IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<MURAT ILDIZ>
<18.01.2023>

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection through API, Web Scraping and Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization and Visual Analytics with Folium
  - Machine Learning Prediction Models
- Summary of all results
  - Exploratory Data Analysis result are given both with numbers and screenshots.
  - Predictive Analytics result are given.

# 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

  - Factors affecting the performance of the rocket.
  - Interaction analysis of various rocket parameters.
  - What must me the best operation conditions for succesful landing.

Section 1

# Methodology

# Methodology

## Executive Summary

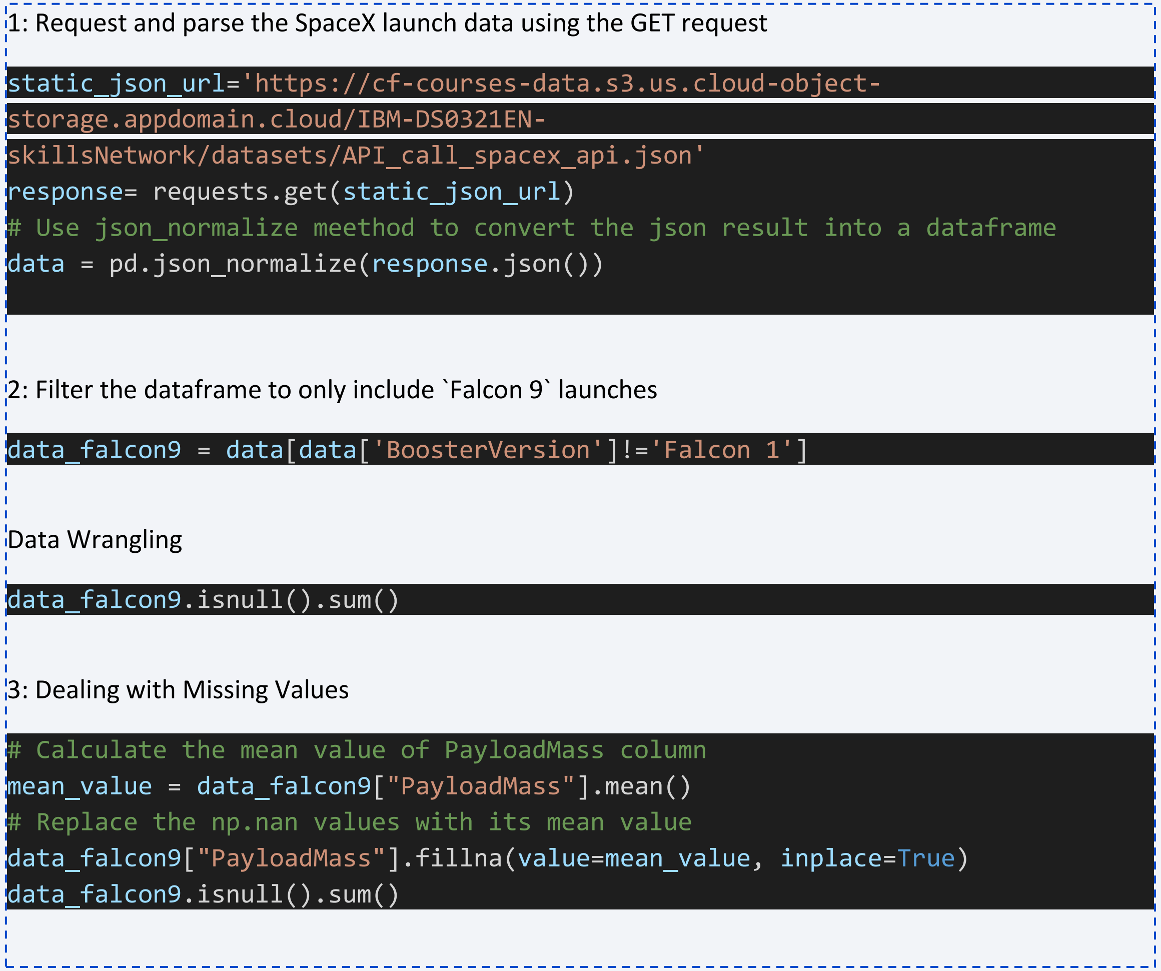
- Data collection methodology:
  - API and Web Scraping Techniques used for collecting data.
- Perform data wrangling
  - Raw data was filltered and null values replaced with suitable values such as mean of related item.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

- Data collected from various sources such as API's and Web Pages.
- We used Python coding and related libraries to extract and wrangle the data from the sources.
- Request Methods, BeautifulSoup Libraries, SQL queries, Panda's library are examples of the data collection methods.

# Data Collection – SpaceX API

- We used SpaceX API to collect data. Then data was cleaned and necessary data wrangling and formatting operations were carried out.
- The link to SpaceX API : https://github.com/mildiz/testrepo/blob/master/jupyter-labs-spacex-data-collection-api.ipynb



1: Request and parse the SpaceX launch data using the GET request

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBM-DS0321EN-
skillsNetwork/datasets/API_call_spacex_api.json'
response= requests.get(static_json_url)
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

2: Filter the dataframe to only include `Falcon 9` launches

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

Data Wrangling

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

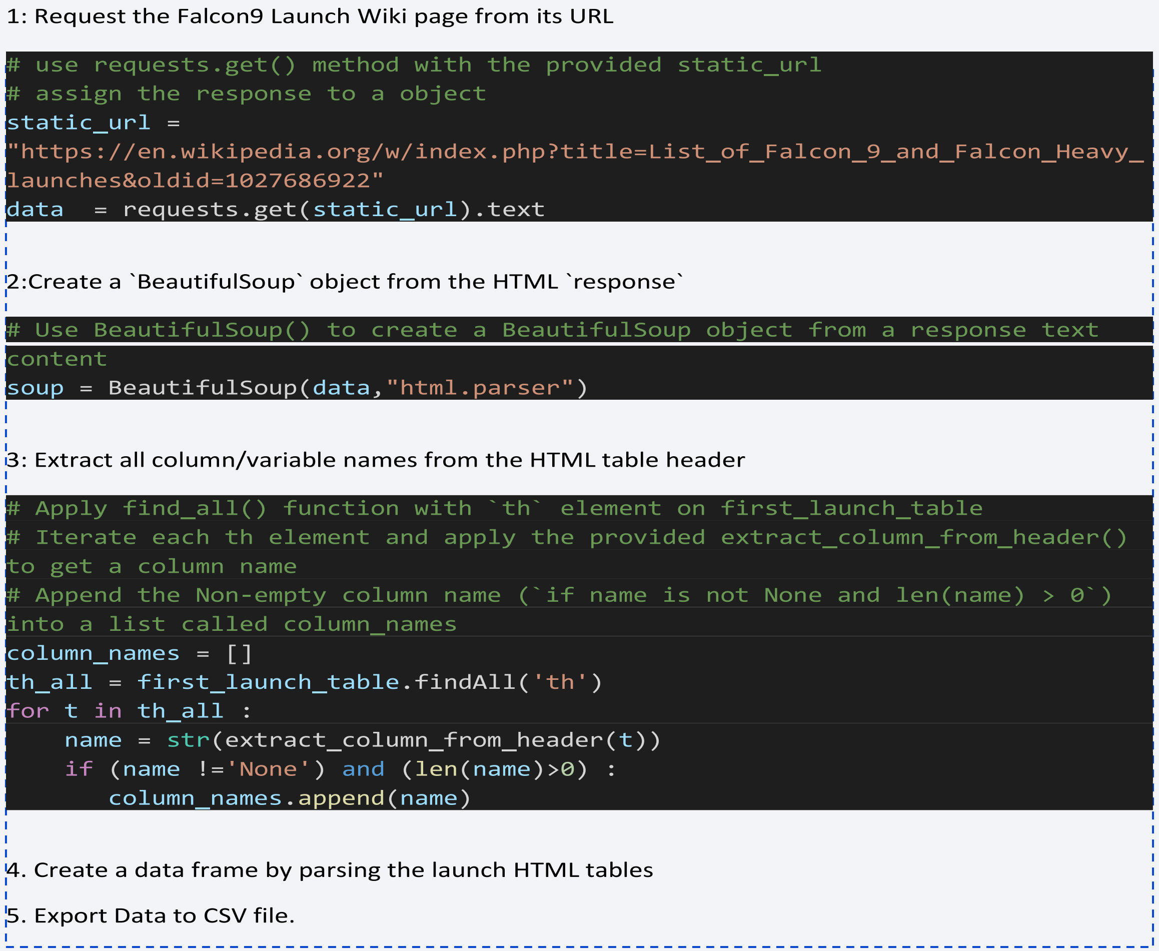
3: Dealing with Missing Values

```
# Calculate the mean value of PayloadMass column
mean_value = data_falcon9["PayloadMass"].mean()
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].fillna(value=mean_value, inplace=True)
data_falcon9.isnull().sum()
```

# Data Collection - Scraping

- Web page Wikipedia scraped with Beautiful Soup, from the table data extracted and then recorded to a CSV file with a format to use later analysis.

