

# Winning Space Race with Data Science

Samaneh Sharifi Golru  
Jan. 2024

GitHub: <https://github.com/ssharifigolru/Applied-Data-Science-Capstone-IBM-Course>



# Outline

---



- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

# Executive Summary

---



## Summary of methodologies

- Data collection (request from SpaceX API) by Webscraping
- Data wrangling
- Exploratory Data Analysis (EDA) Using Data Visualization and SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Classification Predictive Analysis

## Summary of all results

- EDA Results
- ML Results

# Introduction

---



## ➤ Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of \$62 million, a significantly lower cost compared to the \$165 million estimated by other providers. The distinctive feature of SpaceX lies in the ability to reuse the rocket following a successful first-stage landing. Hence, increasing the frequency of successful first-stage landings is crucial for maximizing rocket reuse and minimizing costs.

## ➤ Problems you want to find answers

The objective of the project is to forecast the successful landing of the first stage of the SpaceX Falcon 9 rocket. Rather than tackling complex mathematical and physics equations, our approach involves leveraging data science to create a machine learning model. This model will be developed using historical observational data from previous Falcon 9 launches.

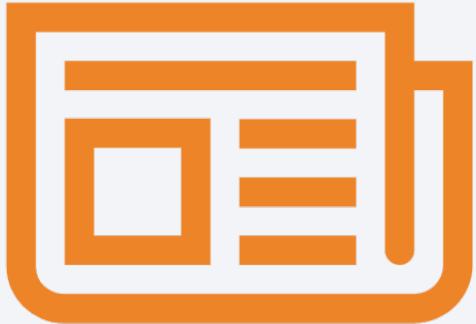
Section 1

# Methodology

# Methodology

---

## Executive Summary:

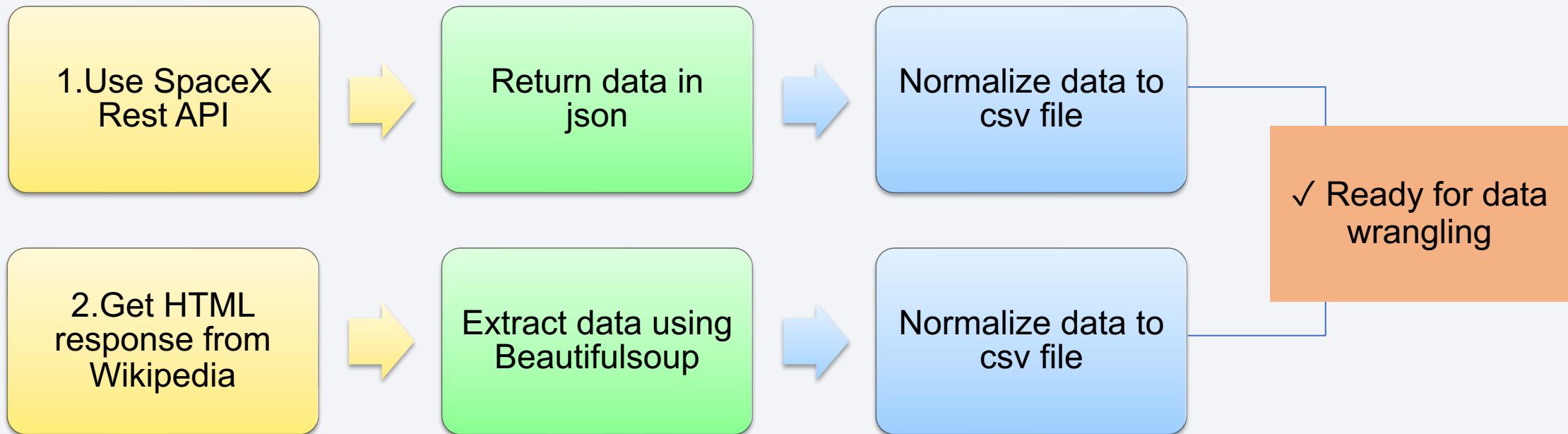


1. Data collection methodology:
  - SpaceX's API Website
  - Web scraping data : Extract a Falcon 9 launch records HTML table from Wikipedia
2. Perform data wrangling
  - Identify the missing values in each attribute
  - Replacing missing numerical data with the mean value of parameter
  - Using One Hot Encoding to transform categorical variables to numerical ones
3. Perform exploratory data analysis (EDA) using visualization and SQL
  - Utilizing categorical, bar, and scatter plots to visually represent data and discern relationships between parameters
  - Using SQL queries to perform the data analysis process
4. Perform interactive visual analytics using Folium and Plotly Dash
  - Interactive Plotly Dashboard to visualize payload and success launch data
  - Using interactive Folium Maps to explore Launch Sites
5. Perform predictive analysis using classification models
  - Standardizing parameters
  - Predictive multiple classification models (Logistic, SVM, Decision Tree, KNN)
  - Finding Best performance model

# Data Collection

SpaceX launch data that is gathered from the SpaceX REST API or web scraping Wiki.

1. Using SpaceX REST API to gather data on rocket launches: <https://api.spacexdata.com/v4>
2. Other data source for collecting Falcon 9 Launch data is web scraping Wikipedia using Beautiful Soup.



# Data Collection – SpaceX API

1- Request data frame from SpaceX API



```
spacex_url="https://api.spacexdata.com/v4/launches/past"  
response = requests.get(spacex_url)
```

2- Transform the json file to a Pandas data frame



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

3- Extract required data and make lists



```
# Takes the dataset and uses the rocket column to call the API and append the data to the list  
def getBoosterVersion(data):  
    for x in data['rocket']:  
        if x:  
            response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()  
            BoosterVersion.append(response['name'])
```

4- Transform lists to a data frame



```
# Create a data from launch_dict  
df=pd.DataFrame(launch_dict)
```

5- Keep Falcon V9 boosters and remove others



```
data_falcon9=df[df['BoosterVersion']!='Falcon 1']  
data_falcon9.head(5)
```

6- Convert data frame to a csv file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

# Data Collection - Scraping

1- Getting response from HTML

```
# use requests.get() method with the provided static_url  
htmlpage=requests.get(static_url)  
# assign the response to a object  
  
htmlpage.status_code  
  
200
```

2- Creating BeautifulSoup object

```
soup = BeautifulSoup(htmlpage.text, 'html.parser')
```

3- Finding Tables

```
html_tables=soup.find_all('table')
```

4- Getting column names

```
column_names = []  
for i in first_launch_table.find_all('th'):  
    if extract_column_from_header(i)!=None:  
        if len(extract_column_from_header(i))>0:  
            column_names.append(extract_column_from_header(i))
```

5- Creating disctionary

```
launch_dict= dict.fromkeys(column_names)  
# Remove an irrelevant column  
del launch_dict['Date and time ()']  
  
# Let's initial the launch_dict with each value to be an empty list  
launch_dict['Flight No.']=[]  
launch_dict['Launch site']=[]  
launch_dict['Payload']=[]  
launch_dict['Payload mass']=[]  
launch_dict['Orbit']=[]  
launch_dict['Customer']=[]  
launch_dict['Launch outcome']=[]  
  
# Add some new columns  
launch_dict['Version Booster']=[]  
launch_dict['Booster landing']=[]  
launch_dict['Date']=[]  
launch_dict['Time']=[]
```

6- Appending data to keys

```
extracted_row = 0  
#Extract each table  
for table_number,table in enumerate(soup.find_all('table','wikitable plainrowheaders collapsible')):  
    # get table rows  
    for rows in table.find_all("tr"):  
        #check to see if first table heading is as number corresponding to launch a number  
        if rows.th:  
            if rows.th.string:  
                flight_number=rows.th.string.strip()  
                flag=flight_number.isdigit()
```

7-Create a dataframe from launch\_dict

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

8-Export data frame to a CSV file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

# Data Wrangling

## Missing values

- 1.Finding missing values in each attribute

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

```
FlightNumber      0  
Date             0  
BoosterVersion   0  
PayloadMass      5  
Orbit            0  
LaunchSite        0  
Outcome           0  
Flights          0  
GridFins          0  
Reused            0  
Legs              0  
LandingPad       26  
Block             0  
ReusedCount      0  
Serial            0  
Longitude         0  
Latitude          0  
dtype: int64
```

- 2.Imputing missing values → replace missing values with Mean

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

## Exploratory data analysis

- 1.Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()  
  
CCAFS SLC 40    55  
KSC LC 39A     22  
VAFB SLC 4E     13  
Name: LaunchSite, dtype: int64
```

- 2.Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()  
  
GTO    27  
ISS    21  
VLEO   14  
PO     9  
LEO    7  
SSO    5  
MEO    3  
ES-L1   1  
HEO    1  
SO     1  
GEO    1  
Name: Orbit, dtype: int64
```

- 3.Calculate the number and occurrence of mission outcome per orbit type

```
landing_outcomes=df['Outcome'].value_counts()  
landing_outcomes  
  
True ASDS    41  
None None    19  
True RTLS    14  
False ASDS   6  
True Ocean   5  
False Ocean  2  
None ASDS   2  
False RTLS   1  
Name: Outcome, dtype: int64
```

- 4.Create landing outcome label from Outcome column

```
def function(item):  
    if item in bad_outcomes:  
        return 0  
    else:  
        return 1  
landing_class = df["Outcome"].apply(function)  
landing_class
```

