IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anuradha Edirisuriya

May 27,2024

Outline

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

Executive Summary

- **Summary of methodologies:**
  - Data collection through API.
  -Data collection with Web Scraping.
  - Data Wrangling
  - Exploratory Data Analysis with SQL.
  -Exploratory Data Analysis with Data Visualization
  -Interactive Visual Analytics with Folium.
  -Machine Learning Prediction.
- **Summary of all results:**
  -Exploratory Data Analysis result.
  -Interactive analytics in screenshots
  -Predictive Analytics result.

Introduction

- **Project background and context**

SpaceX advertises Falco 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. This goad of the project is to create a machine pipeline to predict if the first stag will land successfully.

- Problems we want to find answers:

-- What factors determine if the rocket will land successfully?

-- The interaction amongst various features that determine the success rate of a successful landing.

-- What operating conditions needs to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Open-Source Rest API
  - Web Scraping from Wikipedia page List of Falcon 9 and Falcon Heavy Launches
- Perform data wrangling
  - Transforming categorical data using One Hot Encoding for machine learning algorithms and removing any empty or unnecessary information from the data set.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Tree Models have been developed to determine the most effective classification method.

Data Collection

The data sets are collected using 2 methods:

1). **Request to the SpaceX API**

- Gathered SpaceX's past launch data via their open-source API.
- Retrieved and process this data with GET request.
- Ensured the data included only falcon 9 launches.
- Filled in missing payload weights from secret mission with average values.

2).Web Scraping:

- Requested past Falcon 9 and Falcon Heavy Launch data from Wikipedia's relevant page.
- Accessed the Falcon 9 Launch page via its direct Wikipedia link.
- Extracted all the column names from the HTML table.
- Parsed and transformed the table into a Pandas data frame suitable for analysis.

Data Collection – SpaceX API

** We use 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 is: https://github.com/eanuradha2024/DS-1/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1.Getting response from API:

spacex_url=https://api.spacexdata.com/v4/launches/past

response = requests.get(spacex_url)

2.Converting Response to a .json file:

respjson = response.json()

data = pd.json_normalize(respjson)

3.Apply custom functions to clean data:

getLaunchSite(data)

getPayloadData(data)

getCoreData(data)

getBoosterVersion(data)

4.Assign list to dictionary then dataframe:

5.Filter dataframe and export to flat file(.csv)

Data Collection - Scraping

1.We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup.

2.Then we parse the table and converted into a pandas dataframe.

** The link to the notebook is: https://github.com/eanuradha2024/DS-1/blob/main/jupyter-labs-webscraping.ipynb

1. Getting response from HTML.:

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

response = requests.get(static_url)

2. Creating BeautifulSoup object.:

soup = BeautifulSoup(response.content, 'html.parser')

3. Extract all column names from HTML table header:

column_names = []

table = first_launch_table.find_all('th')

for row in table:

    name = extract_column_from_header(row)

    if name is not None and len(name) > 0:

        column_names.append(name)

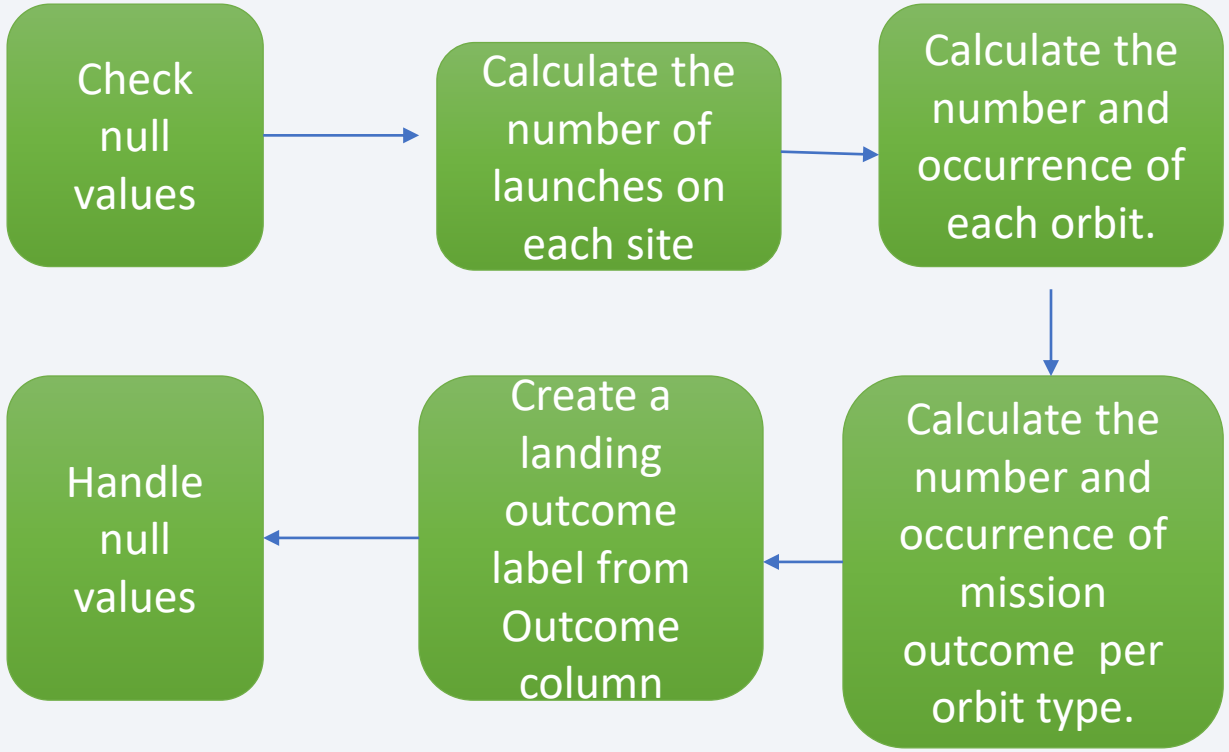
4. Create a dataframe by parsing the launch HTML tables. 5.Export data to CSV

Data Wrangling

We **performed exploratory data analysis and determine the training labels. Then we calculate the number of launches at each site, and the umber and occurrence of each orbits. Then we created landing outcome label from outcome column and exported the results to csv.

