Loaded data, then split into Train and Test sets, used these sets on Logistical Regression, SVM, Decision Tree, and K Nearest Neighbor to determine the best model for the data
- The GitHub URL is: https://github.com/keeharr/courseracapprepo/blob/master/Capstone%20Machine%20Learning%20Predictions.ipynb

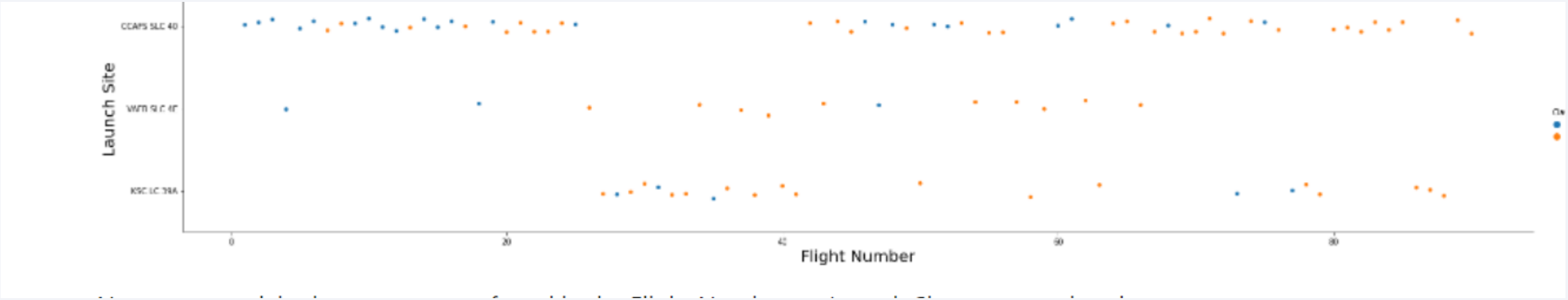
Results

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


Section 2
Insights drawn from EDA

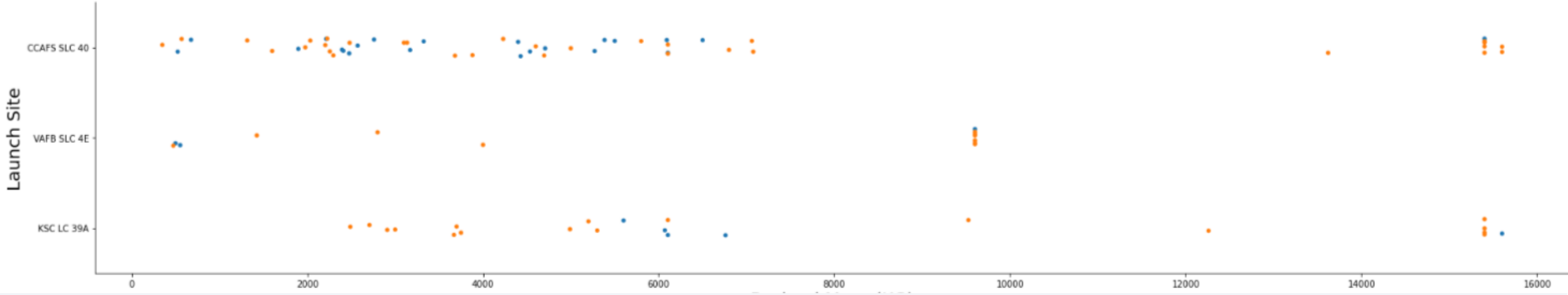
Flight Number vs. Launch Site

- As the Flight Number increases, the frequency of successful landings increases.



