IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

İlkin Berker Topuz
07.12.2021

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

# Introduction

- This project is conducted whether the first stage of Falcon 9 rocket will land successfully or not. SpaceX advertises Falcon 9 rocket saves approximately 100 million dollars because of the reusability of the first stage.

- To save the first stage of Falcon 9, significant variables are analyzed and determined.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - One hot encoding of categorical variables, and dropping irrelevant data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardization of variables, hyperparameter tuning with grid search cross validation, accuracy scores of different algorithms & models, and confusion matrix

# Data Collection

- Using the SpaceX Rest API, the data is obtained in json format, then relevant information such as date, booster versions, orbit, launch site, payload mass is collected as a data frame. Missing values of the payload masses are replaced with mean.
- Other important information, for example, booster landing is gained from web scrapping of the SpaceX Wikipedia page using beautifulsoap4.

# Data Collection – SpaceX API

- SpaceX API URL
- Get Request
- Data as Json
- Json_normalize to get DataFrame
- Data Cleaning
- Filtering Falcon 9
- Writing CSV

GitHub URL of the notebook

# Data Collection – Scraping

Get Request

BeautifulSoup to Parse HTML

Finding Tablle Elements

Getting Column Names

Appending the Data to Dictionary

Converting Dictionary to DataFrame

Writing CSV

GitHub URL of the notebook

# EDA with Data Visualization

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
    - Success rate is visualized for 3 different launch site with flight numbers
- Payload vs Launch Site
- Orbit vs Flight Number
    - Different orbits have various success rate, some of them improves with flight numbers
- Payload vs Orbit
- Orbit vs Payload Mass
- GitHub URL of the notebook

# 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 'CCA'
- 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 where the successful landing outcome in drone ship was achieved.
- GitHub URL of the notebook

# EDA with SQL

- 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 site for the months in year 2017
- Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- GitHub URL of the notebook

# Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map. Using the Latitude and Longitude Coordinates at each launch site and a Circle Marker around each launch site with a label of the name of the launch site added
- Launch outcomes are color coded to classes 0 and 1 with Green and Red markers on the map using MarkerCluster()
- Lines are drawn on the map to measure distance to landmarks.
- GitHub URL of the notebook

# Build a Dashboard with Plotly Dash

- Pie Chart showing the total launches by a certain site or all sites
- Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions
- GitHub URL to the Python file

# Predictive Analysis (Classification)

- **Building The Model**
  - Reading the data from a csv file and transforming the data with Standard Scaler
  - Splitting the data into training and test data sets
  - Various machine learning algorithms are determined
  - Setting the parameters that are going to be tried in GridSearchCV
  - Fitting training set using the GridSearchCV
- **Evaluating The Model**
  - Checking accuracy for each model
  - Getting tuned hyperparameters for each type of algorithms
  - Plotting Confusion Matrix
- **Finding The Best Performing Classification Model**
  - The model with the best accuracy score is chosen as the best performing model
  - GitHub URL of the notebook

