



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Amin Fauzan  
12<sup>nd</sup> Febuary 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

Space X promotes Falcon 9 rocket launches on its website, priced at 62 million dollars, whereas other providers charge over 165 million dollars per launch. A significant portion of the cost savings is attributed to Space X's ability to recycle the initial stage. Consequently, by assessing the likelihood of the first stage landing, we can ascertain the total launch cost. This data becomes valuable for potential rival companies bidding against Space X for rocket launches. The primary objective of this project is to establish a machine learning pipeline for predicting the successful landing of the first stage.

## Problems you want to find answers

- What elements influence the successful landing of the rocket.
- The interplay of different characteristics that impact the success rate of a landing.
- What operational circumstances must be established to guarantee 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 classification
- 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 – API

---

- Retrieve rocket launch data from SpaceX API through API requests.
- Parse the API response using `.json()` and transform it into a structured dataframe using `.json_normalize()`.
- Utilize custom functions to gather detailed information about the launches from the SpaceX API.
  - Construct a dictionary using the obtained data.
  - Convert the dictionary into a dataframe for further analysis.
  - Filter the dataframe to include only launches related to the Falcon 9 rocket.
  - Fill in any missing values in the Payload Mass column by replacing them with the calculated mean.
- Save the processed data to a CSV file for external use.

# Data Collection - Scraping

---

- Retrieve Falcon 9 launch data from Wikipedia through data requests.
- Generate a BeautifulSoup object from the HTML response for effective parsing.
- Extract column names by parsing the HTML table header.
- Gather relevant data by parsing the HTML tables.
- Construct a dictionary using the extracted data.
- Transform the dictionary into a dataframe for structured analysis.
- Save the processed data to a CSV file for external use.



# Data Wrangling

---

Perform EDA and determine data labels

- Calculate:

# of launches for each site

# and occurrence of orbit

# and occurrence of mission outcome per orbit type

- Create binary landing outcome column (dependent variable)
- Export data to csv file

Landing Outcome

- Landing was not always successful
- True Ocean: mission outcome had a successful landing to a specific region of the ocean

Landing Outcome Count

False Ocean: represented an unsuccessful landing to a specific region of ocean

- True RTLS: meant the mission had a successful landing on a ground pad
- False RTLS: represented an unsuccessful landing on a ground pad
- True ASDS: meant the mission outcome had a successful landing on a drone ship
- False ASDS: represented an unsuccessful landing on drone ship
- Outcomes converted into 1 for a successful landing and 0 for an unsuccessful landing

# EDA with Data Visualization

---

## Charts

- Flight Number vs. Payload
- Flight Number vs. Launch Site
- Payload Mass (kg) vs. Launch Site
- Payload Mass (kg) vs. Orbit type EDA with Visualization

## Analysis

- View relationship by using scatter plots. The variables could be useful for machine learning if a relationship exists
- Show comparisons among discrete categories with bar charts. Bar charts show the relationships among the categories and a measured value.

# EDA with SQL

---

Display:

- Names of unique launch sites
- 5 records where launch site begins with 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1.

List:

- Date of first successful landing on ground pad
- Names of boosters which had success landing on drone ship and have payload mass greater than 4,000 but less than 6,000
- Total number of successful and failed missions
- Names of booster versions which have carried the max payload
- Failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015
- Count of landing outcomes between 2010-06-04 and 2017-03-20 (desc)

# Build an Interactive Map with Folium

---

## Markers Indicating Launch Sites :

- Added blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name using its latitude and longitude coordinates
- Added red circles at all launch sites coordinates with a popup label showing its name using its name using its latitude and longitude coordinates

## Colored Markers of Launch Outcomes :

- Added colored markers of successful (green) and unsuccessful (red) launches at each launch site to show which launch sites have high success rates

## Distances Between a Launch Site to Proximities :

- Added colored lines to show distance between launch site CCAFS SLC40 and its proximity to the nearest coastline, railway, highway, and city

# Build a Dashboard with Plotly Dash

---

Dropdown List with Launch Sites :

- Allow user to select all launch sites or a certain launch site

Pie Chart Showing Successful Launches :

- Allow user to see successful and unsuccessful launches as a percent of the total

Slider of Payload Mass Range :

- Allow user to select payload mass range

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version :

- Allow user to see the correlation between Payload and Launch Success

# Predictive Analysis (Classification)

---

- Create NumPy array from the Class column
- Standardize the data with StandardScaler. Fit and transform the data
- Split the data using train\_test\_split
- Create a GridSearchCV object with cv=10 for parameter optimization
- Apply GridSearchCV on different algorithms: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), K-Nearest Neighbor (KNeighborsClassifier())
- Calculate accuracy on the test data using .score() for all models
- Assess the confusion matrix for all models
- Identify the best model using Jaccard\_Score, F1\_Score and Accuracy



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue and red on the right. These streaks are layered over a fine, light-colored grid, creating a sense of depth and movement, reminiscent of a digital or data visualization theme.

