Winning Space Race with Data Science

Jerry Yu Zhang
28/02/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with API
  - Data collection with web scraping
  - Data wrangling
  - Exploratory data analysis with SQL
  - Exploratory data analysis with Pandas and Matplotlib
  - Interactive visual analytics with Folium
  - Machine learning prediction
- Summary of all results
  - Exploratory data analysis results
  - Interactive dashboard with Ploty Dash
  - Predictive analytics results

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - What interaction amongst various features that determine the success rate of a successful landing?
  - What operation conditions needs to be in place to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied 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
  - GridSearchCV was used to find the best parameters of classification algorithms.
  - Find the best method by comparing the accuracy and confusion matrix.

Data Collection

- The data was collected using various methods
  - Data collection was done using the GET request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values by mean where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 Launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the GET request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Collection%20with%20API.ipynb

1. Get request for rocket launch data using API

```
In [2]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [3]: response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

```
In [4]: data = pd.json_normalize(response.json())
```

3. We performed data cleaning and filling in the missing values

```
In [5]: # Calculate the mean value of PayloadMass column
PayloadMassMean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(
    np.nan, PayloadMassMean, inplace = True)
```

Data Collection - Scraping

- We applied web scrapping to collect Falcon 9 launch records with Beautifulsoup.
- We parsed the table and converted it into pandas dataframe.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [ ]:
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_
response = requests.get(static_url).text
```

2. Create a BeautifulSoup object from the HTML response

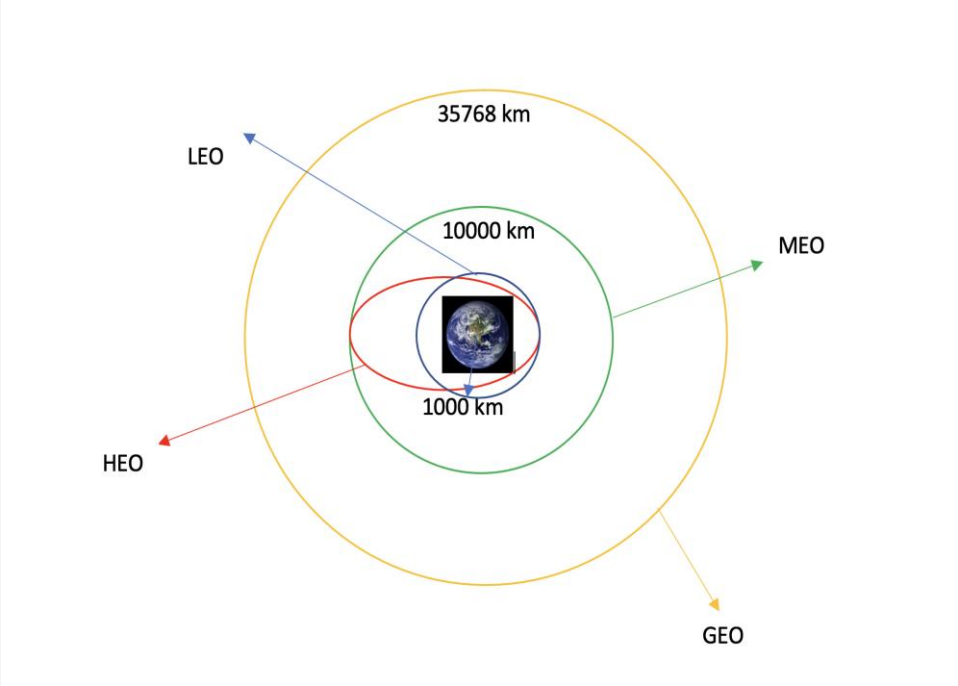
```
In [ ]:
soup = BeautifulSoup(response, 'html5lib')
```

