IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chee Chung Lam
21 April 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection via API, Web Scraping
  - Data wrangling
  - Exploratory Data Analysis with SQL, data visualization
  - Interactive visual analytics using Folium
  - Machine learning prediction
- Summary of all results
  - Exploratory Data Analysis
  - Interactive visual analytics
  - Predictive analysis

Introduction

SpaceX is an aerospace company which provides space transportation services and communication. It has the ability to reuse its booster rockets hence able to lower down cost of launching rockets.

The challenge is to predict safe landing of booster rockets

Section 1

Methodology

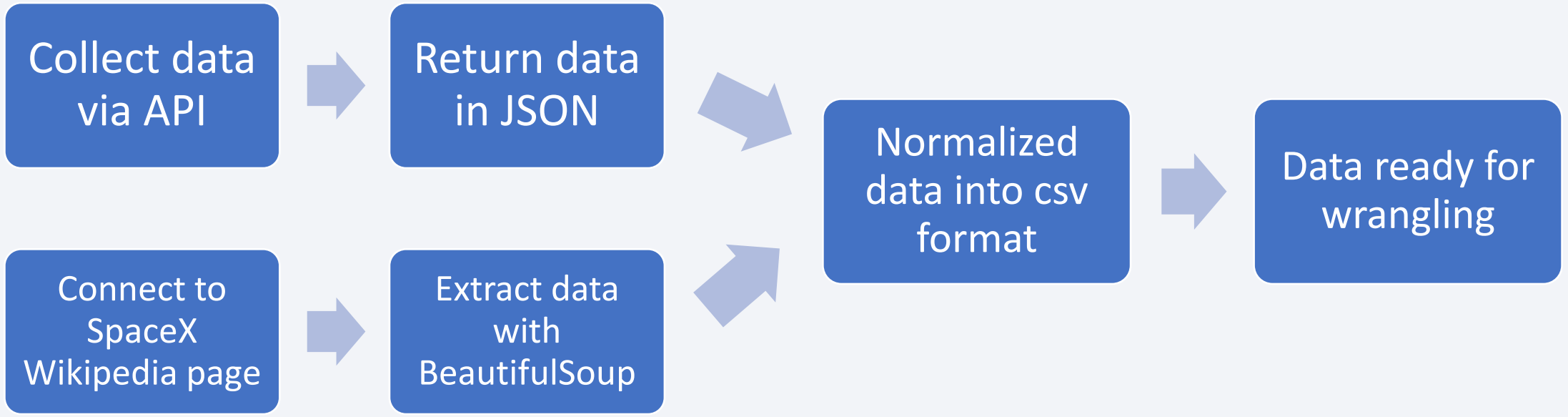
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected via API and web scraping SPACEX Wikipedia page
- Perform data wrangling
  - Data was cleaned by replacing null values, changing data type and recategorized categorical data as integers
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Several classification models (LR, KNN, etc) were used

Data Collection

- Data was collected via API function (SpaceX REST API) which gets launch related data from SpaceX website and web scraping was used via BeautifulSoup to get launch data from SpaceX Wikipedia page



Data Collection – SpaceX API

1. Collect data via API

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

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

2. Convert JSON to data frame

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

3. Save data to csv format

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

https://github.com/CC-70300/SpaceY/blob/master/Data%20Collection%20API.ipynb

Data Collection - Scraping

1. Get HTML response from SpaceX Wiki page

```
# use requests.get() method with the provided static_url

response = requests.get(static_url)

# assign the response to a object
response.status_code
```

2. Create BeautifulSoup object from HTML response

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

3. Save data to csv format

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

https://github.com/CC-70300/SpaceY/blob/master/Web%20Scraping%20lab.ipynb

Data Wrangling

- Data wrangling is a process of cleaning up the source data, modify existing data into more useable form or preprocess them into more meaningful data for analysis

https://github.com/CC-70300/SpaceY/blob/master/Web%20Scraping%20lab.ipynb

EDA with Data Visualization

- Charts used in the data visualization:
  - Scatter plot: to visualize/show relationship between independent & dependent variables
  - Bar chart: show success rate for each types of orbits
  - Line chart: time series chart to present success rate over period of years

https://github.com/CC-70300/SpaceY/blob/master/EDA%20with%20Visualization%20cc.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Display unique launch sites
- Display only 5 launch sites beginning ‘CCA’
- Display total payload mass launch by NASA
- Display average payload carried by F9v1.1
- Find date for 1st successful landing outcome in ground pad
- Show booster names with drone ship success with payload between 4000 & 6000
- Show total number of mission outcome status (success or failure)
- Show booster versions that carried maximum payload mass
- List out mission outcome status, booster versions, launch sites by month for 2015
- List out successful mission outcome in descending order between date 04-06-2010 and 20-03-2017

https://github.com/CC-70300/SpaceY/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20cc(1).ipynb

