



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

AFIF JETHAM  
14/02/2022



# 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

Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you 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:
  - 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
  - How to build, tune, evaluate classification models.

# Data Collection

---

- The data was collected using various methods
  - Data collection was done using 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 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 is

[https://github.com/afifjetham/CapstoneProject/blob/main/Capstone\\_DataCollection\\_API.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/Capstone_DataCollection_API.ipynb)

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
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'])
```

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

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

```
response = requests.get(spacex_url)
```

Check the content of the response

```
print(response.content)
```



# Data Collection - Scraping

---

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup.
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is

[https://github.com/afifjetham/CapstoneProject/blob/main/Capstone\\_DataCollection\\_Webscrapping.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/Capstone_DataCollection_Webscrapping.ipynb)

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_L
```

Next, request the HTML page from the above URL and get a `response` object

## TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, "html.parser")
```

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

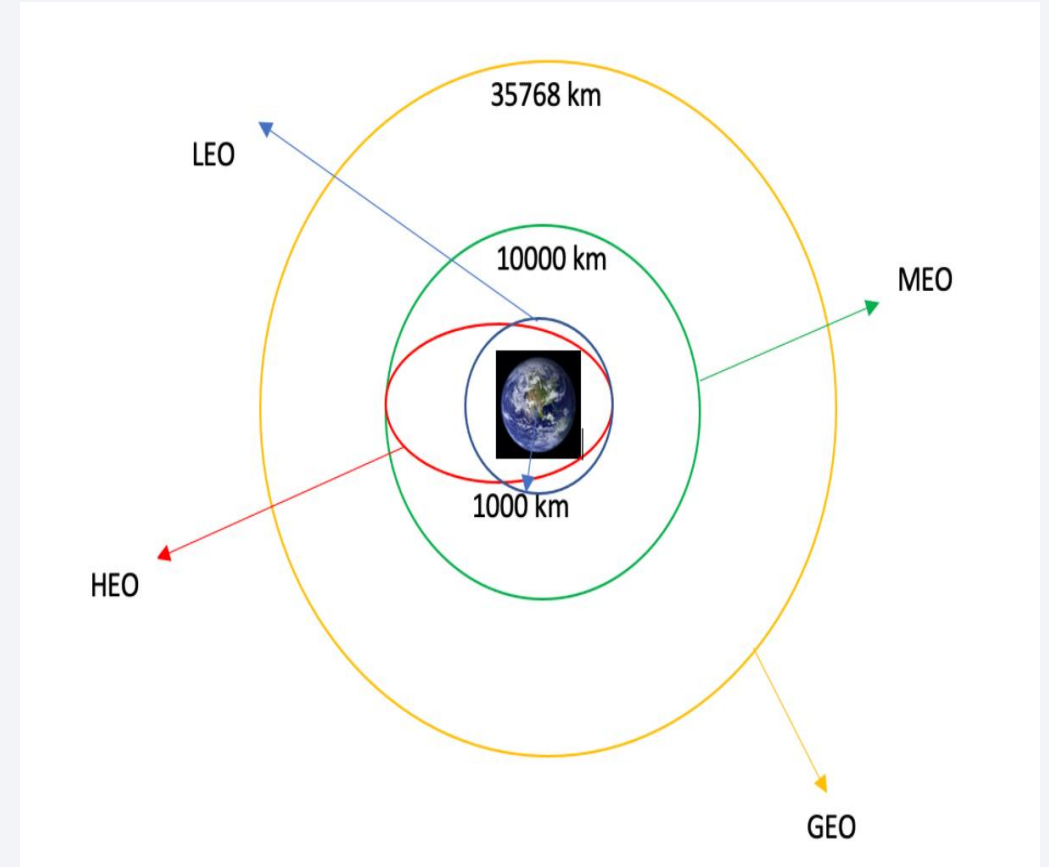
```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

# Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- 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 is

[https://github.com/afifjetham/CapstoneProject/blob/main/Capstone\\_DataWrangling\\_EDA.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/Capstone_DataWrangling_EDA.ipynb)

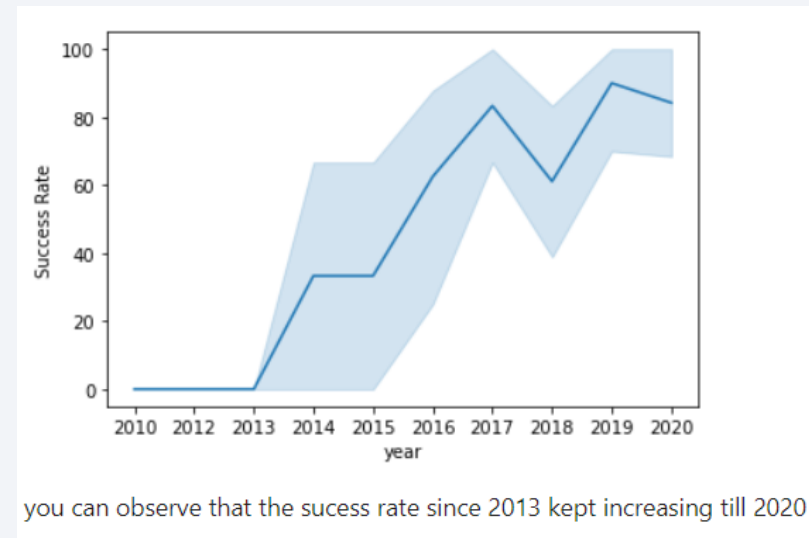
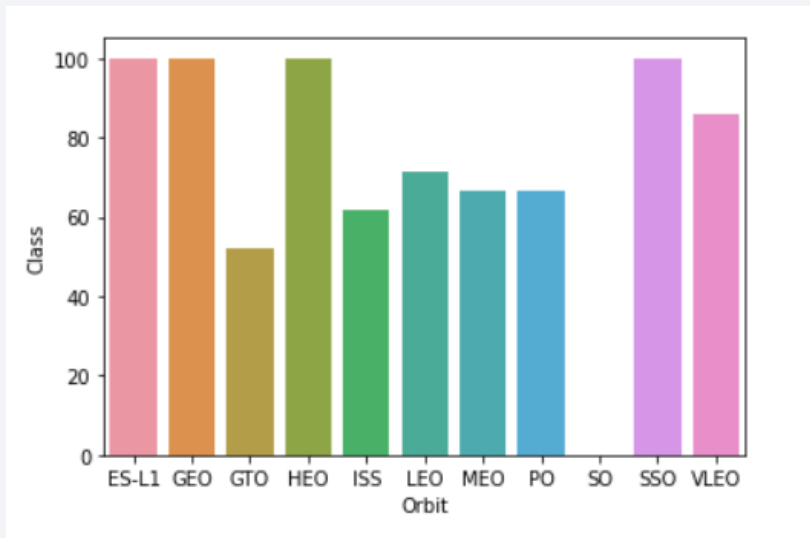