- The link to the notebook is: https://github.com/mildiz/testrepo/blob/master/jupyter-labs-webscraping.ipynb



1: Request the Falcon9 Launch Wiki page from its URL

```
# use requests.get() method with the provided static_url
# assign the response to a object
static_url =
"https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_
launches&oldid=1027686922"
data  = requests.get(static_url).text
```

2: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")
```

3: Extract all column/variable names from the HTML table header

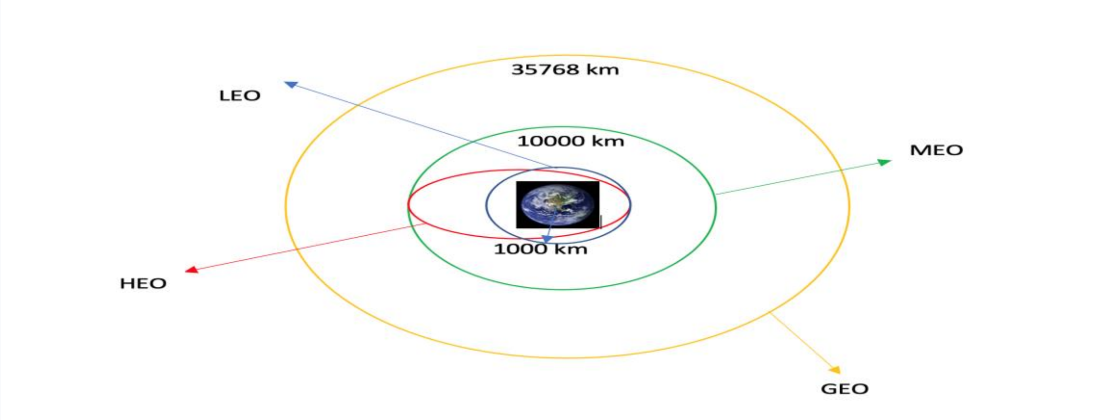
```
# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header()
to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`)
into a list called column_names
column_names = []
th_all = first_launch_table.findAll('th')
for t in th_all :
    name = str(extract_column_from_header(t))
    if (name !='None') and (len(name)>0) :
       column_names.append(name)
```

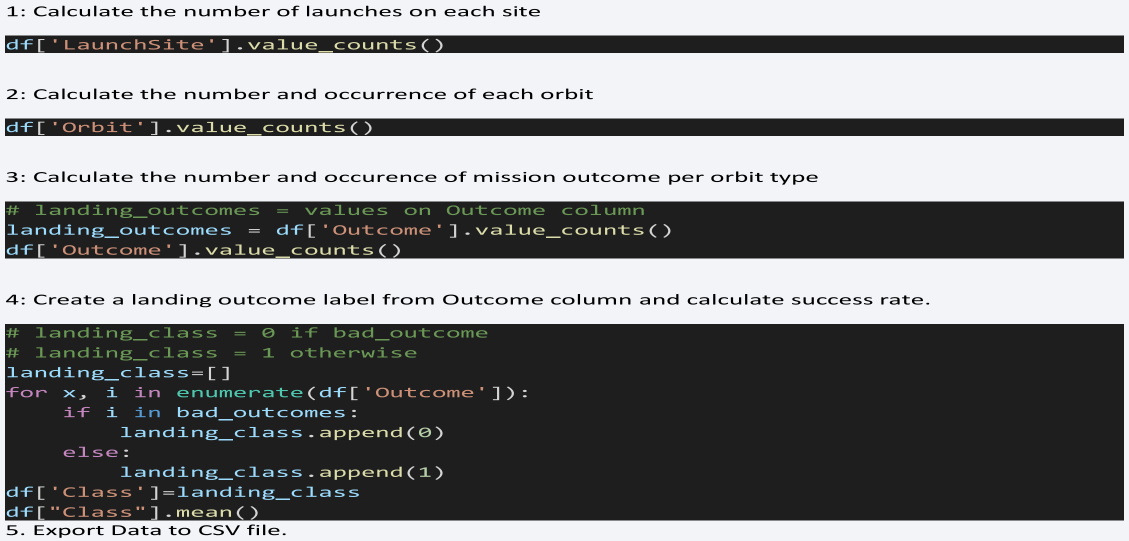
4. Create a data frame by parsing the launch HTML tables

5. Export Data to CSV file.

# Data Wrangling

- We performed exploratory  data analysis and determined training labels.
- We added a column to dataframe as Class to define successfull landings where 1 is for successful 0 is not successfull.
- The link to the notebook is https://github.com/mildiz/testrepo/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb.




1: Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()
```

2: Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

3: Calculate the number and occurence of mission outcome per orbit type

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
df['Outcome'].value_counts()
```

4: Create a landing outcome label from Outcome column and calculate success rate.

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for x, i in enumerate(df['Outcome']):
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
df["Class"].mean()
```

5. Export Data to CSV file.

# EDA with Data Visualization

- FlightNumber vs. PayloadMass Plot : We see as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
- FlightNumber vs. Launch Site Plot: We see that increase in flight number increase in succes at launch site.
- Class vs. OrbitType Plot: We observe success per orbit.
- Orbit vs. FlightNumber Plot: We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- Orbit vs. PayloadMass Plot: With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) .
- Class vs. Date : We can observe that the sucess rate since 2013 kept increasing till 2020.
- The link to notebook is : https://github.com/mildiz/testrepo/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

# EDA with SQL

- SQL Querries carried out:
- The names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- The total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- The date when the first succesful landing outcome in ground pad was acheived.
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes
- The names of the booster_versions which have carried the maximum payload mass.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

- The link to Notebook is :https://github.com/mildiz/testrepo/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- We marked all launch sites, and added map objects  circles, markers, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes 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.
- The link to the notebook is: https://github.com/mildiz/testrepo/blob/master/lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches of a launch site or all sites which we can choose it from a dropdown menu.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version with Payload Range Slider which helped us change Payload Mass interactively
- The link to the notebook is https: https://github.com/mildiz/testrepo/blob/master/spacex_dash_app.py

# Predictive Analysis (Classification)

- By using numpy, pandas, sklearn, we loaded, transformed the data and split our data into training and testing after standartising.
- With the help of different machine learning models and tuning different hyperparameters using GridSearchCV, we run the machine learning models such as Logistic Regression, Support Vector Machine, Decision Tree Classifier, K-nearest Neighbours.
- We plotted Confusion Matrix of each model. Finally we have compared each model's AUC, F1-Score,Precision,Recall,Accuracy. Accuracy is our metric to find the best model.
- The link to the notebook is: https://github.com/mildiz/testrepo/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

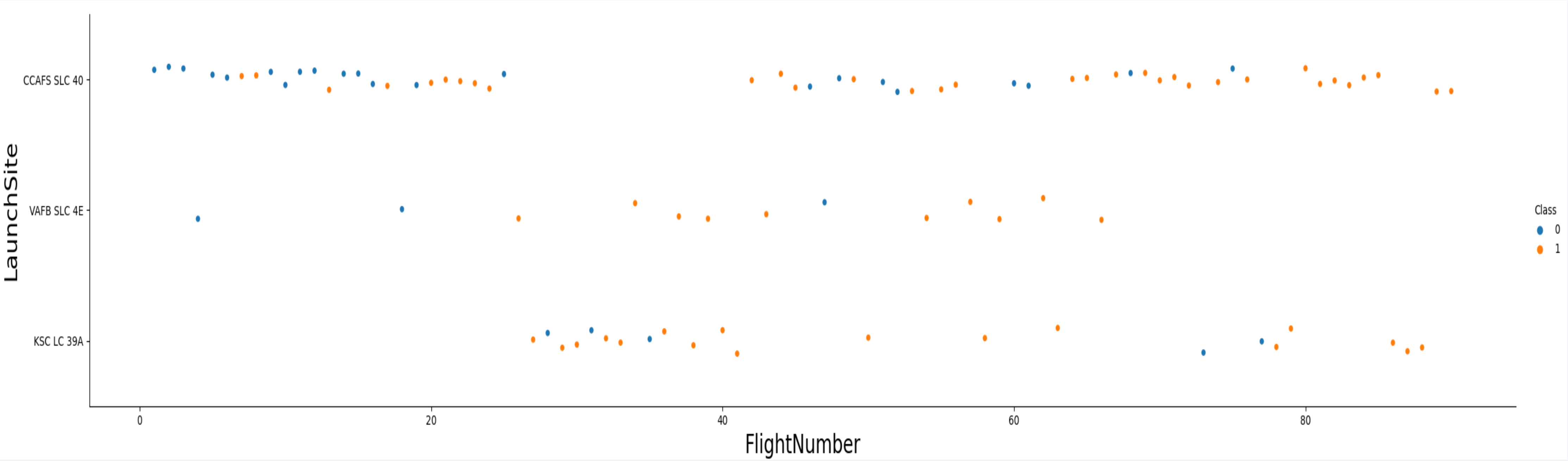
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