Section 2

# Insights drawn from EDA

# Results

---

## Exploratory Data Analysis

- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate Results Summary

## Visual Analytics

- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities

## Predictive Analytics

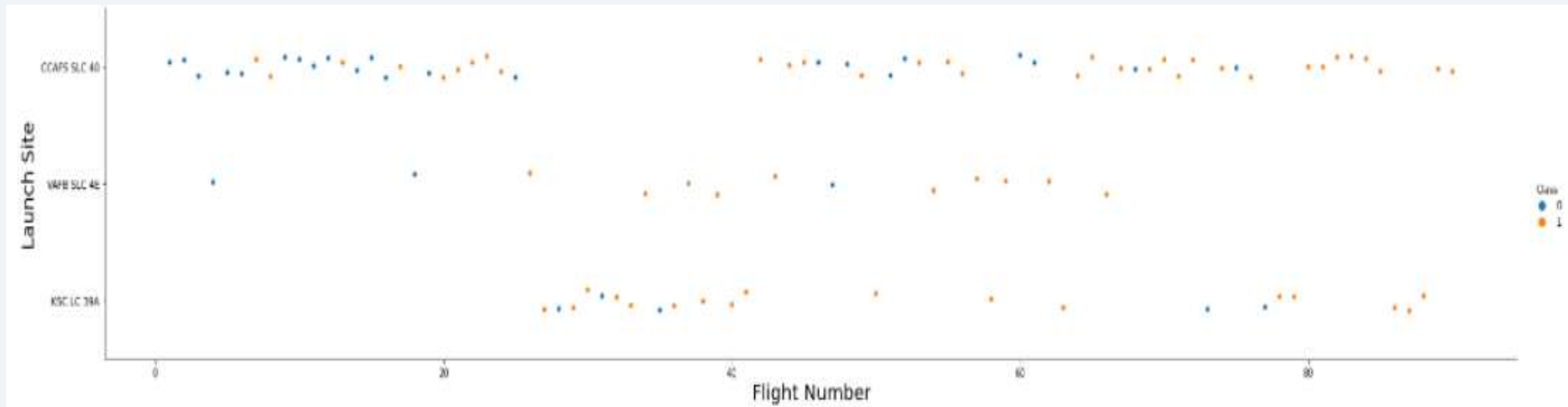
- Decision Tree model is the best predictive model for the dataset



# Flight Number vs. Launch Site

---

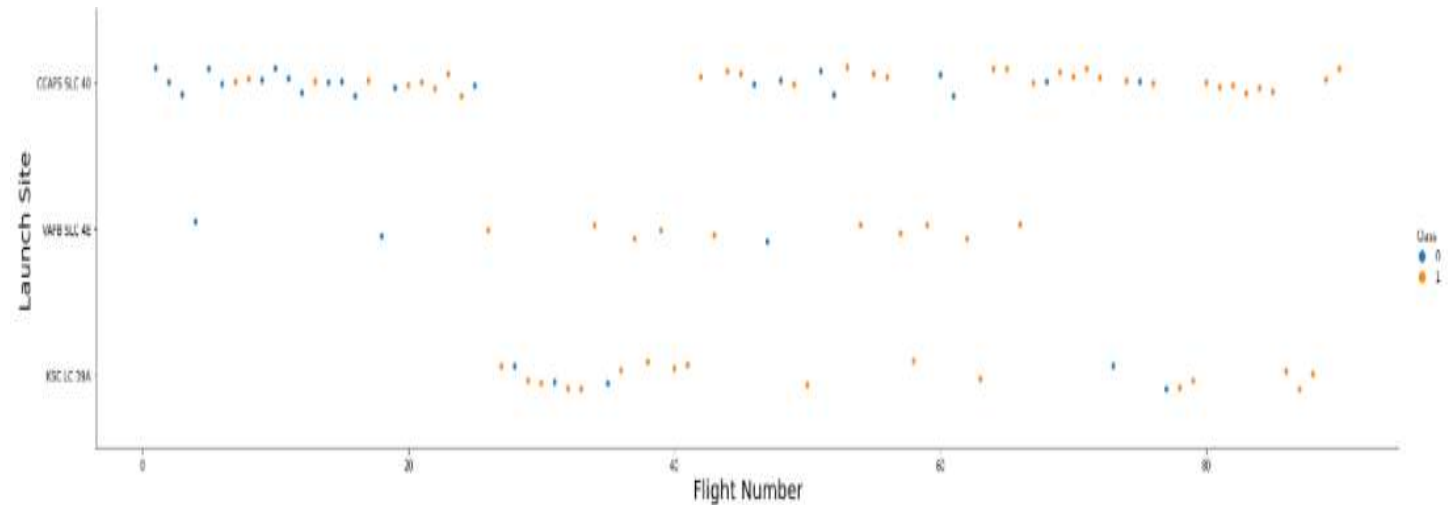
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