# EDA with Data Visualization

---

- 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 is

[https://github.com/afifjetham/CapstoneProject/blob/main/EDA\\_Using\\_Python.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/EDA_Using_Python.ipynb)



# EDA with SQL

---

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - 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 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 the notebook is

[https://github.com/afifjetham/CapstoneProject/blob/main/EDA\\_Using\\_SQL.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/EDA_Using_SQL.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.



# 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/afifjetham/CapstoneProject/blob/main/DataVisualization\\_Using\\_Folium.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/DataVisualization_Using_Folium.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 is

[https://github.com/afifjetham/CapstoneProject/blob/main/Machine\\_Learning\\_Prediction.ipynb](https://github.com/afifjetham/CapstoneProject/blob/main/Machine_Learning_Prediction.ipynb)

# Results

---

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



The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and teal on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

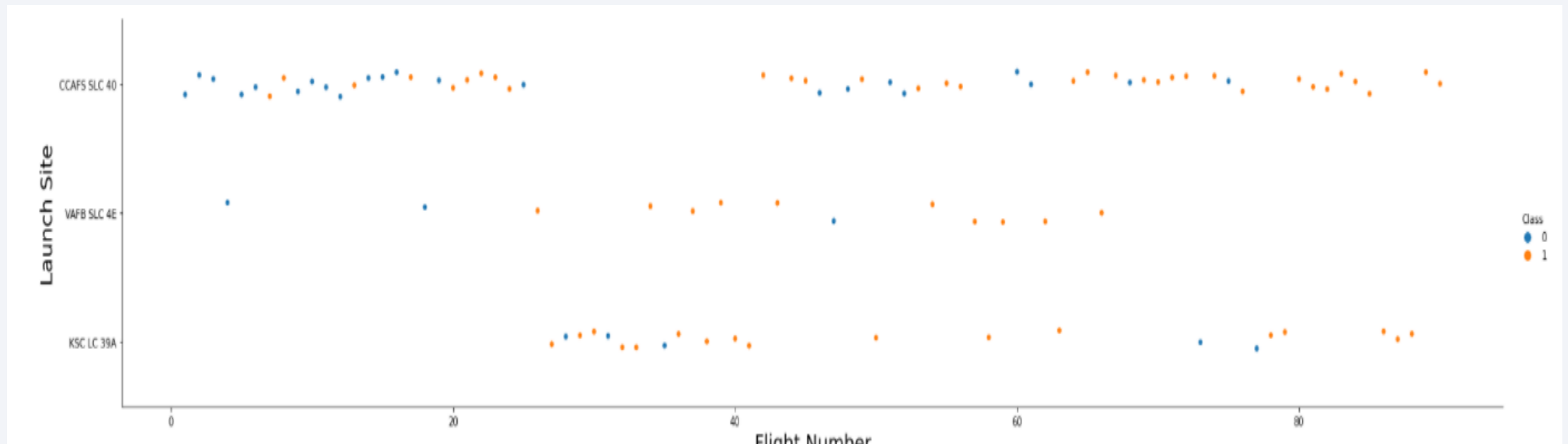
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

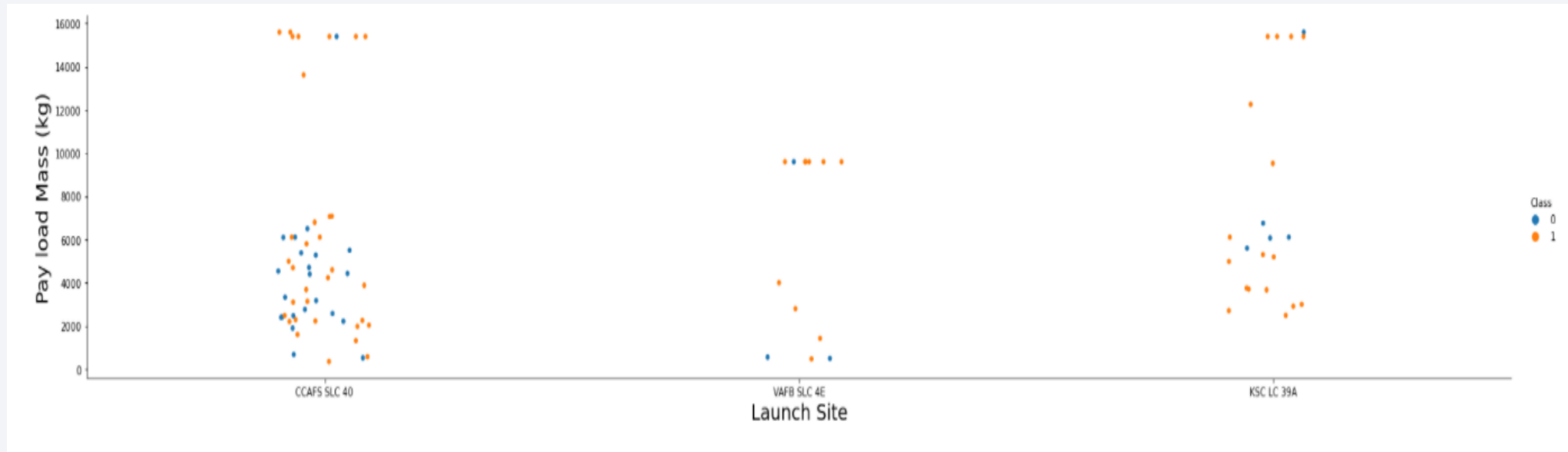
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.