# Insights drawn from EDA

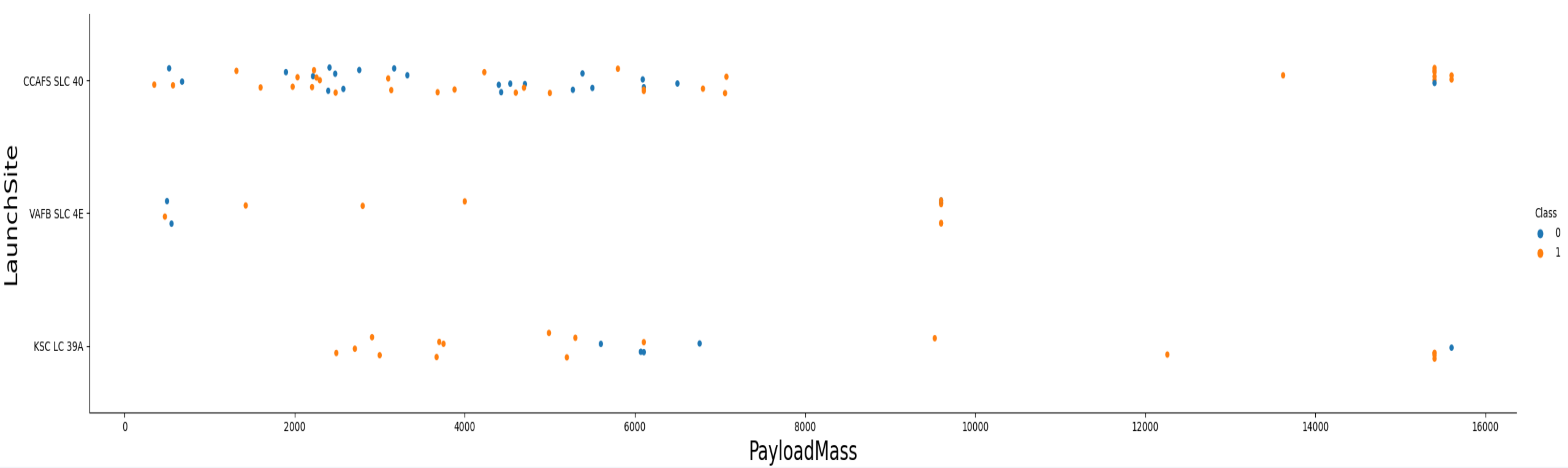
# Flight Number vs. Launch Site

- As increase in Flight Number we see increase in success each launch site.



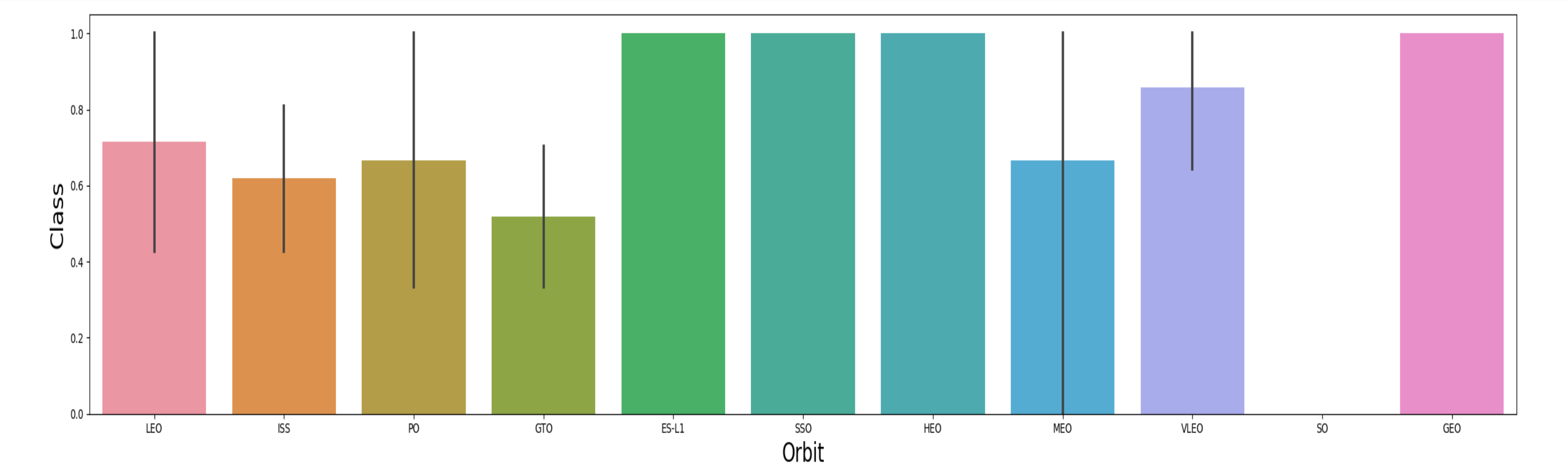
# Payload vs. Launch Site

- CCAFS SLC 40 has great success rates as payload increases where as VAFB SLC 4E has stabil success rate in the range of  wide payloadmass.



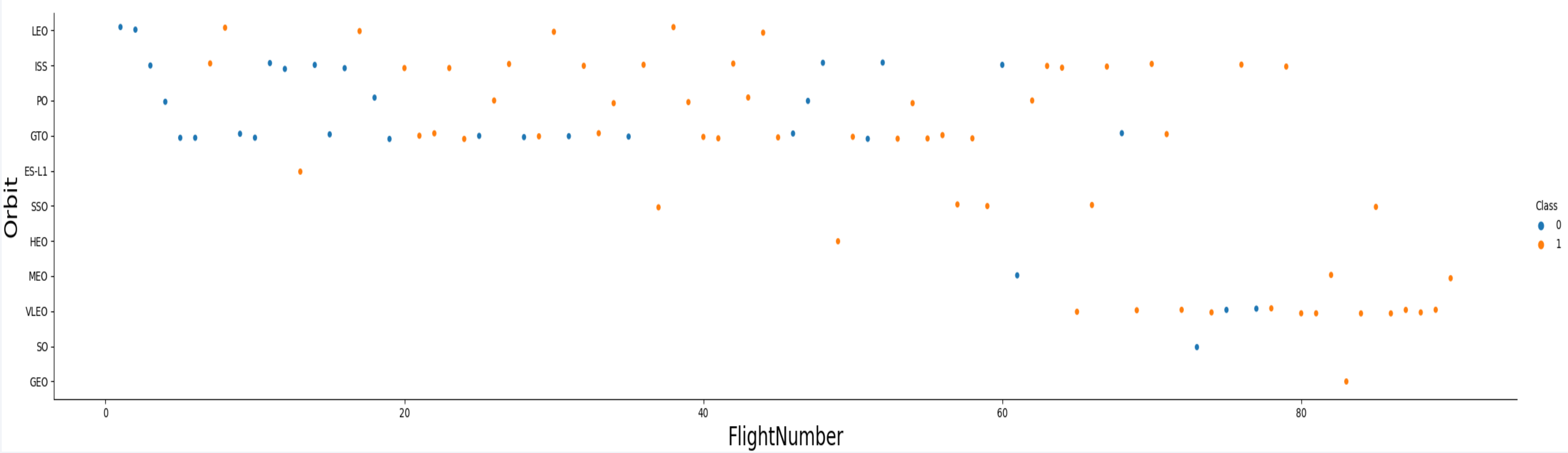
# Success Rate vs. Orbit Type

- From the plot, we can understand that ES-L1, SSO, HEO, VLEO and GEO have the most success rate.