3. Extract all column names from the HTML table header

```
In [ ]:
column_names = []
for name in first_launch_table.find_all('th'):
    stripped_header = extract_column_from_header(name)
    if (stripped_header is not None and len(str(stripped_header)) > 0):
        column_names.append(stripped_header)
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

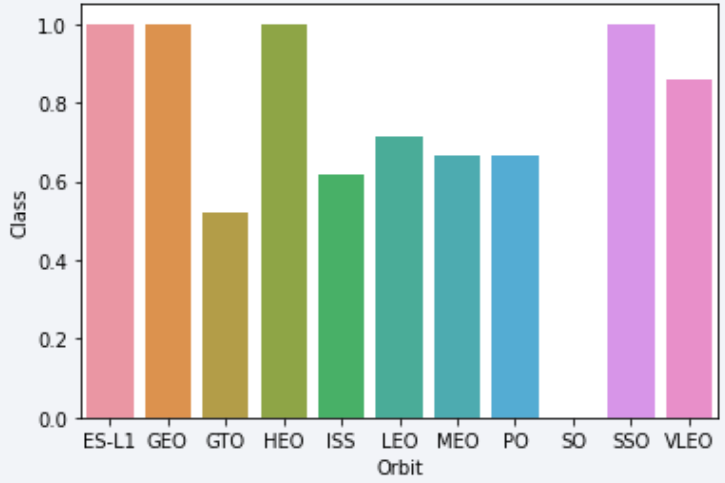
Data Wrangling

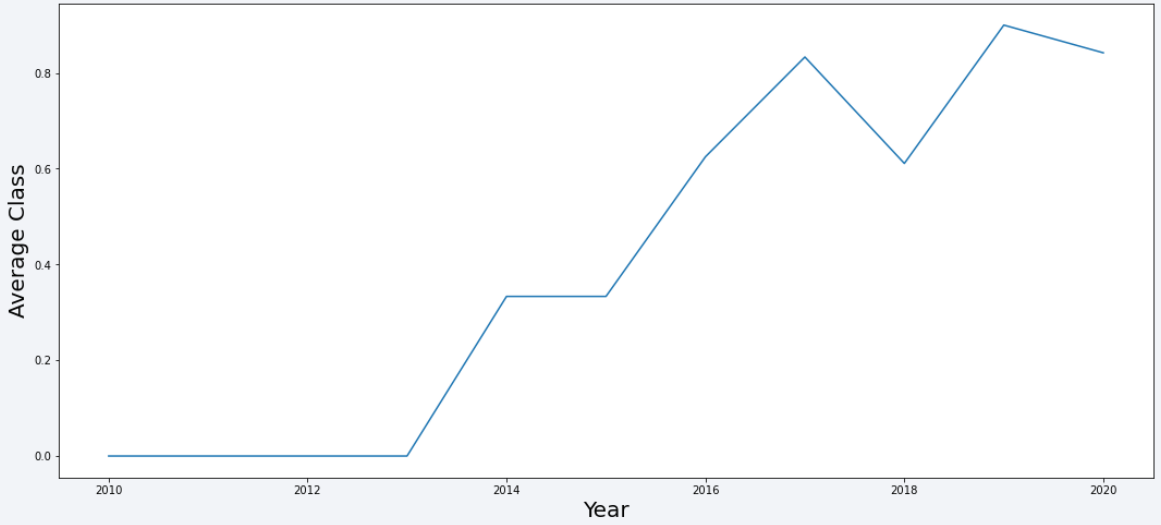


- We performed exploratory data analysis
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Wranglin g.ipynb