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

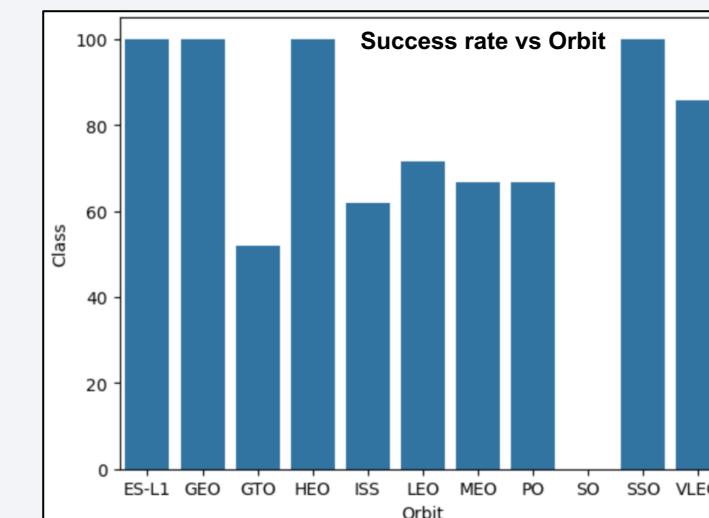
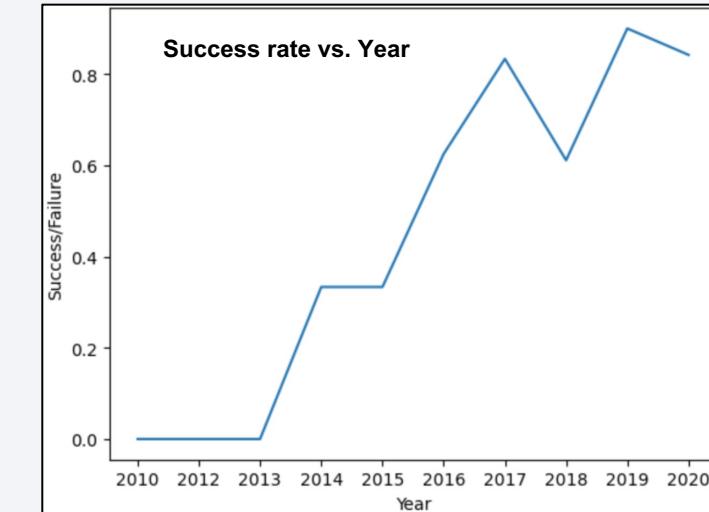
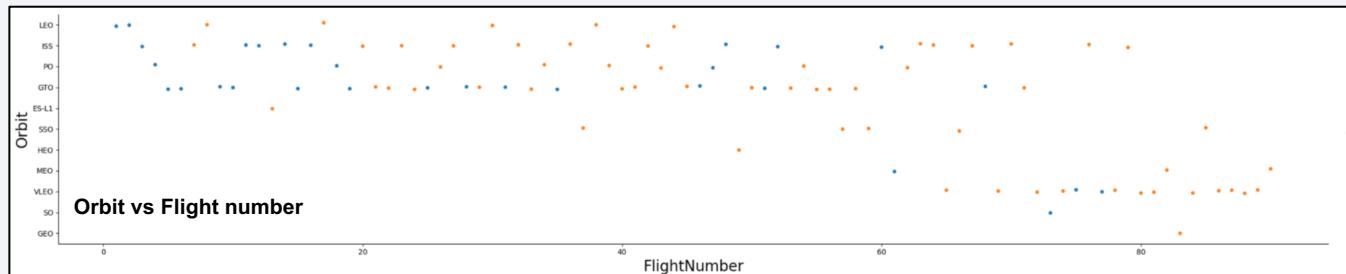
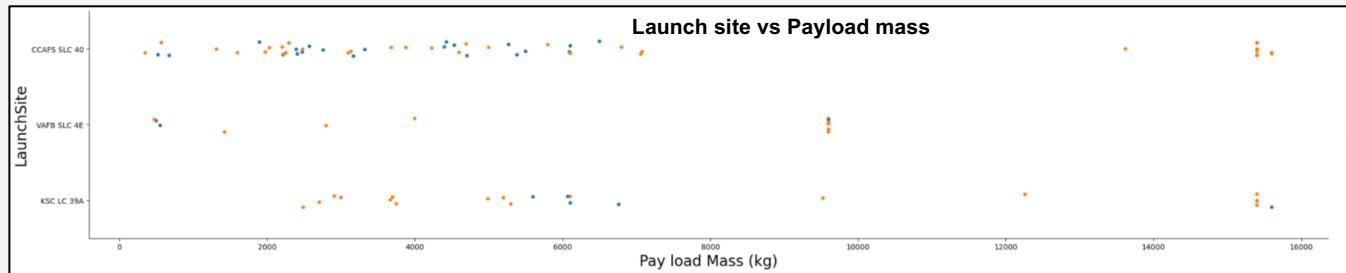
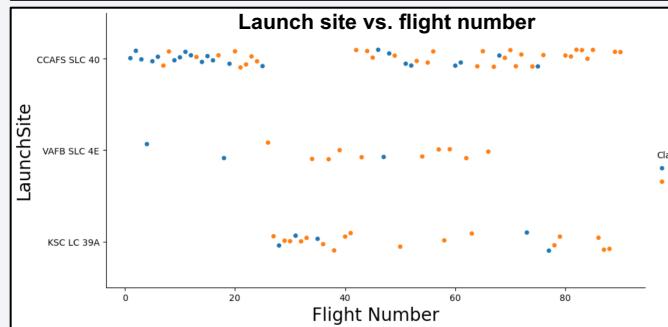
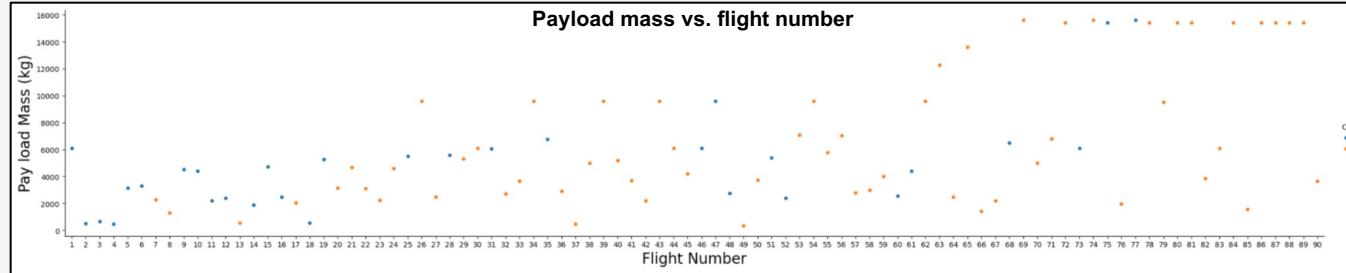
- 5.Export results to CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```

[Link to : Data collection using API](#)

[Link to: Data wrangling](#)

# EDA with Data Visualization



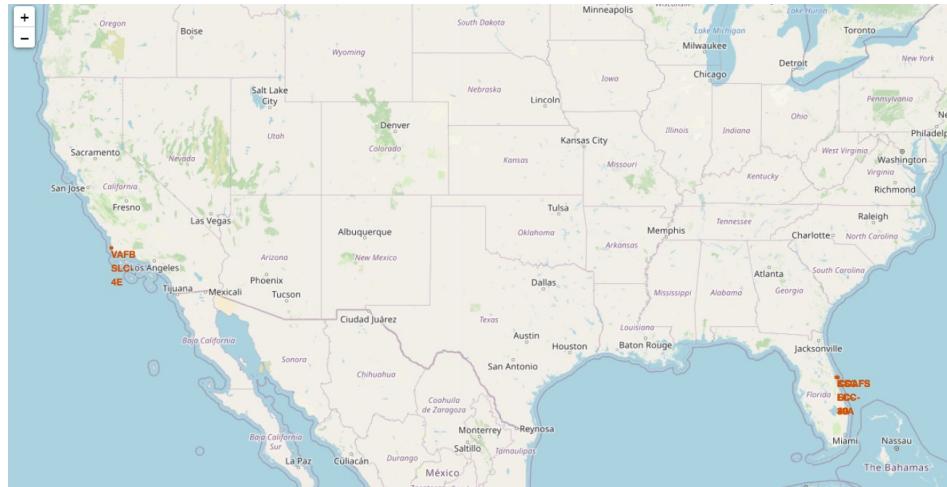
# EDA with SQL

---

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the records which will display the month names, successful landing outcomes in ground pad booster versions, launch months in year 2015
- Ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

# Build an Interactive map with Folium

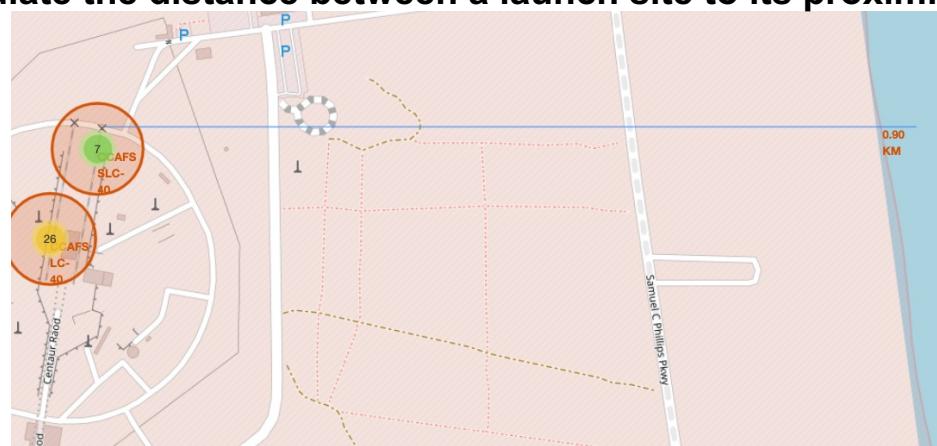
Launches sites on a map



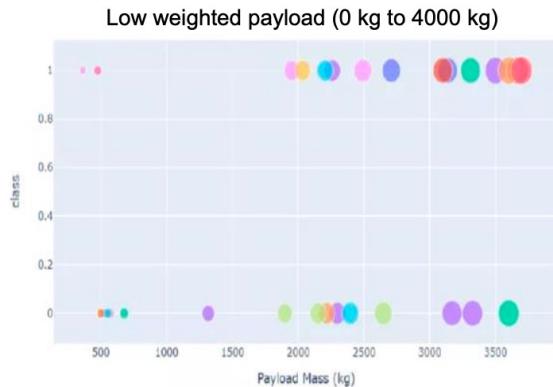
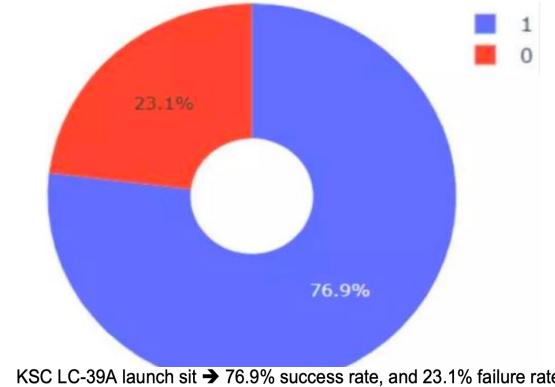
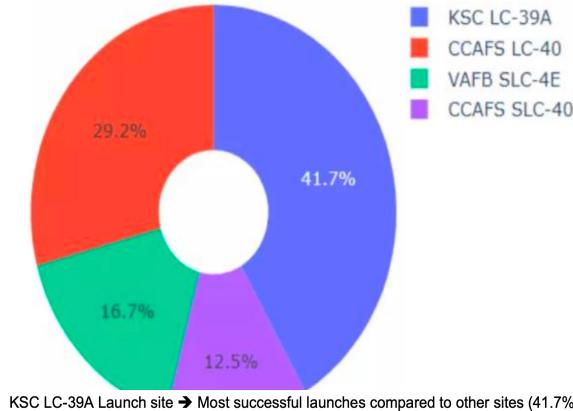
Success/failed launches for each site