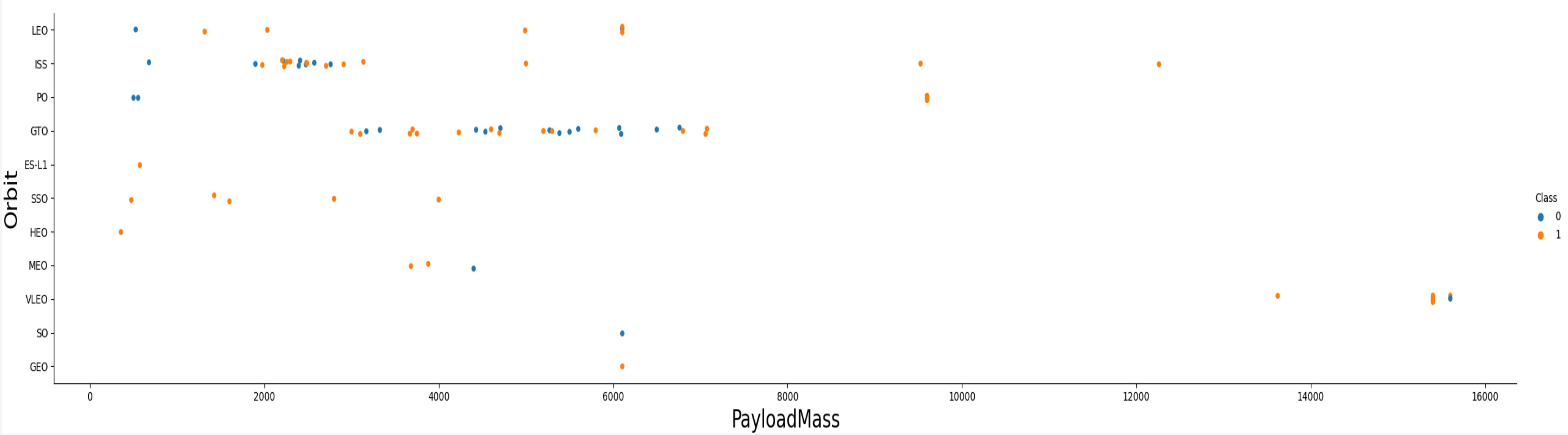
# Flight Number vs. Orbit Type

- We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



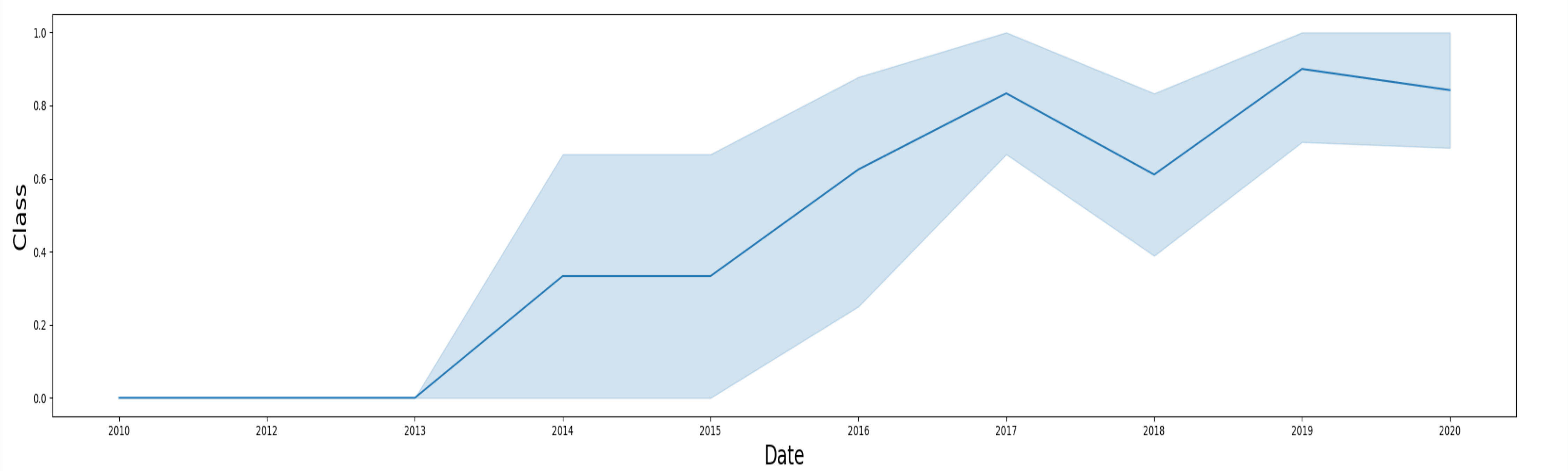
# Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission).



# Launch Success Yearly Trend

- We can observe that the sucess rate since 2013 kept increasing till 2020.



# All Launch Site Names

- Name of Launch_Sites
- CCAFS LC-40
- VAFB SLC-4E
- KSC LC-39A
- CCAFS SLC-40
- We use following query with DISTINCT to list Launch Sites
- %sql select distinct launch_site from SPACEXTBL

# Launch Site Names Begin with 'CCA'

- Following querry will give 5 records with Launch Site Names begin with ’CCA’
- %sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- Calculated Total Payload Mass of NASA with the following querry is 107.010 KG

- %sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like '%NASA%’

```
SUM(PAYLOAD_MASS__KG_)

107010
```

- In querry customer NASA is choosen with WHERE / LIKE parameter then payload mass is calculated with SUM function.

# Average Payload Mass by F9 v1.1

- The calculated average payload mass carried by booster version F9 v1.1with the following querry is 2534.6 KG
- %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_version like '%F9 v1.1%'
- In querry Booster_Version F9 v1.1 is choosen with WHERE / LIKE parameter then payload mass is calculated with AVG function.

# First Successful Ground Landing Date

- The calculated date of the first successful landing outcome on ground pad with the following querry is 01.05.2017.
- %sql select MIN(Date) from SPACEXTBL where (Landing_Outcome like '%ground pad%') and (Mission_Outcome like '%Success%’)
- In querry Date is choosen with WHERE / LIKE parameter Landing_outcome equals to ground pad and Mission_Outcome equals to Success then MIN function is used to calculate Date.

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Following querry is used to calculate Succesful Drone Ship Landing with Payload between 4000 and 6000 Kg. AND operator is used to querry with more than one value and condition.
- %sql select Booster_Version, Landing_Outcome , Mission_Outcome, PAYLOAD_MASS__KG_ from SPACEXTBL

  where Landing_Outcome like ('%Success (drone ship)%') and (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)

| Booster_Version | Landing_Outcome | Mission_Outcome | PAYLOAD_MASS__KG_ |
|---|---|---|---|
| F9 FT B1022 | Success (drone ship) | Success | 4696 |
| F9 FT B1026 | Success (drone ship) | Success | 4600 |
| F9 FT B1021.2 | Success (drone ship) | Success | 5300 |
| F9 FT B1031.2 | Success (drone ship) | Success | 5200 |

# Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes calculated with the following queery.
- %sql SELECT COUNT(Mission_Outcome) AS 'SuccessOutcome' FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Success%’

```
SuccessOutcome
100
```

- The total number of failure mission outcomes.
- %sql SELECT COUNT(Mission_Outcome) AS 'FailureOutcome' FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%'

```
FailureOutcome
1
```

# Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass are calculated with the following querry. A sub query is used as input for querryin maksimum payload.
- %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY Booster_Version

| 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

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 are calculated with the following querry. Substr(DATE,7,4) is used for querrying the YEAR in the DATE and YEAR condition is filtered. Substr(Date,4,2) is used for listing the months of the date.
- %sql SELECT substr(Date,4,2) as 'Month', Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure (drone ship)' AND (substr(Date,7,4) ='2015')

| Month | Booster_Version | Launch_Site | Landing_Outcome |
|---|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