**The link to the notebook is:

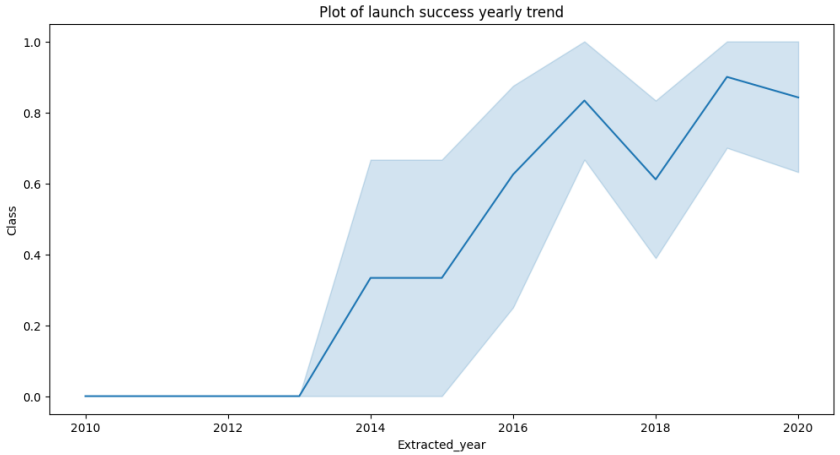
https://github.com/eanuradha2024/DS-1/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

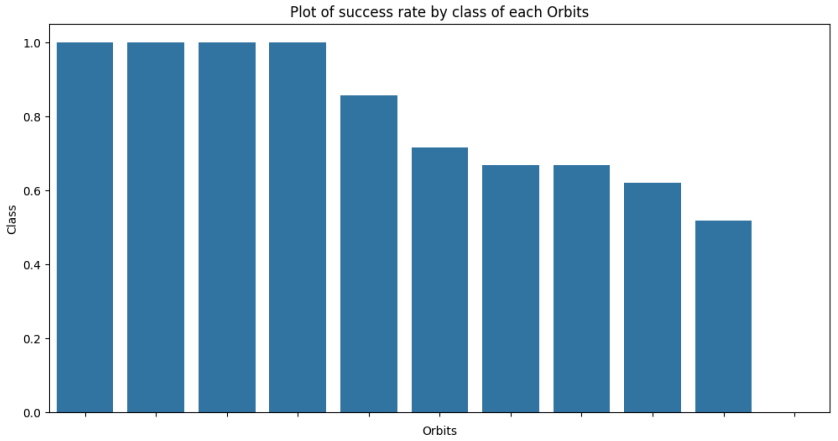


EDA with Data Visualization

**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 between the notebook is: https://github.com/eanuradha2024/DS-1/blob/main/EDA_Visualization.ipynb





EDA with SQL

- After loading SapceX dataset into PostgreSQL database , Applied EDA with SQL to get insight from the data. I wrote queries to find out the instance:

  * The name 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 v 1.1

  *The total number of successful and failure mission outcomes.

  *The failed landing outcomes in drone ship, their booster version and launch site names.

The link to notebook is: https://github.com/eanuradha2024/DS-1/blob/main/jupyter-labs-eda-sql-coursera_sqllite.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 launch outcomes and o for failure, 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distance between a launch site to its proximities. We focus some questions for instance:

** Are launch sites near railways, highways and coastlines.

** Do launch sites keep certain distance away from cities.

The link to notebook URL: https://github.com/eanuradha2024/DS-1/blob/main/Visual%20Analytics%20with%20Folium%20Lab.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.

**the link to the notebook is:
https://github.com/eanuradha2024/DS-1/blob/main/spacex_Plotty_dash_app.py

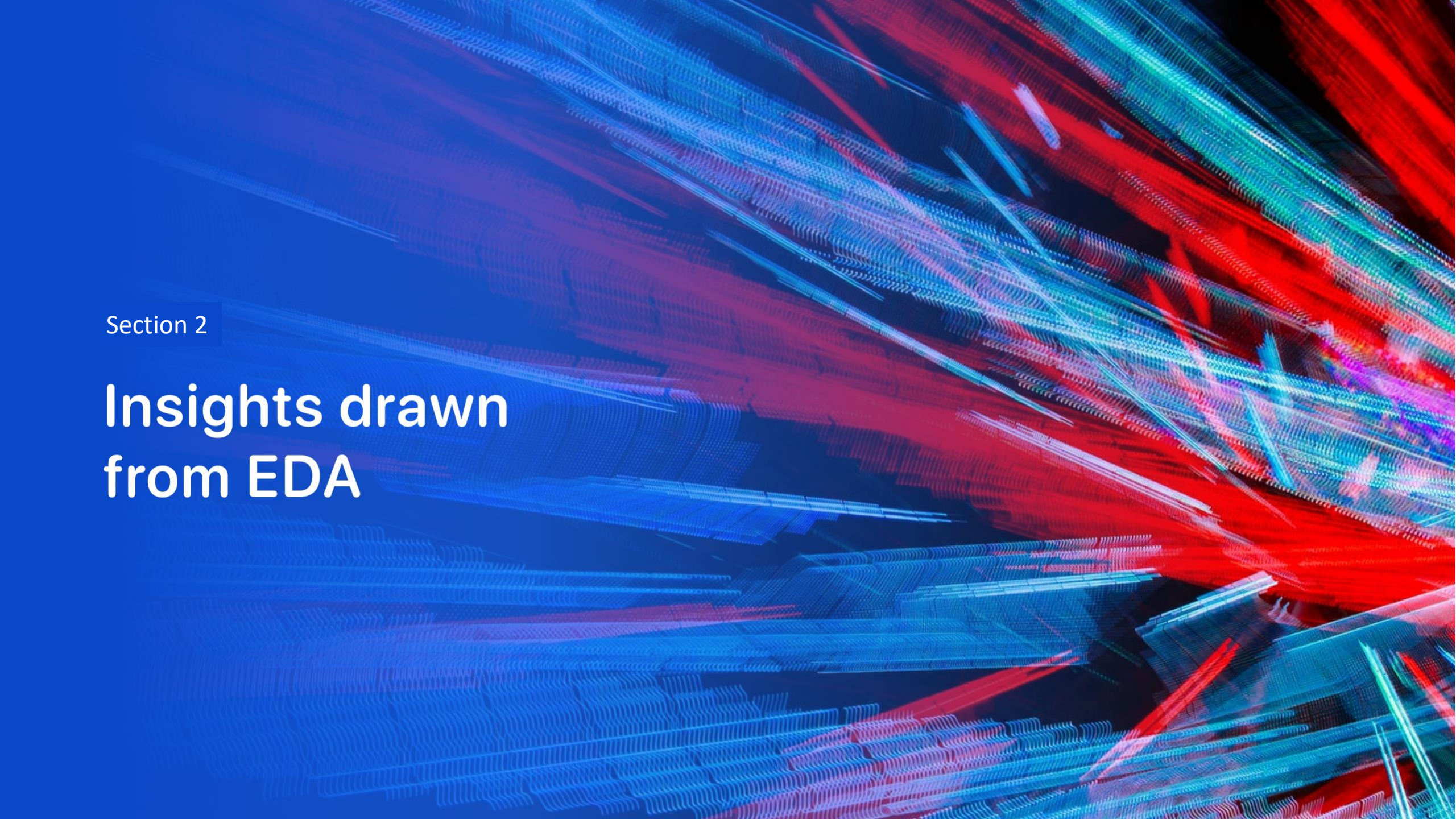
Predictive Analysis (Classification)

- 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 is: https://github.com/eanuradha2024/DS-1/blob/main/Machine%20Learning%20Prediction.ipynb