# Payload vs. Launch Site

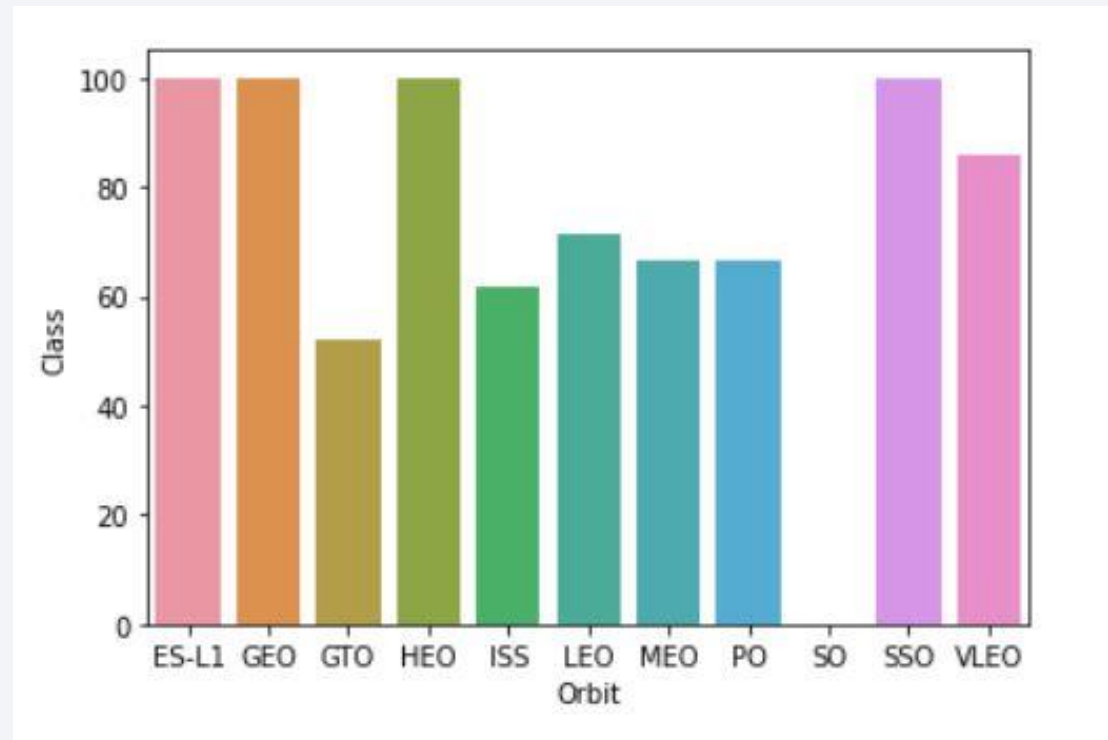
- The greater the payload mass for launch site CCAFS SLC 40, the higher the success rate for the rocket.



# Success Rate vs. Orbit Type

---

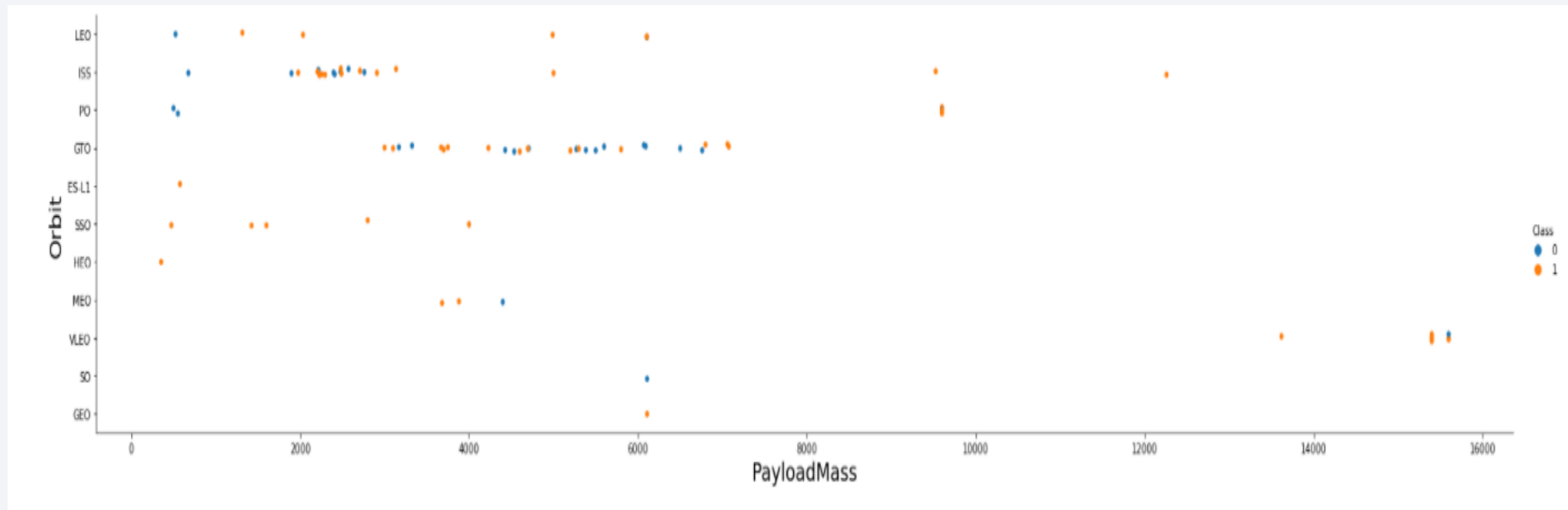
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.