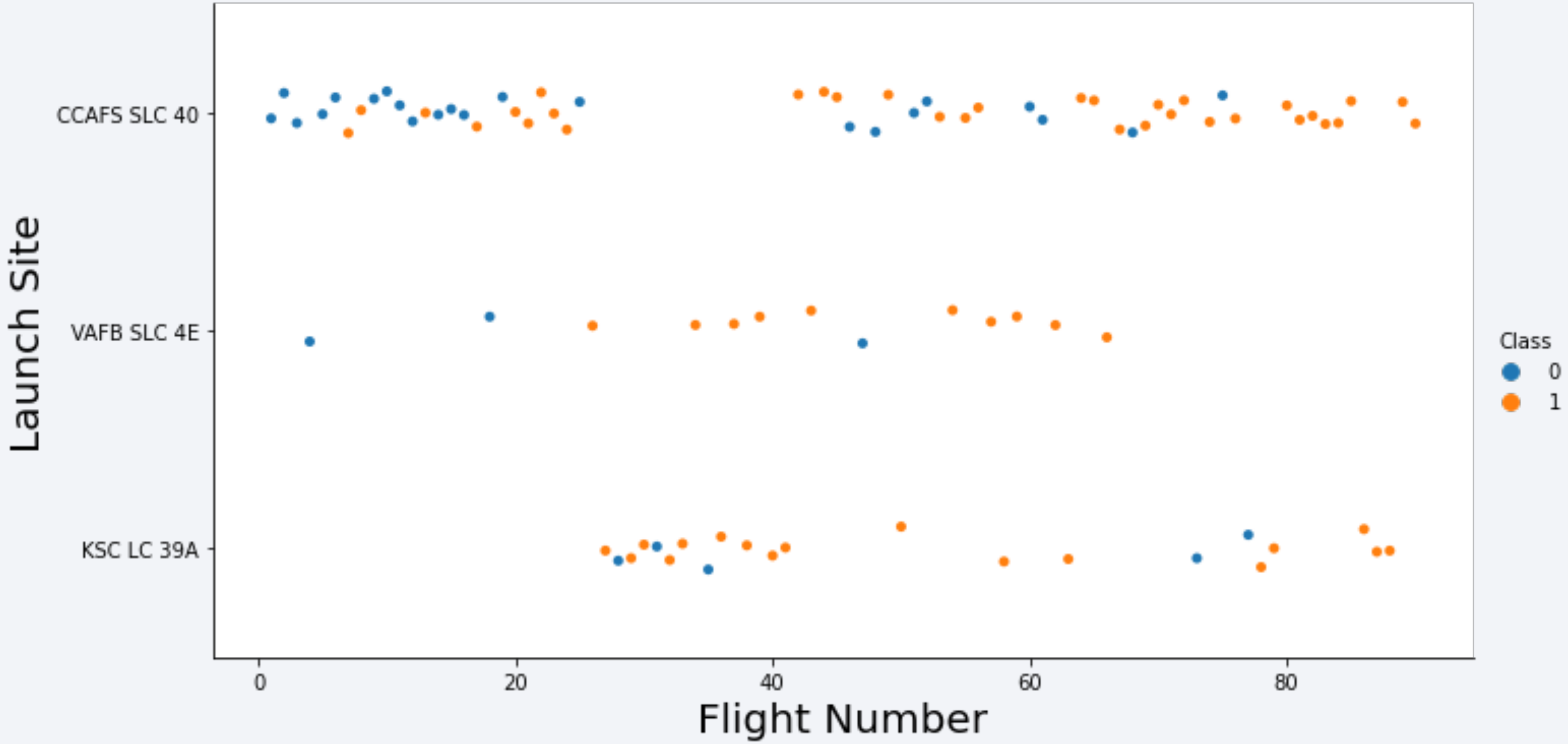
# Results

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

Section 2

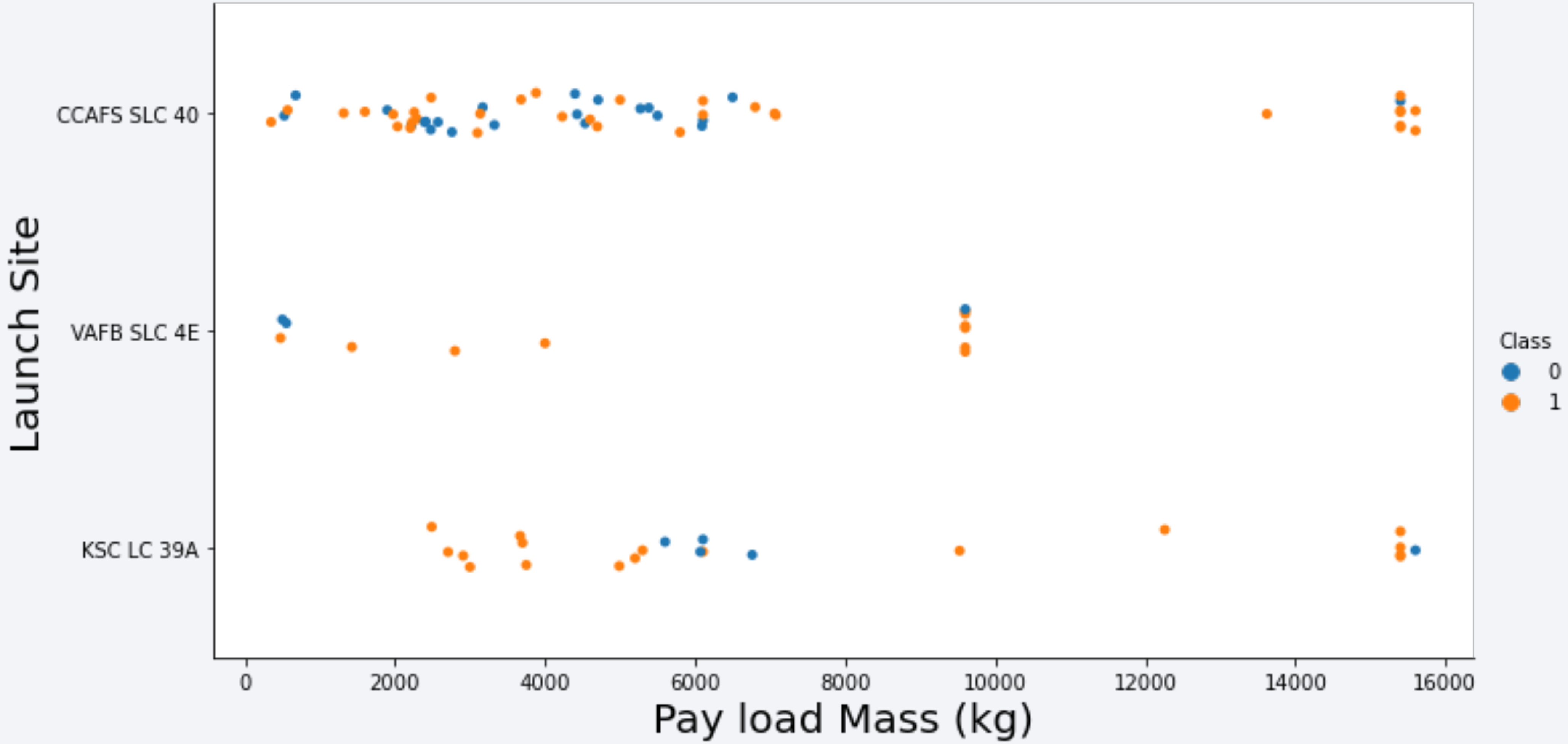
# Insights drawn from EDA

# Flight Number vs. Launch Site



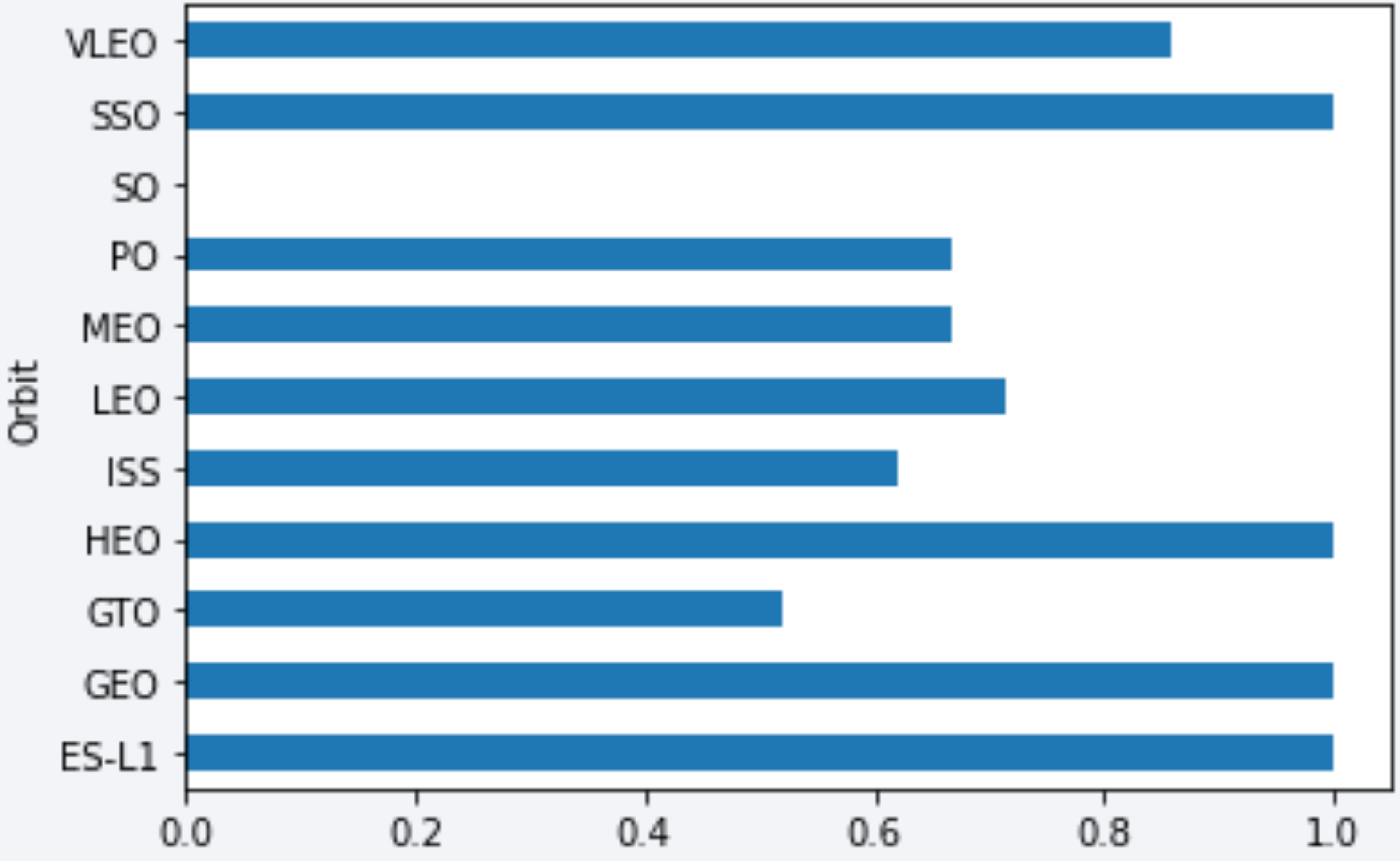

- With the increasing flight numbers, success rate for respective launch site increases.