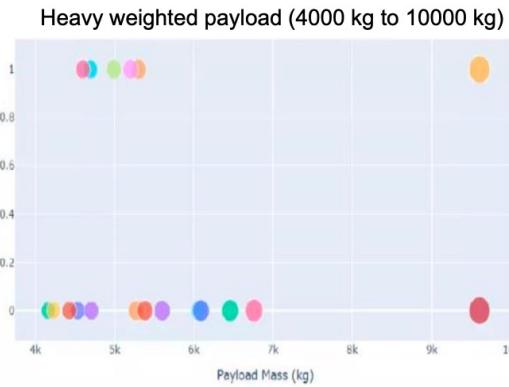
Calculate the distance between a launch site to its proximities



# Build a Dashboard with Plotly Dash

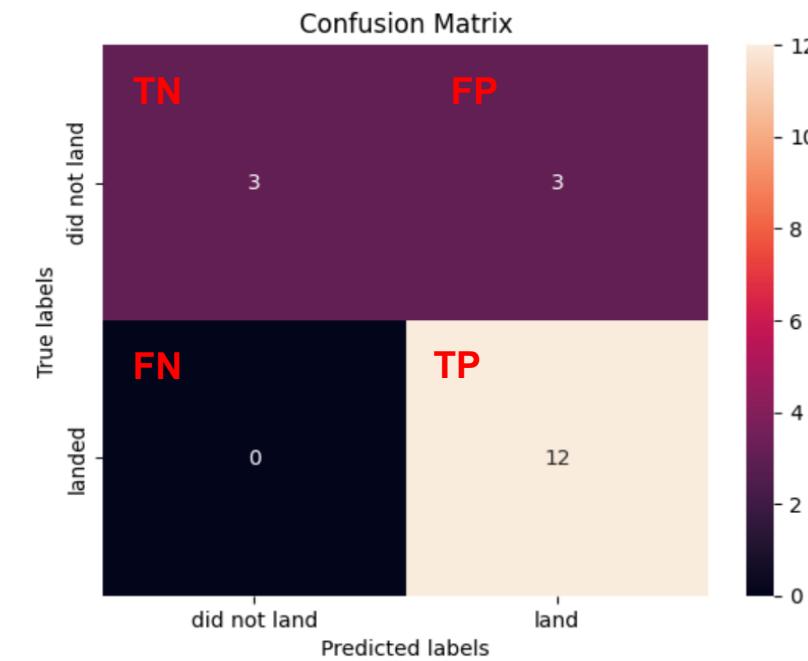
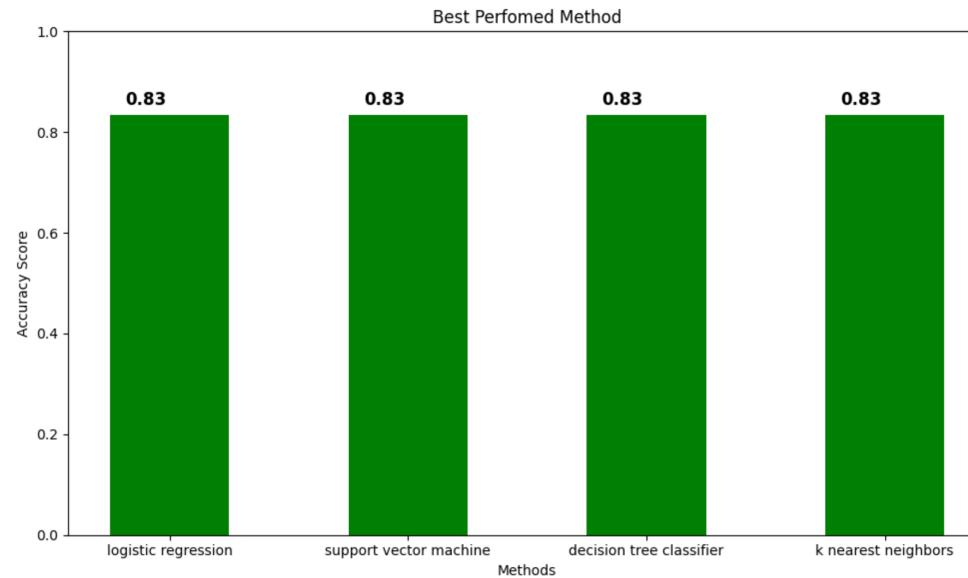


Success rates for low weighted is higher than heavy weighted payload!



- Built Plotly Dashboard ➔ To make an interactive web app to visualize launch data
- Pie Charts ➔ to visualize launch landing success broken down by Launch Site
- If all sites are selected, we get the proportion successful launch landings each site accounts
- If we select an individual site, we see the proportion of all launches at that site which landed successfully
- Includes Scatter Plot of Payload Mass (kg) vs. landing Success Rating (0 for Failure, 1 for Success) color coded by booster version
- Selecting a single site removes points from other sites, and selecting All includes all data
- Plot Payload Mass Range can be selected by a slider for Min and Max

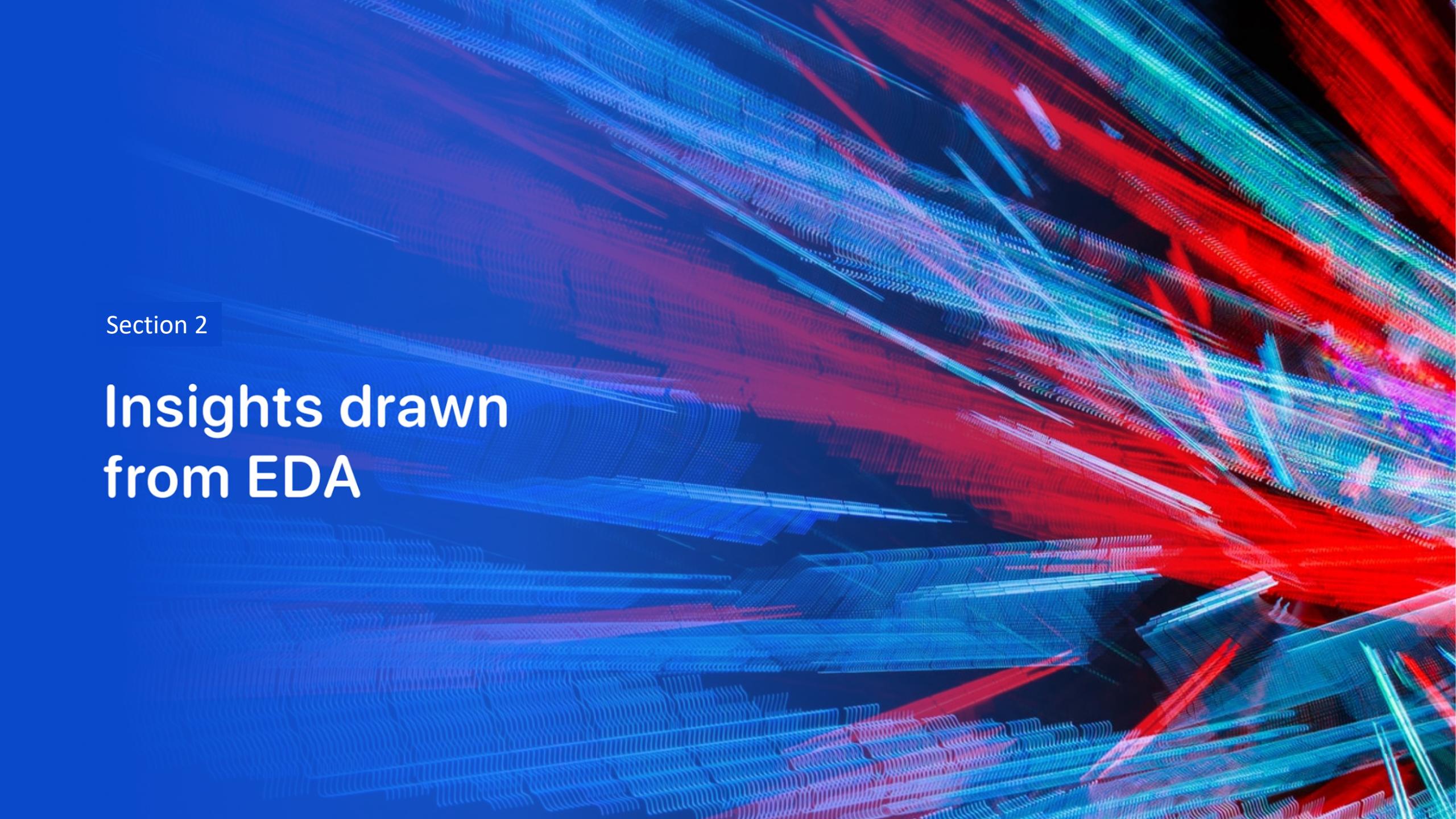
# Predictive analysis (Classification)



Accuracy scores and confusion matrix for all four models (SVM, KNN, Logistic Regression, and Decision Tree) are the same. Accuracy score is 83% for all models studied here.