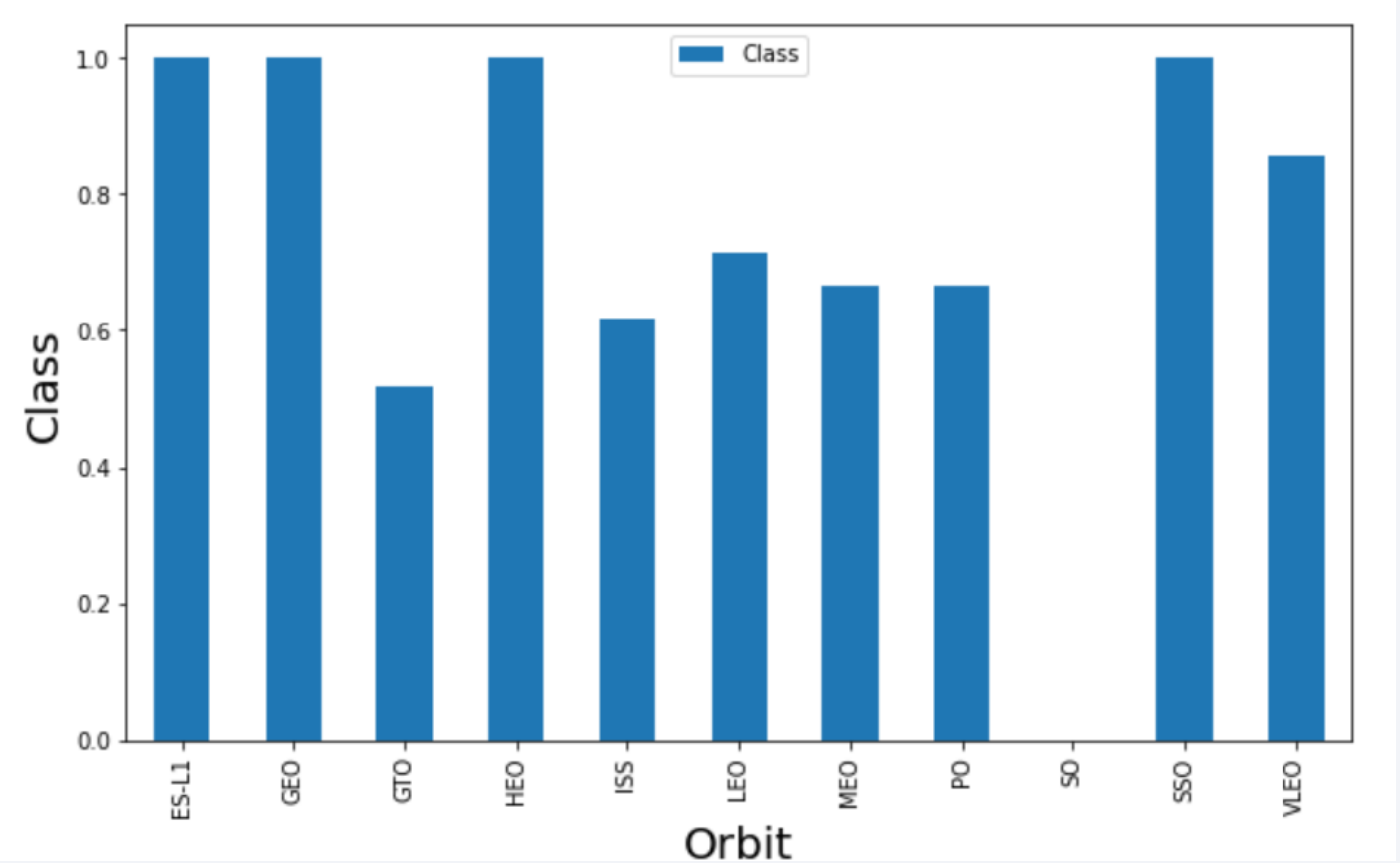
Payload vs. Launch Site

- Generally, the success goes up as the Payload reaches “heavy” or over 10,000 kg. It is important to not that VAFB SLC 4E does not have any heavy launches



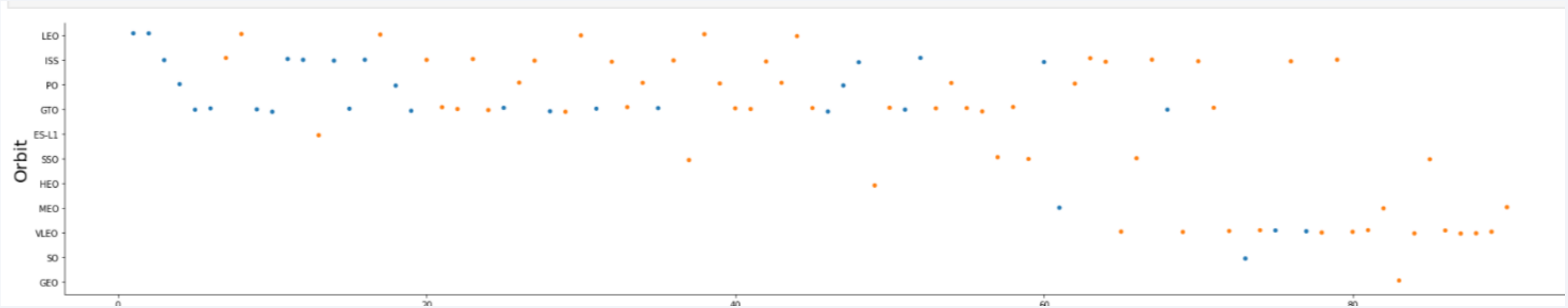
Success Rate vs. Orbit Type

- No orbit has less than .500, and 4 orbits are sitting at 1.000 when this data was recorded.