# Payload vs. Launch Site



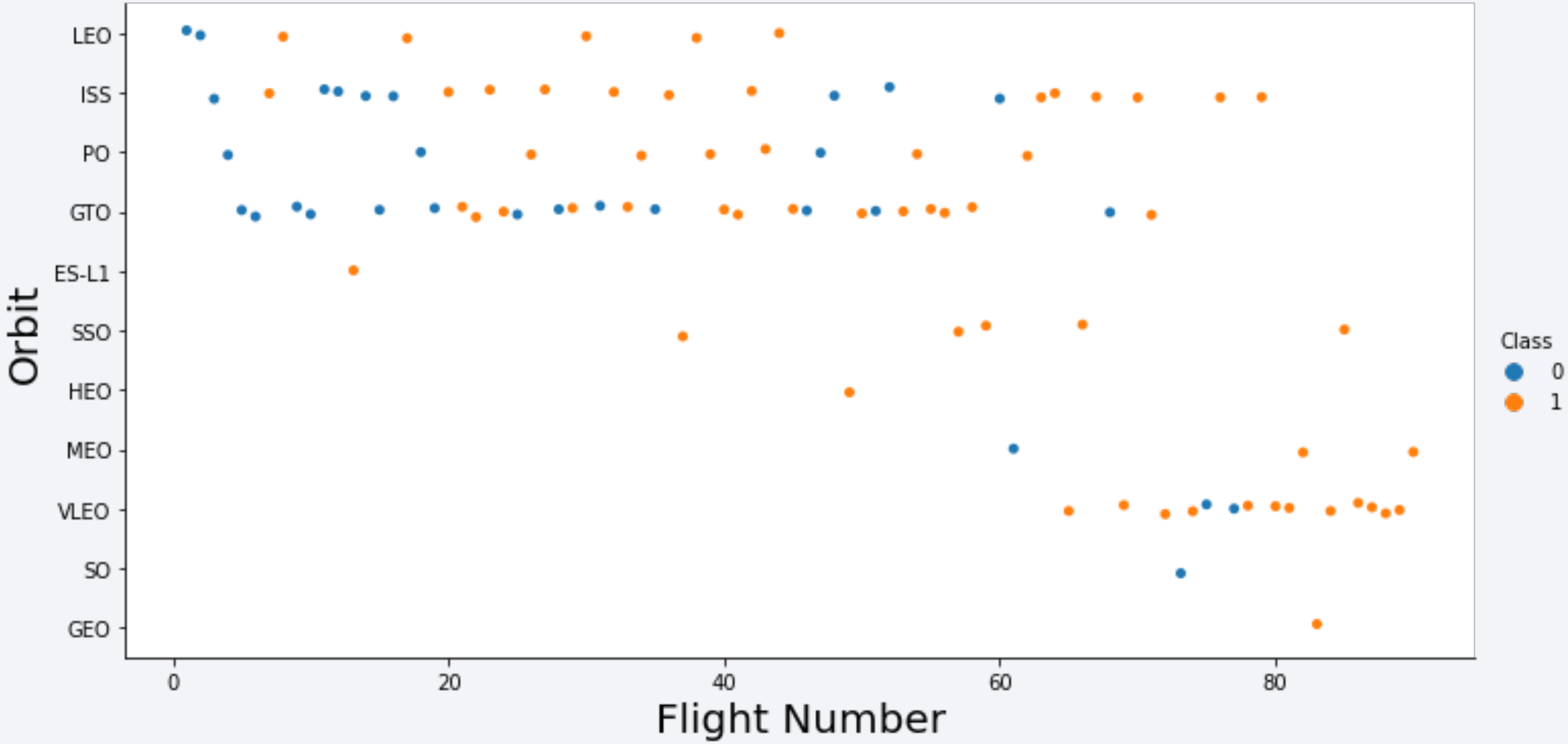

- With the increasing payload mass, success rate for respective launch site increases.

# Success Rate vs. Orbit Type



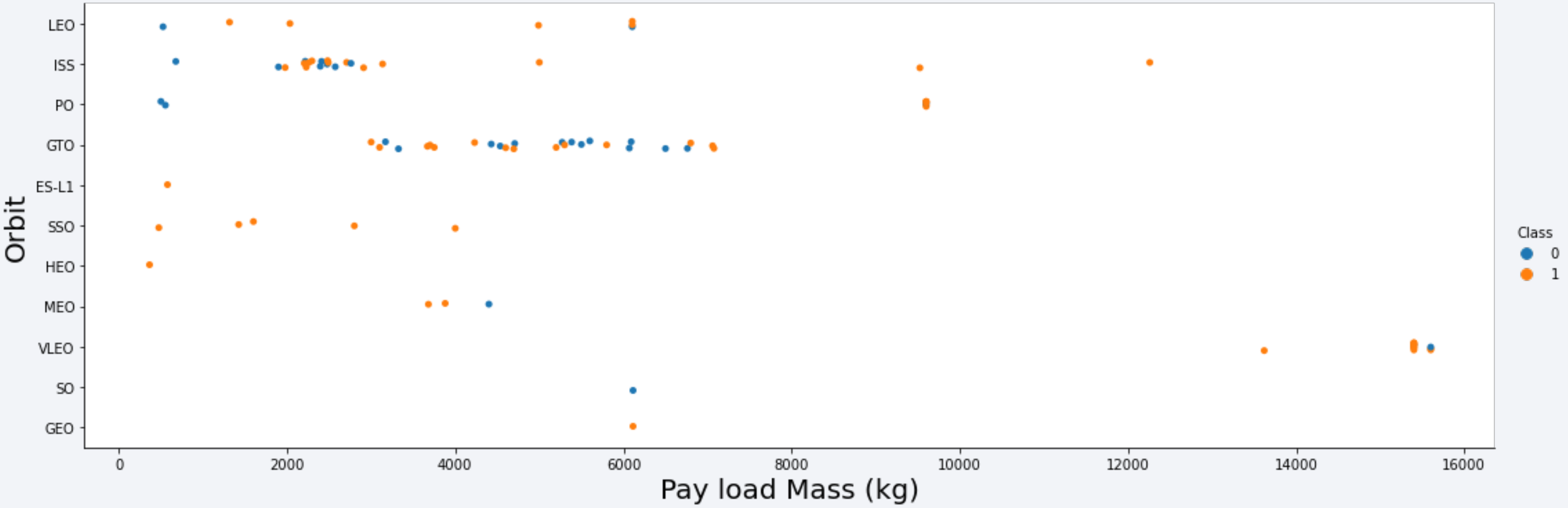

- Different orbits have various success rates. GTO has the lowest success rate.