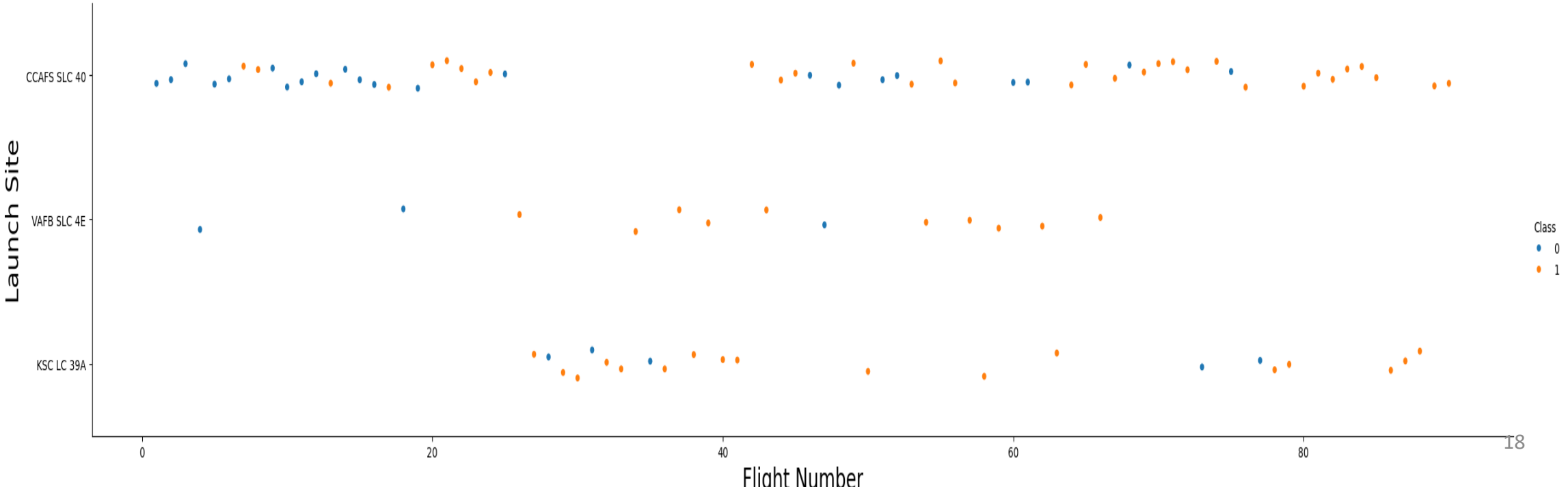
Results

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


Section 2
Insights drawn from EDA

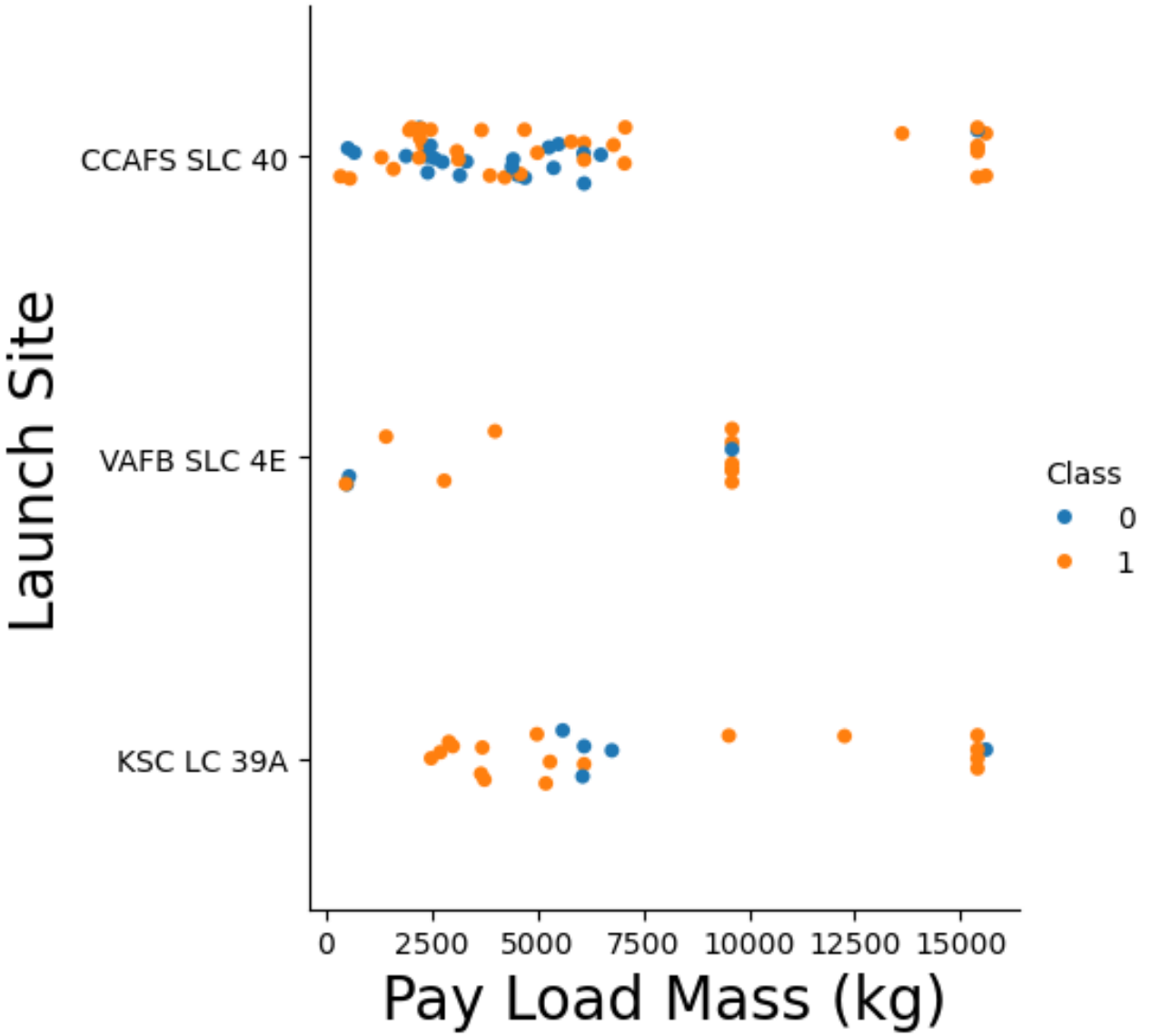
Flight Number vs. Launch Site

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



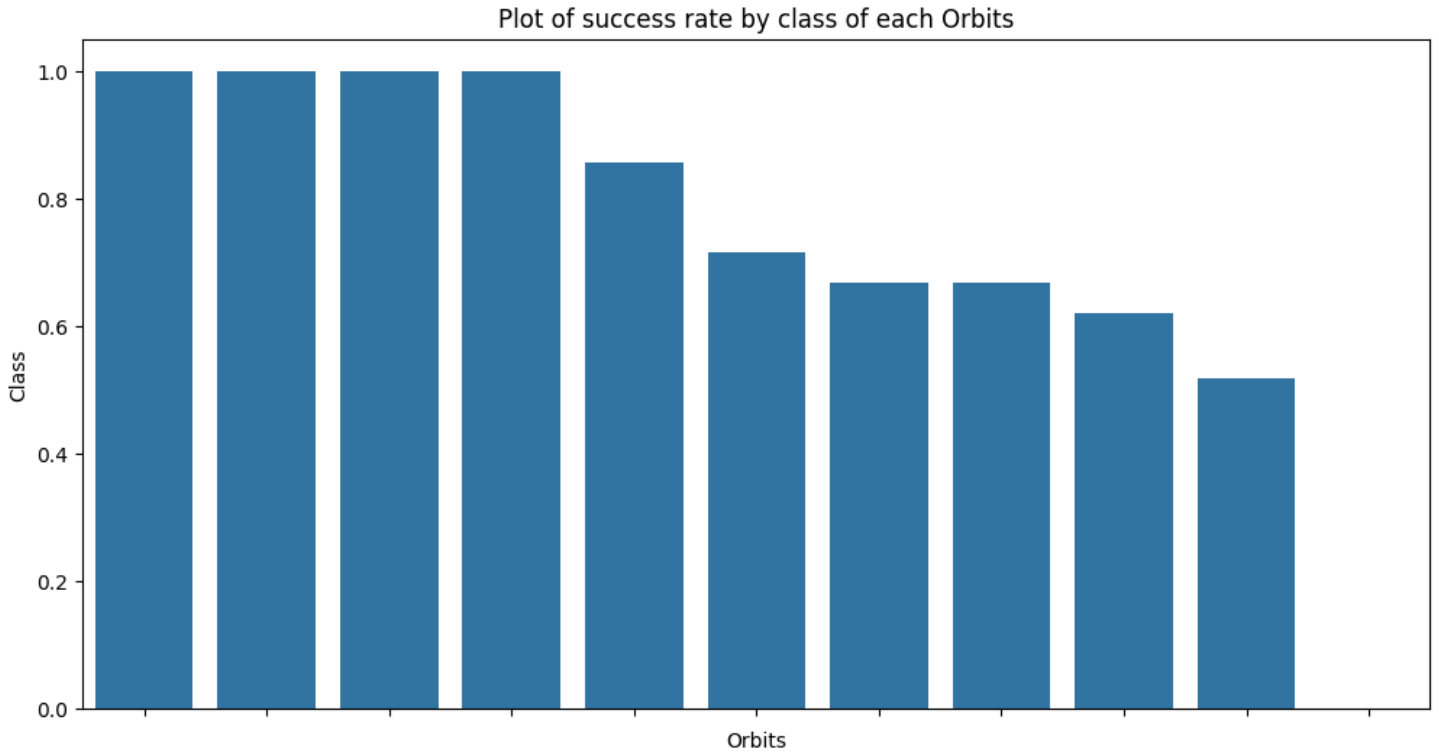
Payload vs. Launch Site

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



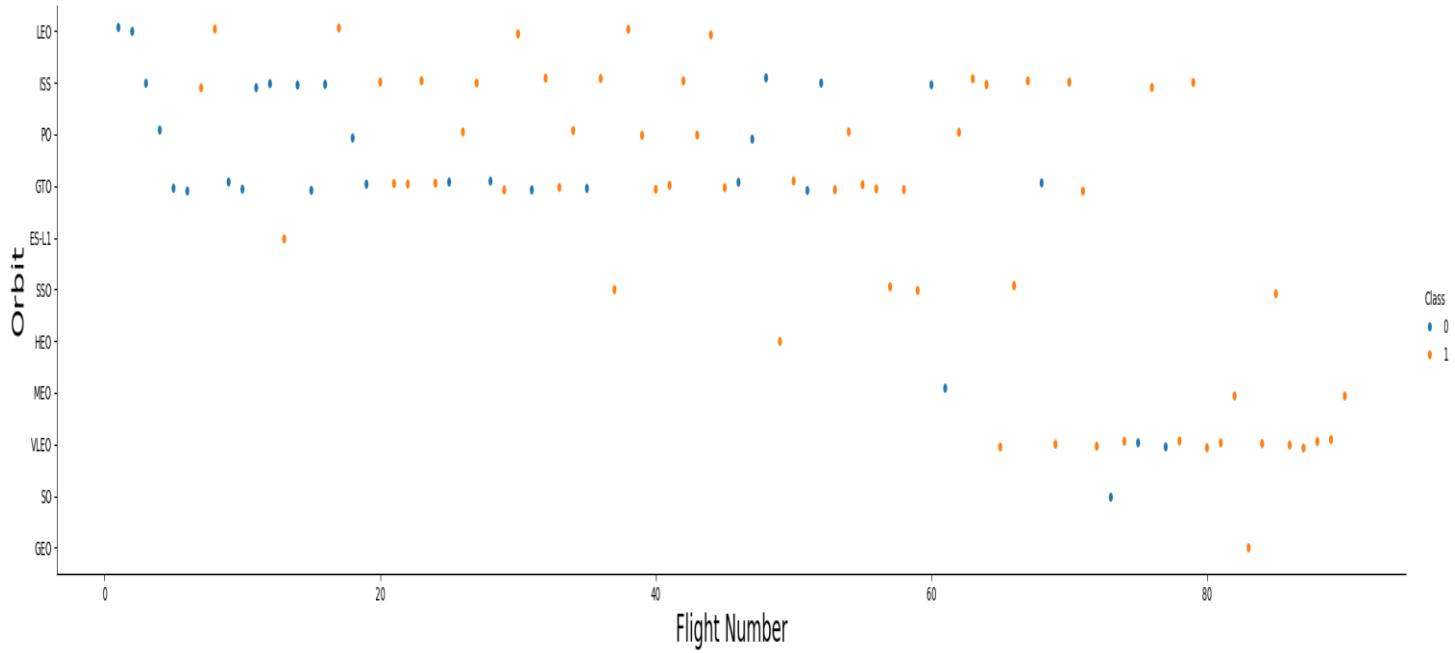
Success Rate vs. Orbit Type

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



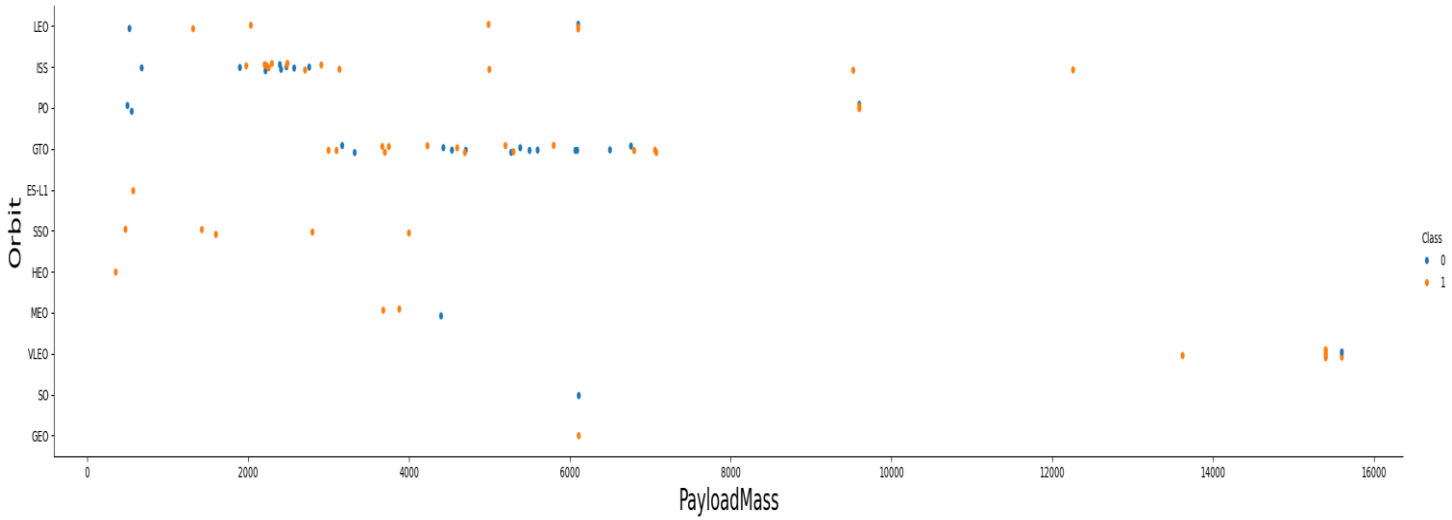
Flight Number vs. Orbit Type

** 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 orbit.



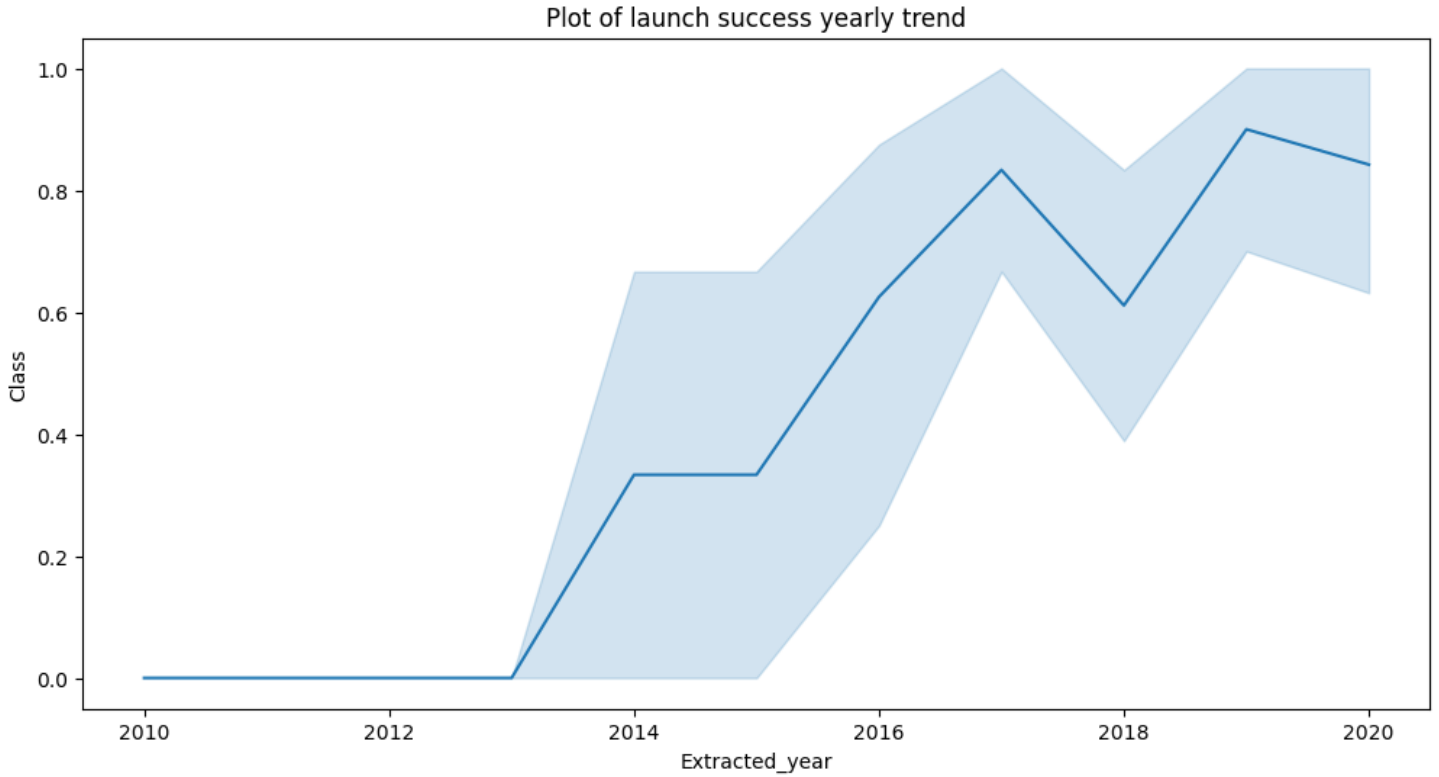
Payload vs. Orbit Type

**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 SPACEX data.

Task 1

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

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

**we used the query to display 5 records where launch sites begin with “CCA”.

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

```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