# Payload vs. Orbit Type

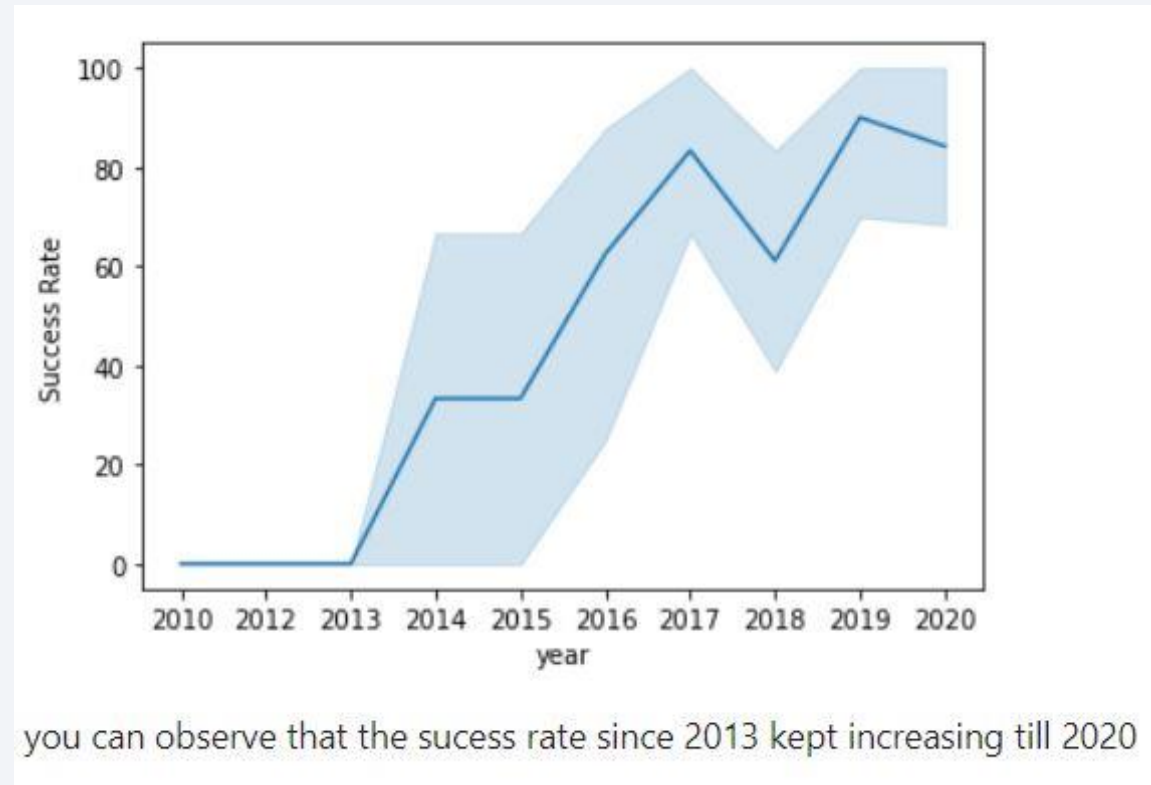
- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



# 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 SPACEXTBL;

* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87!
Done.

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

# Launch Site Names Begin with 'CCA'

- We used the below 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;
```

```
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8l1cg.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 calculated the total payload carried by boosters from NASA as 45596 using the query below.

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

```
%sql SELECT SUM(payload_mass__kg_) AS Sum FROM SPACEXTBL WHERE customer LIKE 'NASA (CRS)';
```

```
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases  
Done.
```

```
SUM
```

```
45596
```

# Average Payload Mass by F9 v1.1

---

- We used the below query to display the average payload mass carried by booster version F9 v1.1.

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

: %sql SELECT AVG(payload_mass__kg_) AS Average FROM SPACEXTBL
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87
Done.
: average
2534
```

# First Successful Ground Landing Date

---

- We observed 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 achieved.

*Hint: Use min function*

```
%sql SELECT MIN(date) AS Date FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)';  
  
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.c.  
Done.  
  
DATE  
2015-12-22
```

```
%sql SELECT date FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)' ORDER BY date LIMIT 1;  
  
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.c.  
Done.  
  
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 than 6000.