# Flight Number vs. Orbit Type



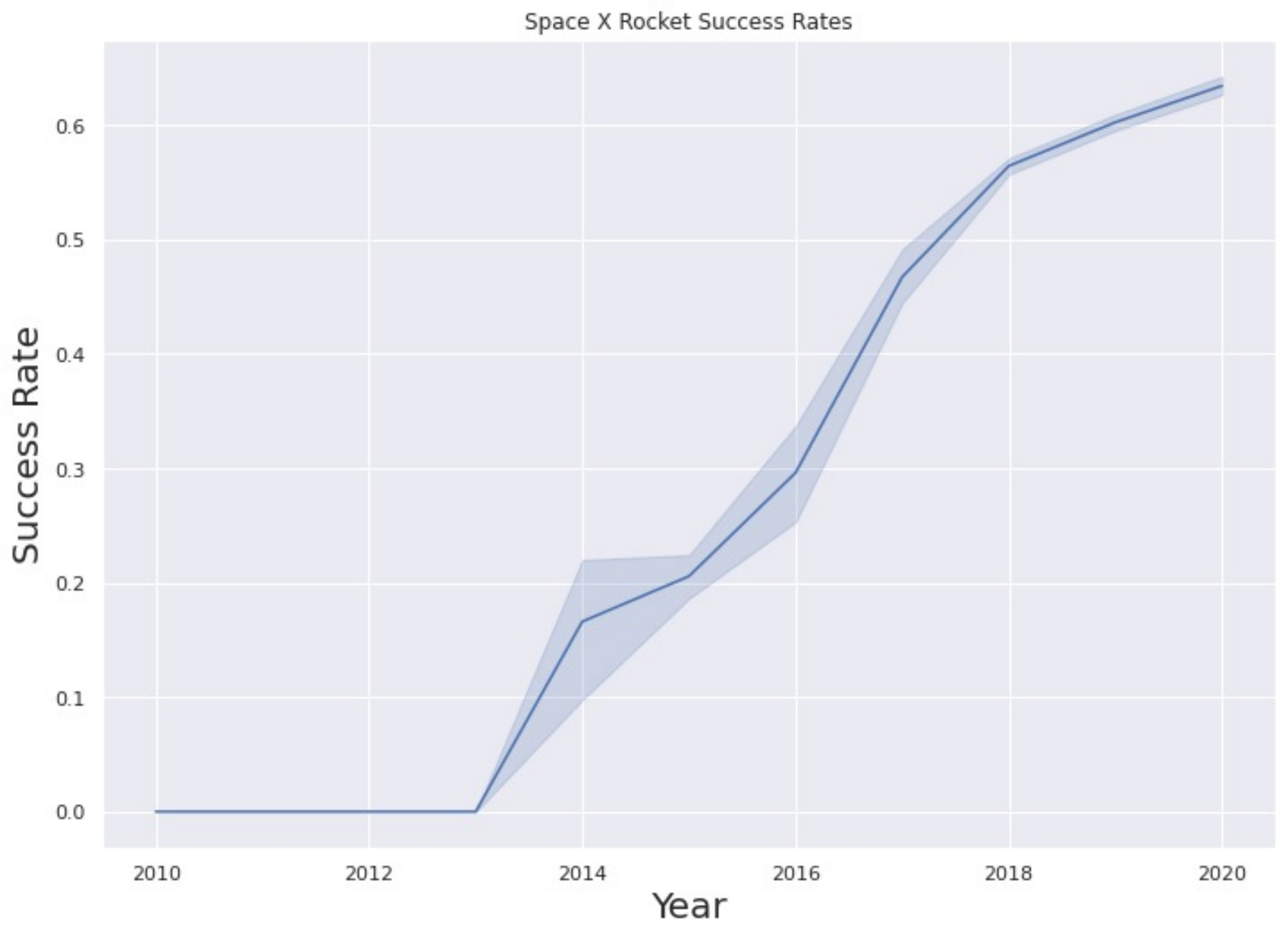

- LEO orbit the success appears related to the number of flights.

# Payload vs. Orbit Type




- Success rate of GTO orbit appears negatively related to pay load mass.

# Launch Success Yearly Trend




- Success rate increases with years.

# All Launch Site Names

## Task 1

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

In [7]:

```
%sql select distinct launch_site from SPACEXTBL
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e
Done.
```

Out[7]:

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

# Launch Site Names Begin with 'CCA'

## Task 2

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

In [9]:
```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[9]:

| 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

## Task 3

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

In [19]:
```
%%sql
select * from spacextbl
where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[19]:

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 2014-04-18 | 19:25:00 | F9 v1.1 | CCAFS LC-40 | SpaceX CRS-3 | 2296 | LEO (ISS) | NASA (CRS) | Success | Controlled (ocean) |
| 2014-09-21 | 05:52:00 | F9 v1.1 B1010 | CCAFS LC-40 | SpaceX CRS-4 | 2216 | LEO (ISS) | NASA (CRS) | Success | Uncontrolled (ocean) |
| 2015-01-10 | 09:47:00 | F9 v1.1 B1012 | CCAFS LC-40 | SpaceX CRS-5 | 2395 | LEO (ISS) | NASA (CRS) | Success | Failure (drone ship) |
| 2015-04-14 | 20:10:00 | F9 v1.1 B1015 | CCAFS LC-40 | SpaceX CRS-6 | 1898 | LEO (ISS) | NASA (CRS) | Success | Failure (drone ship) |
| 2015-06-28 | 14:21:00 | F9 v1.1 B1018 | CCAFS LC-40 | SpaceX CRS-7 | 1952 | LEO (ISS) | NASA (CRS) | Failure (in flight) | Precluded (drone ship) |
| 2016-04-08 | 20:43:00 | F9 FT B1021.1 | CCAFS LC-40 | SpaceX CRS-8 | 3136 | LEO (ISS) | NASA (CRS) | Success | Success (drone ship) |
| 2016-07-18 | 04:45:00 | F9 FT B1025.1 | CCAFS LC-40 | SpaceX CRS-9 | 2257 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-02-19 | 14:39:00 | F9 FT B1031.1 | KSC LC-39A | SpaceX CRS-10 | 2490 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-06-03 | 21:07:00 | F9 FT B1035.1 | KSC LC-39A | SpaceX CRS-11 | 2708 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-08-14 | 16:31:00 | F9 B4 B1039.1 | KSC LC-39A | SpaceX CRS-12 | 3310 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| 2017-12-15 | 15:36:00 | F9 FT B1035.2 | CCAFS SLC-40 | SpaceX CRS-13 | 2205 | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |

# Average Payload Mass by F9 v1.1

## Task 4

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

```
%%sql
select avg(payload_mass__kg_) as average_payload_mass
from spacextbl
where booster_version like '%F9 v1.1%'
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b(
Done.
```

| average_payload_mass |
|---|
| 2534 |

# First Successful Ground Landing Date

## Task 5

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

*Hint:Use min function*

```
%%sql
select * from spacextbl where date =
(select min(date) as min_date from spacextbl
where landing__outcome = 'Success (ground pad)')
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2015-12-22 | 01:29:00 | F9 FT B1019 | CCAFS LC-40 | OG2 Mission 2 11 Orbcomm-OG2 satellites | 2034 | LEO | Orbcomm | Success | Success (ground pad) |

# Successful Drone Ship Landing with Payload between 4000 and 6000

## Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
select * from spacextbl
where payload_mass__kg_ >= 4000
and payload_mass__kg_ <= 6000
and landing__outcome = 'Success (drone ship)'
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2016-05-06 | 05:21:00 | F9 FT B1022 | CCAFS LC-40 | JCSAT-14 | 4696 | GTO | SKY Perfect JSAT Group | Success | Success (drone ship) |
| 2016-08-14 | 05:26:00 | F9 FT B1026 | CCAFS LC-40 | JCSAT-16 | 4600 | GTO | SKY Perfect JSAT Group | Success | Success (drone ship) |
| 2017-03-30 | 22:27:00 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO | SES | Success | Success (drone ship) |
| 2017-10-11 | 22:53:00 | F9 FT B1031.2 | KSC LC-39A | SES-11 / EchoStar 105 | 5200 | GTO | SES EchoStar | Success | Success (drone ship) |

# Total Number of Successful and Failure Mission Outcomes

```
%%sql
select landing_outcome, count(*) as cnt
from (
select a.*, case when landing__outcome like '%Success%' then 'Success'
            when landing__outcome like '%Failure%' then 'Failure'
    end as landing_outcome
from spacextbl a)
where landing_outcome is not null
group by landing_outcome
order by cnt desc
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kq
Done.
```

| landing_outcome | cnt |
|---|---|
| Success | 61 |
| Failure | 10 |

# Boosters Carried Maximum Payload

## Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
select distinct booster_version, payload_mass__kg_
from spacextbl
where payload_mass__kg_ = (
    select max(payload_mass__kg_) as max_payload_mass
    from spacextbl)
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.datab
Done.
```

| booster_version | payload_mass__kg_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1049.7 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1060.3 | 15600 |

# 2015 Launch Records

## Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select date, booster_version, launch_site, landing__outcome
from spacextbl
where landing__outcome = 'Failure (drone ship)'
and substr(date,1,4) = '2015'
```