## Payload vs. Launch Site

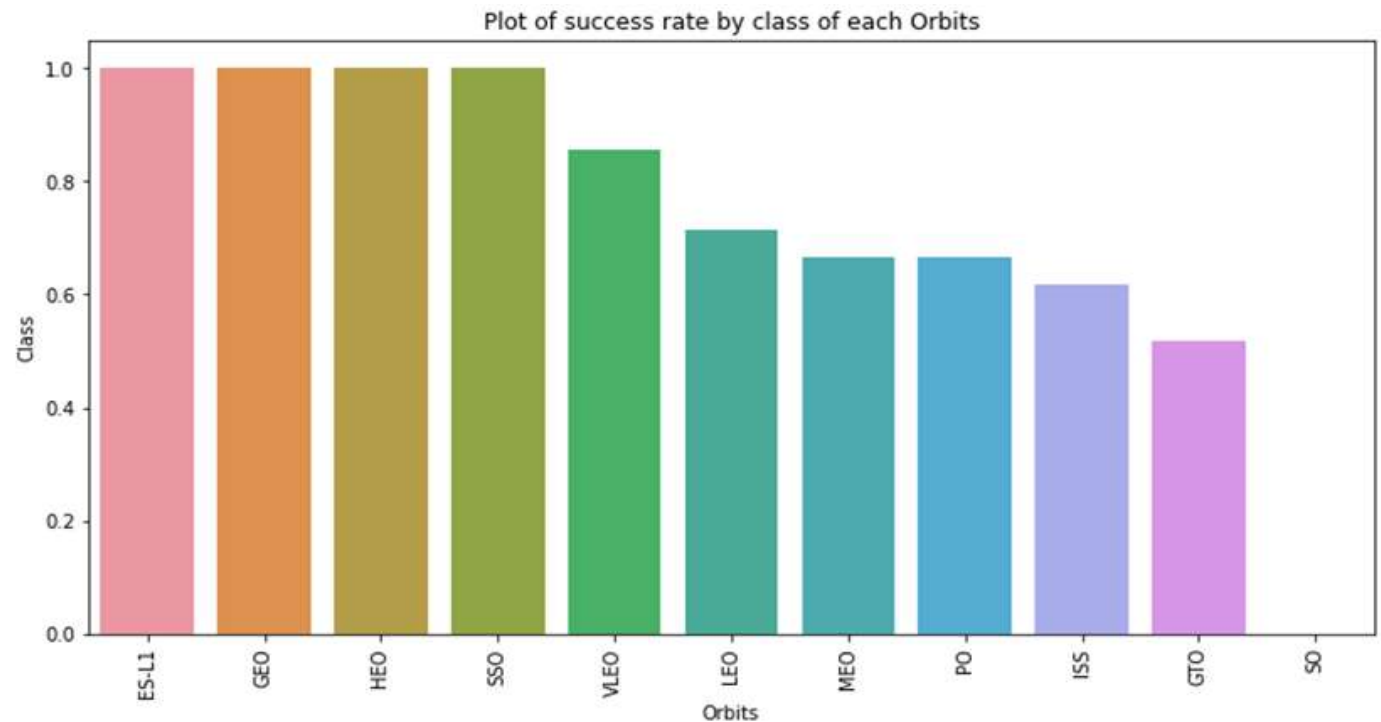


The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



# Success Rate vs. Orbit Type

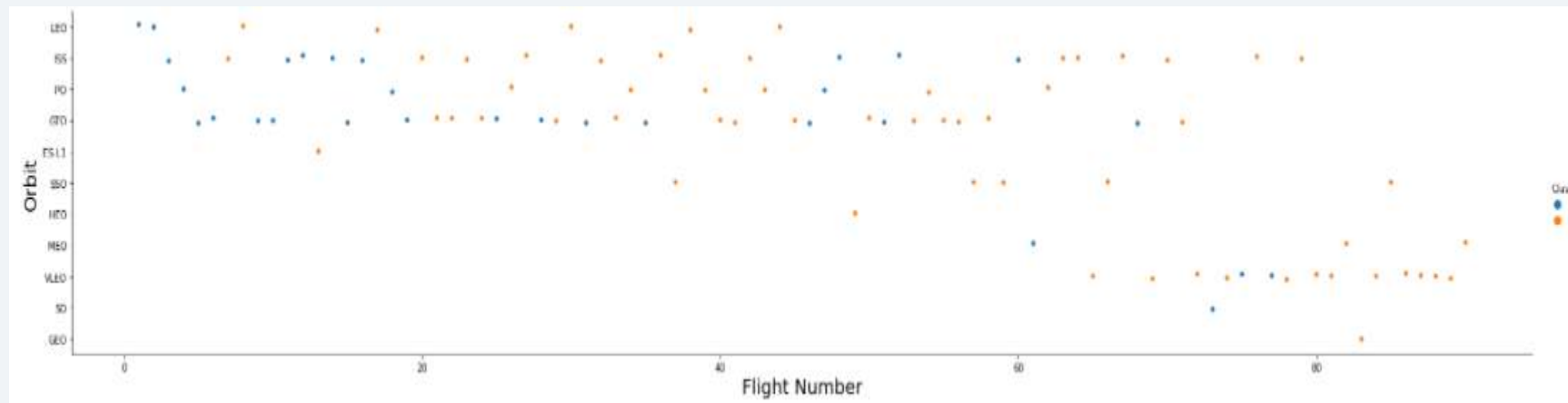
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