# Results

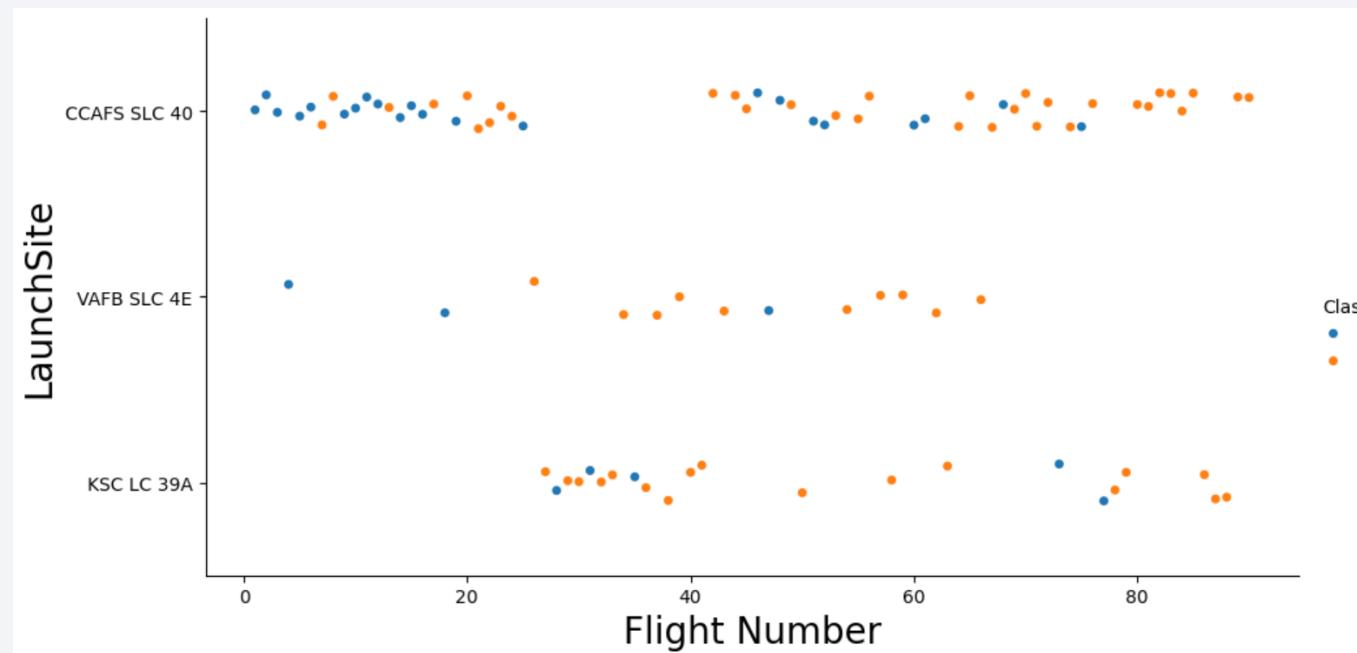
- All Models showed the same accuracy (%83) and confusion matrix in this analysis.
- Low-weighted payloads perform better than the heavier payloads.
- KSC LC-39A Launch Site has the highest probability of success rate.
- ES-L1, SSO, HEO and GEO Orbits have the highest rate of landing successfully.
- The success rate for SpaceX launches has improved by time from years 2013 to 2020.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a 3D wireframe or a network of data points. The overall effect is futuristic and dynamic, suggesting concepts like data flow, digital communication, or complex systems.

Section 2

## Insights drawn from EDA

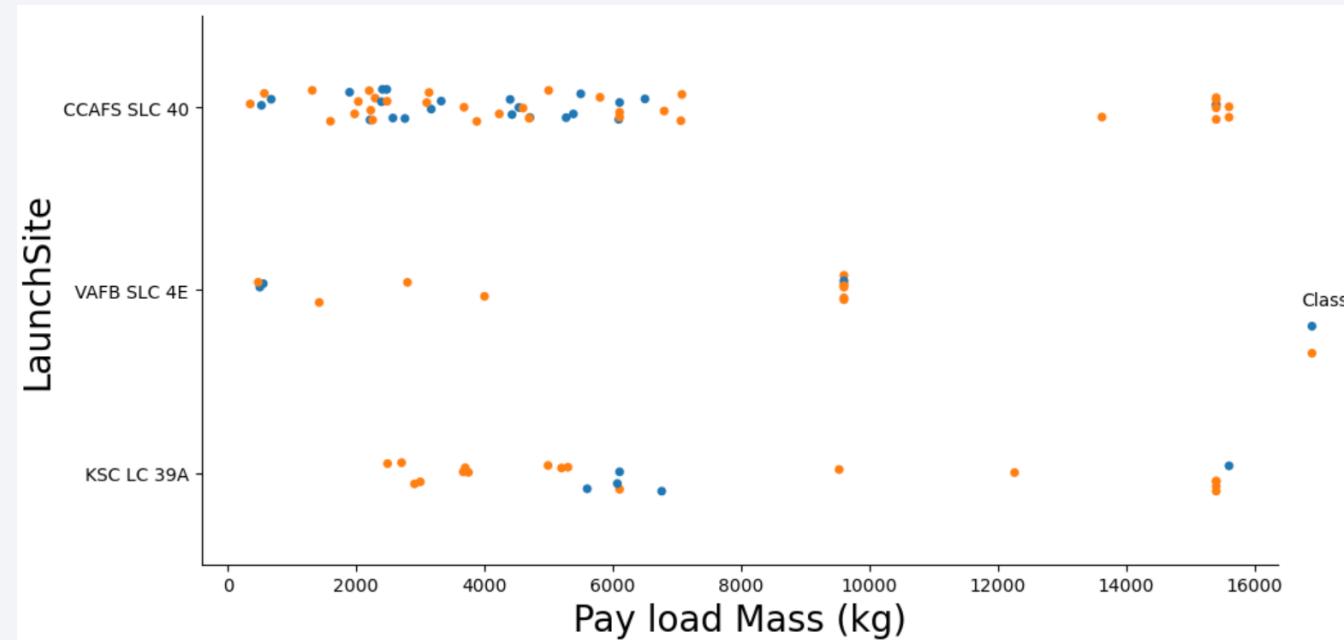
# Flight Number vs. Launch Site



CCAFS LC-40 has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%.

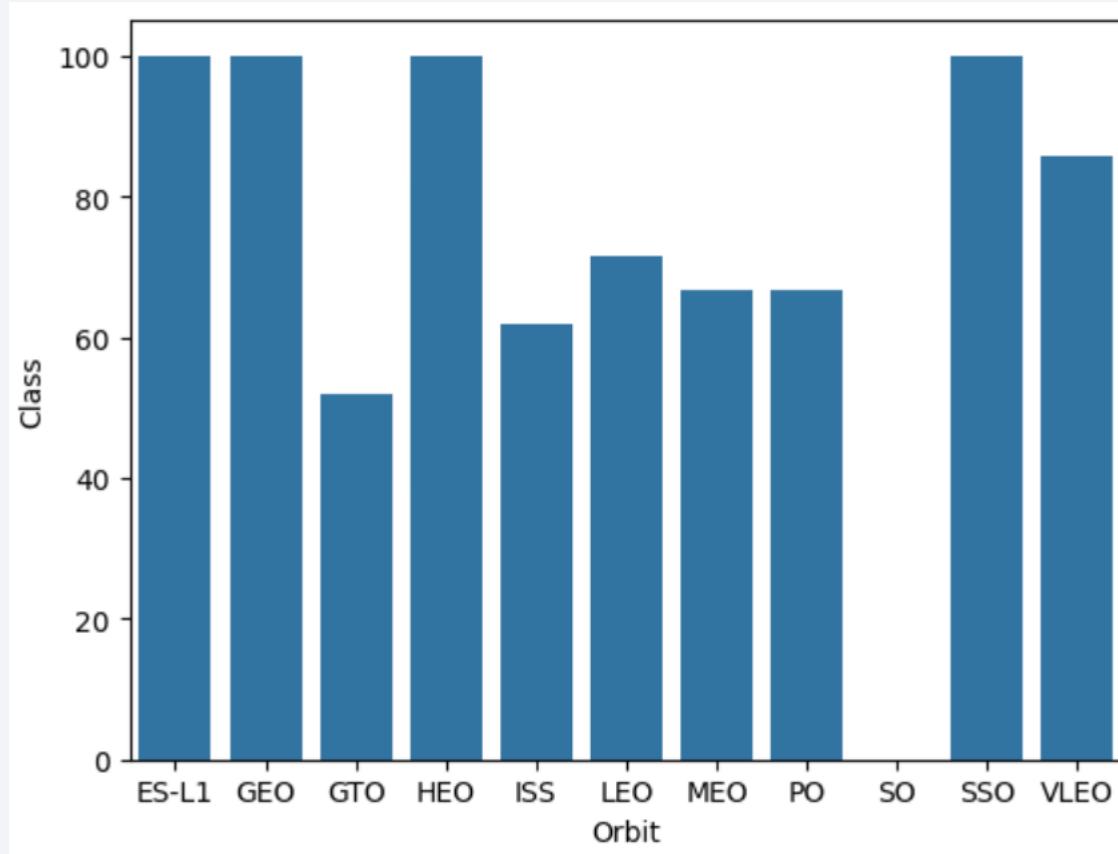
- **Launch Site: CCAFS SLC 40:** Flight numbers <20 → mostly failed, 80<Flight numbers → mostly successful, This launch site has the highest number of launches compared to other sites.
- **Launch Site\_VAFB SLC 4E:** 20<Flight numbers → mostly successfully
- **Launch Site\_KSL LC 39A:** 20<Flight numbers<40 → mostly successful