**We calculated the total payload carried by boosters from NASA as 45596 using the query below.

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_payloadmass from SPACEXTBL where CUSTOMER like 'NASA (CRS)'
```

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

Total_payloadmass
45596

Average Payload Mass by F9 v1.1

* We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) as AVERAGE_PAYLOADMASS from SPACEXTBL where BOOSTER_VERSION like "F9 v1.1"
```

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

AVERAGE_PAYLOADMASS
2928.4

First Successful Ground Landing Date

- We observed that that the dated of the first successful landing outcome on ground pad was 22nd December 2015.

Task 5

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = "Success (ground pad)"
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

** We 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 4000, but less that 6000.

Task 6

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 SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000
```

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

]:

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

Total Number of Successful and Failure Mission Outcomes

- **We used wildcard like “%” to filter for WHERE MissionOutCome was a success or a failure.

Task 7

List the total number of successful and failure mission outcomes

```
%sql select COUNT(MISSION_OUTCOME) AS TOTAL_MISSION_OUTCOME  FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE "Success%" or MI
```

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

TOTAL_MISSION_OUTCOME
101

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  BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) F
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT BOOSTER_VERSION,LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE "failure%" and DATE
```

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

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

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 WHRE 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 gr

- output landing outcome in descending order.

Task 10

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) FROM SPACEXTBL WHERE DATE BETWEEN "2010-06-04" AND "2017-03-20" GRO
```