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Wrangling.ipynb

EDA with SQL

- We loaded the SpaceX dataset into the corresponding table in a Db2 database.
- We loaded the SQL extension in jupyter notebook and established a connection with the Db2 database.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance like:
  - The names of unique launch sites in the space mission
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The failed landing outcomes in drone ship, their booster version and launch site
  - The rank of the count of landing outcomes between the date 2010-06-04 and 2017-03-20
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Exploratory%20Data%20Analysis%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1, i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

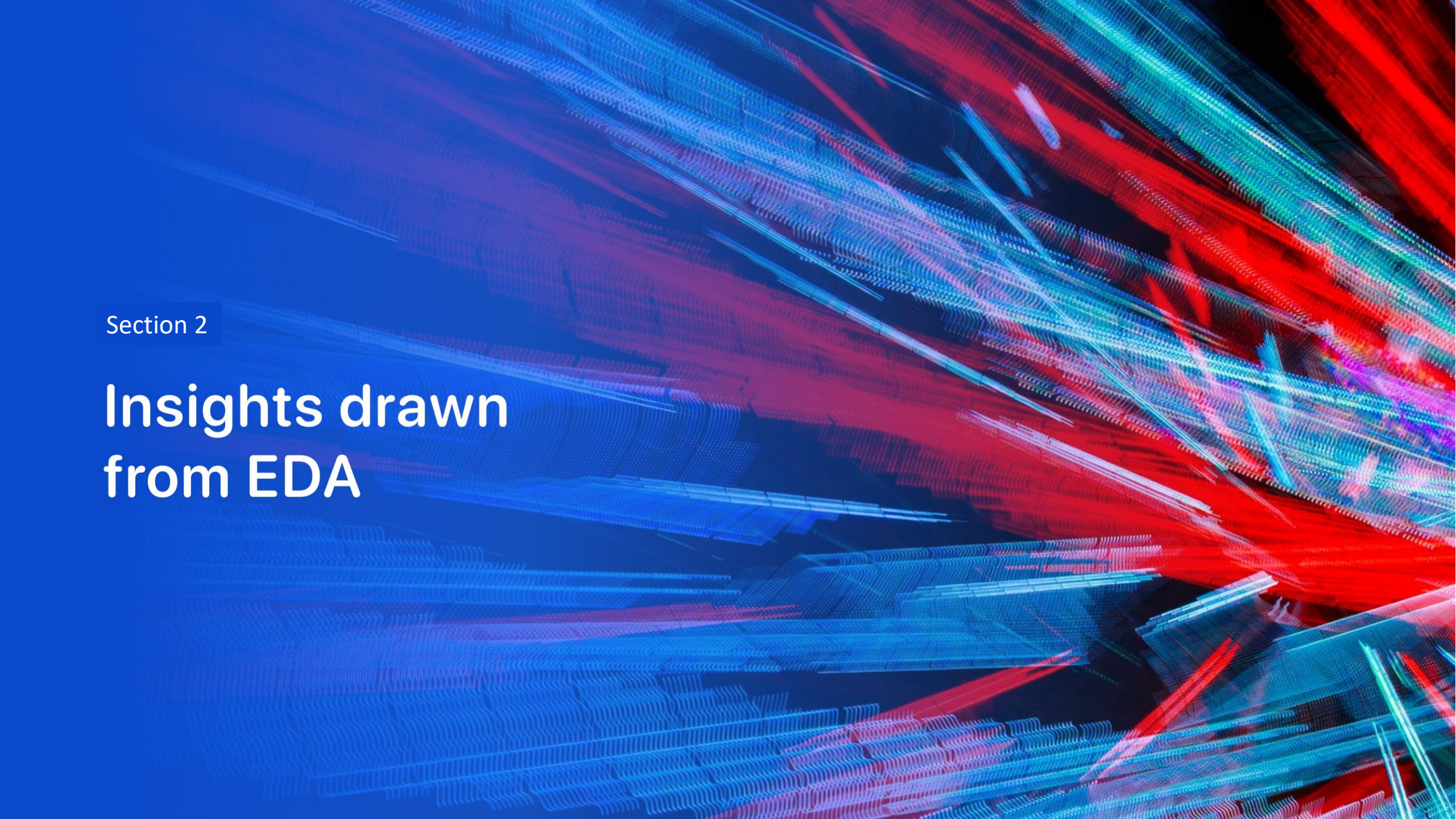
- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- This interactive visual analytics enables stakeholders to explore and manipulate data in an interactive and real-time way.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

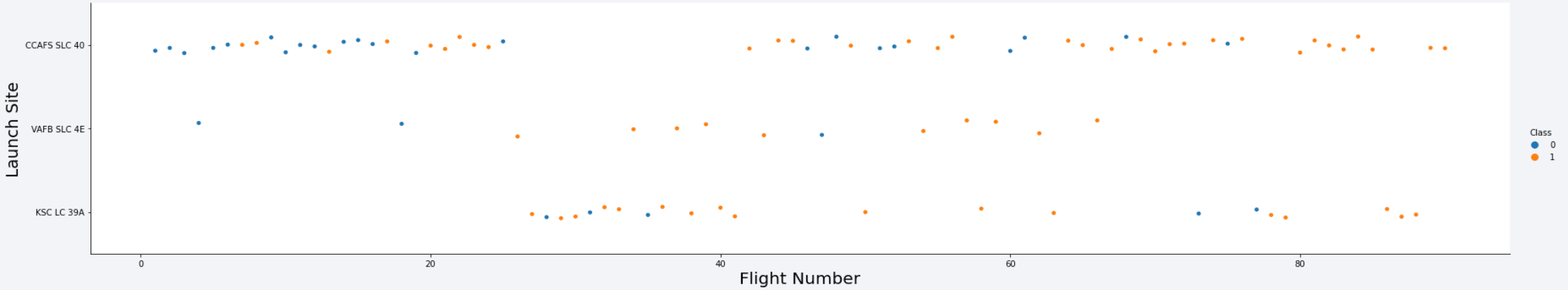
Results

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


Section 2
Insights drawn from EDA