```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.database
Done.
```

| DATE | booster_version | launch_site | landing__outcome |
|---|---|---|---|
| 2015-01-10 | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| 2015-04-14 | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

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

## Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql
select landing__outcome, count(*) as cnt
from spacextbl
where date between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by cnt desc
```

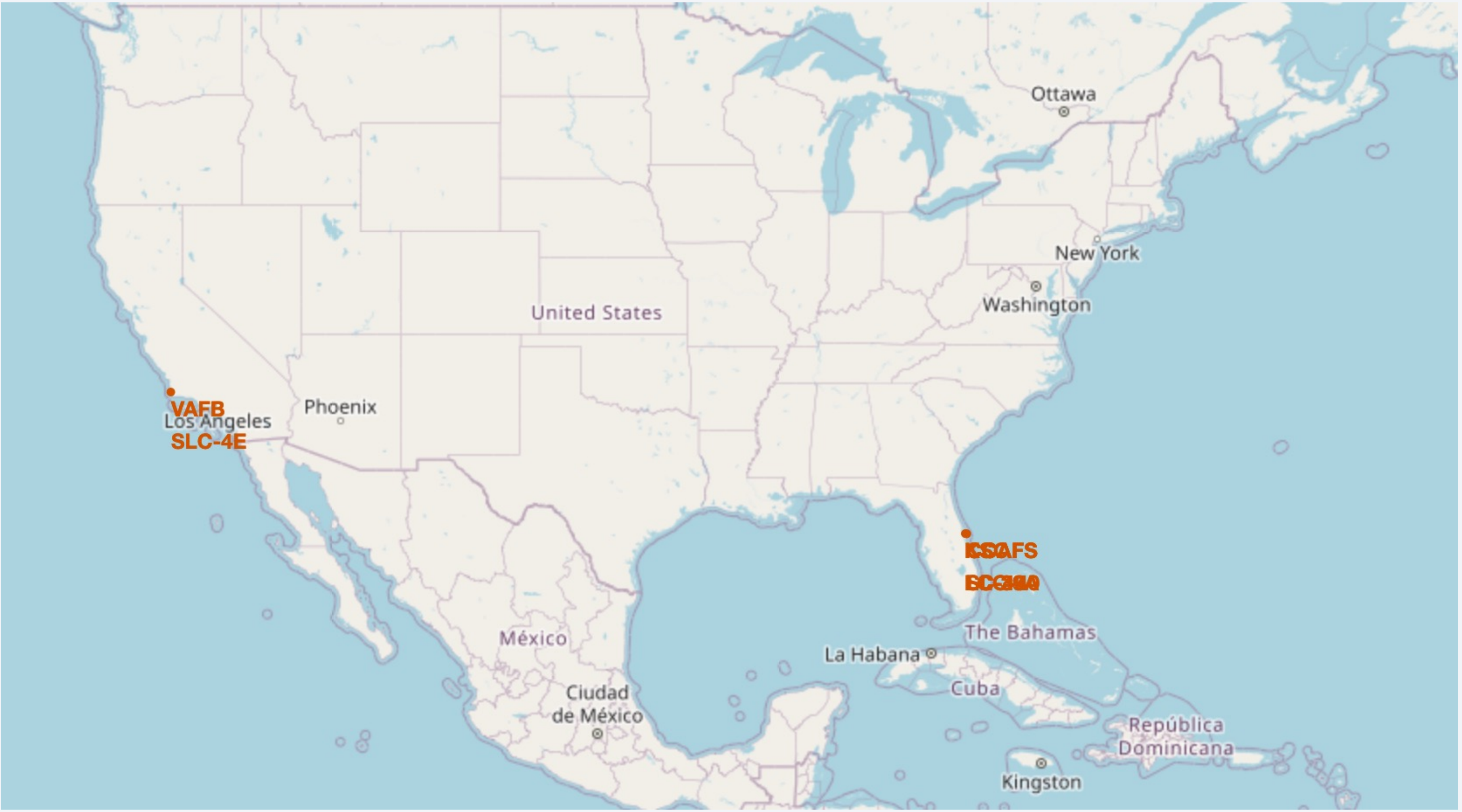
```
 * ibm_db_sa://rgs09380:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

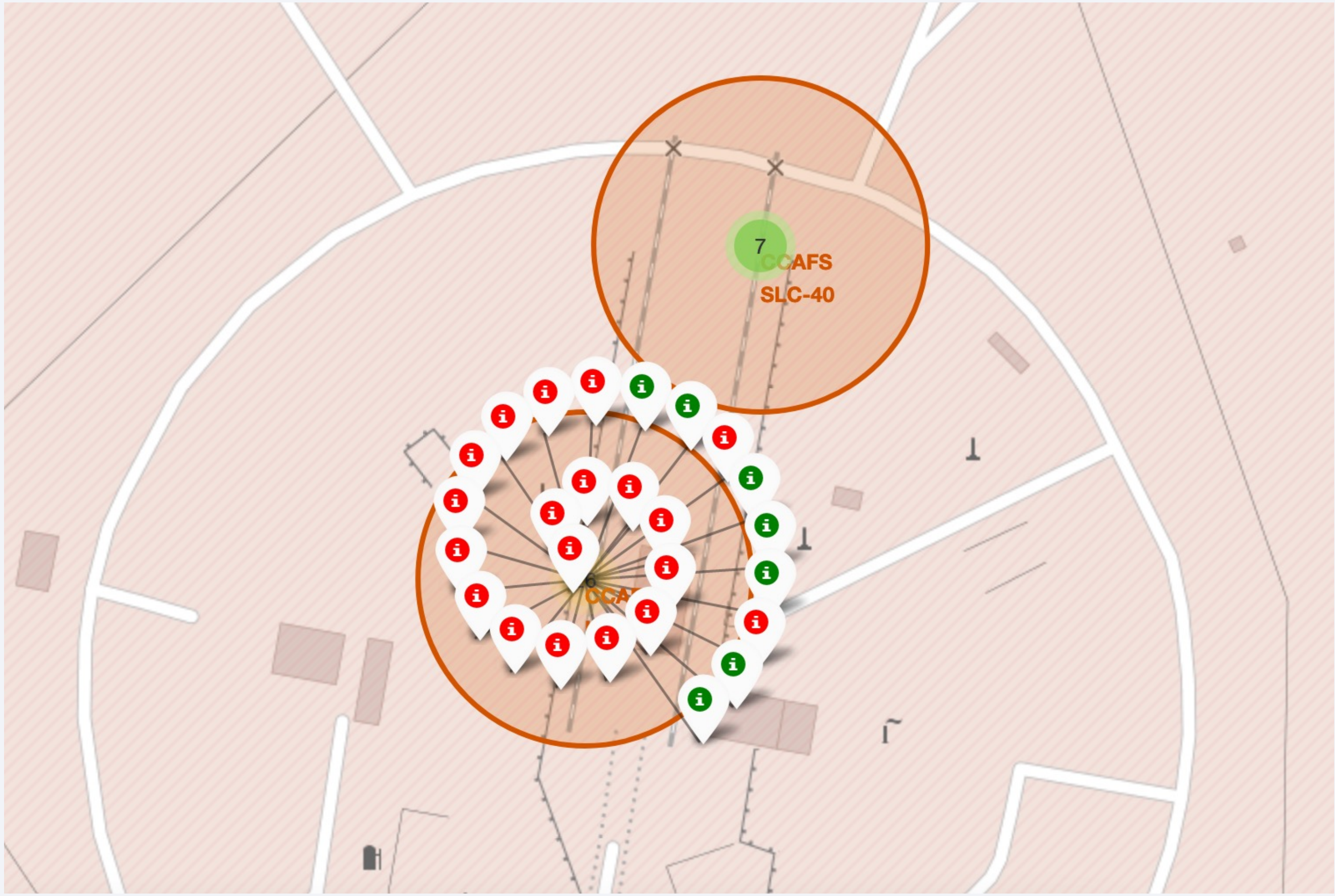
Section 4

# Launch Sites Proximities Analysis

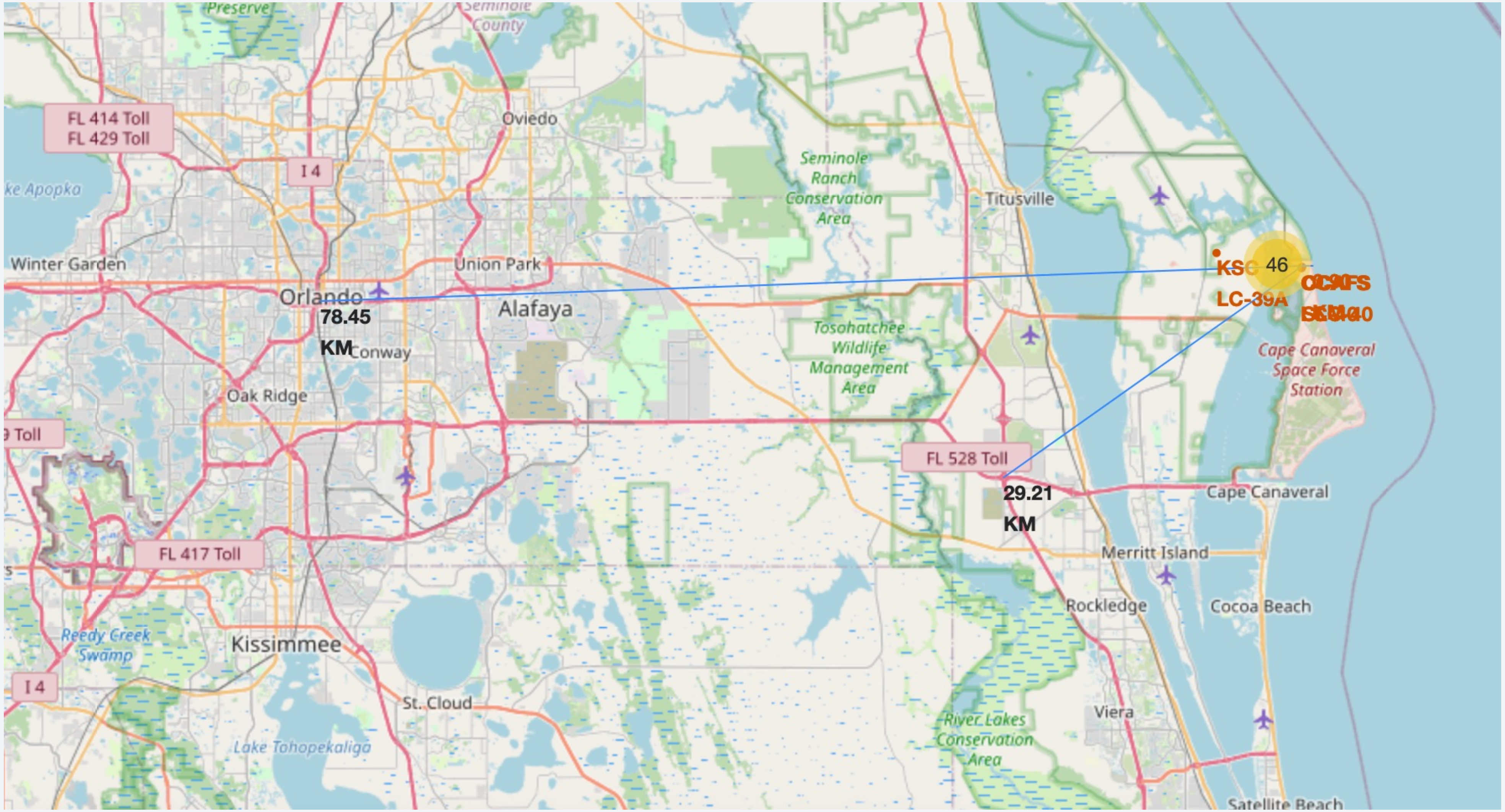
# Launch Sites Map with Markers



# Launch Site Success Status Color Coded



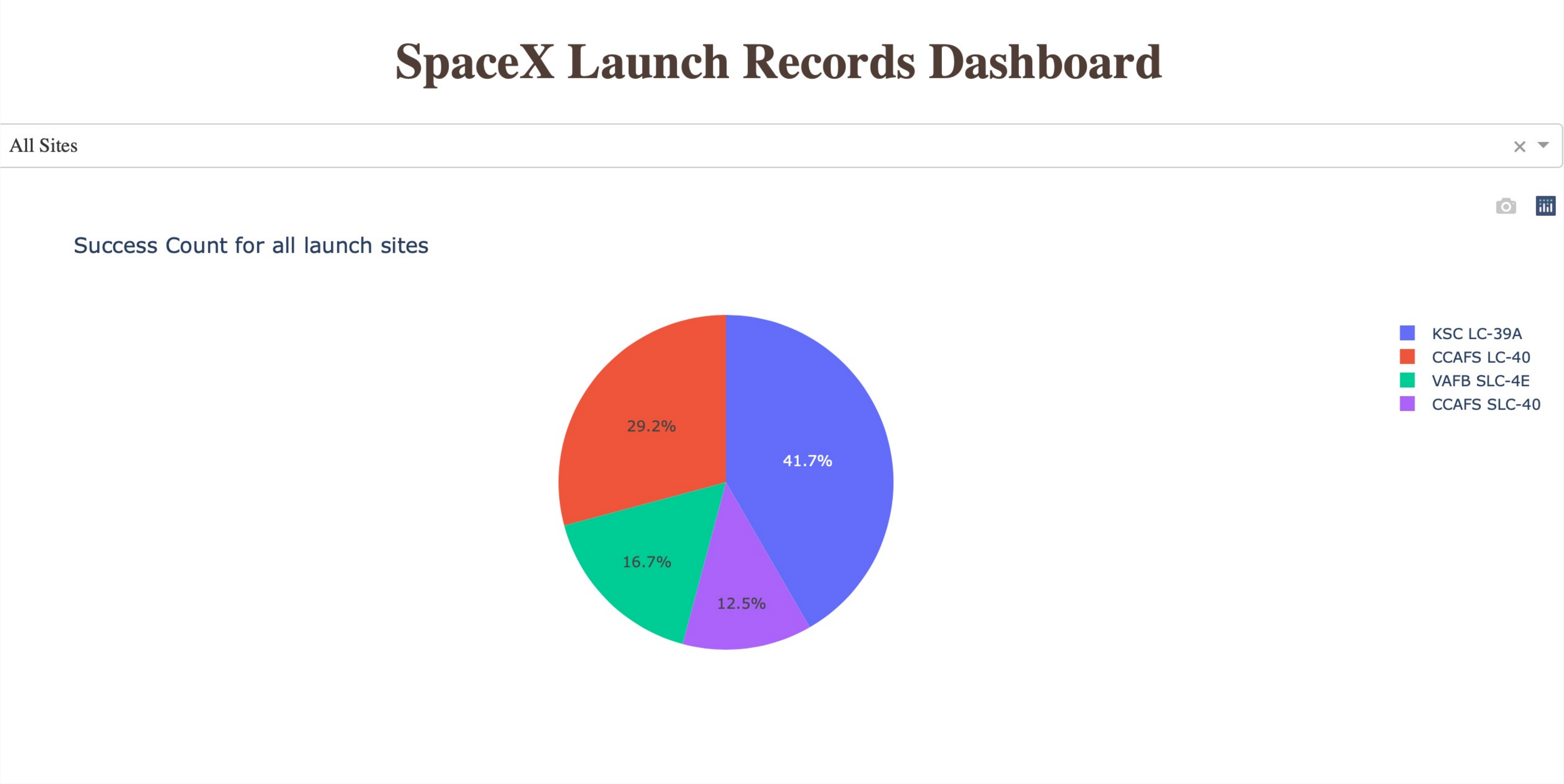
# Launch Site Distances to Highway & Railway Station



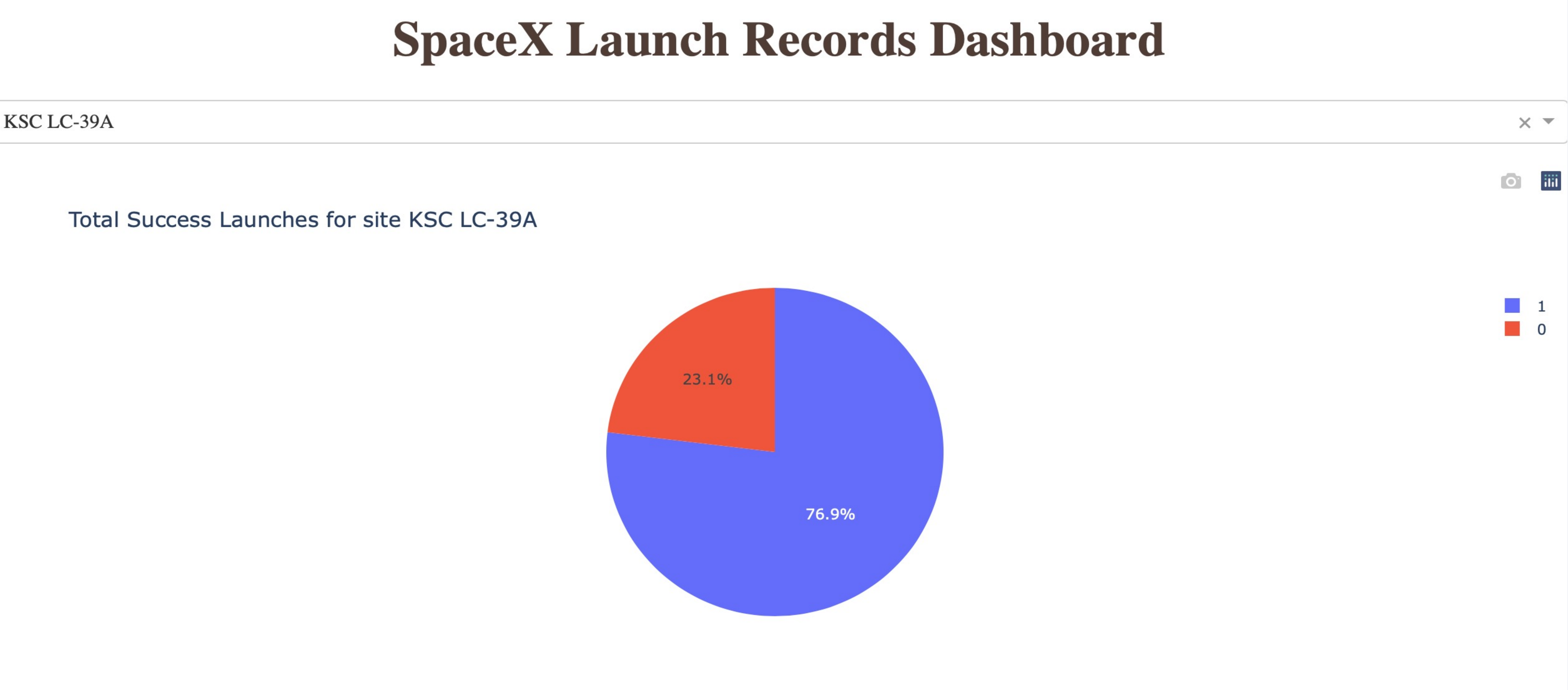