# Payload vs. Launch Site



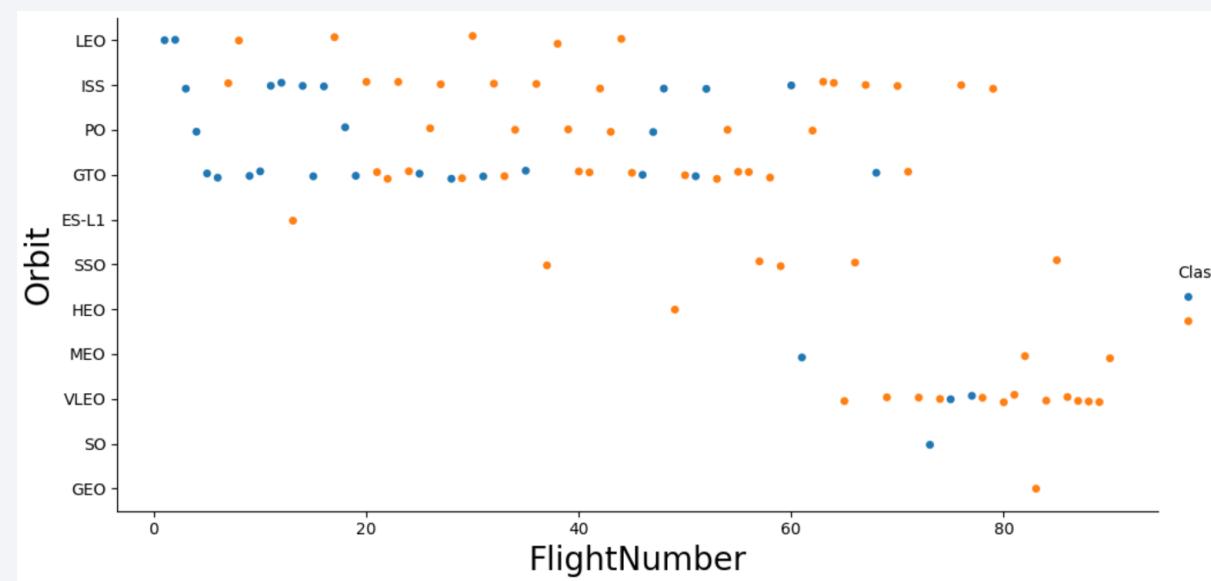
- CCAFS SLC 40: launches with higher pay load mass have higher success probability.
- VAFB-SLC 4E launch site: No rockets launched for payload mass greater than 10000.

# Success Rate vs. Orbit Type



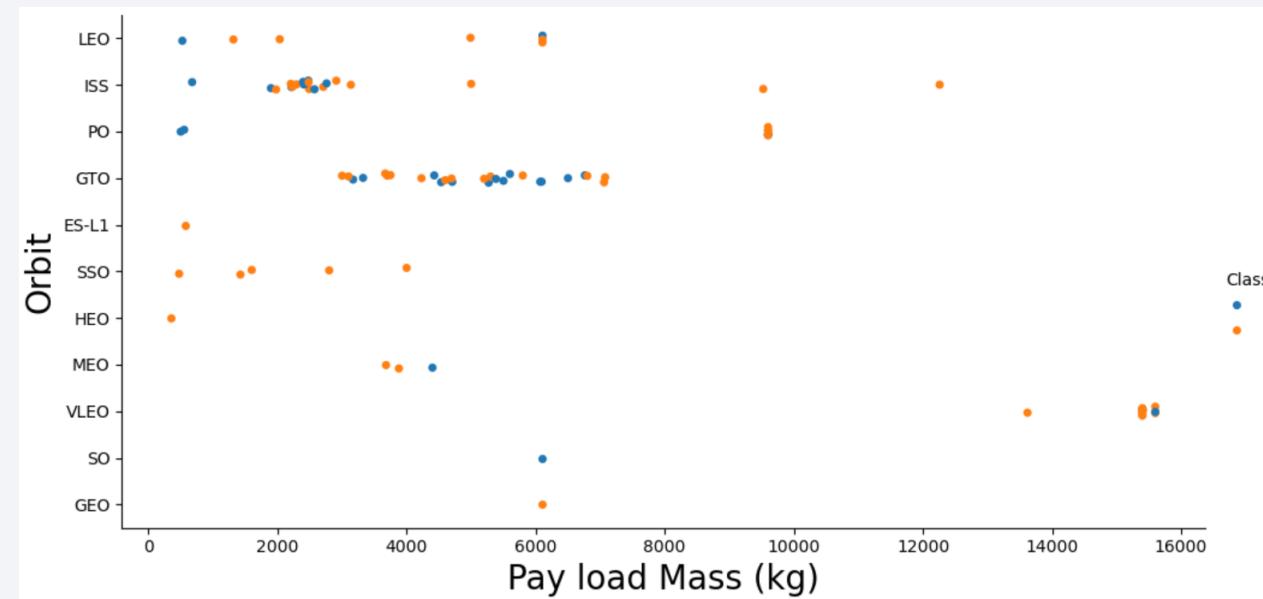
- ES-L1, GEO, HEO, and SSO ➔ Success rate of 100%
- VLEO ➔ > 80% success rate
- LEO ➔ > 70% success

# Flight Number vs. Orbit Type



- ISS, GTO, LEO, MEO, & VLEO orbits: higher flight number → higher chance of success
- SSO, ES-L1, GEO & HEO orbits: 100% success for all flight numbers

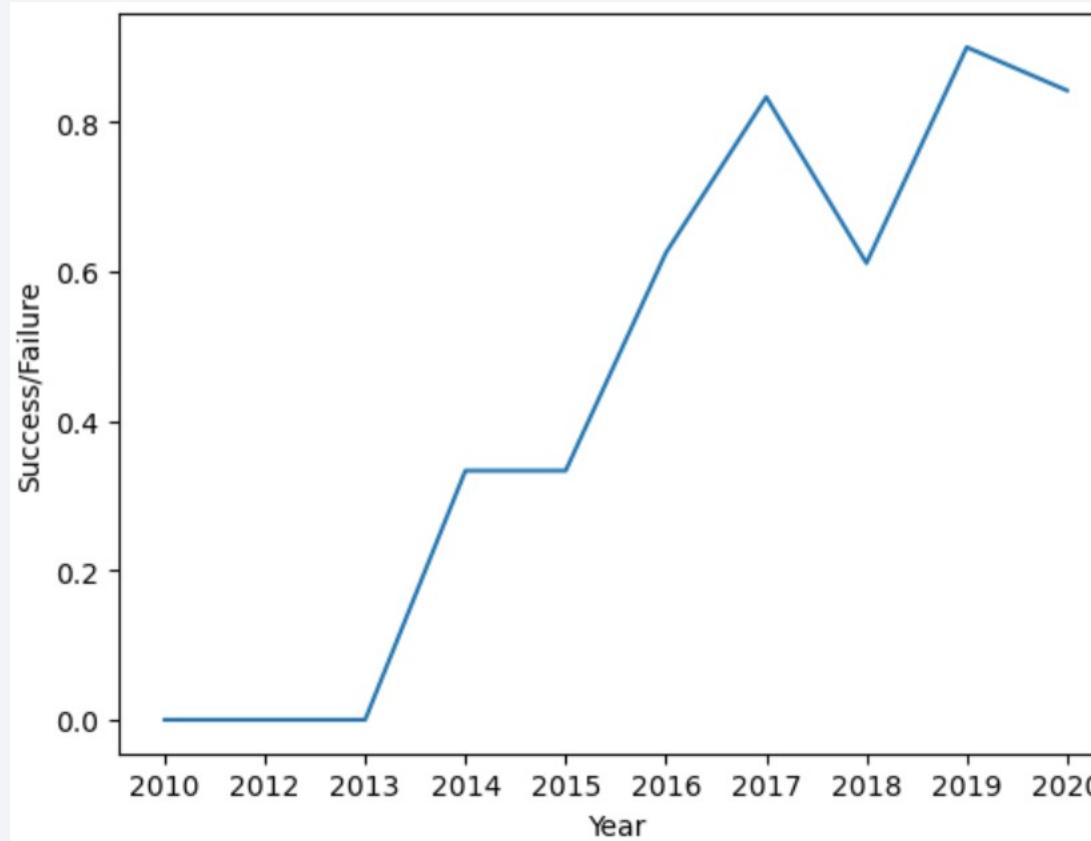
# Payload vs. Orbit Type



- Most of launches for ISS orbit have pay load mass of 2000 to 4000kg.
- Most of launches for GTO orbit have pay load mass of 3000 to 8000kg.

# Launch Success Yearly Trend

---



- The success rate was zero before 2013, and it has increased over time. However, a decline in the success rate was observed in 2018 and 2020.
- Maximum success rate (0.9) happened in 2019.

# All Launch Site Names

---

```
%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

- "DISTINCT" is used to list the unique categories or values present in a specific column of data set.

# Launch Site Names 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_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (recovery)
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 (recovery)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

# Total Payload Mass

---

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

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

<b>SUM(PAYLOAD_MASS__KG_)</b>
45596

- SUM function is used to calculate The total payload mass for customers with the name 'NASA (CRS)'  
.

# Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION ='F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

- AVG function was used to calculate the average payload mass carried by booster version F9 v1.1.

# First Successful Ground Landing Date

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

- SQL query was used to find the date of the first successful ground pad landing.

# Successful Drone Ship Landing with Payload between 4000 and 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_ >
```

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

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

- Using the keywords “BETWEEN” and “AND”, the names of boosters that had payload masses between 4000kg and 6000kg and had successfully landed on a drone ship were shown.

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

Mission_Outcome	Total_Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- COUNT function was used to calculate the total number of successful and failed missions.
- Results show that there were 100 successful missions and 1 failed mission.

# Boosters Carried Maximum Payload

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)
```

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

- The boosters that carried the maximum payload were listed using a sub-query with the MAX function, and a total of 12 were found.

# 2015 Launch Records

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 substr(Date,4,2) as month, DATE, BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome FROM SPACEXTBL where (La
```

```
* sqlite:///my_data1.db
(sqlite3.OperationalError) incomplete input
[SQL: SELECT substr(Date,4,2) as month, DATE, BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome FROM SPACEXTBL where (Lan
ding_Outcome = 'Failure (drone ship)' and substr(Date,7,4)='2015')
(Background on this error at: http://sqlalche.me/e/e3q8)
```

```
%sql SELECT substr(Date,6,2) AS Month, substr(Date,1,4) AS Year, DATE, BOOSTER_VERSION, LAUNCH_SITE, Landing_Outc
```

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

Done.