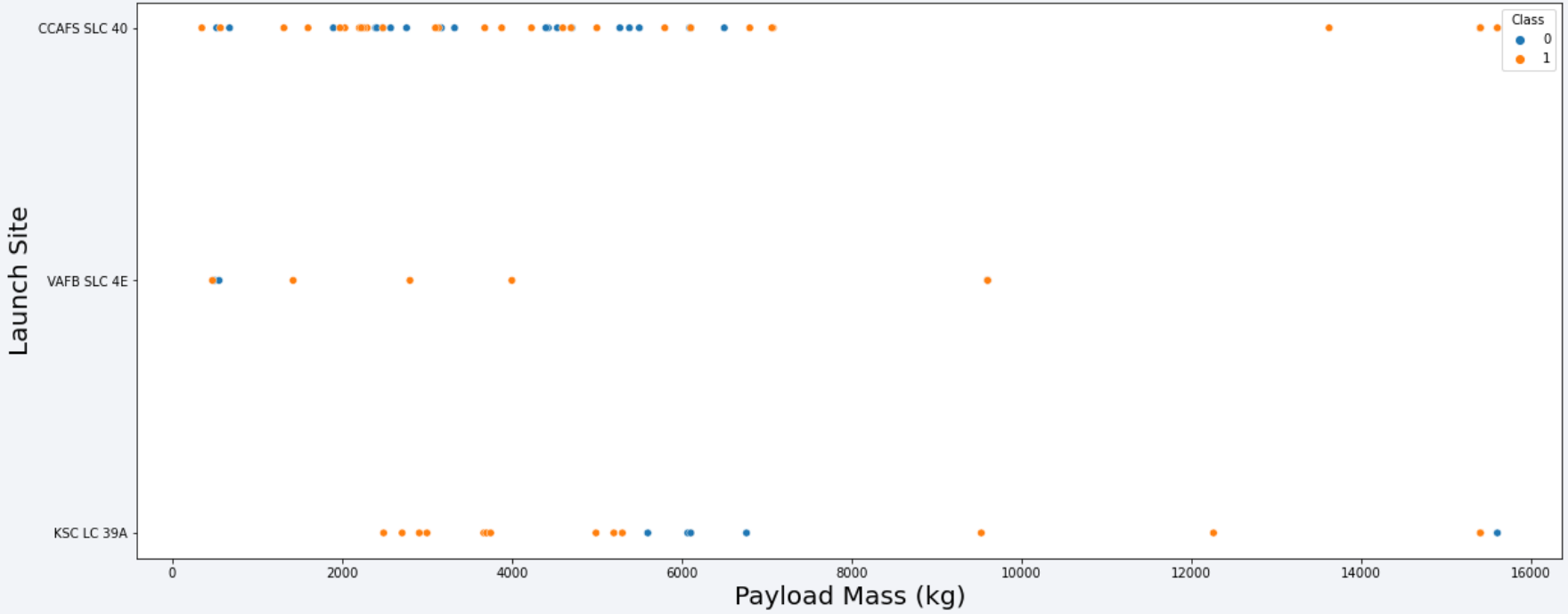
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



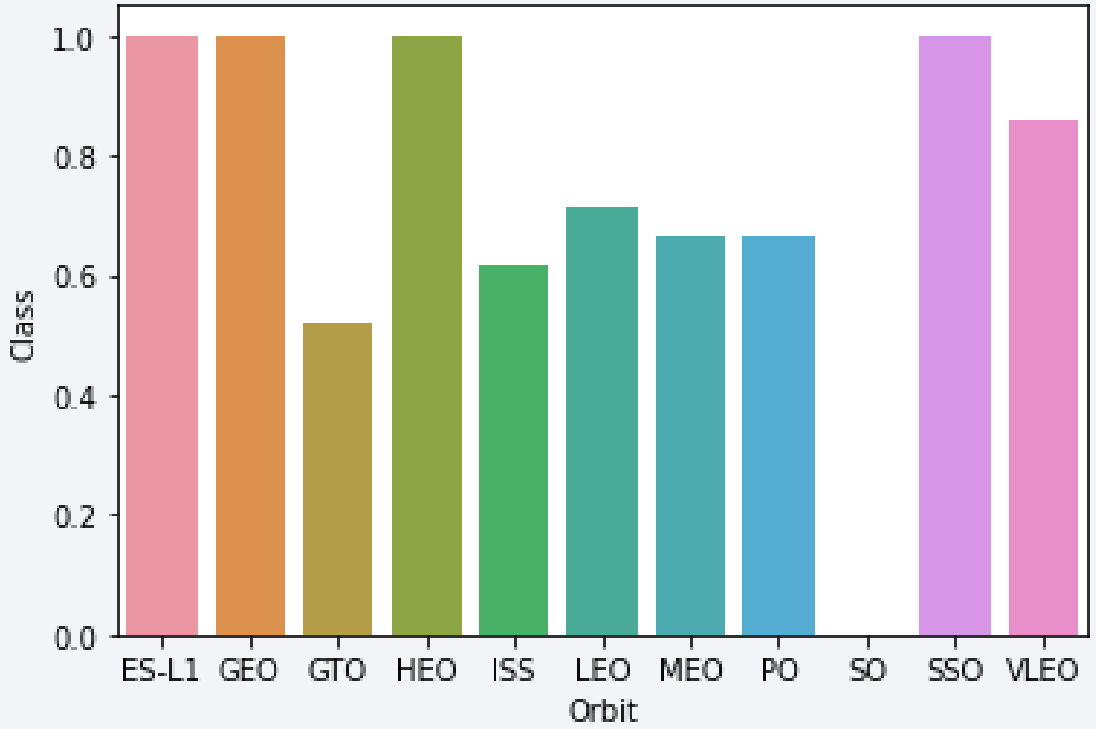
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



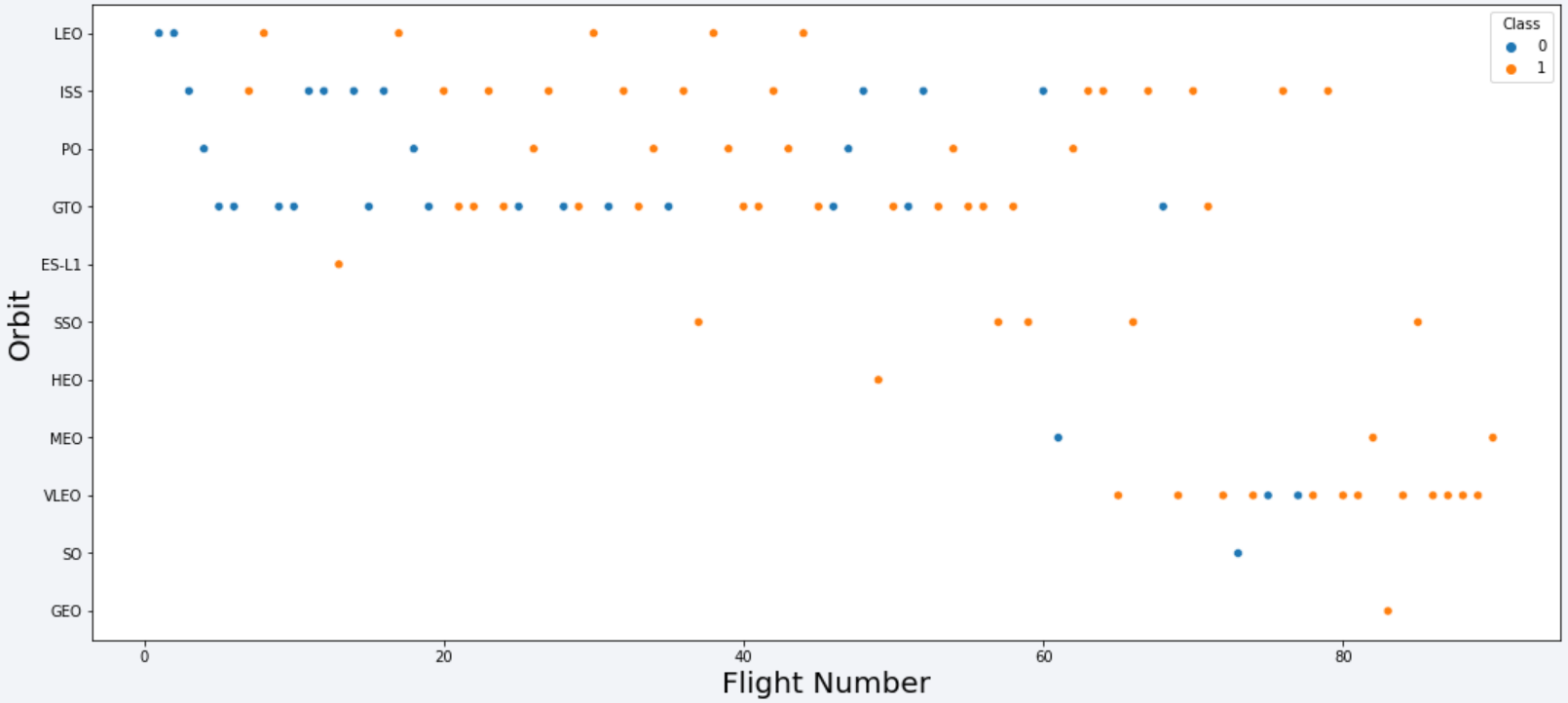
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



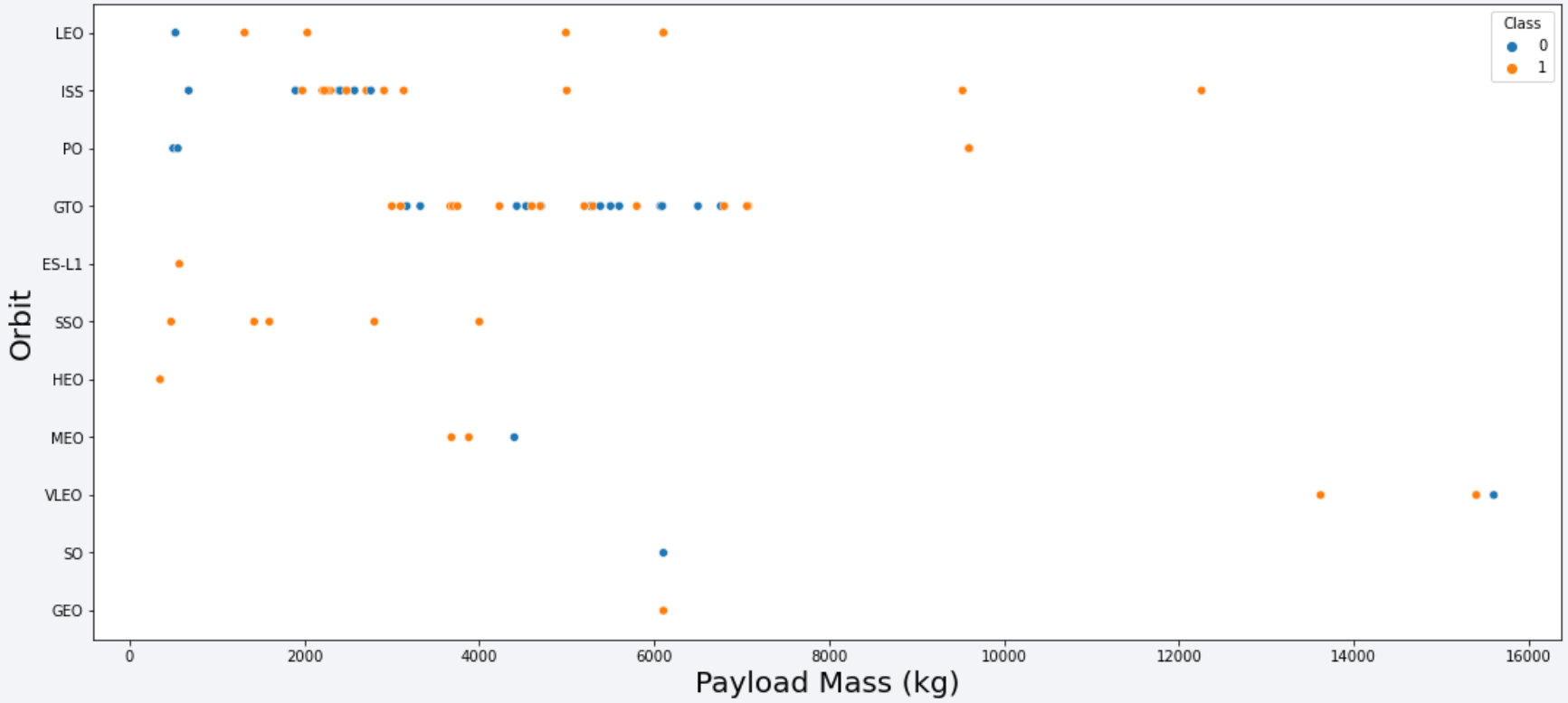
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



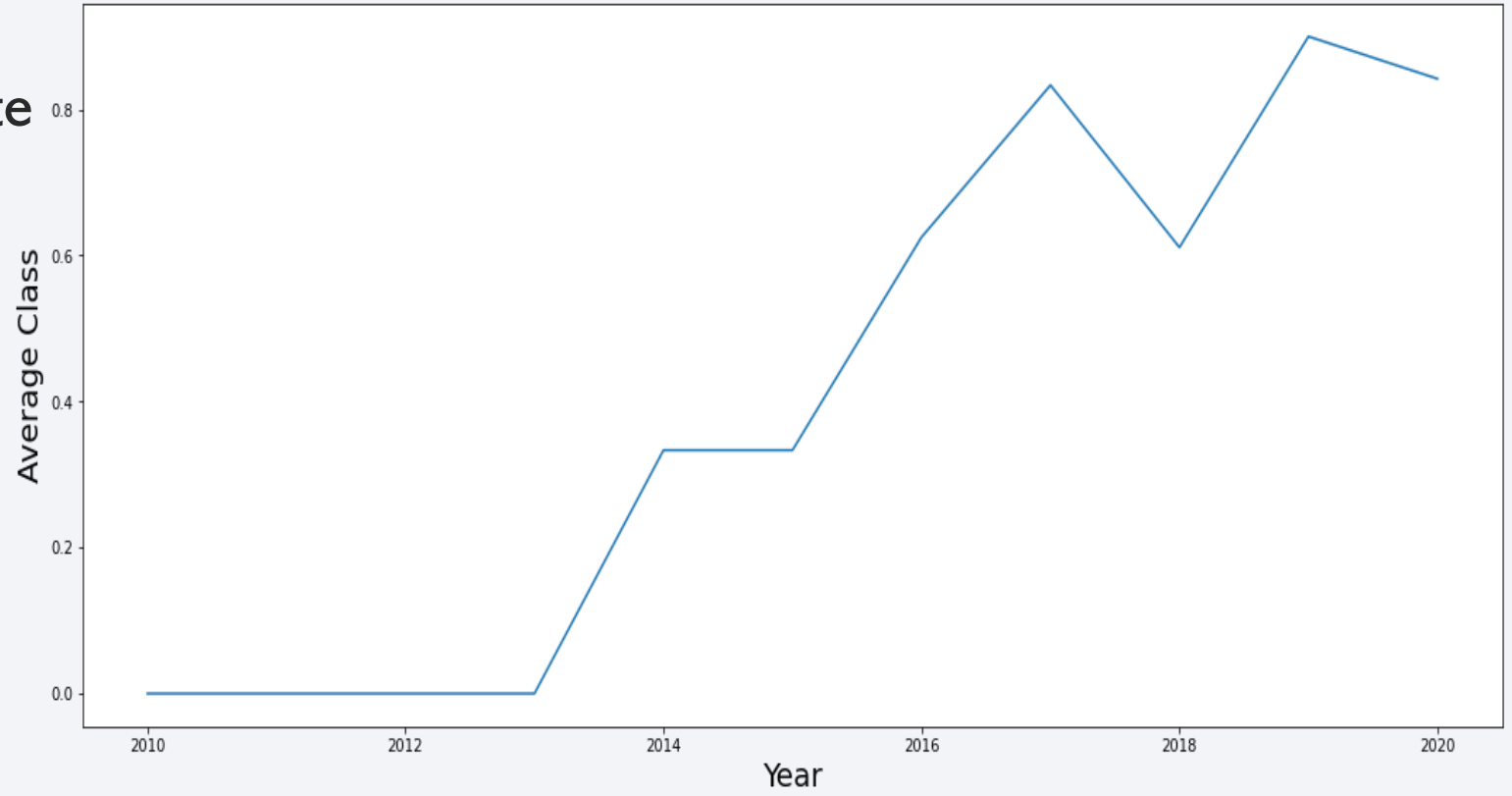
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

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

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b
Done.
```