Section 5

# Build a Dashboard with Plotly Dash

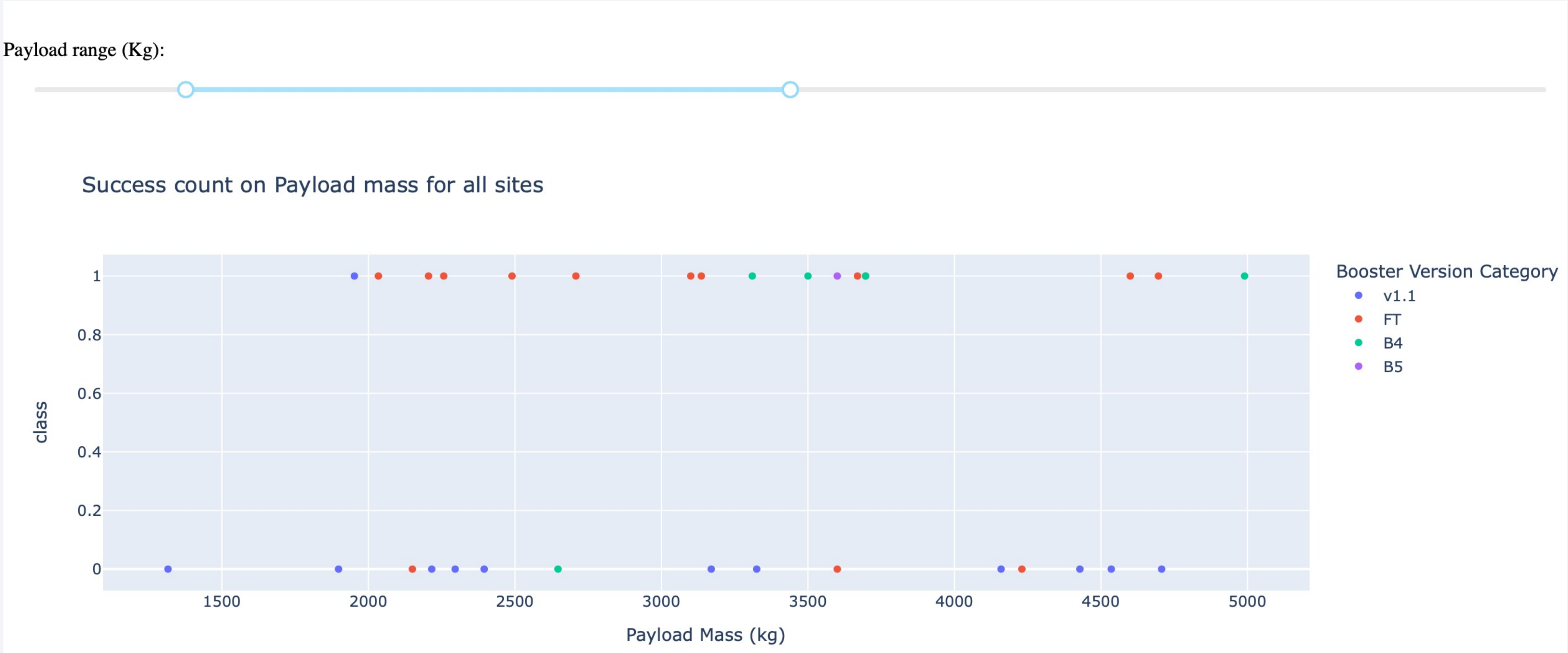
# SpaceX Launch Records Dashboards Pie Chart



# Highest Launch Success Ratio



# Payload vs. Launch Outcome Scatter Plot

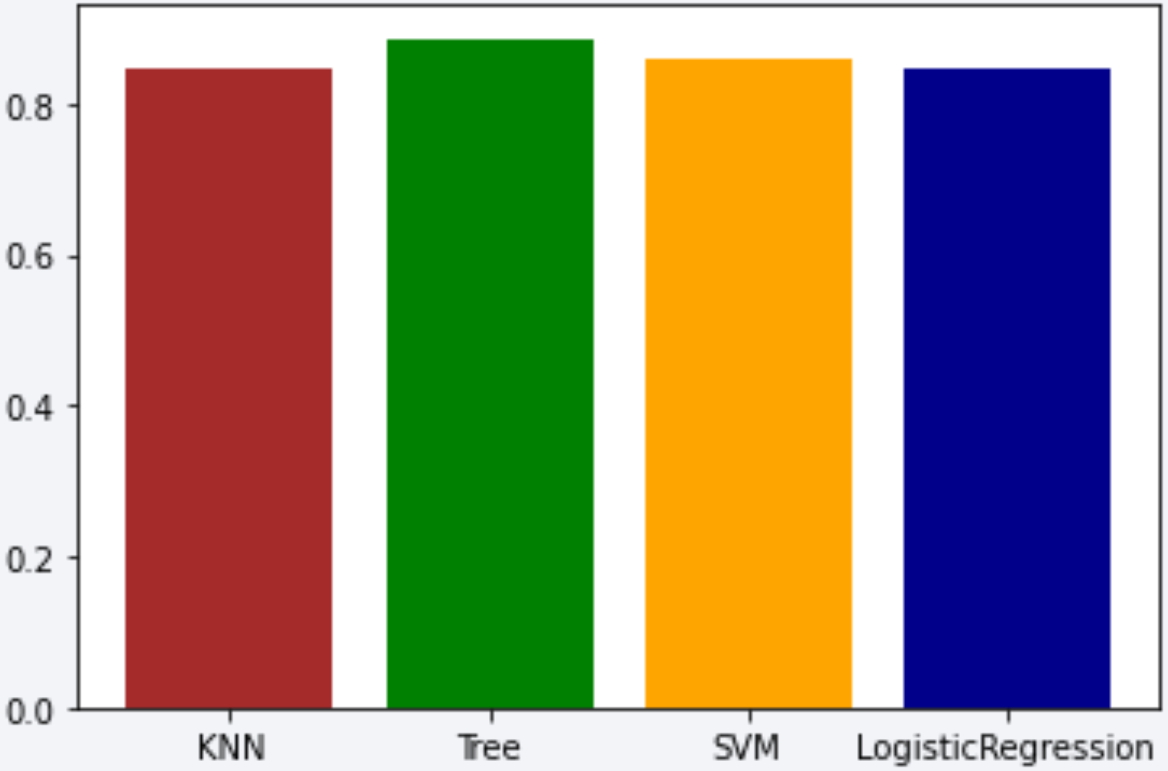


Section 6

# Predictive Analysis (Classification)

# Classification Accuracy

- All models perform close to one another; however, decision tree classifier has the highest accuracy score.