Month	Year	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- Results show that landing outcome of Failure (Drone Ship) and launch date in the year 2015

# 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)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
%sql SELECT Date, Landing_Outcome, COUNT(*) AS Count_Landing_Outcome FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY Count_Landing_Outcome DESC
```

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

Date	Landing_Outcome	Count_Landing_Outcome
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

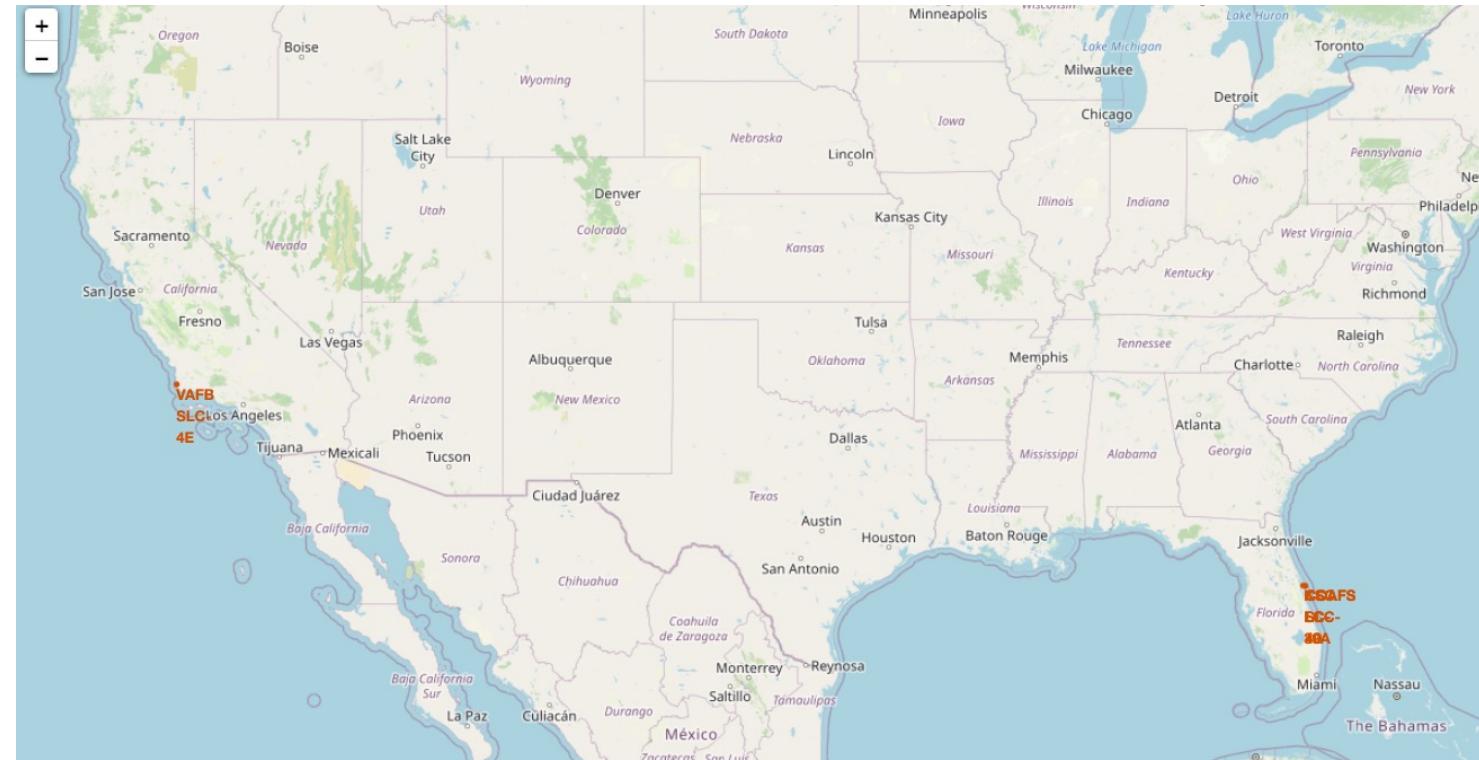
- Landing outcomes from 2010-06-04 to 2017-03-20 were listed and sorted in descending order.

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against a dark blue and black sky. City lights are visible as small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper left quadrant, the green and yellow glow of the Aurora Borealis (Northern Lights) is visible.

Section 3

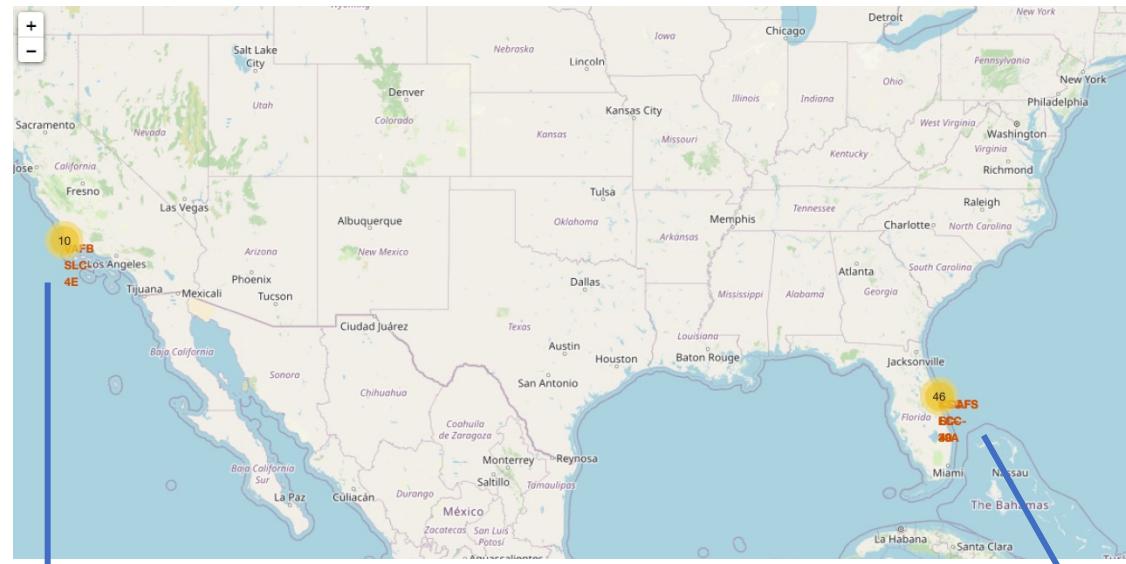
# Launch Sites Proximities Analysis

## Interactive map with Folium: Launch site locations for SpaceX Falcon 9

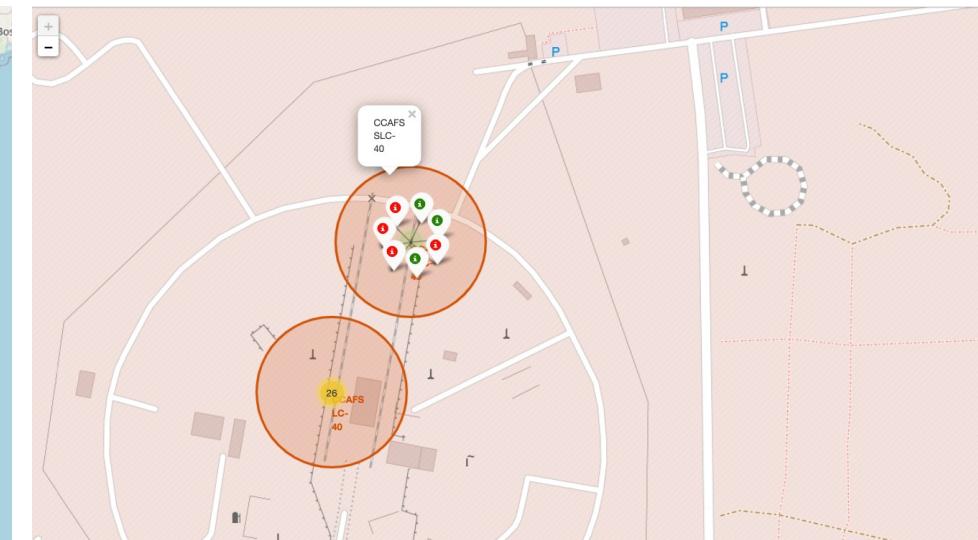