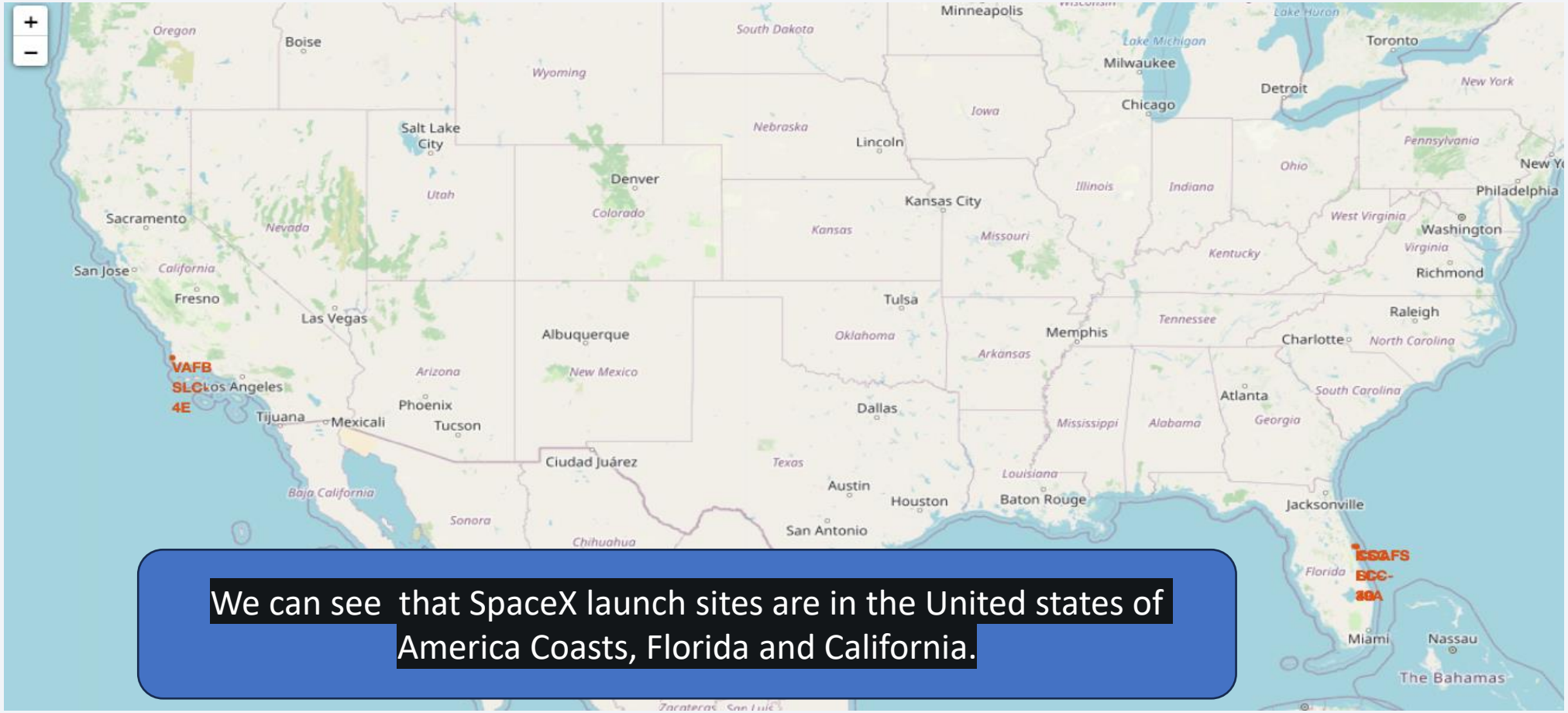
```
* sqlite:///my_data1.db
Done.
```

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

Section 3

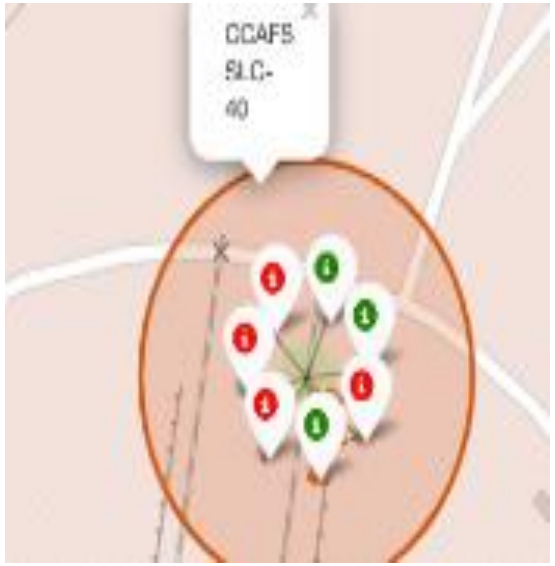
Launch Sites Proximities Analysis

All Launch sites global map markers



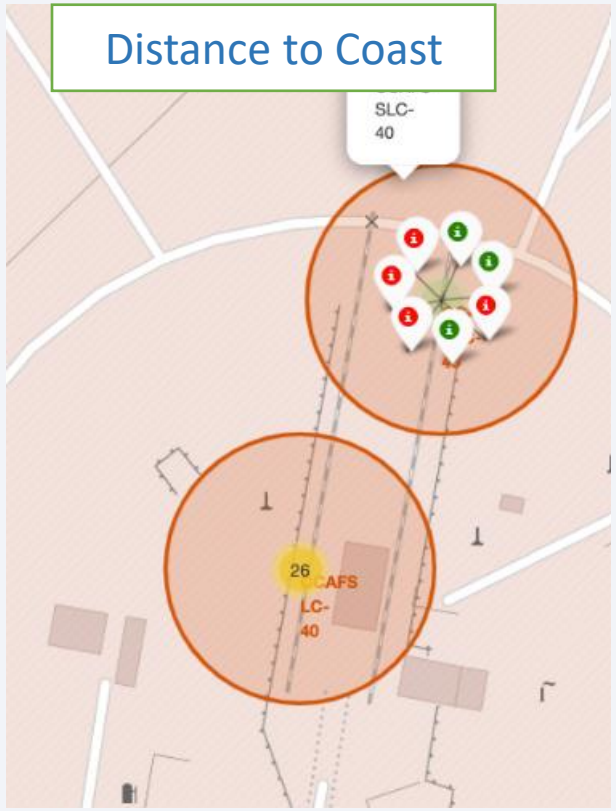
We can see that SpaceX launch sites are in the United states of America Coasts, Florida and California.

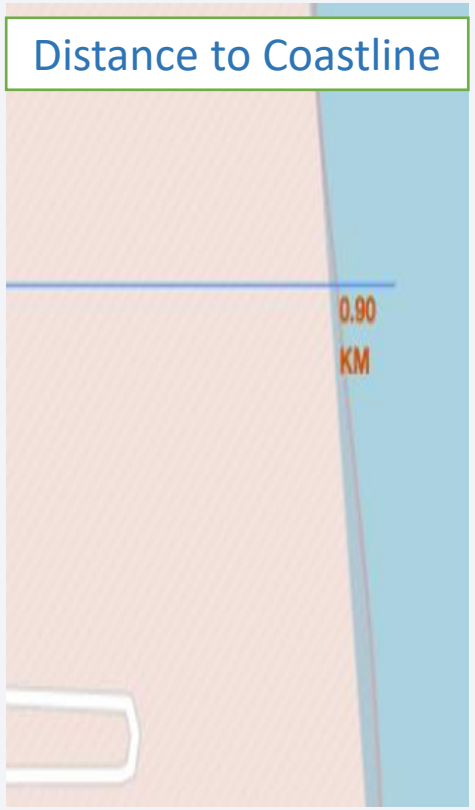
Markers showing launch sites with color labels