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_ BETWEEN 4000 AND 6000;
```

```
* ibm_db_sa://tdb86782:***@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 GROUP BY to filter MissionOutcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTBL GROUP by mission_outcome ORDER BY mission_outcome;
```

```
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198
Done.
```

mission_outcome	COUNT
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 booster_version FROM SPACEXTBL WHERE payload_mass__kg_=(SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8l1cg.databases.appdomain.cloud:31198/t
Done.
: booster_version
  F9 B5 B1048.4
  F9 B5 B1049.4
  F9 B5 B1051.3
  F9 B5 B1056.4
  F9 B5 B1048.5
  F9 B5 B1051.4
  F9 B5 B1049.5
  F9 B5 B1060.2
  F9 B5 B1058.3
  F9 B5 B1051.6
  F9 B5 B1060.3
  F9 B5 B1049.7
```

# 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 booster_version, launch_site, date, mission_outcome, landing__outcome FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' AND date LIKE '2015%';
```

```
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8l1cg.databases.appdoma:
Done.
```

booster_version	launch_site	DATE	mission_outcome	landing_outcome
F9 v1.1 B1012	CCAFS LC-40	2015-01-10	Success	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	2015-04-14	Success	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 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 ascending 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(*) AS Count FROM SPACEXTBL WHERE date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY Count
```

```
* ibm_db_sa://tdb86782:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8l1cg.databases.appdomain.cloud:31198/bludb
Done.
```

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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a solid blue background on the left and a satellite image of Earth on the right. The Earth's surface is dark blue, with numerous bright yellow and orange lights representing cities and urban areas. The lights are concentrated in the lower right portion of the image, following the curve of the Earth's horizon. The overall composition suggests a global or space-related theme.