This analysis shows that launch sites are located on the west coast in California, and the east coast in Florida.

## Interactive map with Folium: Successful/failed launches for each site

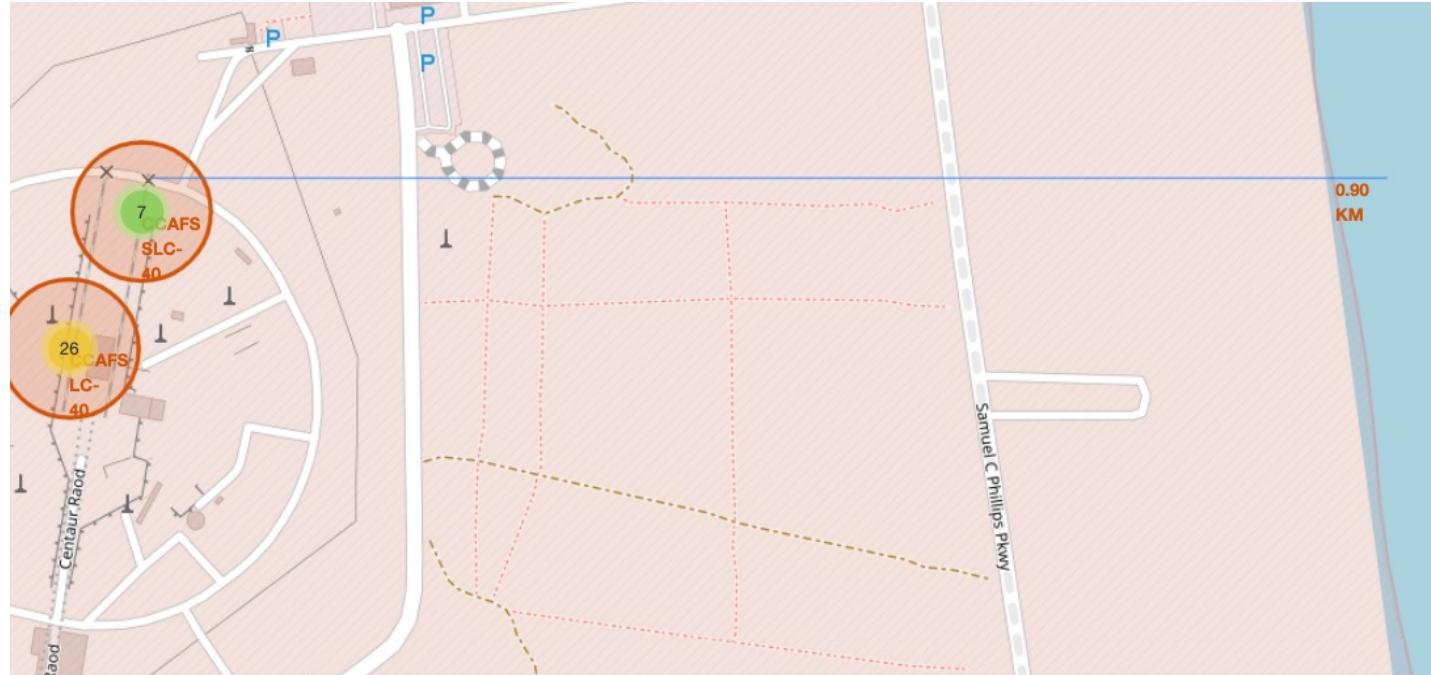


- SpaceX VAFB SLC-4E Launch Site has a single Launch Site Stationed of California.
- Our Cluster Marker indicates 10 Falcon 9 launches have taken place at this site.



SpaceX Launch Sites KSC LC-39A , CCAFS LC40 and CCAFS SLC-40 are located in Florida and 46 lunches have taken place.

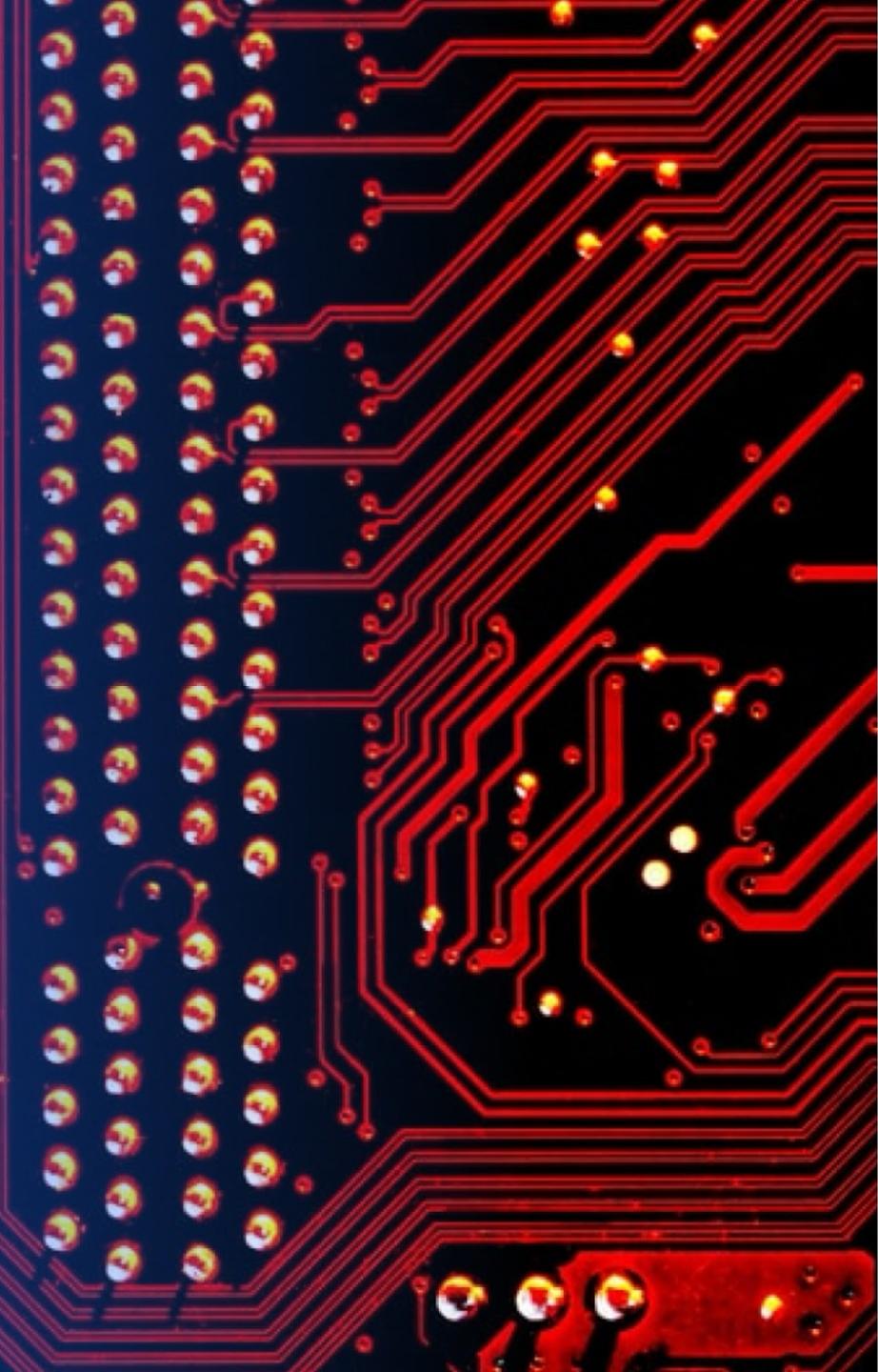
# Interactive map with Folium: Launch site distance from coastline



We can see that the nearest coastline to the Cape Canaveral sites is within 1 kilometer and exactly measured to 0.90km for the LC-40 site.

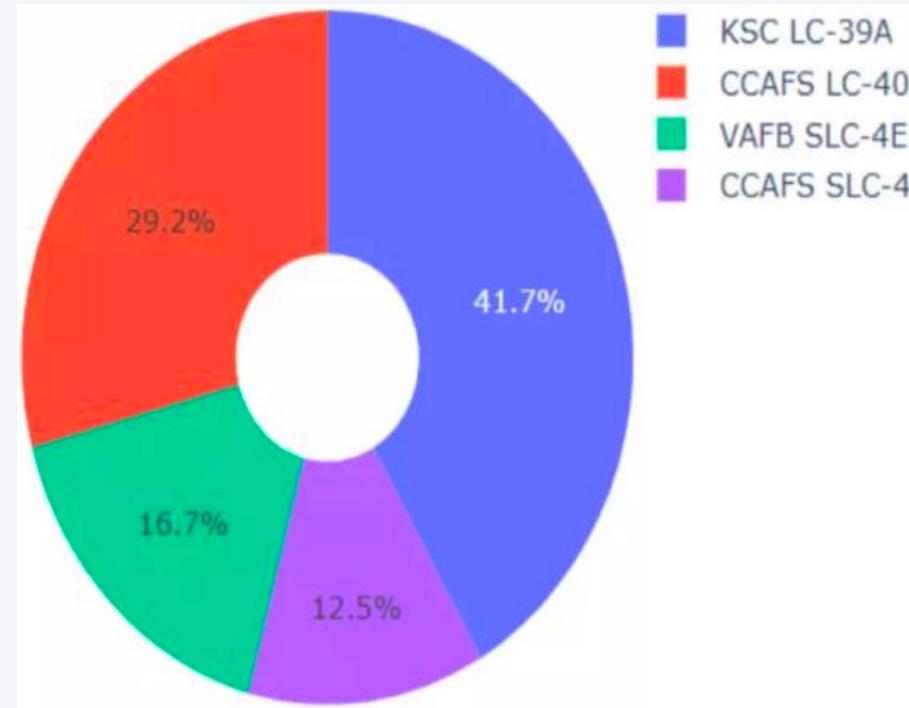
Section 4

# Build a Dashboard with Plotly Dash



# Pie chart for launch success count for all sites

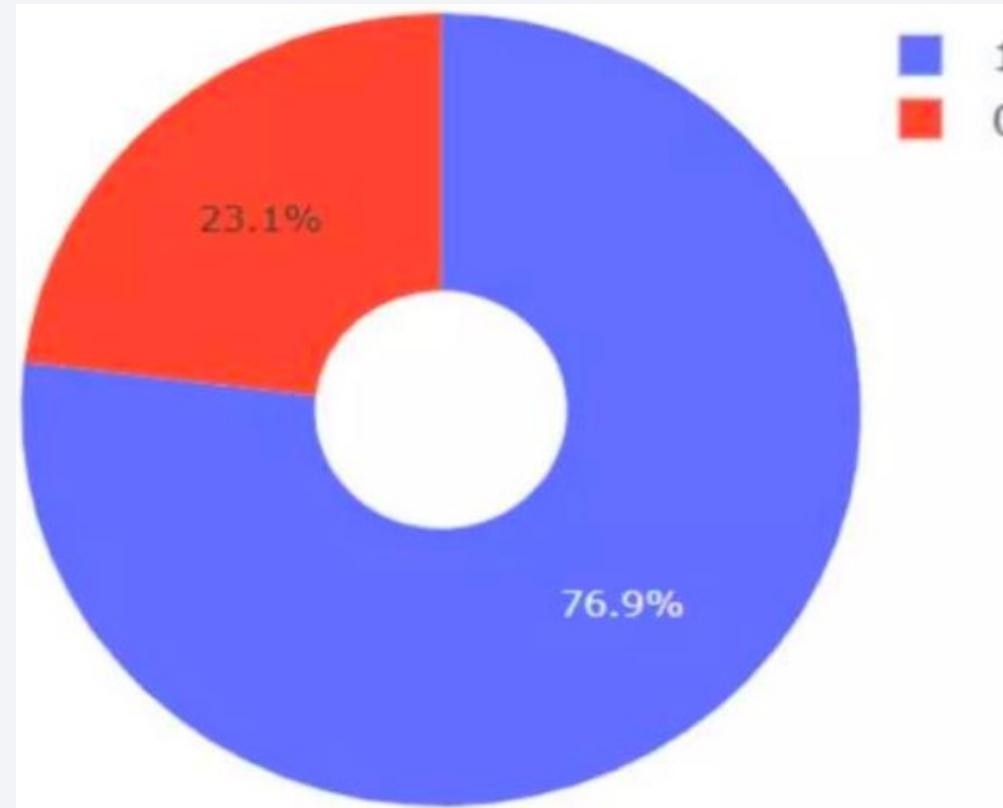
---



KSC LC-39A Launch site → Most successful launches compared to other sites (41.7%)

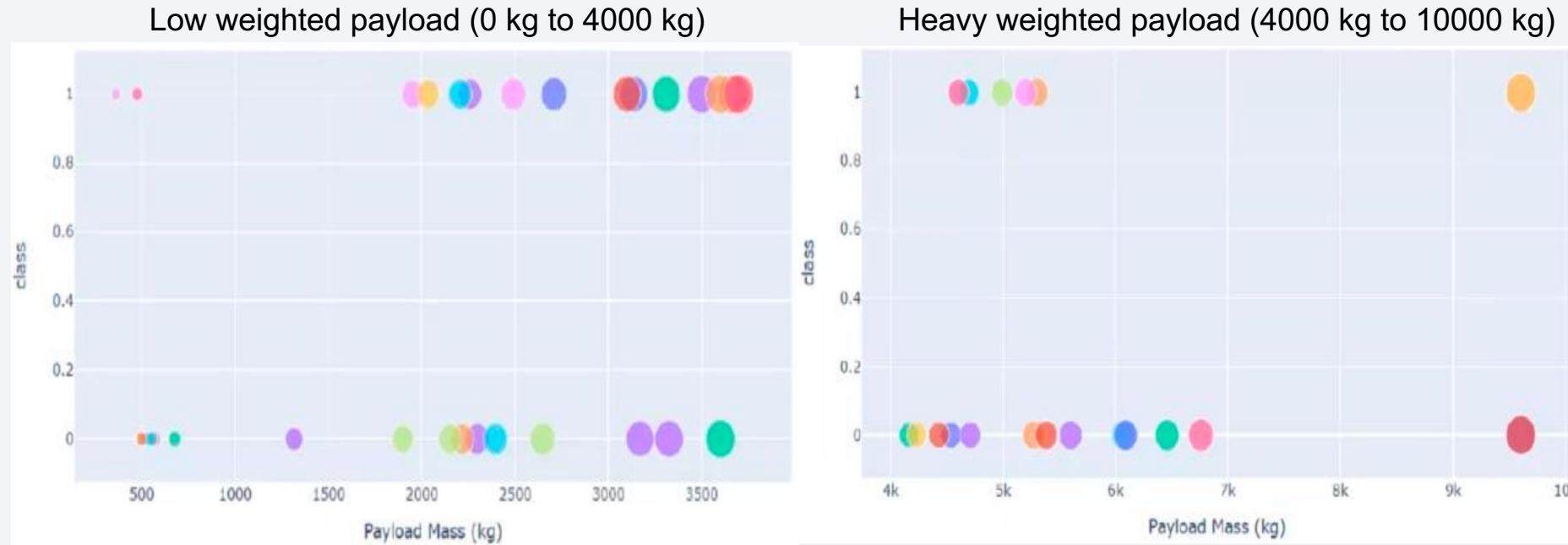
## Pie chart for KSC LC-39A launch site which has the highest launch success ratio

---