# 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.

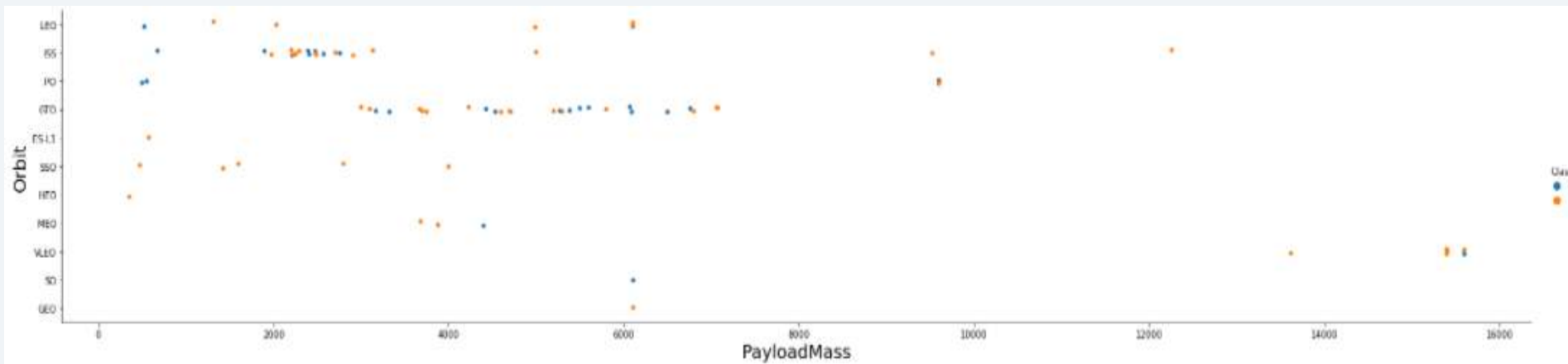




# Payload vs. Orbit Type

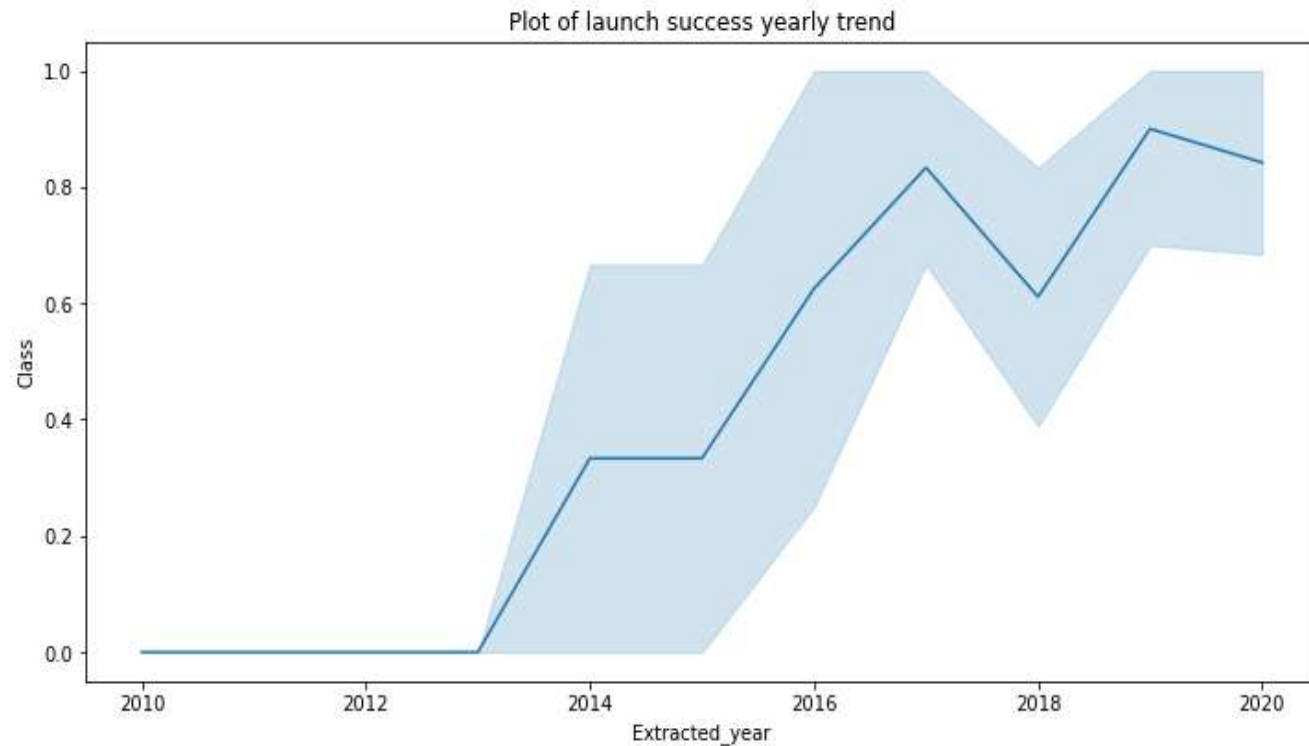
---

- 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

```
In [10]: task_1 = '''
          SELECT DISTINCT LaunchSite
          FROM SpaceX
          ...
          create_pandas_df(task_1, database=conn)
```

```
Out[10]:
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

# Launch Site Names Begin with 'CCA'

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

In [11]:

```
task_2 = '''
SELECT *
FROM SpaceX
WHERE LaunchSite LIKE 'CCA%'
LIMIT 5
'''

create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- We used the query above to display 5 records where launch sites begin with 'CCA'

# 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)
```

```
In [12]: task_3 = '''
          SELECT SUM(PayloadMassKG) AS Total_PayloadMass
          FROM SpaceX
          WHERE Customer LIKE 'NASA (CRS)'
          '''
          create_pandas_df(task_3, database=conn)
```

```
Out[12]:
```

	total_payloadmass
0	45596

# Average Payload Mass by F9 v1.1

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

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

```
In [13]: task_4 = '''
          SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
          FROM SpaceX
          WHERE BoosterVersion = 'F9 v1.1'
          '''
          create_pandas_df(task_4, database=conn)
```

```
Out[13]:
```

	avg_payloadmass
0	2928.4



# First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22<sup>nd</sup> December 2015

```
In [14]: task_5 = '''
          SELECT MIN(Date) AS FirstSuccessfull_landing_date
          FROM SpaceX
          WHERE LandingOutcome LIKE 'Success (ground pad)'
          '''

          create_pandas_df(task_5, database=conn)
```

```
Out[14]:
```

	firstsuccessfull_landing_date
0	2015-12-22

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [15]: task_6 = '''
          SELECT BoosterVersion
          FROM SpaceX
          WHERE LandingOutcome = 'Success (drone ship)'
             AND PayloadMassKG > 4000
             AND PayloadMassKG < 6000
          ...
          create_pandas_df(task_6, database=conn)