Section 3

# Launch Sites Proximities Analysis

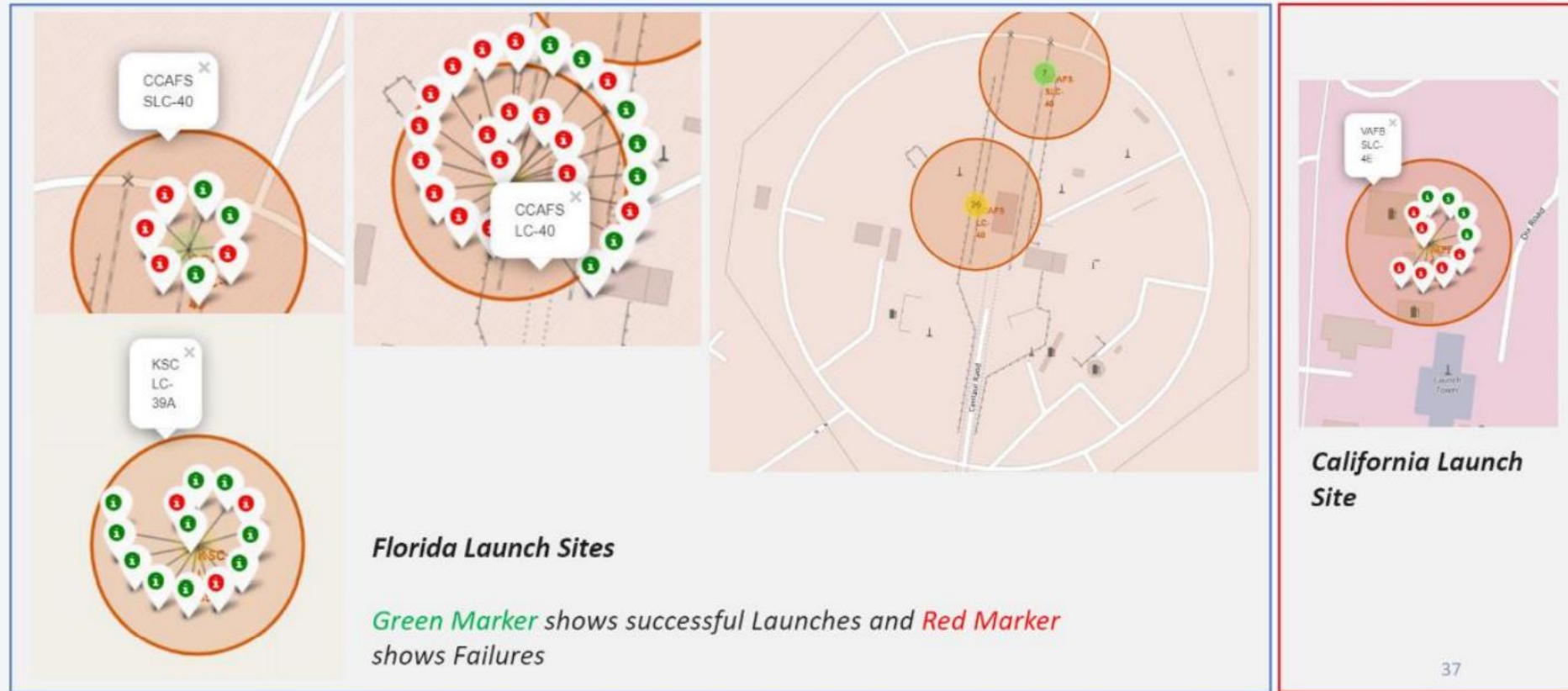
# All launch sites global map markers

---



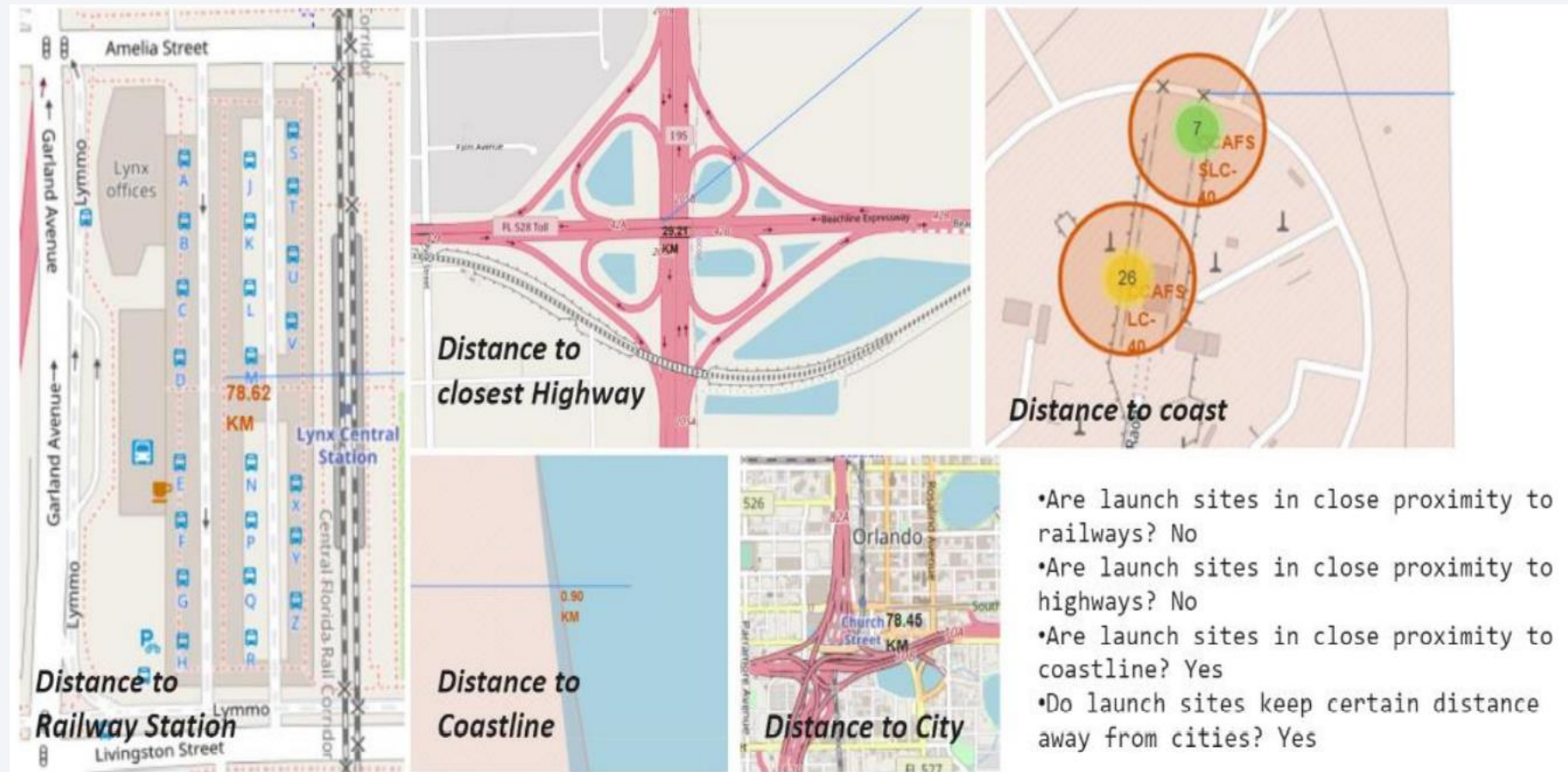


# Markers showing launch sites with color labels





# Launch