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

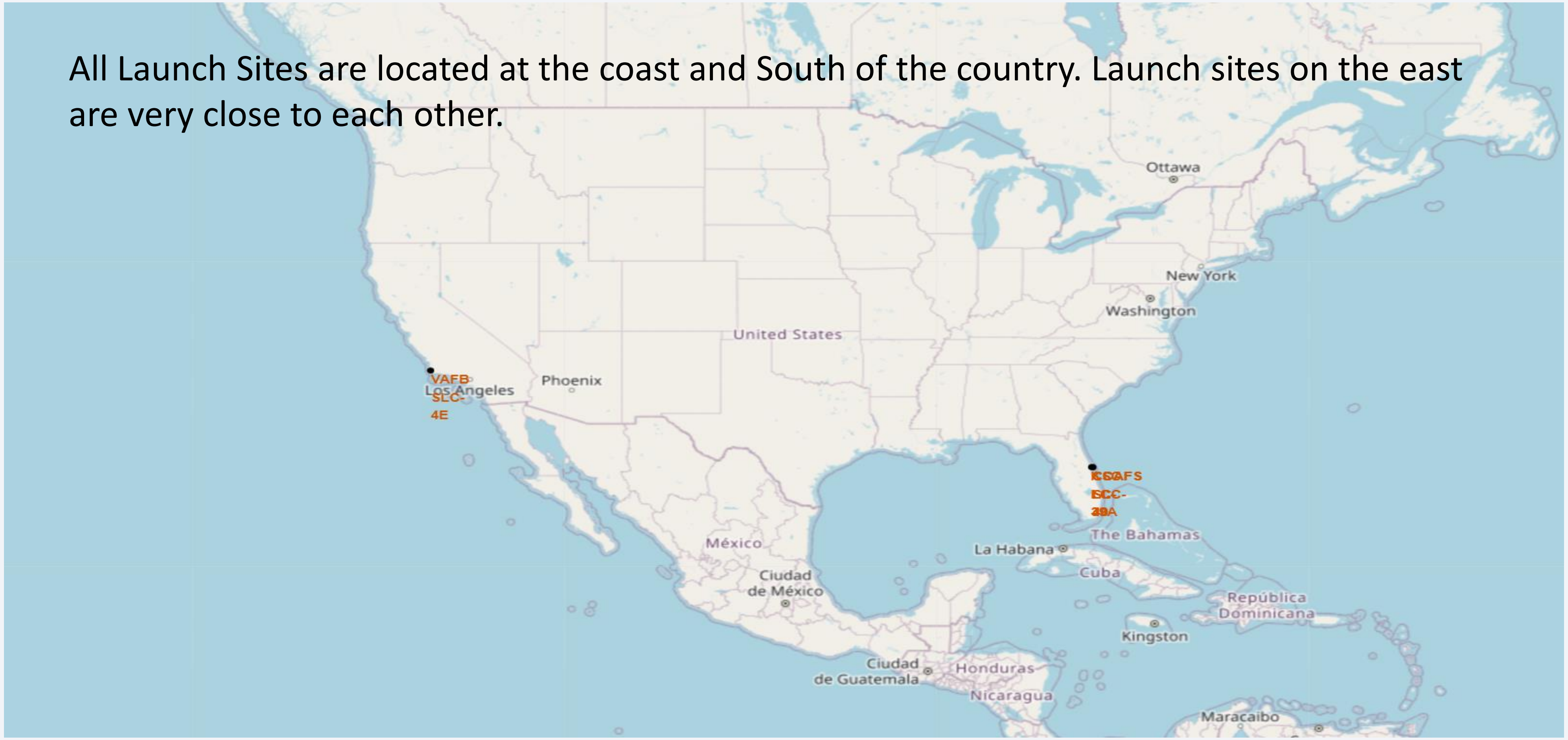
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order is calculated with the following querry. DATE is querried BETWEEN function and, the querry groupped by Landing Outcome and sorted descending.
- %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC

| Landing_Outcome | COUNT(Landing_Outcome) |
|---|---|
| Success | 20 |
| No attempt | 10 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |
| Failure (drone ship) | 4 |
| Failure | 3 |
| Controlled (ocean) | 3 |
| Failure (parachute) | 2 |
| No attempt | 1 |

Section 3

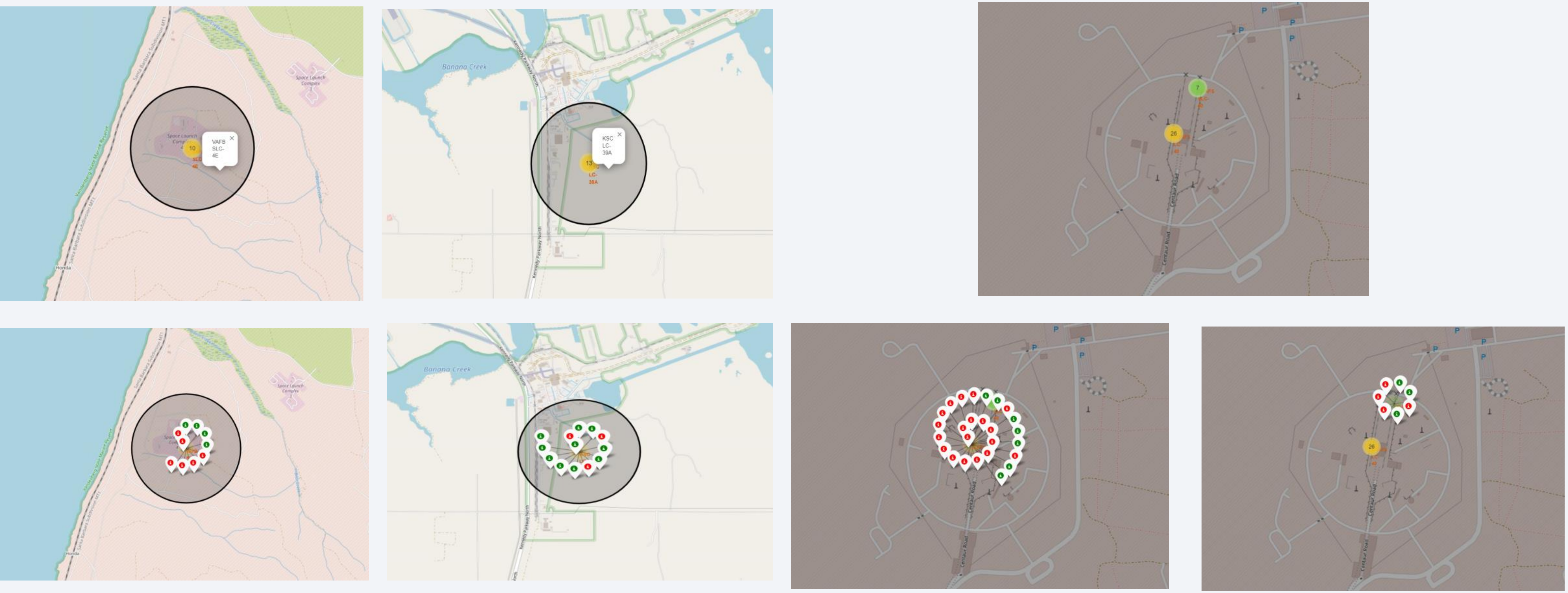
# Launch Sites Proximities Analysis

# SpaceX Launch Sites Location



All Launch Sites are located at the coast and South of the country. Launch sites on the east are very close to each other.