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

Launch Site Names Begin with 'CCA'

- We used key words **LIKE** and **LIMIT** to display 5 records where launch sites begin with `CCA`.

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

```
%sql select * from spacexdataset where launch_site like 'CCA%' limit 5;
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We used the function **SUM()** to find the total payload carried by boosters launched by NASA (CRS) as 45596kg.

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

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- We used the function **AVG()** to find the average payload mass carried by booster version F9 v1.1 as 2928kg.

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://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.dat
Done.
```

1
2928

First Successful Ground Landing Date

- We used the key word **LIKE** and function **MIN()** to find that the dates of the first successful landing outcome on ground pad was 22nd December 2015.

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

Hint:Use min function

```
%sql select min(date) from spacexdataset where "Landing _Outcome" like '%Success%';
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lc
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000kg and 6000kg

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000kg but less than 6000kg.

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 booster_version from spacexdataset where "Landing _Outcome" = 'Success (drone ship)' and payload_mass__kg_ between 4
000 and 6000;
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
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 used key words **COUNT** and **GROUP BY** to find the total number of successful and failure mission outcomes.

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(mission_outcome) as count_outcomes from spacexdataset group by mission_outcome;
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

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

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

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

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

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

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

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

```
%sql select "Landing _Outcome", booster_version, launch_site from spacexdataset where "Landing _Outcome" = 'Failure (drone shi
p)' and extract(year from date) = 2015;
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

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

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

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("Landing _Outcome") as Count_Landing_outcome from spacexdataset
    where date between '2010-06-04' and '2017-03-20' group by "Landing _Outcome" order by Count_Landing_outcome desc;
```

```
 * ibm_db_sa://qgp41604:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

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

Section 3

Launch Sites Proximities Analysis

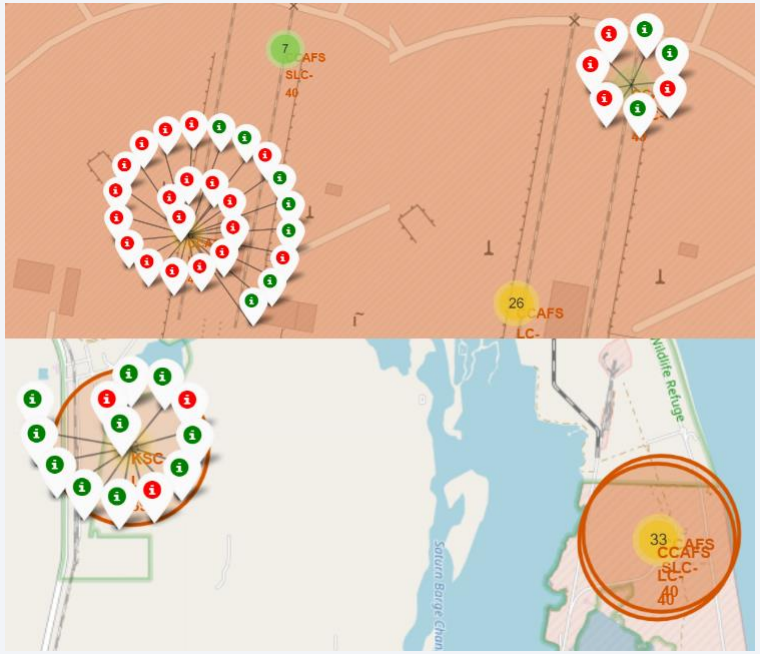
All launch sites global map markers

- We can see that the SpaceX launch sites are at the coasts of USA, Florida and California.

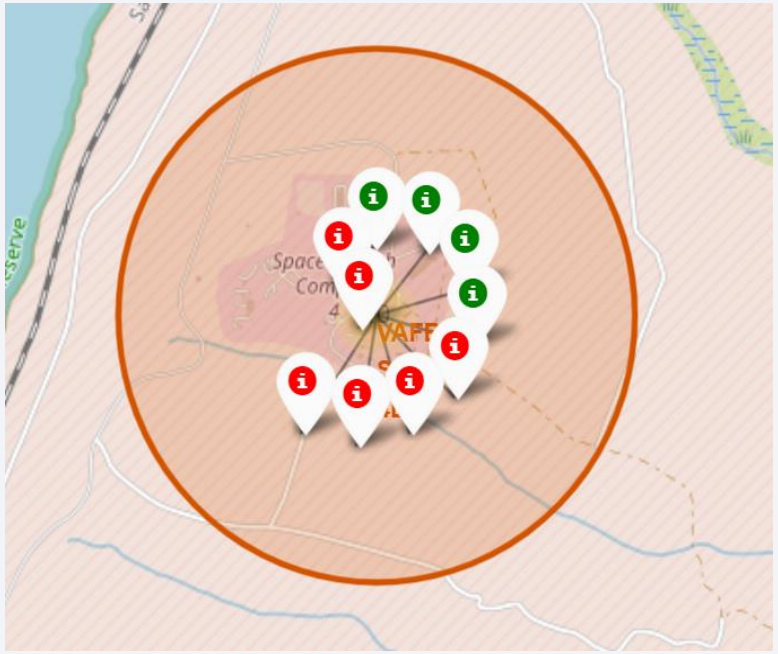
Markers showing launch sites with color labels

- The Green Marker shows the successful launch, and the Red Marker shows the failed launch.