# Confusion Matrix

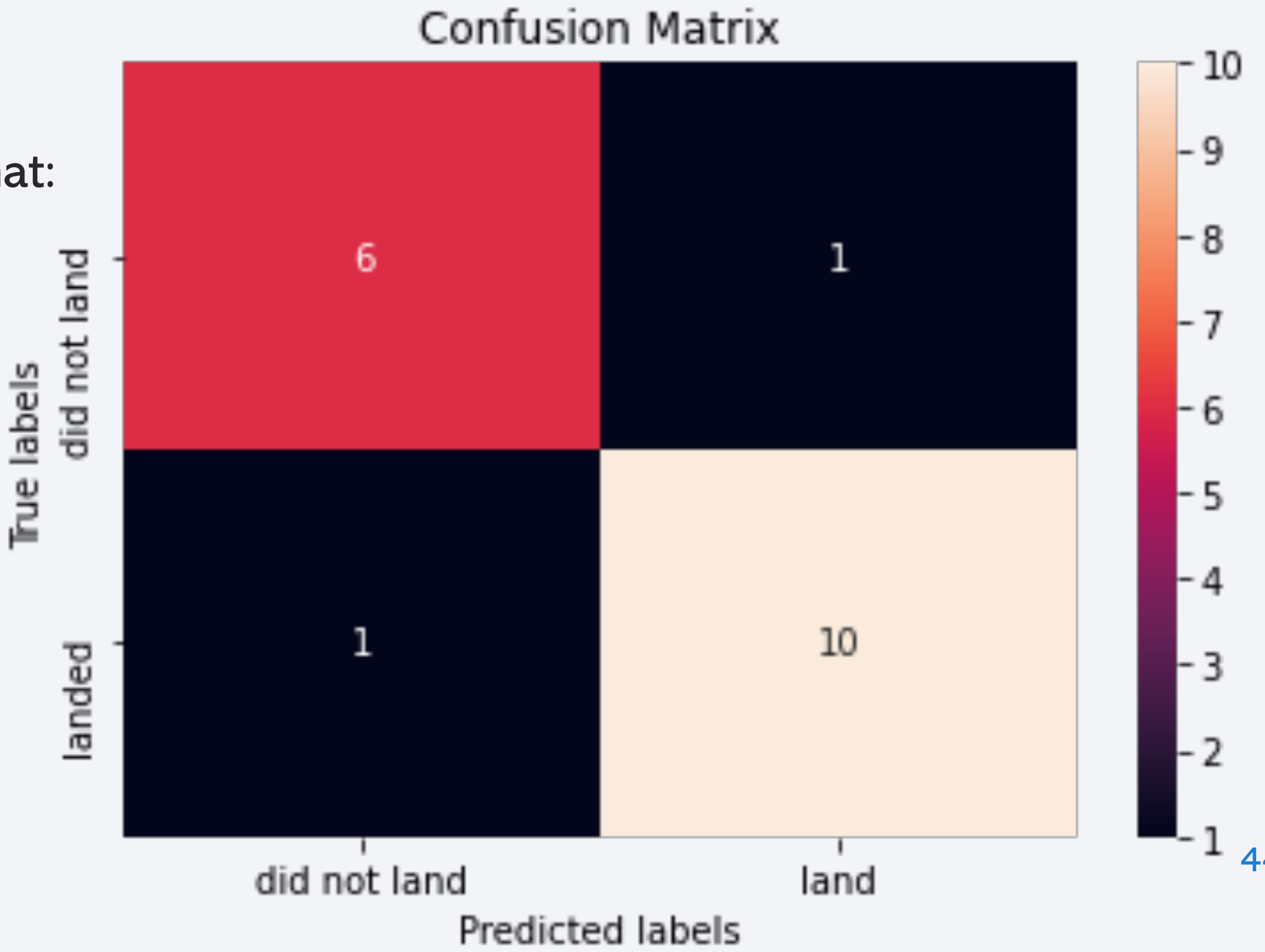
- Confusion matrix shows that:

Accuracy = 83.3%

Recall = 90.9%

Specificity = 71.4%

Precision = 83.3%



# Conclusions

- The Decision Tree Classifier Algorithm is the best performing algorithm for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional with years
- KSC LC-39A had the most successful launches from all the sites with 76.9% success rate
- Orbit GEO, HEO, SSO, ES-L1 has the best success rate which is 100%

# Thank you!