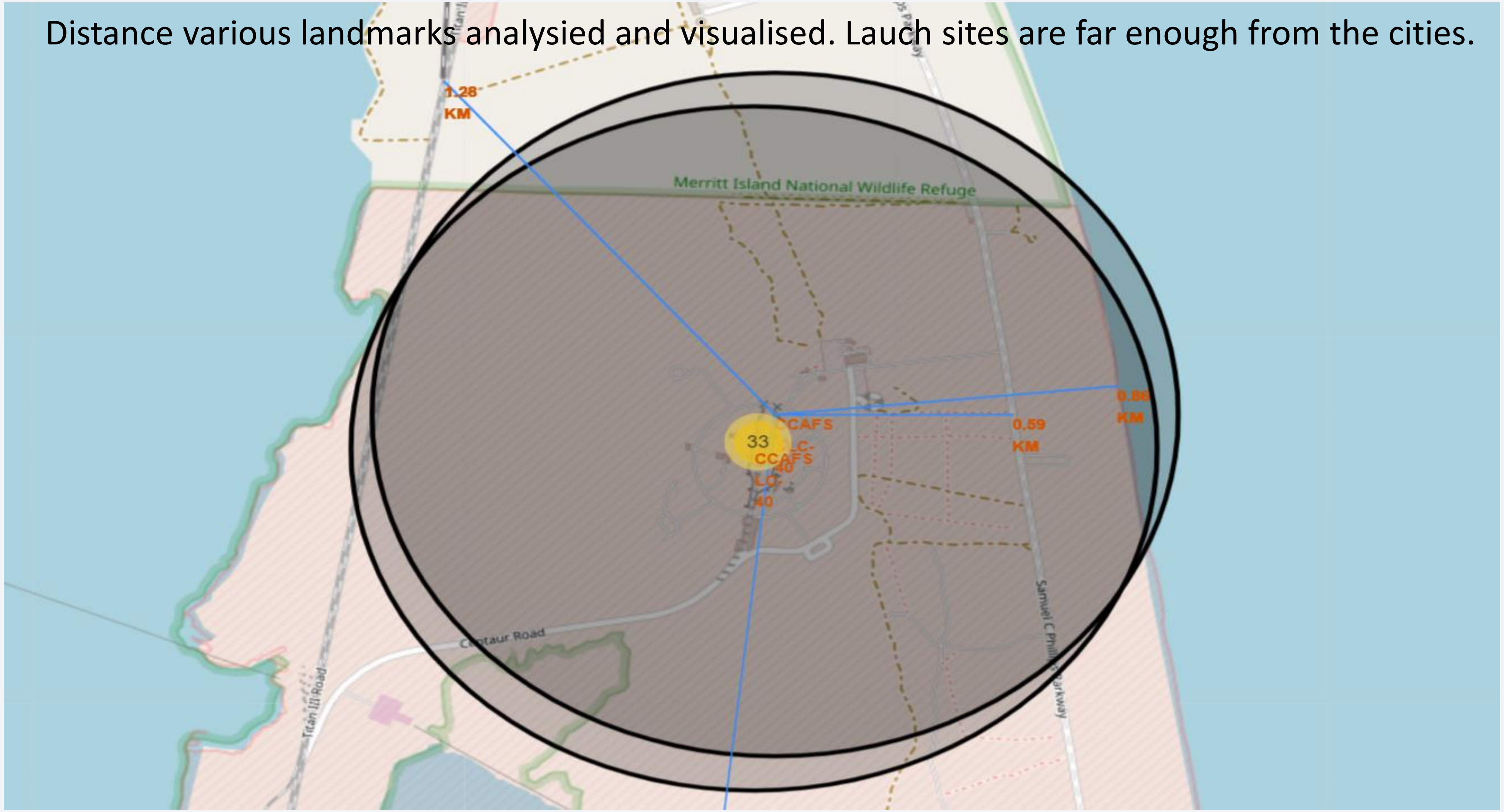
# Launch Performances By Sites Location

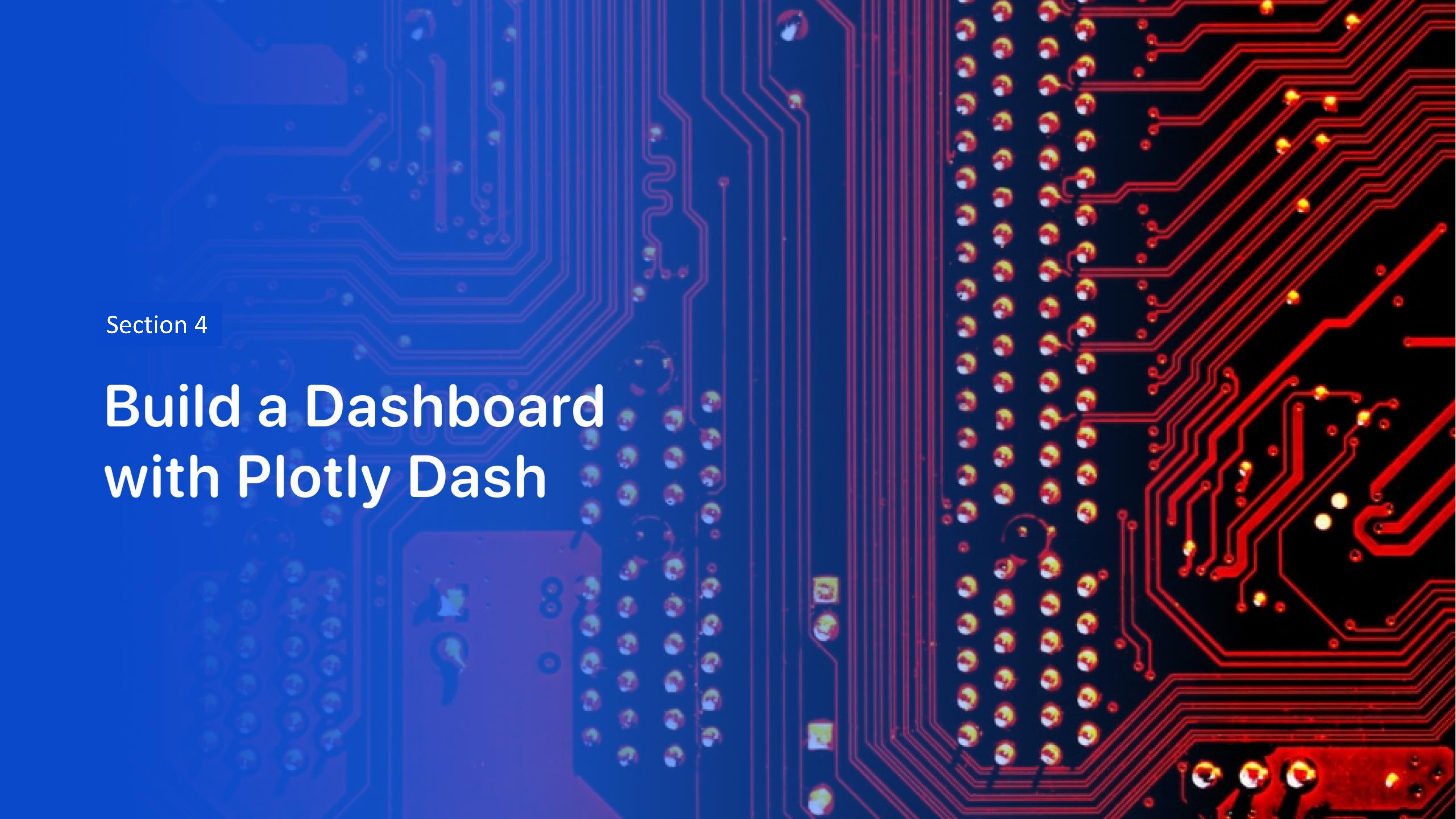


Mission completed witg success are shown with green flag. Name of the locations are also given on the map.

# Launch Site Distance To Landmarks

Distance various landmarks analysied and visualised. Lauch sites are far enough from the cities.