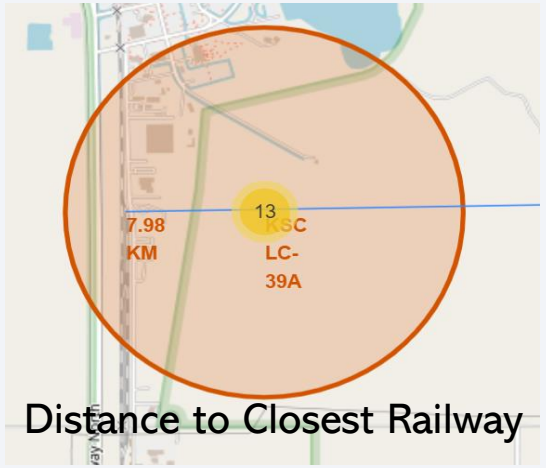
Florida Launch Sites



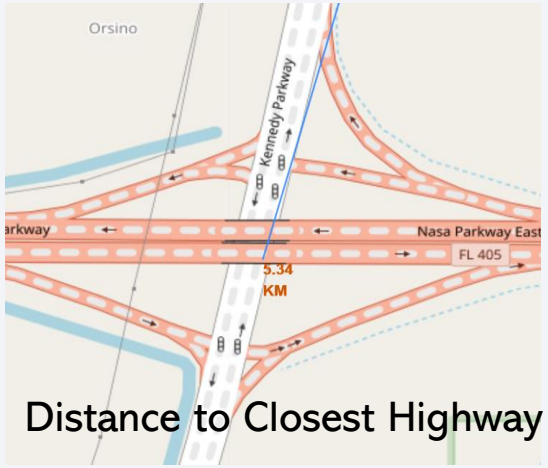
California Launch Site



Launch Site distance to landmarks

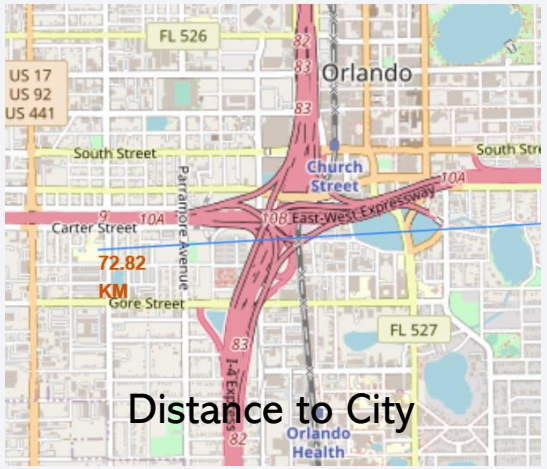


Distance to Closest Railway



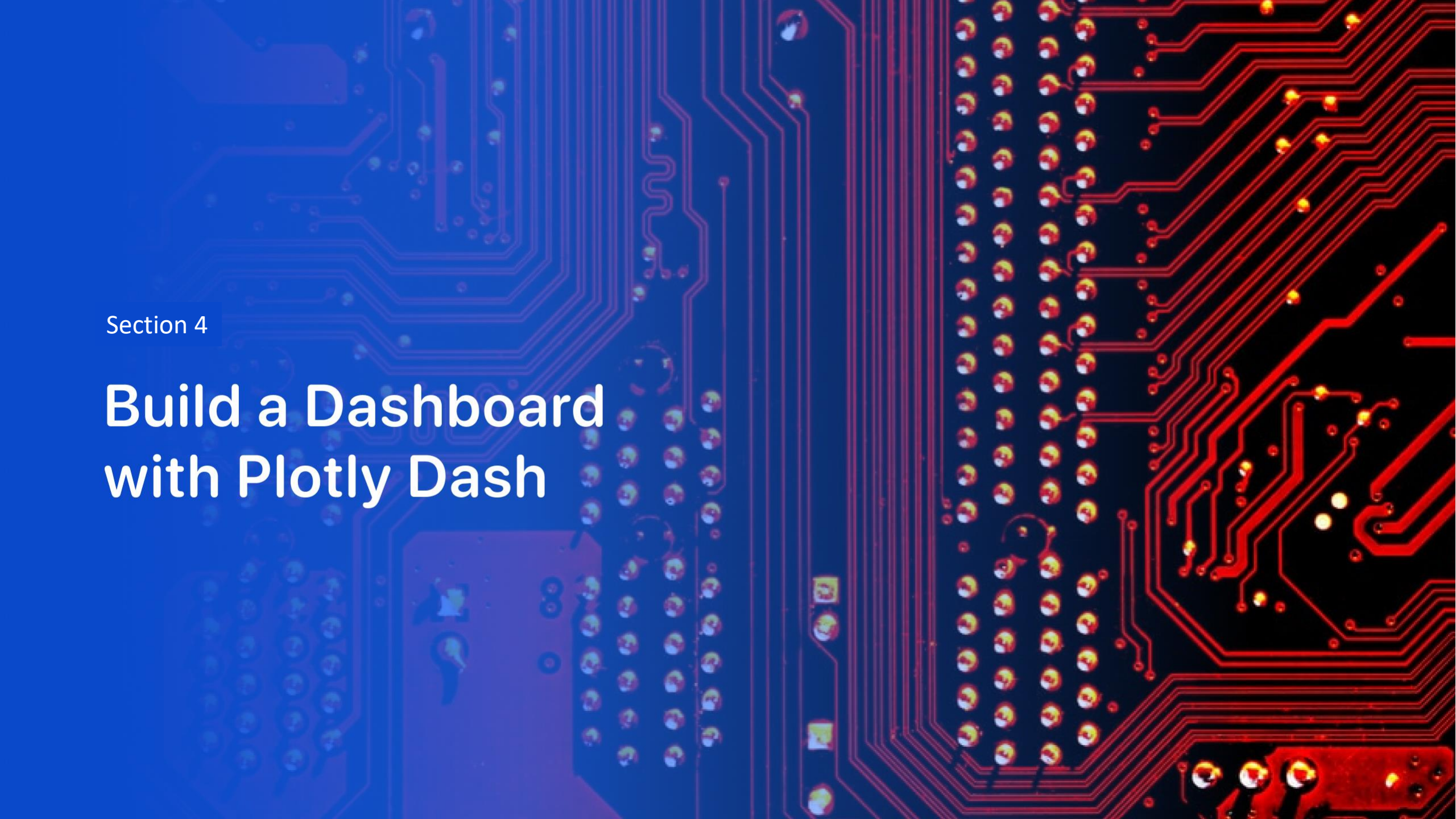
Distance to Closest Highway

Distance to Coast

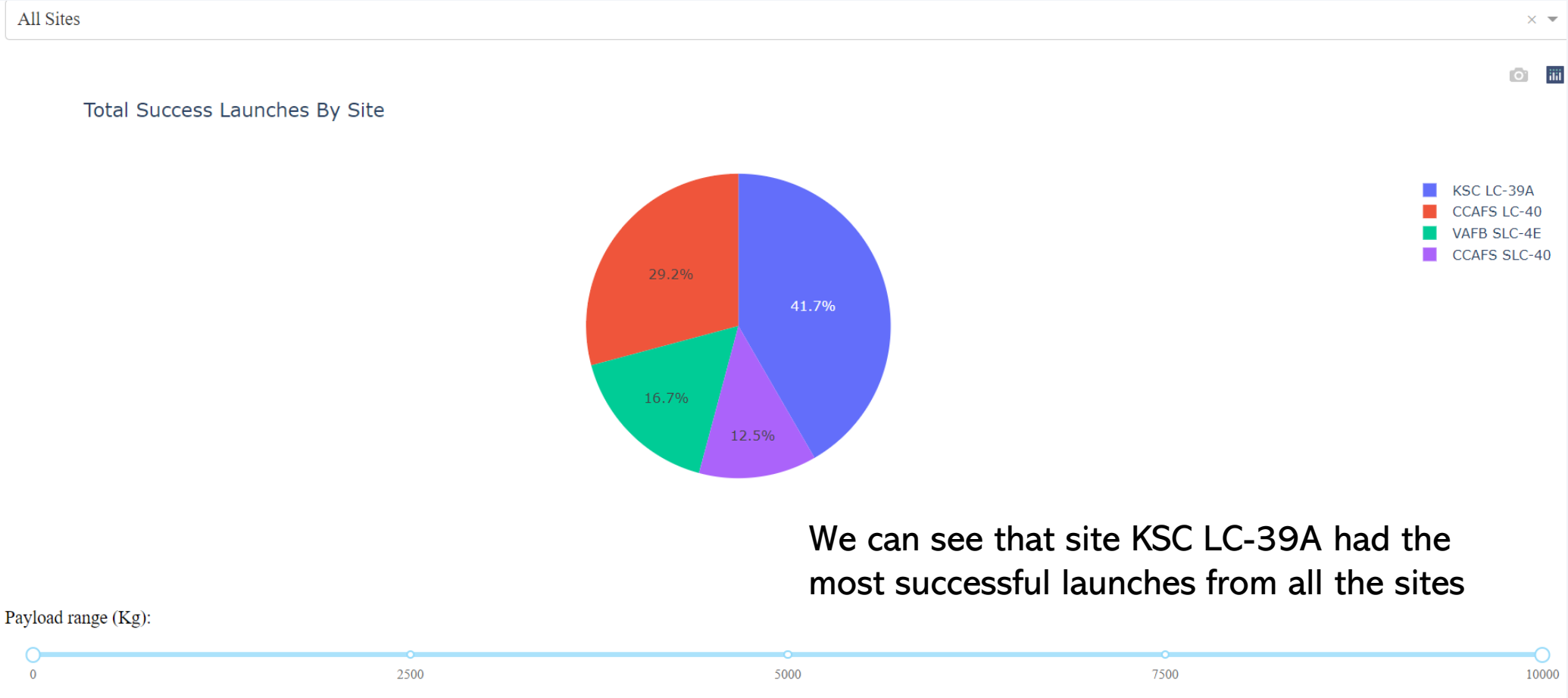


Distance to City

- Are launch sites in close proximity to railways? Yes.
- Are launch sites in close proximity to highways? Yes.
- Are launch sites in close proximity to coastline? Yes.
- Do launch sites keep certain distance away from cities? No.