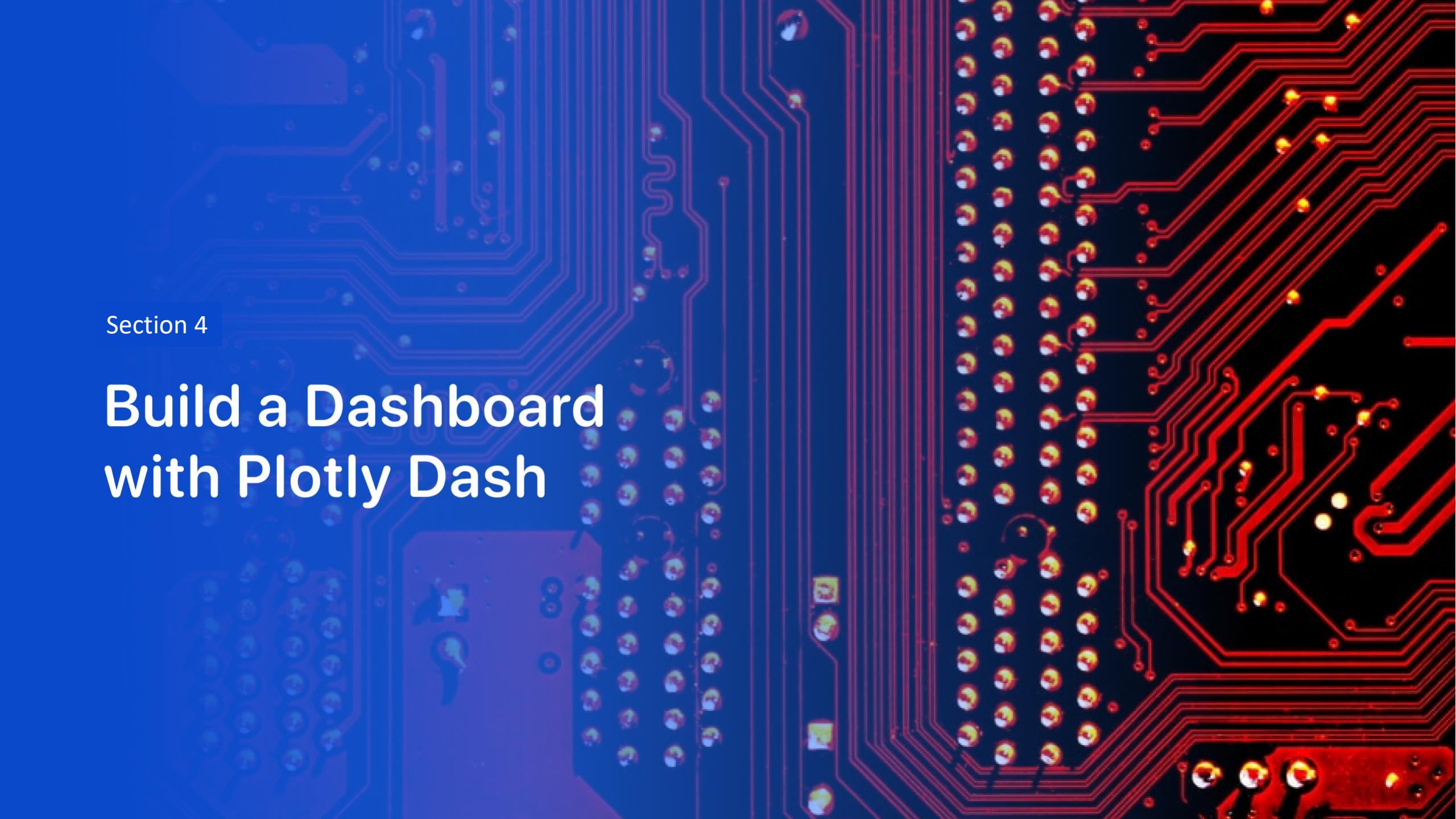
** Green Markers shows SUCCESSFUL LAUNCHES and Red Markers shows FAILURE LAUNCHES.

Launch Site distance to Landmarks:

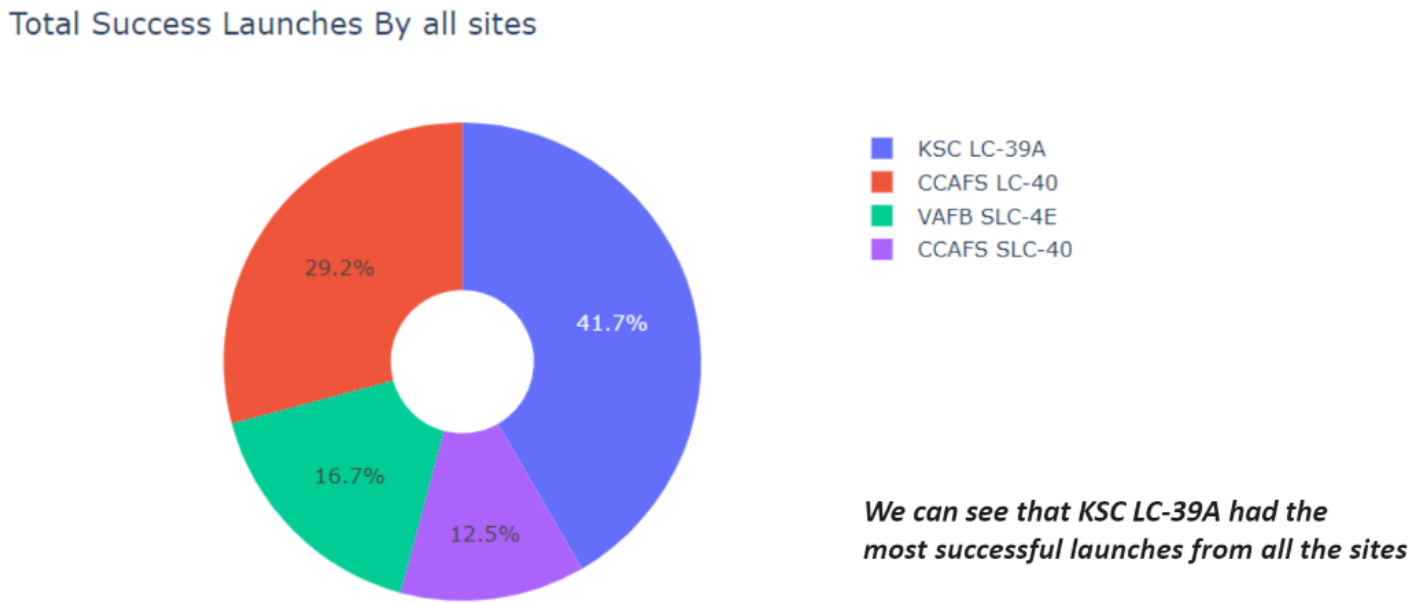




*Are Launch sites in close proximity to railways?
No
*Are Launch sites in close proximity to highways?
No
* Are launch sites in close proximity to coastline?
Yes
*Do launch sites keep certain distance away from cities? Yes