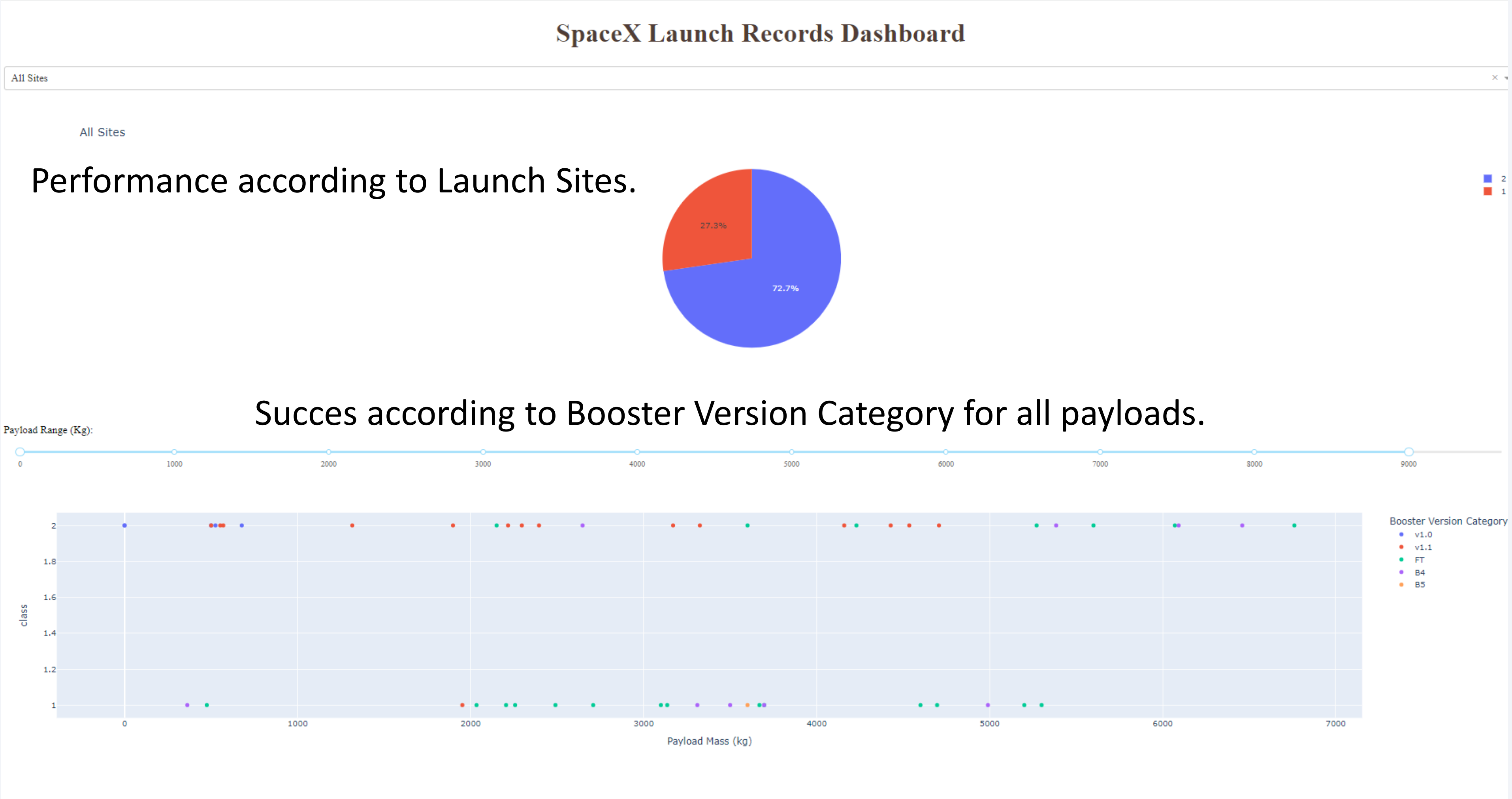




Section 4

# Build a Dashboard with Plotly Dash

# Total Succes Rates Of Launch Sites >



SpaceX Launch Records Dashboard

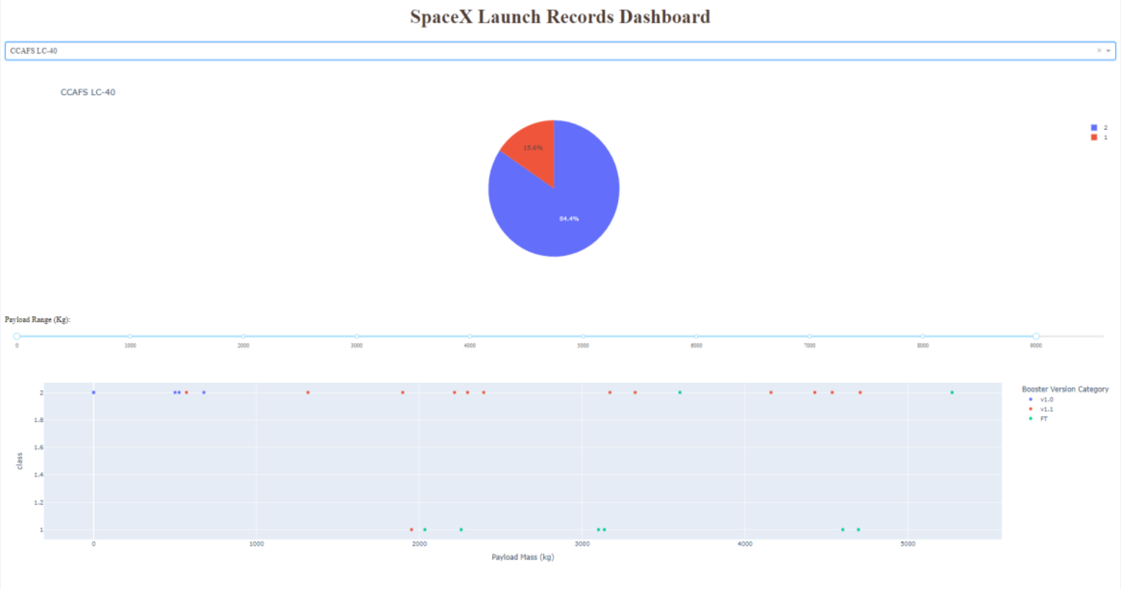
All Sites

Performance according to Launch Sites.

Succes according to Booster Version Category for all payloads.

Payload Range (Kg):

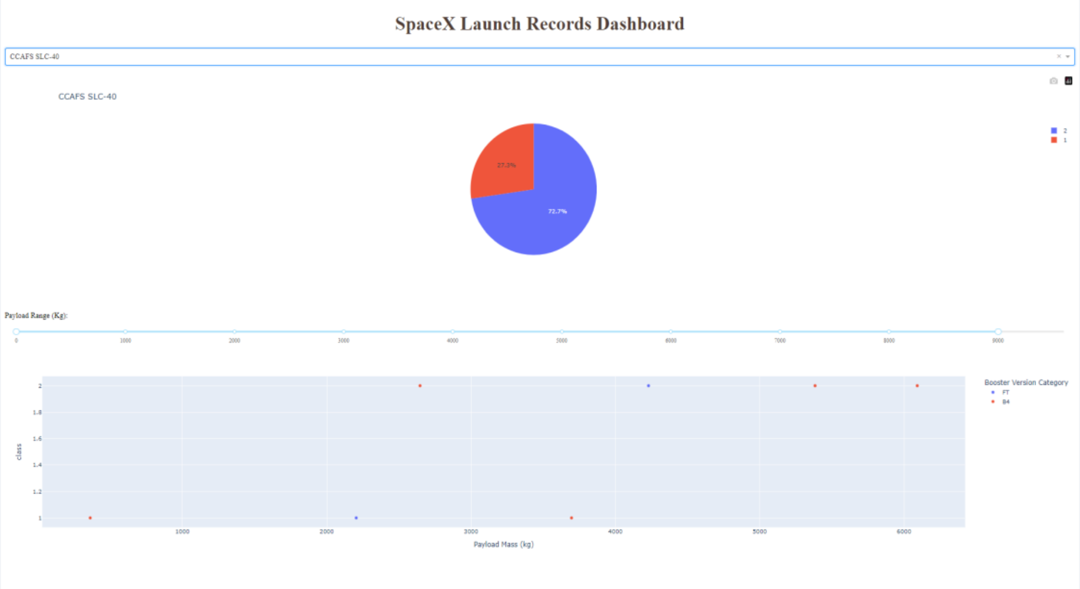
# CCAFS LC-40 & CCAFS SLC-40 Performance



SpaceX Launch Records Dashboard

CCAFS LC-40

CCAFS SLC-40 Launch Site has %27.3 success with the most succesful booster version B4.

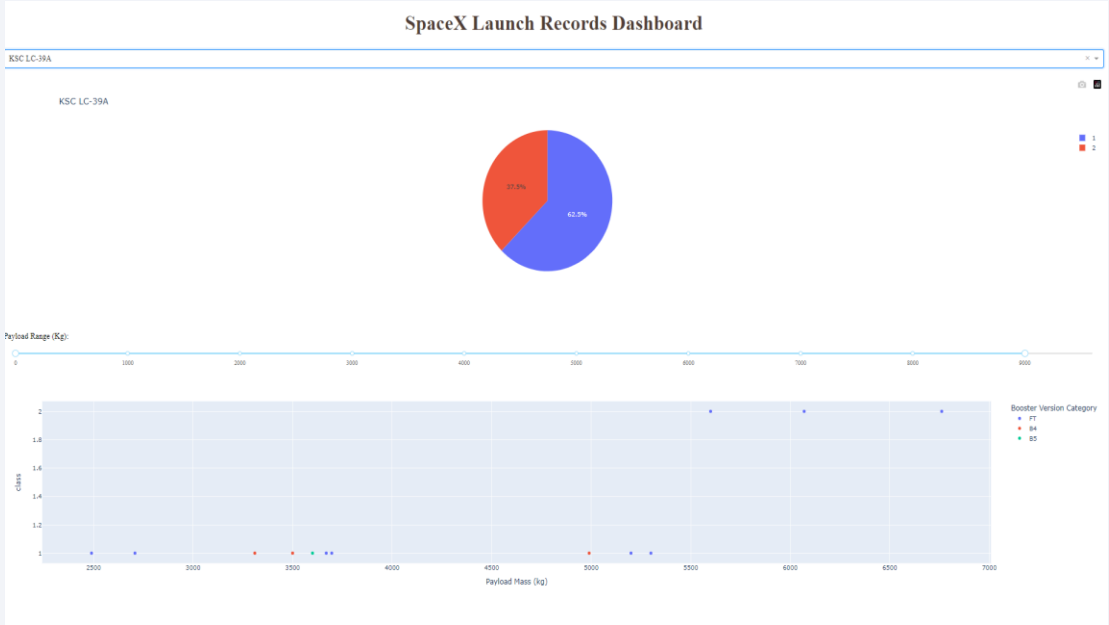


SpaceX Launch Records Dashboard

CCAFS SLC-40

CCAFS LC-40 Launch Site has %15.6 success with the most succesful booster version FT.

# KSC LC 39-A & VAFB SLC-4E Performance



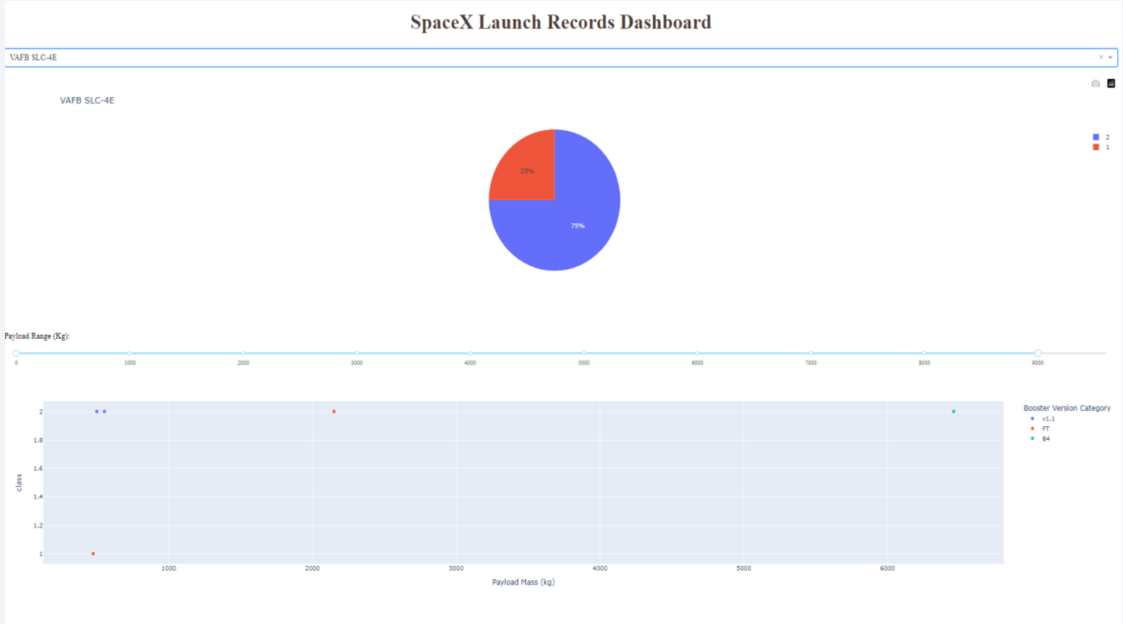
SpaceX Launch Records Dashboard

KSC LC-39A

Payload Range (Kg):

KSC LC 39-A launch site has %37.5 success with the most succesful booster version FT.