```

```
Out[15]:
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

- 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

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
In [16]: task_7a = '''
          SELECT COUNT(MissionOutcome) AS SuccessOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Success%'
          '''

          task_7b = '''
          SELECT COUNT(MissionOutcome) AS FailureOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Failure%'
          '''

          print('The total number of successful mission outcome is:')
          display(create_pandas_df(task_7a, database=conn))
          print()
          print('The total number of failed mission outcome is:')
          create_pandas_df(task_7b, database=conn)
```

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

```
Out[16]:
```

	failureoutcome
0	1

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

# 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

```
In [17]: task_8 = '''
          SELECT BoosterVersion, PayloadMassKG
          FROM SpaceX
          WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
          ORDER BY BoosterVersion
          '''
          create_pandas_df(task_8, database=conn)
```

```
Out[17]:
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	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 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
```

```
In [18]: task_9 = '''
          SELECT BoosterVersion, LaunchSite, LandingOutcome
          FROM SpaceX
          WHERE LandingOutcome LIKE 'Failure (drone ship)'
             AND Date BETWEEN '2015-01-01' AND '2015-12-31'
          ...
          create_pandas_df(task_9, database=conn)
```

```
Out[18]:
```

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

```
In [19]: task_10 = '''
          SELECT LandingOutcome, COUNT(LandingOutcome)
          FROM SpaceX
          WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
          GROUP BY LandingOutcome
          ORDER BY COUNT(LandingOutcome) DESC
          '''
          create_pandas_df(task_10, database=conn)
```

```
Out[19]:
```

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

- 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.