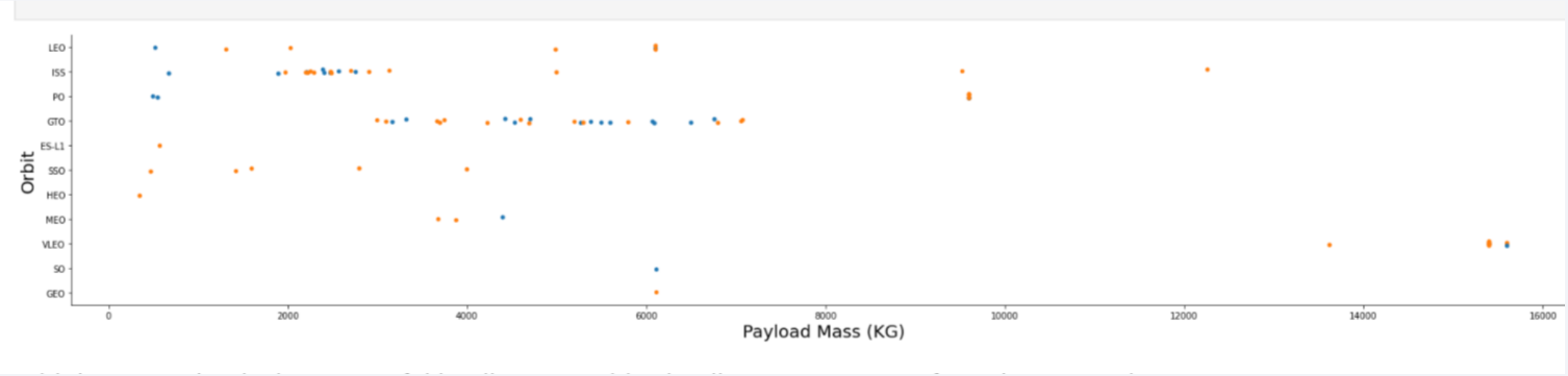
Flight Number vs. Orbit Type

- All the orbits show that as the flight number increases, so does the success rate, but failures are still noted in the higher flight numbers



Payload vs. Orbit Type

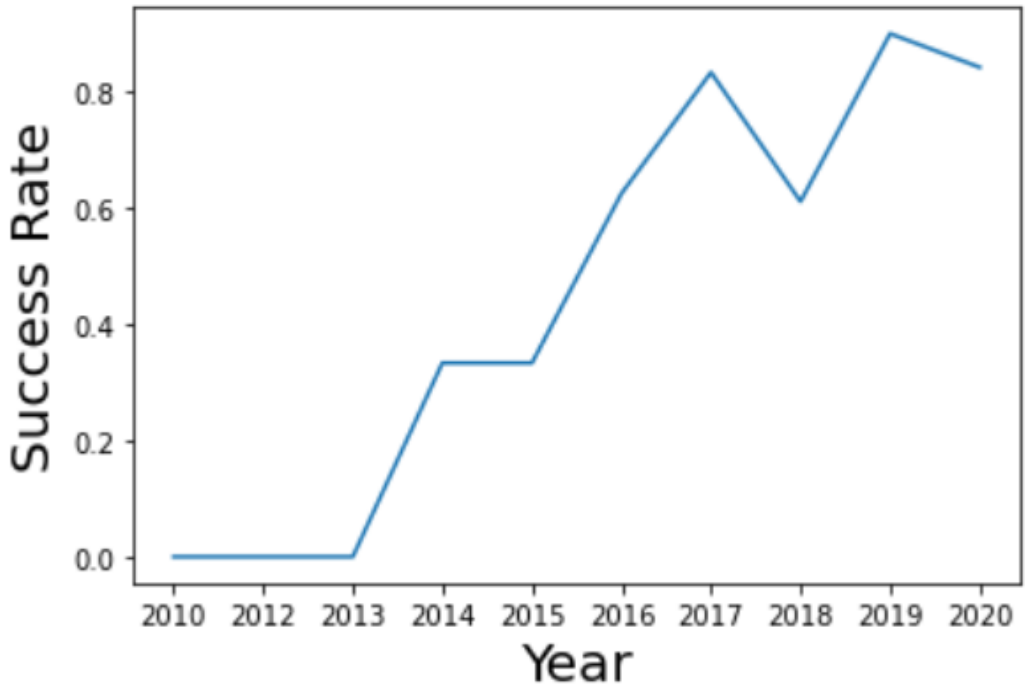
- The success rate increases with Payload Mass for Polar, LEO, and ISS
- For the other Orbits, a correlation cannot be established between success rate and payload mass based on this data