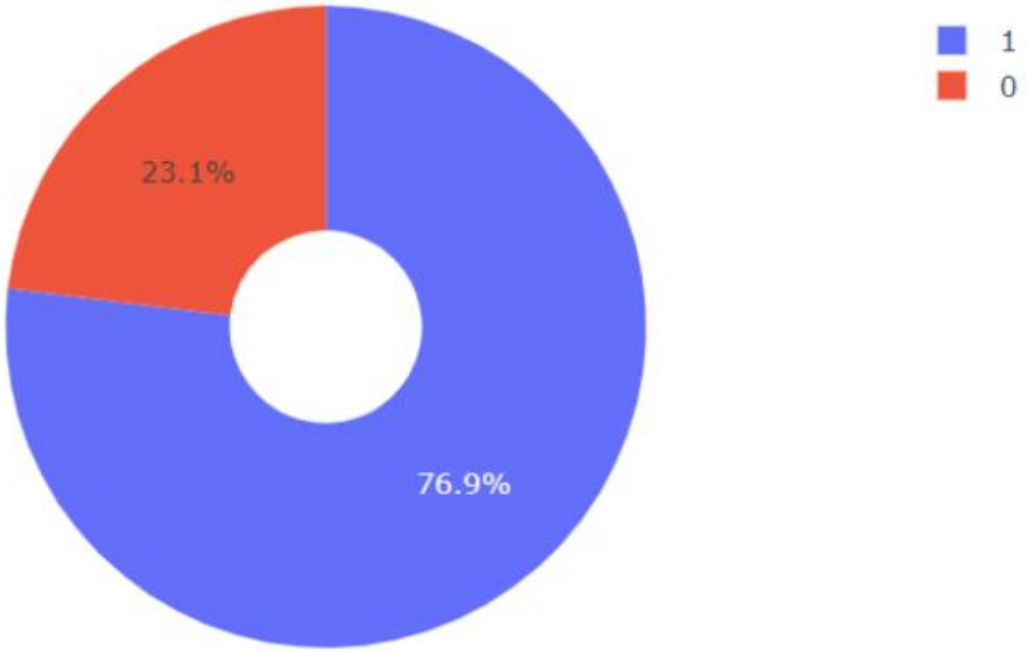

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site:



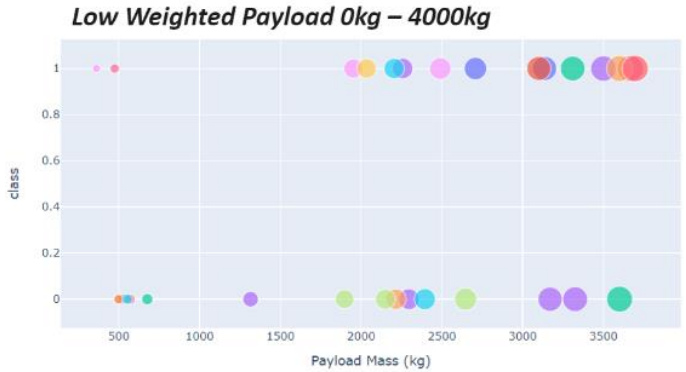
We can see that KSC LC-39A had the most successful launches from all the sites

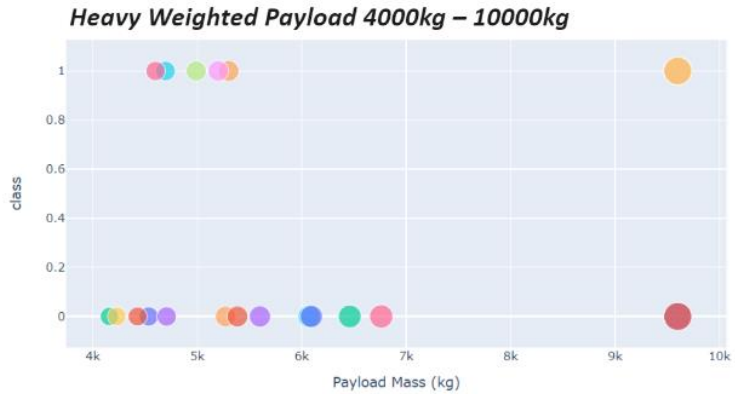
Pie chart showing the Launch site with the highest launch success ratio:



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

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





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

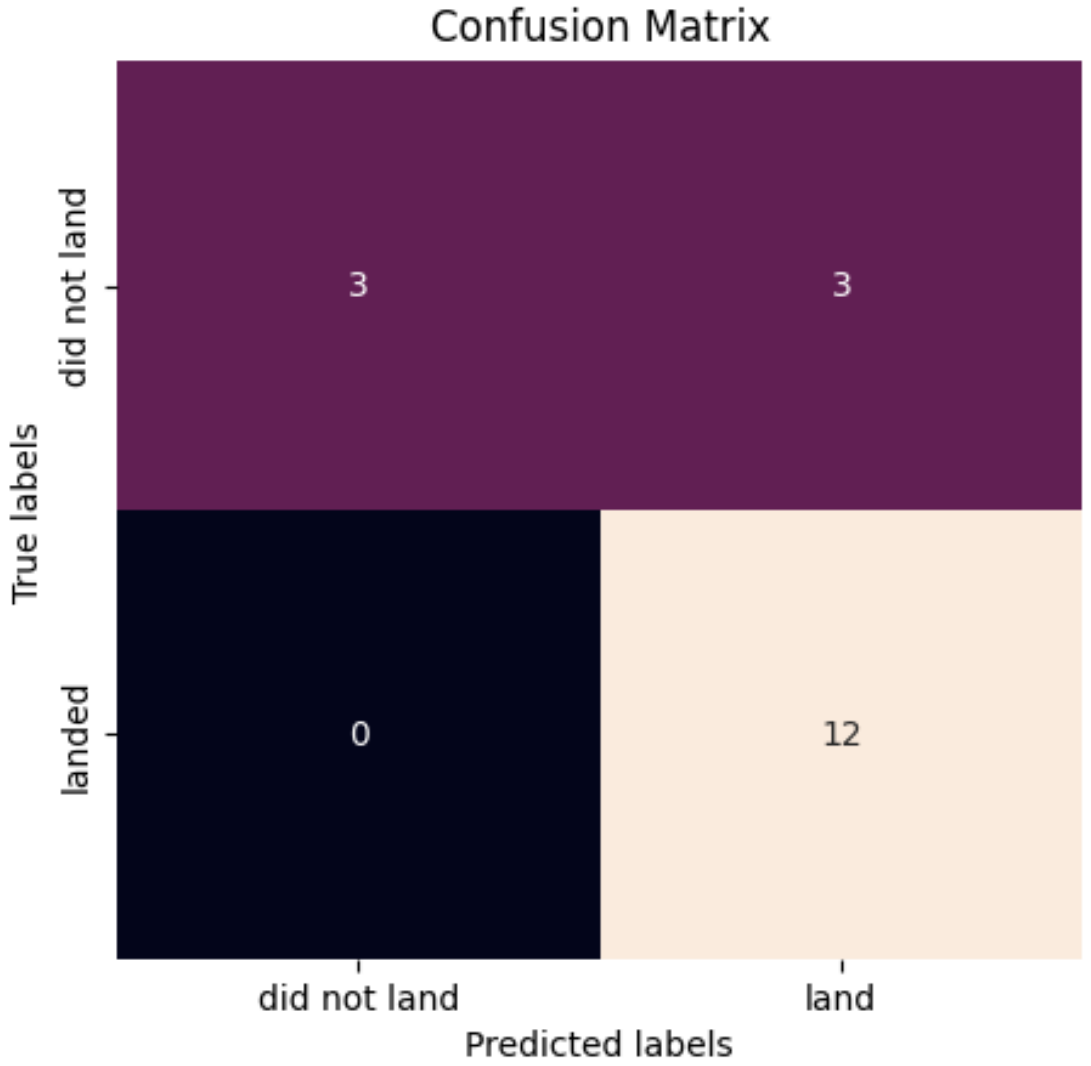
```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Classification Accuracy

* The decision tree classifier is the model with highest classification accuracy.

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. Ex: 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.

Thank you!