Build an Interactive Map with Folium

- Map objects used in folium map:
  - Markers: show location from latitude & longitude
  - Circles: show single location
  - Lines: show distances between 2 points
  - Clusters: group several markers together

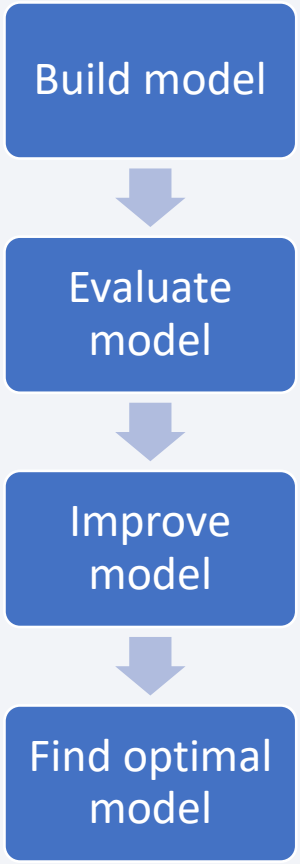
https://github.com/CC-70300/SpaceY/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20cc.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Pie chart: show total launch by sites or success rate of selected site
- Scatter plot: show success rate for selected site by booster version and payload mass. Also allow user to filter by payload mass

https://github.com/CC-70300/SpaceY/blob/master/spacex_dash_app%20cc.py

Predictive Analysis (Classification)