KSC LC-39A launch sit → 76.9% success rate, and 23.1% failure rate.

# Payload vs. Launch Outcome scatter plot for all sites



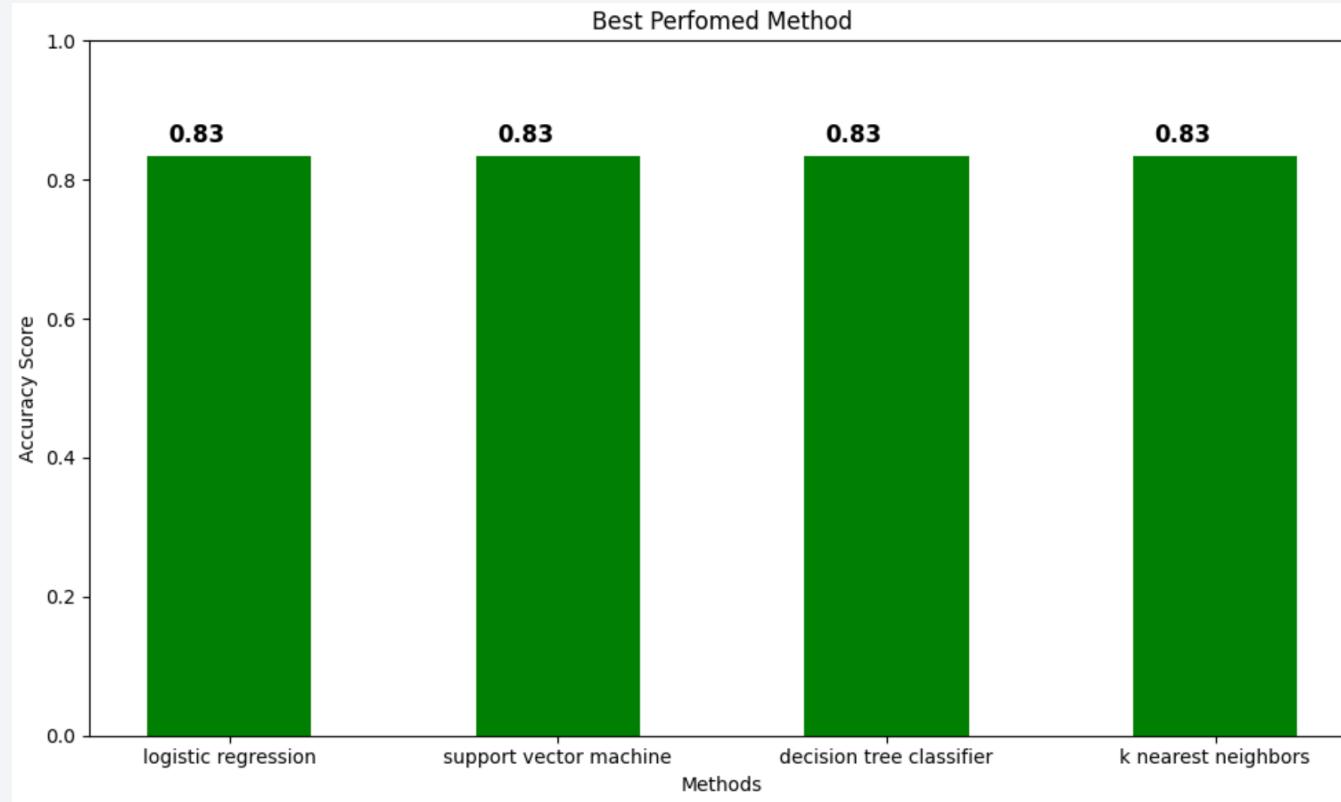
Success rates for low weighted is higher than heavy weighted payload!

# Predictive Analysis (Classification)



# Classification Accuracy

---



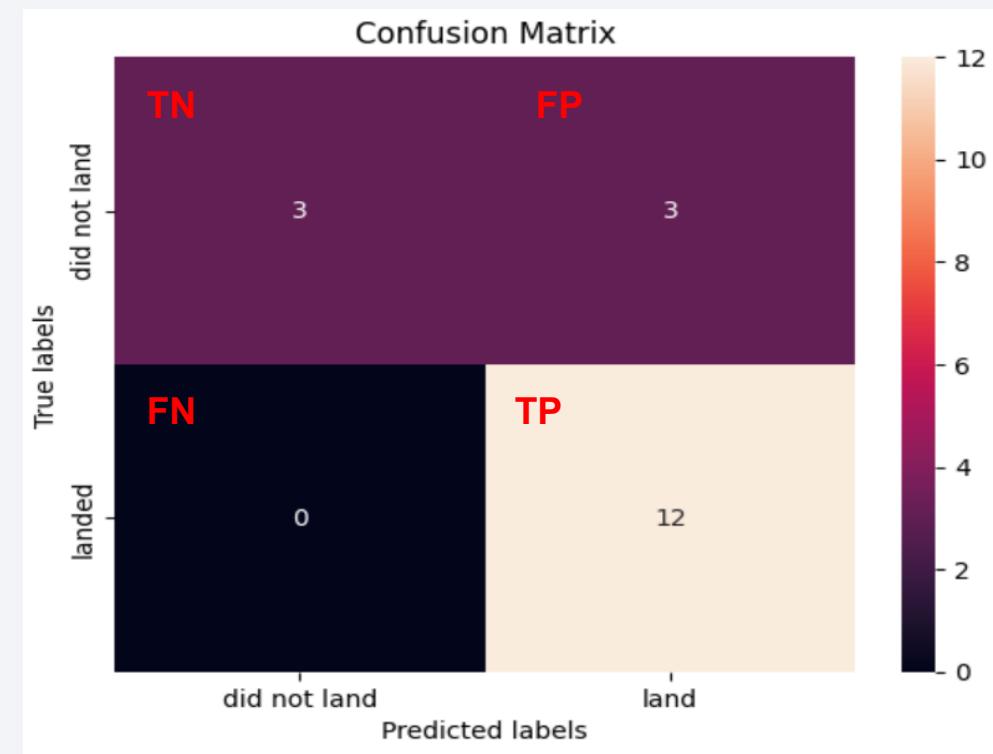
Accuracy scores for all four models are the same.

# Confusion Matrix for all models

Confusion matrix for all four models are also the same.

We see that among 18 samples, 15 sample were correctly predicted ( $3TN + 12TP$ ), and 3 samples were not predicted correctly ( $0FN + 3FP$ ).

- Sensitivity:  $12/(12+0)=100\%$
- Specificity =  $3/(3+3) = 50\%$
- Accuracy:  $(3+12)/(3+3+0+12)=83\%$



# Conclusions

---



- All Models showed the same accuracy (%83) and confusion matrix in this analysis.
- Low-weighted payloads perform better than the heavier payloads.
- KSC LC-39A Launch Site has the highest probability of success rate.
- ES-L1, SSO, HEO and GEO Orbits have the highest rate of landing successfully.
- The success rate for SpaceX launches has improved by time from years 2013 to 2020.

Thank you!