Site distance to landmarks





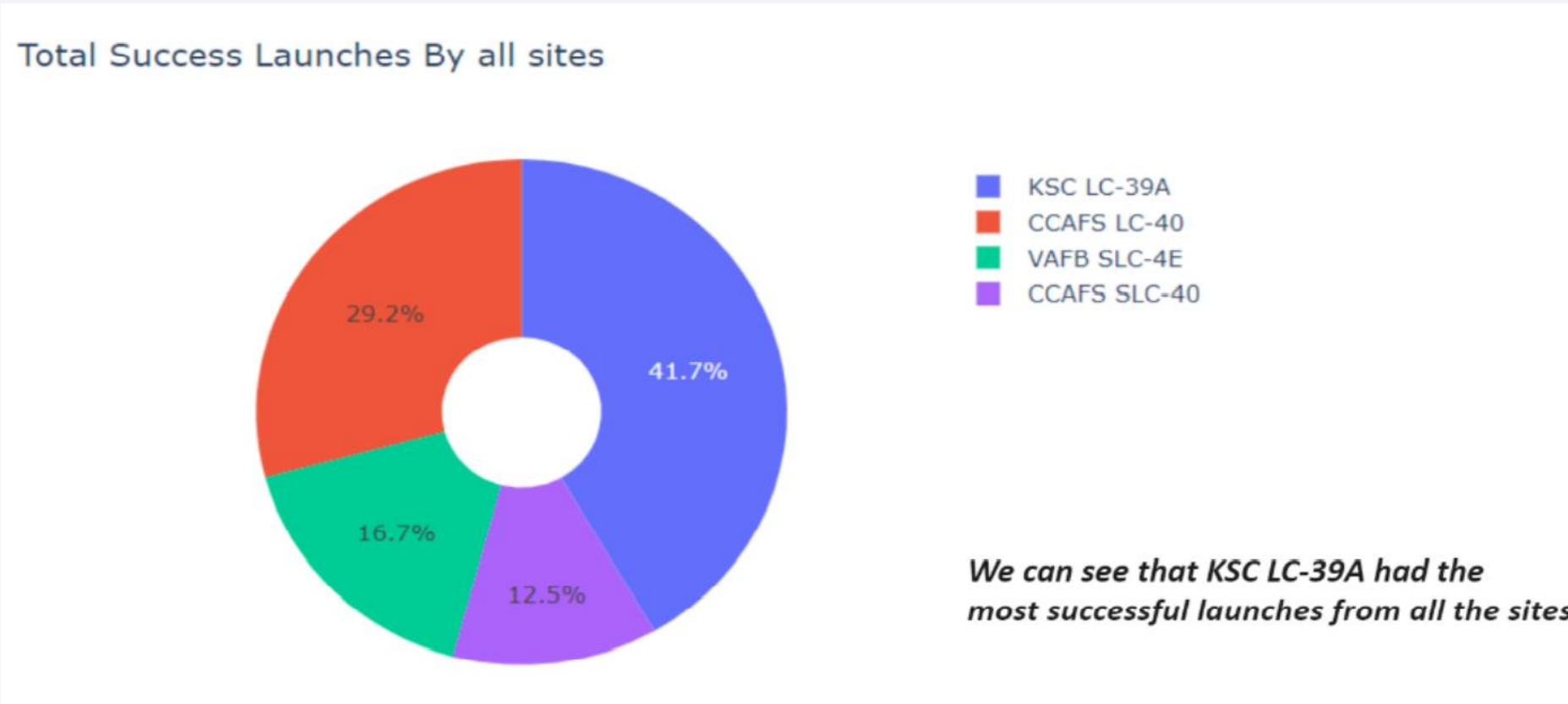
Section 4

# Build a Dashboard with Plotly Dash



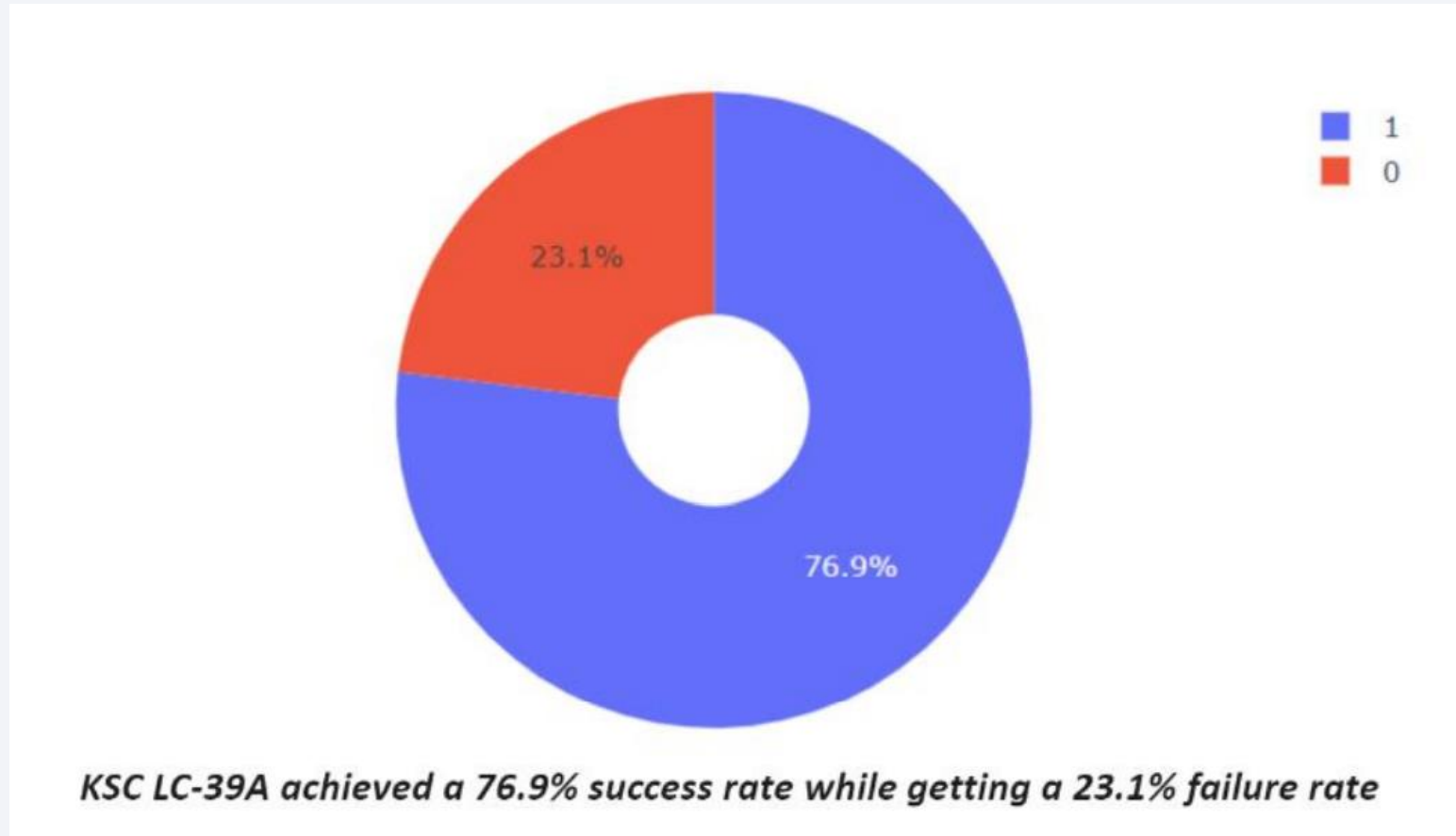
## Pie chart showing the success percentage achieved by each launch site