Launch Success Yearly Trend

- The Success Rate has been generally positive since 2013
- There are clear dips in Success Rate in 2017 and again in 2019

```
plt.show()
```



All Launch Site Names

Used Key Word DISTINCT to ensure that site names would only appear once in the list

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

[9]:
```
%%sql
SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET;
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

ut[9]:

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

Launch Site Names Begin with 'CCA'

Used the LIKE key word to find site names beginning with CCA

Task 2

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

[16]:
```
%%sql
SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[16]:

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
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
2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
2014-07-14	15:15:00	F9 v1.1	CCAFS LC-40	OG2 Mission 1 6 Orbcomm-OG2 satellites	1316	LEO	Orbcomm	Success	Controlled (ocean)
2014-09-21	5:52:00	F9 v1.1 B1010	CCAFS LC-40	SpaceX CRS-4	2216	LEO (ISS)	NASA (CRS)	Success	Uncontrolled (ocean)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

Task 3

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

Total Payload Mass

Used the SUM method to find the total mass of 23,589 kg

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

8]:
```
%%sql
select sum(PAYLOAD_MASS__KG_) from SPACEXDATASET where CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

8]:

1
23589

Task 4

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

Average Payload Mass by F9 v1.1

The Average Payload was found using the avg method and was 1,806 kg

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

```
%%sql
select avg(PAYLOAD_MASS__KG_) from SPACEXDATASET where BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

1
1806

Task 5

List the date when the first successful landing outcome in ground pad was acheived.

First Successful Ground Landing Date

The First Successful Ground Landing was on December 22, 2015

```
%%sql
select min(DATE) from SPACEXDATASET where LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

1
2015-12-22

Task 6

Successful Drone Ship Landing with Payload between 4000 and 6000

Two Boosters have successfully landed on a Drone ship with Payload between 4,000 and 6,000 kg

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

```
%%sql
select distinct BOOSTER_VERSION from SPACEXDATASET where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

booster_version
F9 FT B1021.2
F9 FT B1026

Task 7

Total Number of Successful and Failure Mission Outcomes

There have been 55 missions considered a Success and one Failure(in flight)

```
0]: %%sql
    select MISSION_OUTCOME, count(*) from SPACEXDATASET group by MISSION_OUTCOME;
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

0]:

mission_outcome	2
Failure (in flight)	1
Success	55

Boosters Carried Maximum Payload

A subquery was used to find which Boosters have carried the Maximum Payload

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

[33]:
```
%%sql
select distinct BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPA
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

[33]:

booster_version
F9 B5 B1048.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1060.3

2015 Launch Records

There was only one failure in 2015, having booster version F9 v1.1 B1015 from site CCAF LC -40

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATASET where LANDING__OUTCOME = 'Failure (drone ship)' and extract
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:327
1/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1015	CCAFS LC-40

Task 10

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

Between the Specified Dates, Failure(drone ship) was the most common outcome with Precluded(drone ship) as the least

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

```
%%sql
select LANDING__OUTCOME, count(*) from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20' group by LAND
```

```
 * ibm_db_sa://mdj32074:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
1/bludb
Done.
```

landing__outcome	2
Failure (drone ship)	3
No attempt	3
Success (drone ship)	3
Success (ground pad)	3
Controlled (ocean)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Reference Links

Section 3

Launch Sites Proximities Analysis

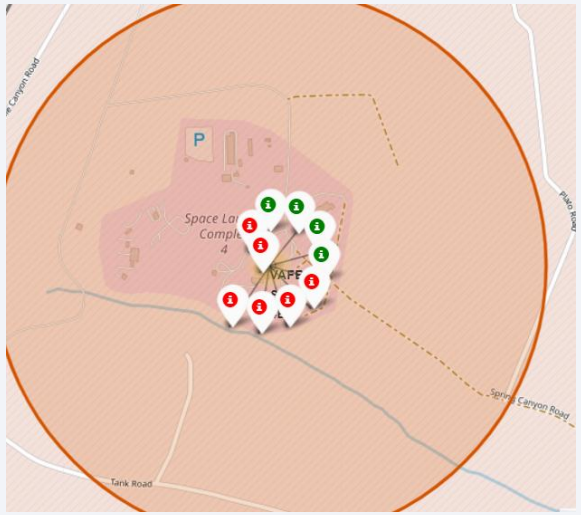
Launch Sites

The launch Sites for Space X are on the coasts of California and Florida in the United States