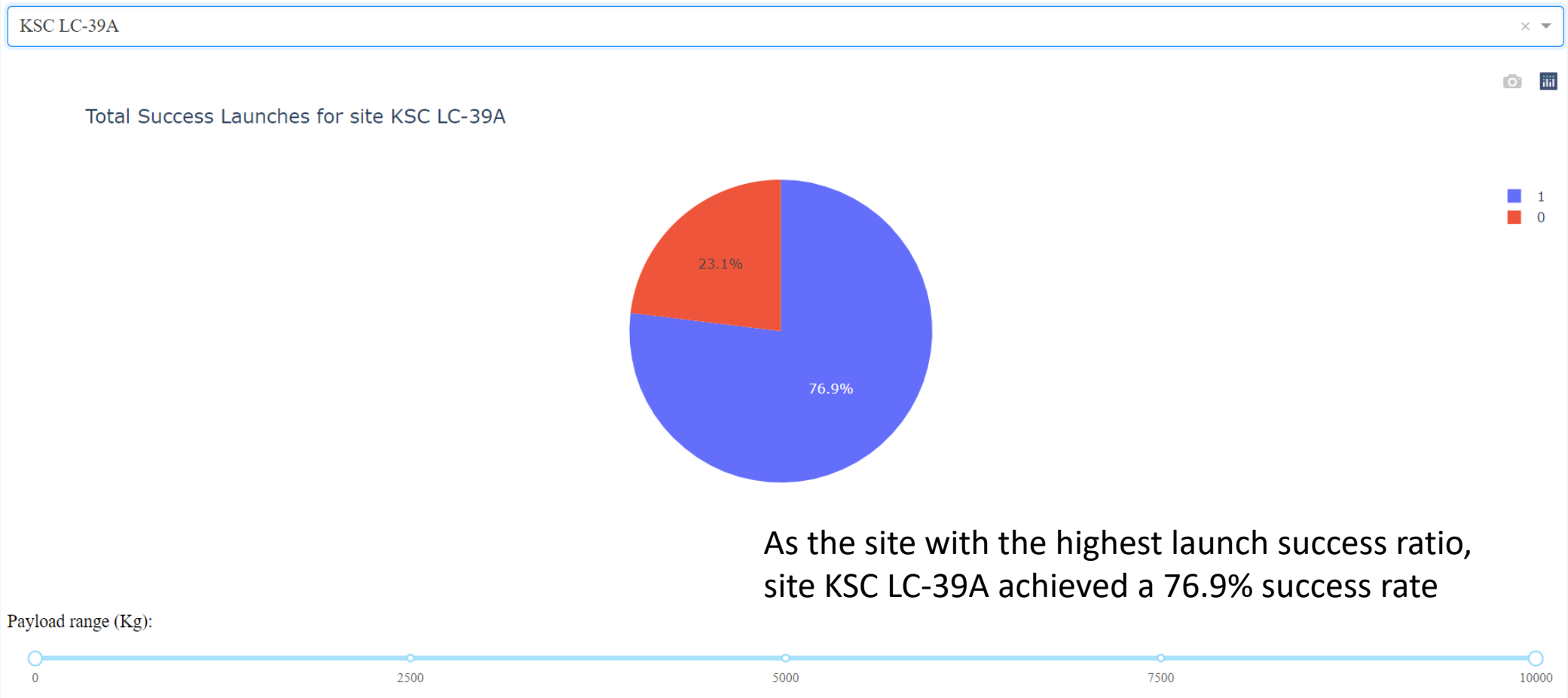

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



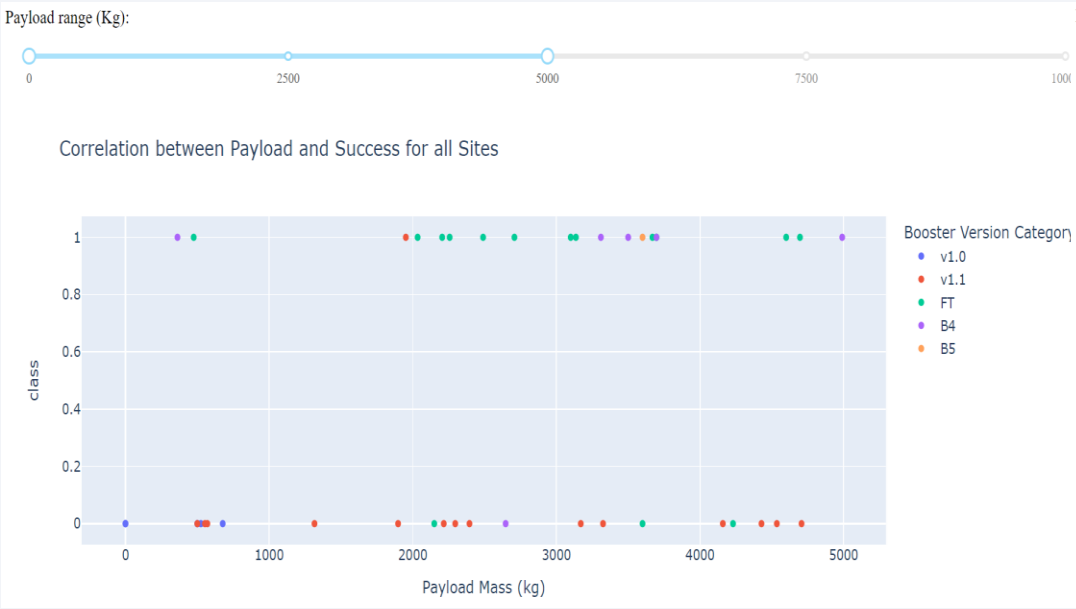
We can see that site KSC LC-39A had the most successful launches from all the sites

Pie chart showing the Launch site with the highest launch success ratio

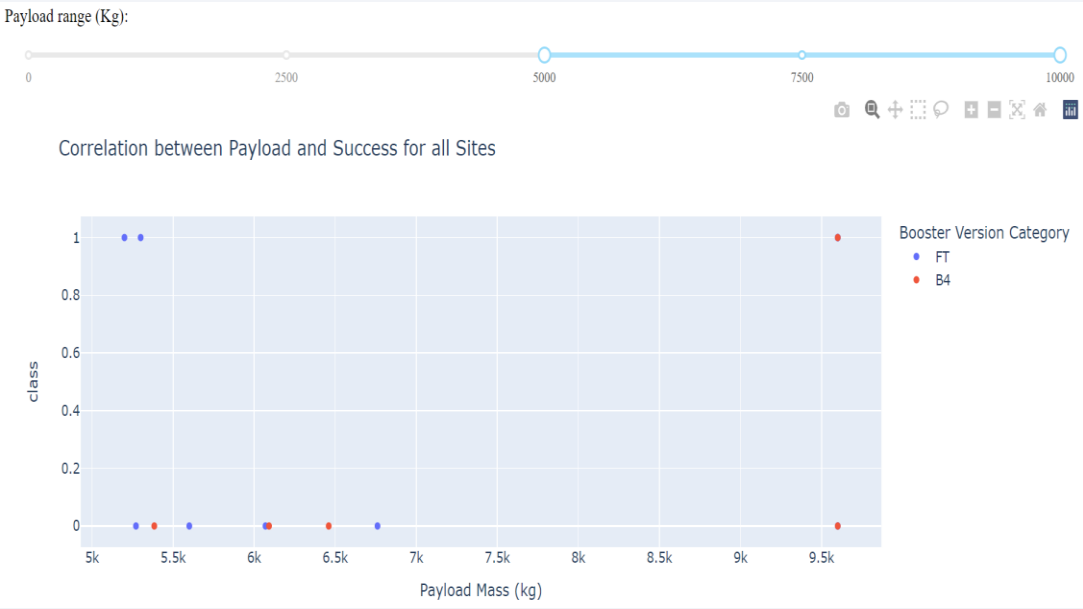


Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

Light Weighted Payload 0kg – 5000kg



Heavy Weighted Payload 5000kg – 10000kg

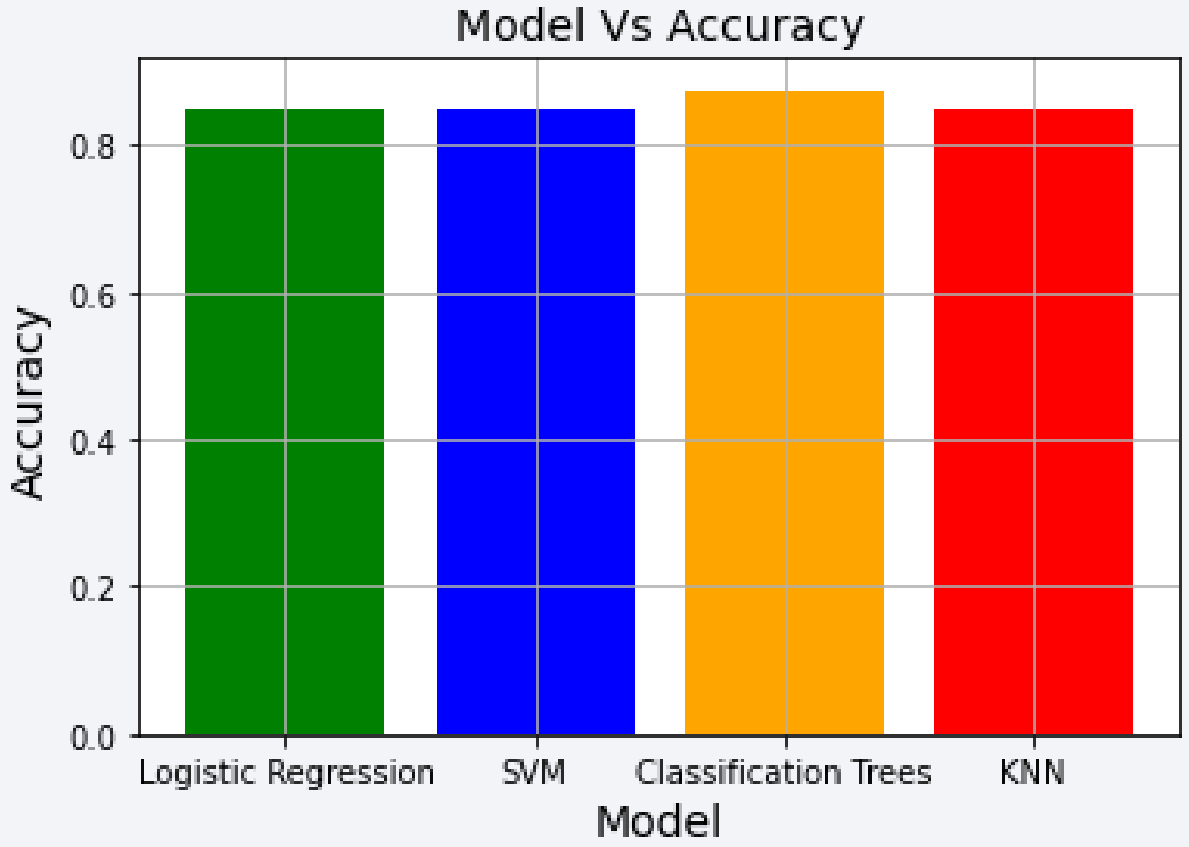


We can see the success rate for light weighted payloads is higher than the success rate for the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy



- We can see the Logistic Regression, SVM, and KNN performed mostly the same while the Classification Trees performed slightly better.

Confusion Matrix

- 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 landing marked as successful landing by the classifier.

Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- GitHub URL: https://github.com/Jerry-Yu-Zhang/IBM-Applied-Data-Science-Capstone.git

Thank you!