Section 4

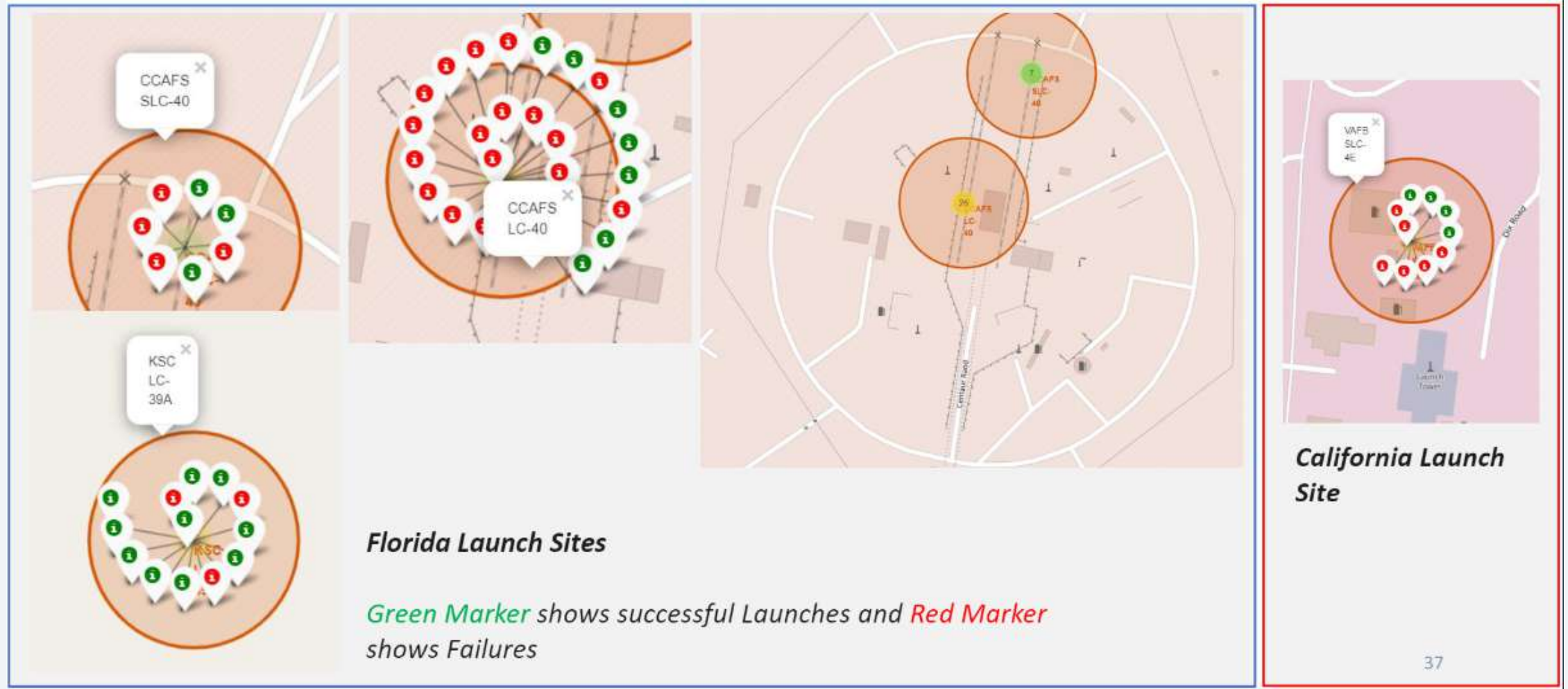
# Launch Sites Proximities Analysis



# All launch sites global map markers

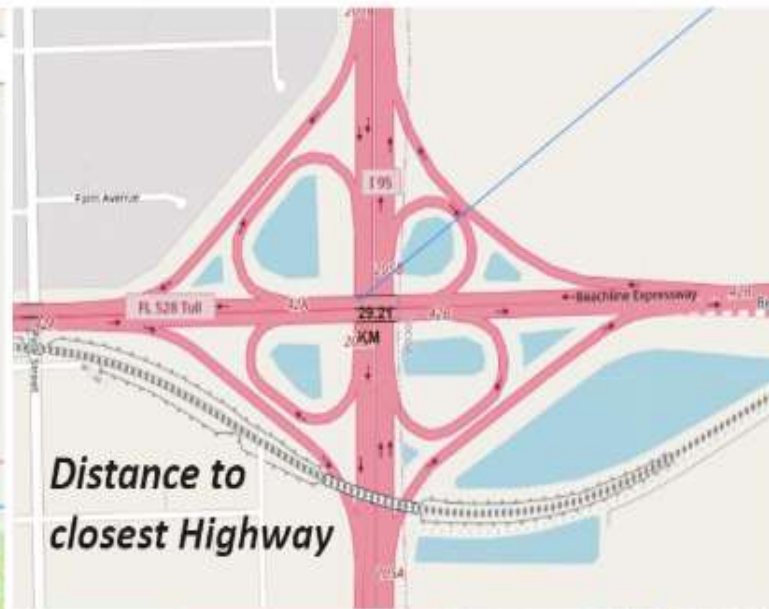


# Markers showing launch sites with color labels





# 