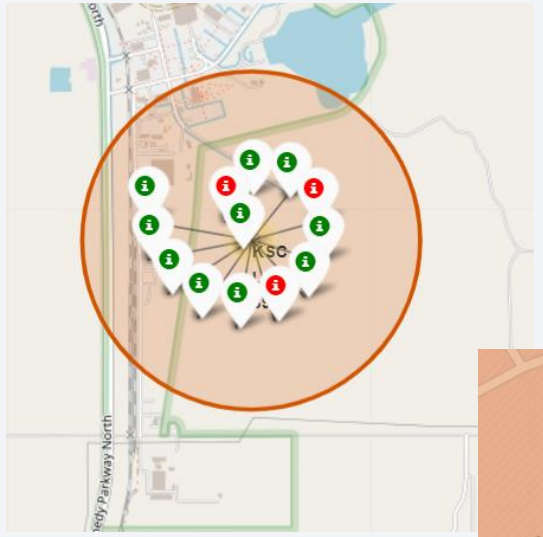
Launch Outcomes

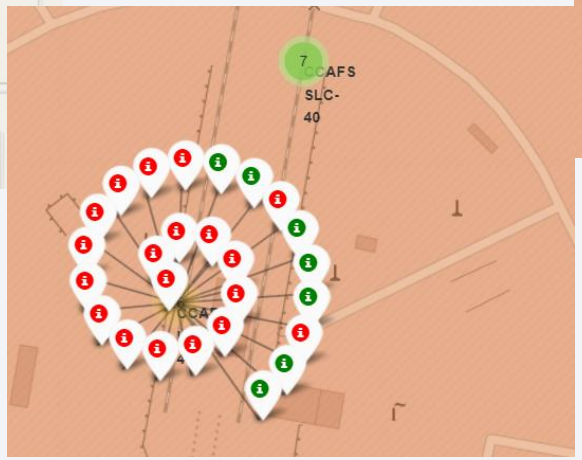
Green markers are success and red markers are failures