VAFB SLC-4E Launch Site has %25.0 success with the most succesful booster version FT.



SpaceX Launch Records Dashboard

VAFB SLC-4E

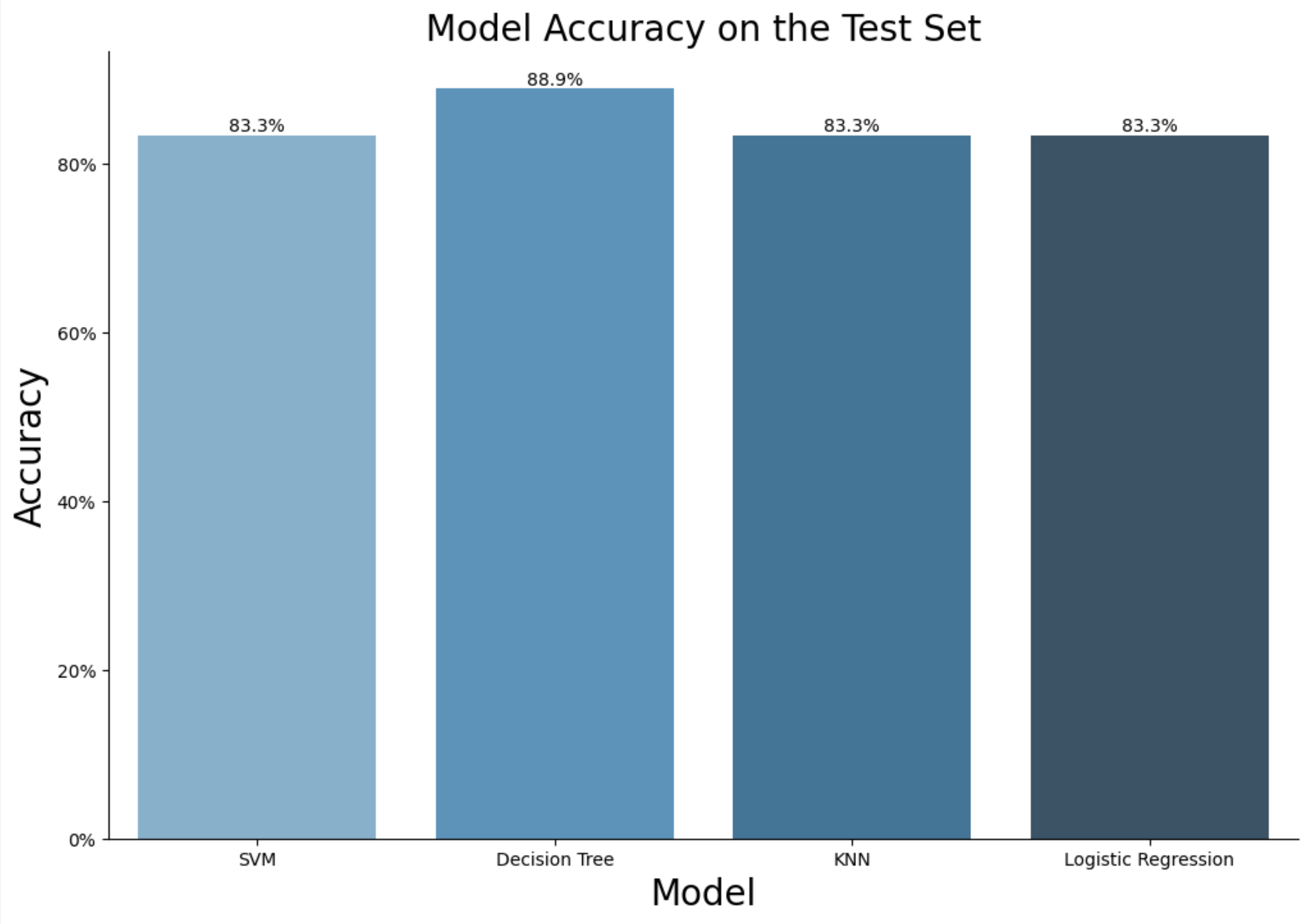
Payload Range (Kg):

Section 5

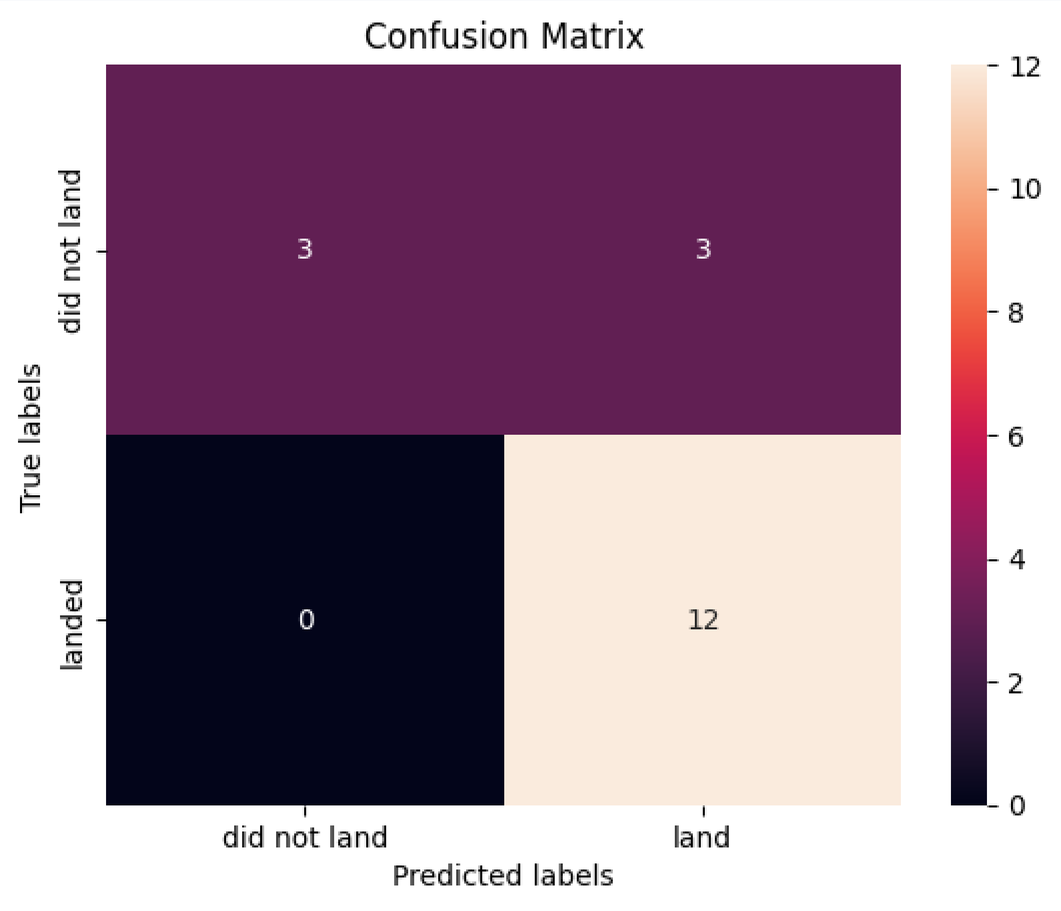
# Predictive Analysis (Classification)

# Classification Accuracy

The Decision Tree Classifier
is the model with higher accuracy



# Confusion Matrix



The confusion matrix for the decision tree classifier tells that the classifier can distinguish between the different classes. But the problem with false positives continues.

# Conclusions

- Increase in  Flight Number of a launch site is resulting with higher success rate of launch site.
- Orbits ES-L1, SSO, HEO, VLEO and GEO are more succesful than the others.
- KSC LC 39-A launch site has the best success rate.
- Launch Success Rate starts to increade after the year 2013.
- The Decision Tree is the best classifier model amongs the others.
- Launch Sites distances to cities are enough to prevent any danger due to accident.

# Appendix

- Watson Studio and Jupiter Notebooks located on local computer were used simultaneously to over come connection and monthly usage limitations of Watson Studio.
- All data downloaded to my computer and studied with it on local computer.

# Thank you!