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



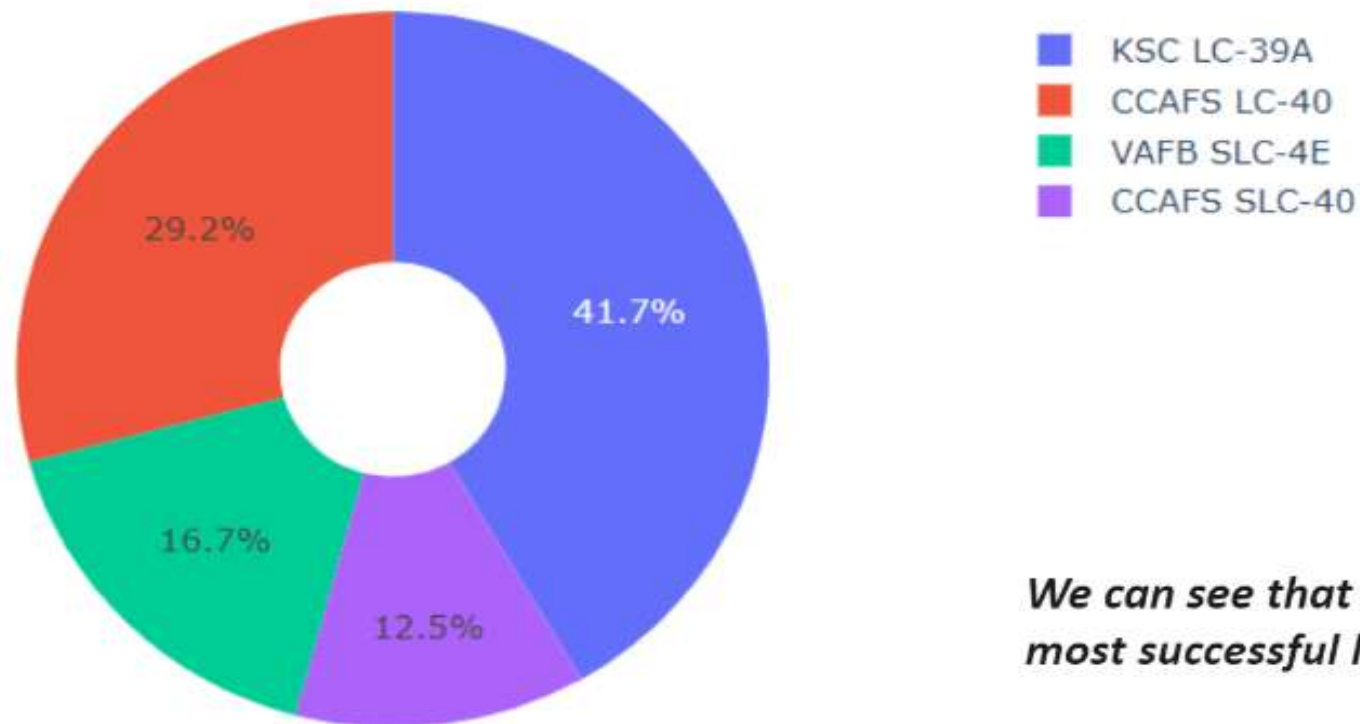
Section 5

# Build a Dashboard with Plotly Dash



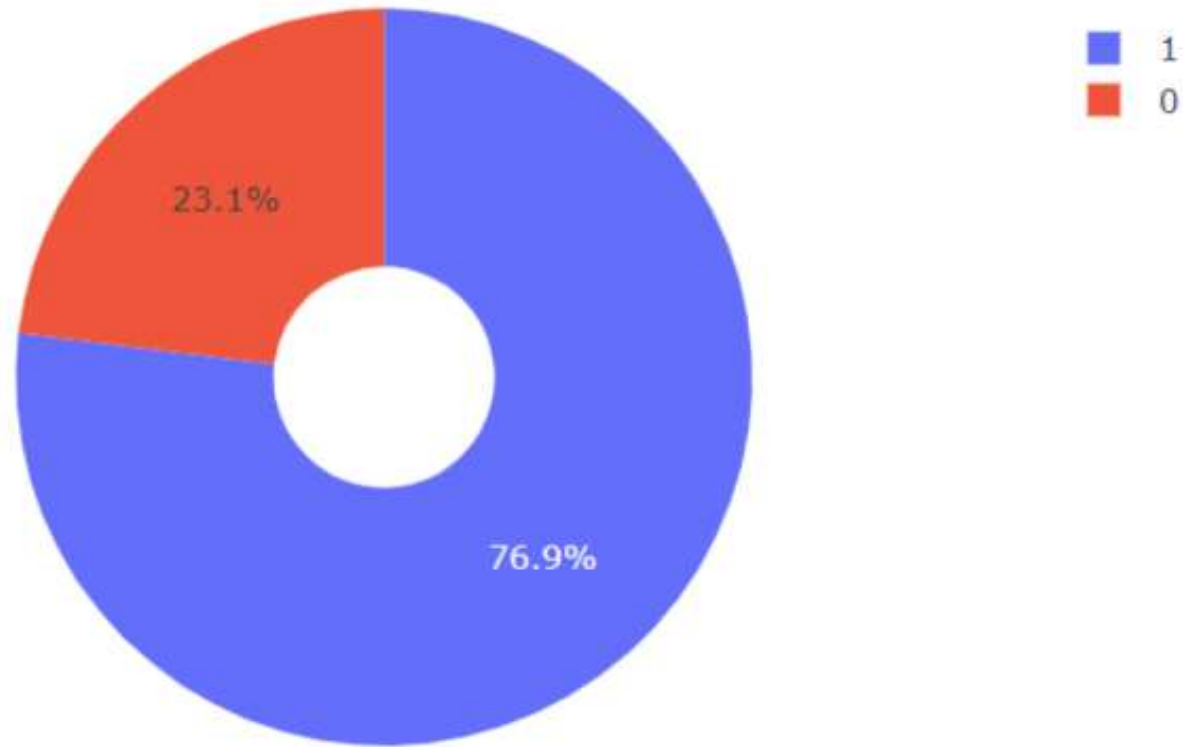
## Pie chart showing the success percentage achieved by each launch site