---



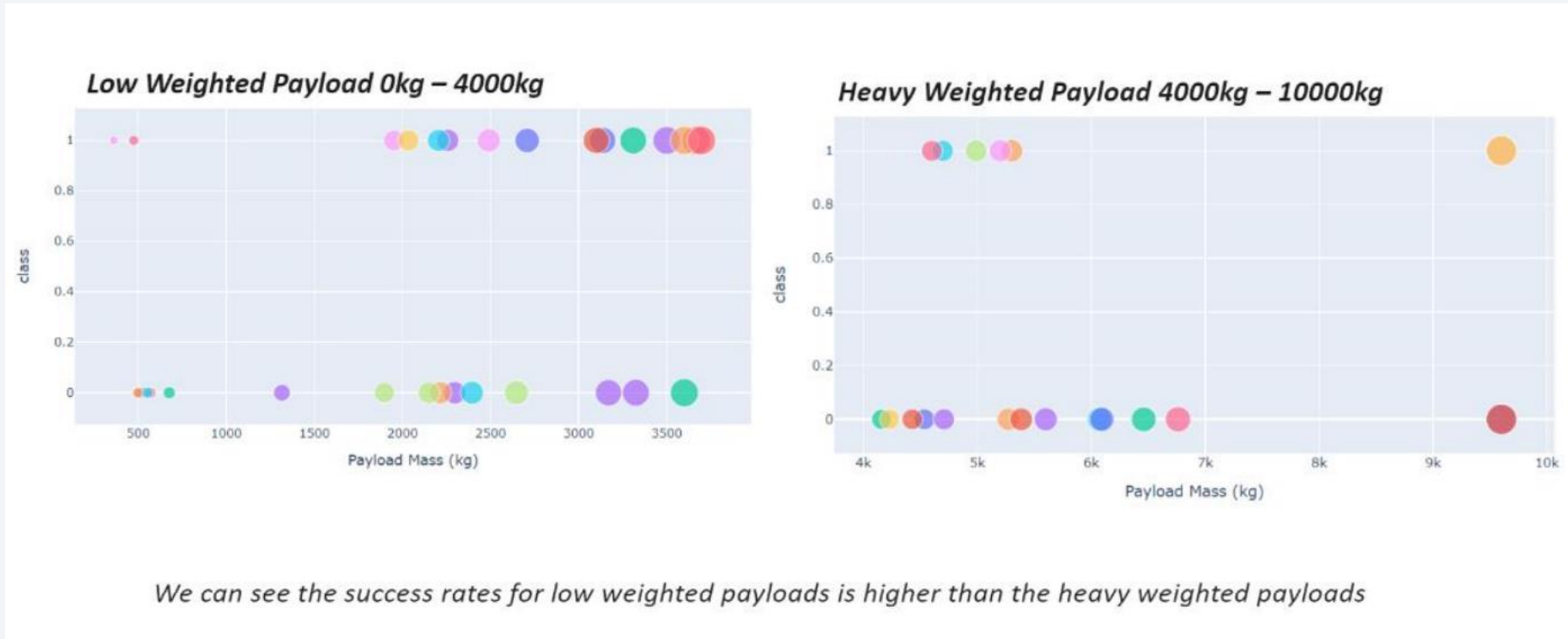
## Pie chart showing the Launch site with the highest launch success ratio

---



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

---



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

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

```
# Examining the scores from the whole Dataset
jaccard_scores = [
    jaccard_score(Y, logreg_cv.predict(X), average='binary'),
    jaccard_score(Y, svm_cv.predict(X), average='binary'),
    jaccard_score(Y, tree_cv.predict(X), average='binary'),
    jaccard_score(Y, knn_cv.predict(X), average='binary'),
]

f1_scores = [
    f1_score(Y, logreg_cv.predict(X), average='binary'),
    f1_score(Y, svm_cv.predict(X), average='binary'),
    f1_score(Y, tree_cv.predict(X), average='binary'),
    f1_score(Y, knn_cv.predict(X), average='binary'),
]

accuracy = [logreg_cv.score(X, Y), svm_cv.score(X, Y), tree_cv.score(X, Y), knn_cv.score(X, Y)]

scores = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]),
                      index=['Jaccard_Score', 'F1_Score', 'Accuracy'],
                      columns=['LogReg', 'SVM', 'Tree', 'KNN'])

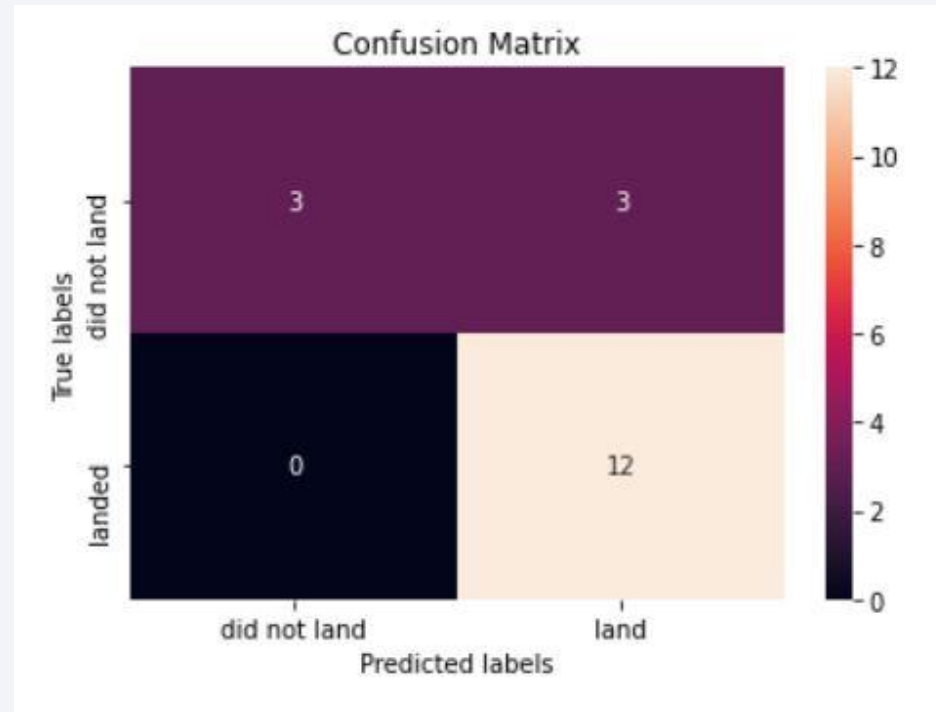
scores
```

	LogReg	SVM	Tree	KNN
<b>Jaccard_Score</b>	0.833333	0.845070	0.857143	0.819444
<b>F1_Score</b>	0.909091	0.916031	0.923077	0.900763
<b>Accuracy</b>	0.866667	0.877778	0.888889	0.855556



# 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.

Thank you!