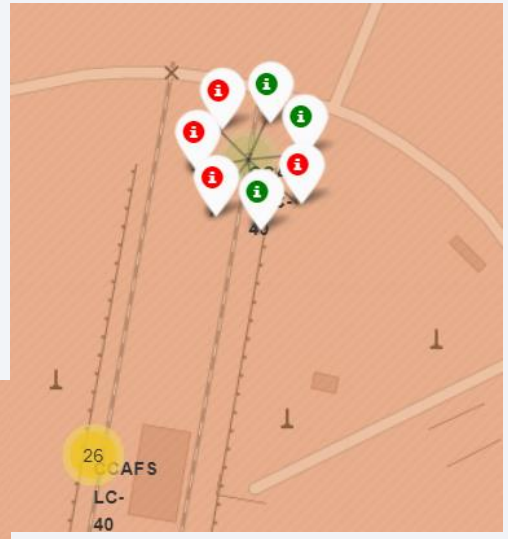
California Launch Site



Florida Launch Sites



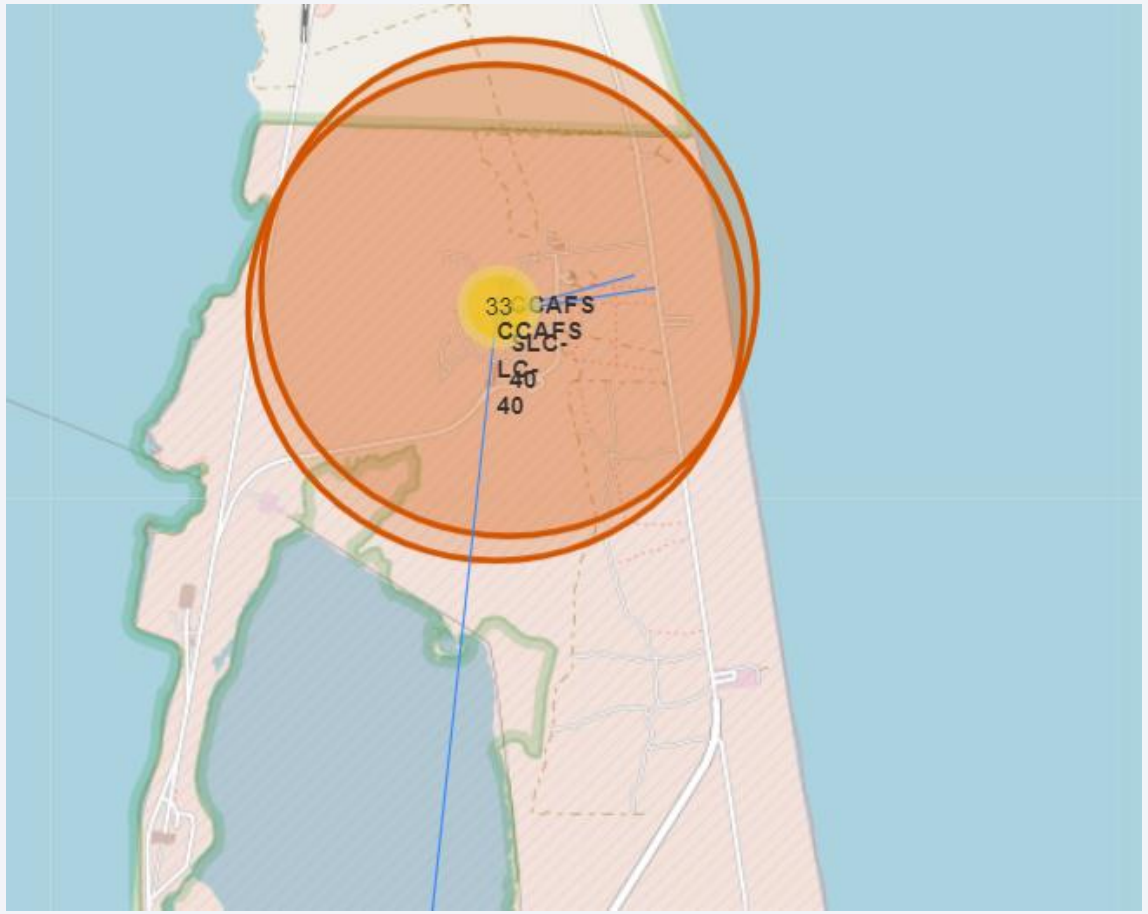


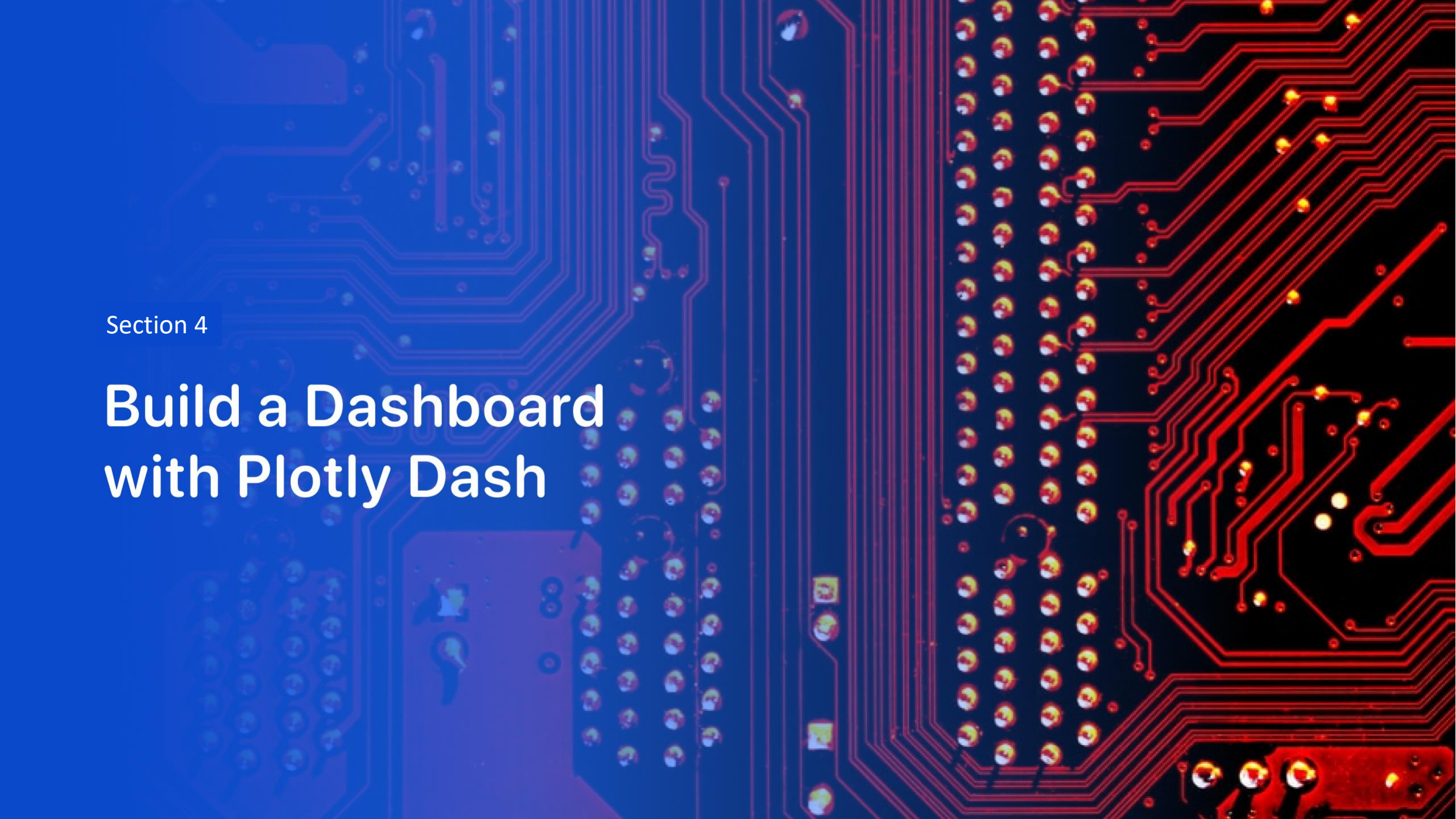


Launch Site Location and Surroundings