Total Success Launches By all sites



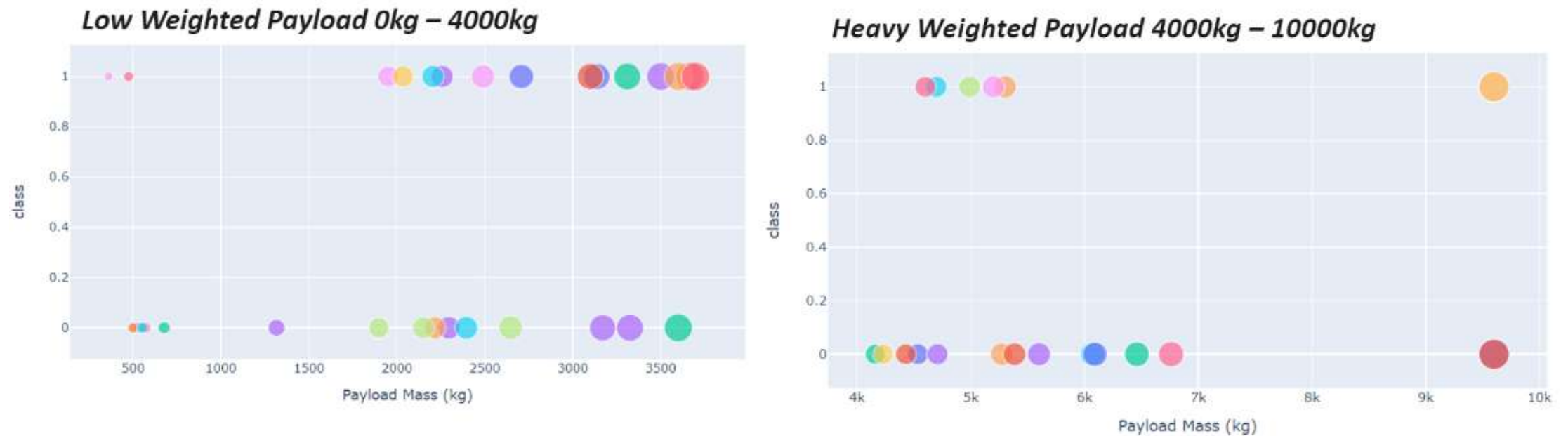
*We can see that KSC LC-39A had the most successful launches from all the sites*

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 6

# Predictive Analysis (Classification)

# Classification Accuracy

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

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

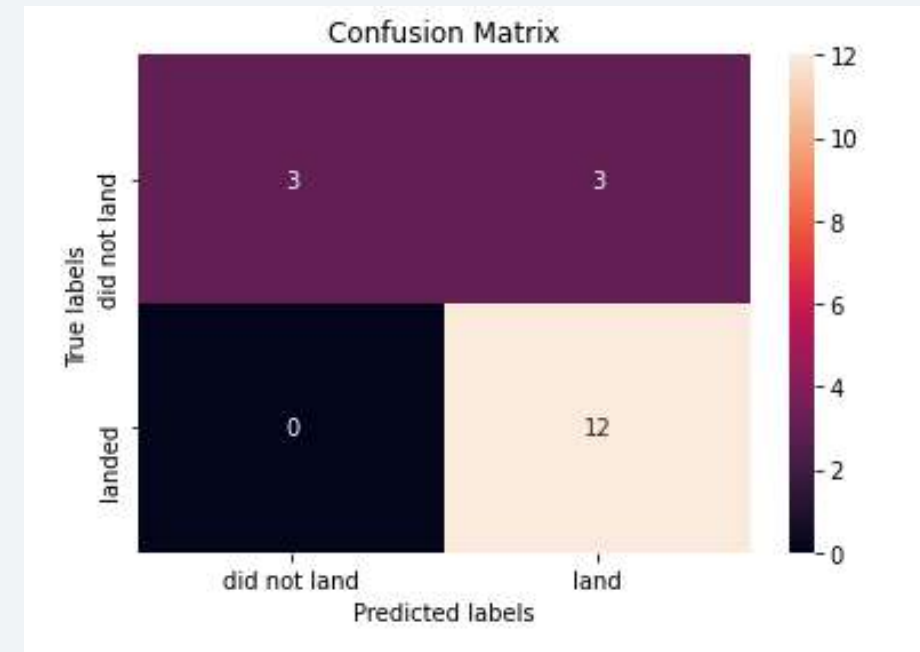
bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm, 'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm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

# 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!