The launch site with lines leading to RailRoads, Highways, and Cities nearby.

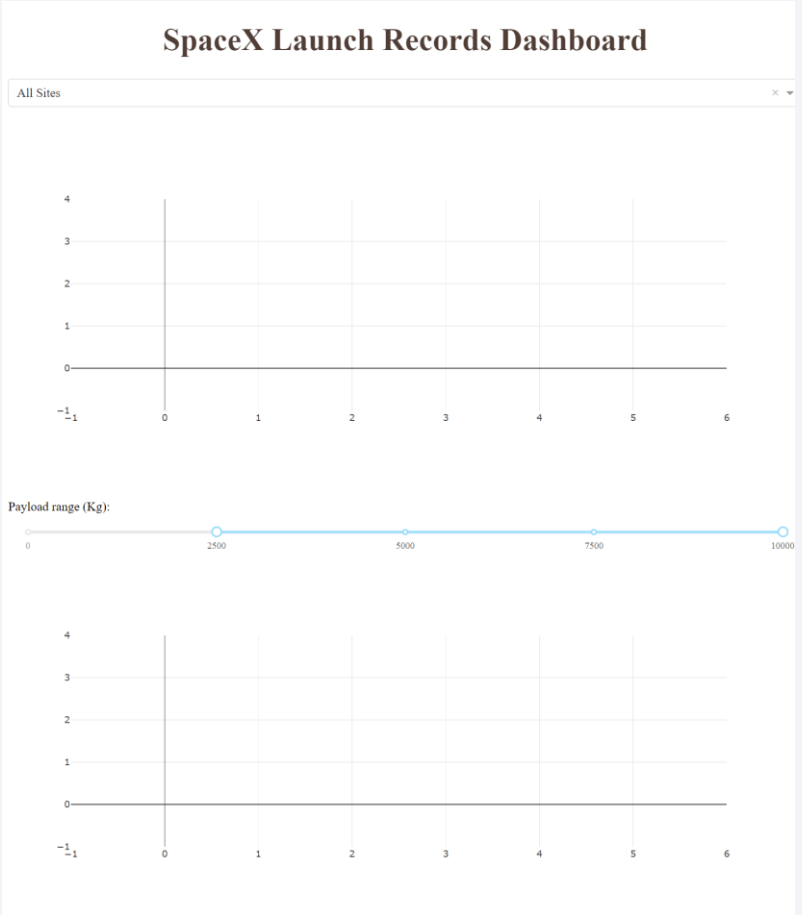
The Launch Site is near RailRoads and close to highways, but are kept farther away from Cities




Section 4
Build a Dashboard with Plotly Dash

Plotly Dropdown

Dropdown menu on the Plotly Dashboard.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The models tested were Logistic Regression, Support Vector Machine, Decision Trees, and K Nearest Neighbor.
- The Support Vector Machine had the highest Accuracy on the Test Data

[34]:

```
modelScores = {'LogisticRegression':logreg_cv.score,
               'SupportVector':svm_cv.score,
               'DecisionTree':tree_cv.score,
               'KNearestNeighbors':knn_cv.score}
bestModel = max(modelScores)
print('The best model is ', bestModel)
```

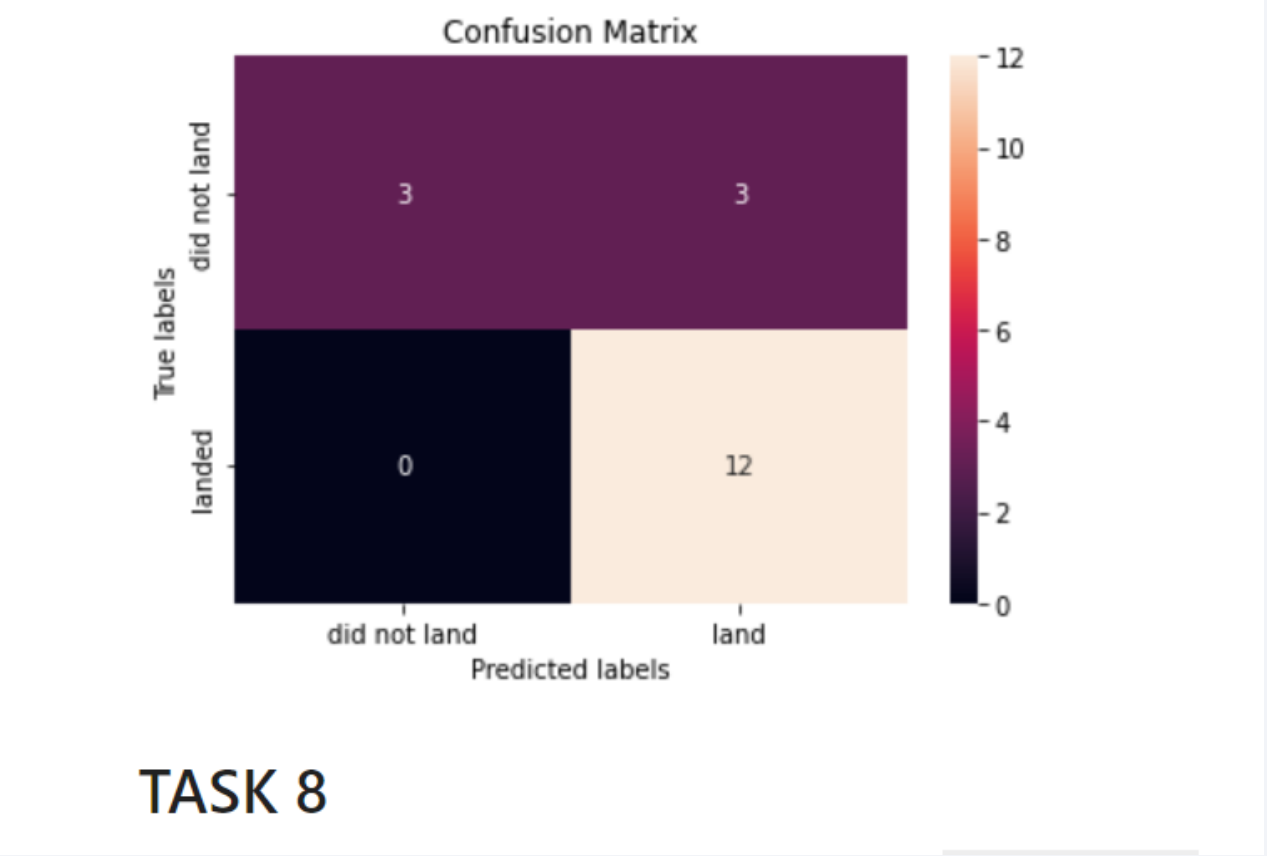
```
The best model is  SupportVector
```

Authors

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and

Confusion Matrix

- The Support Vector Machine's Confusion Matrix.
- SVM struggled most with false positives, but had no false negatives.



Conclusions

We can see that

- The higher the number of flights a particular site launches, the higher the success rate will be
- Landing Success has been increasing since 2013
- The Launch Sites are purposefully chosen to be safe
- The Support Vector Model is the best model to predict success

Thank you!