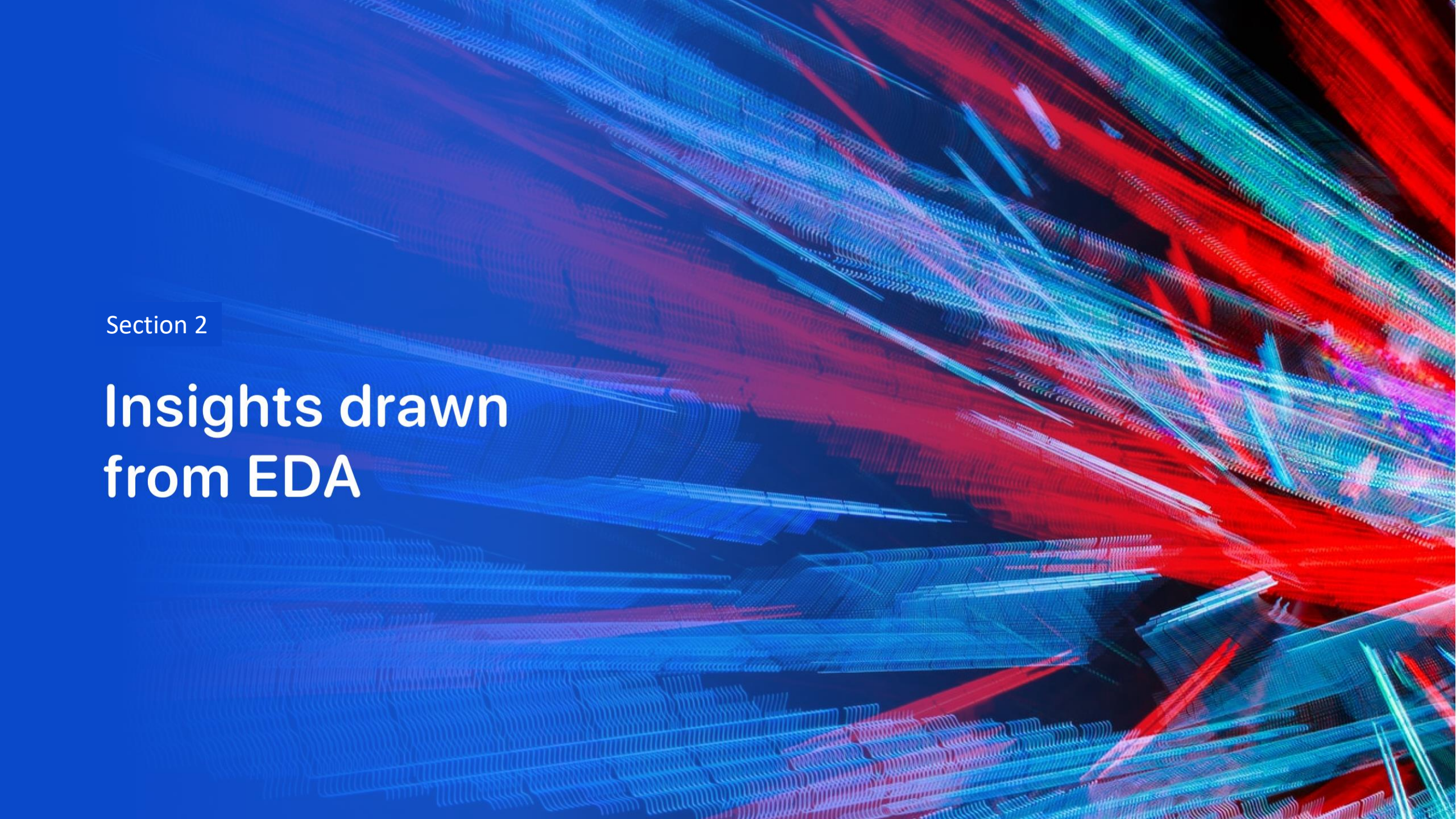


- Load, transform dataset then split to train & test sets; set parameters to GridSearch CV and fit dataset
- Check accuracy & confusion matrix of each model
- Fine tune hyperparameters for each algorithms
- Compare all the results & select the model with best accuracy

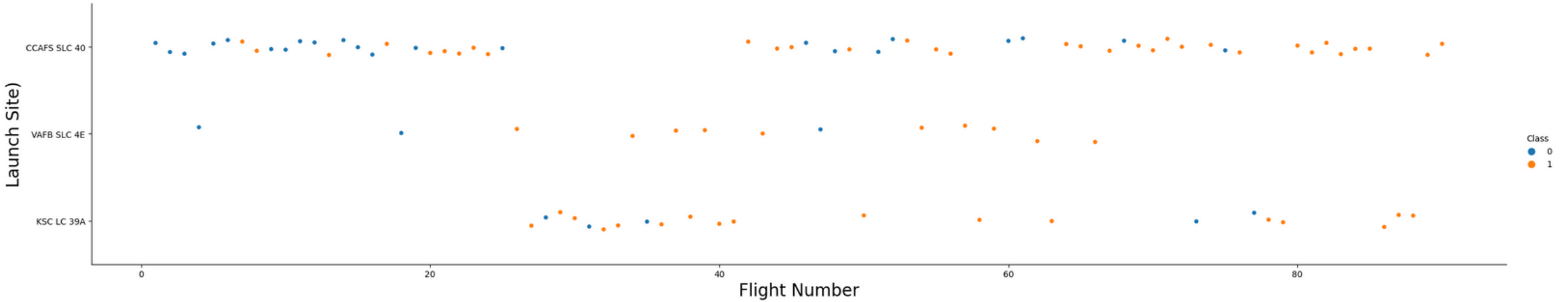
https://github.com/CC-70300/SpaceY/blob/master/Machine%20Learning%20Prediction%20lab%20cc.ipynb

Results

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

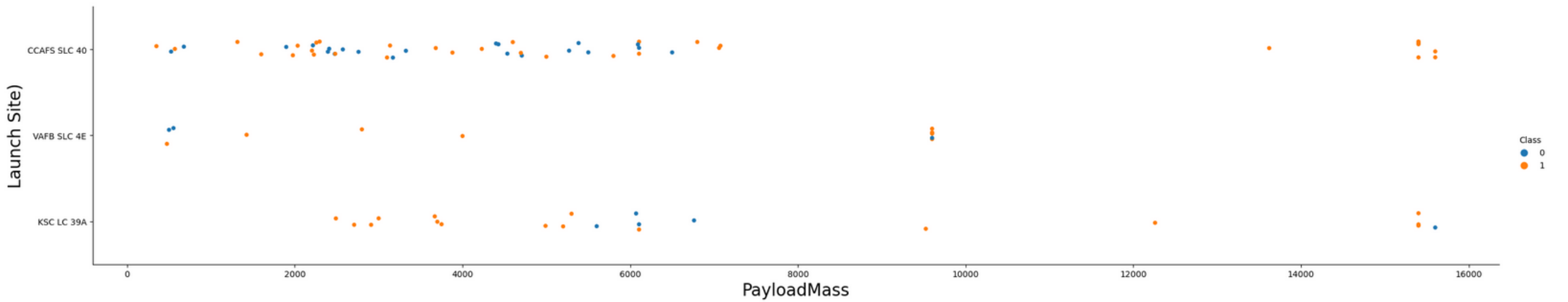

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- Higher flight numbers from CCAFS LC-40 has more success rate than lower flight numbers

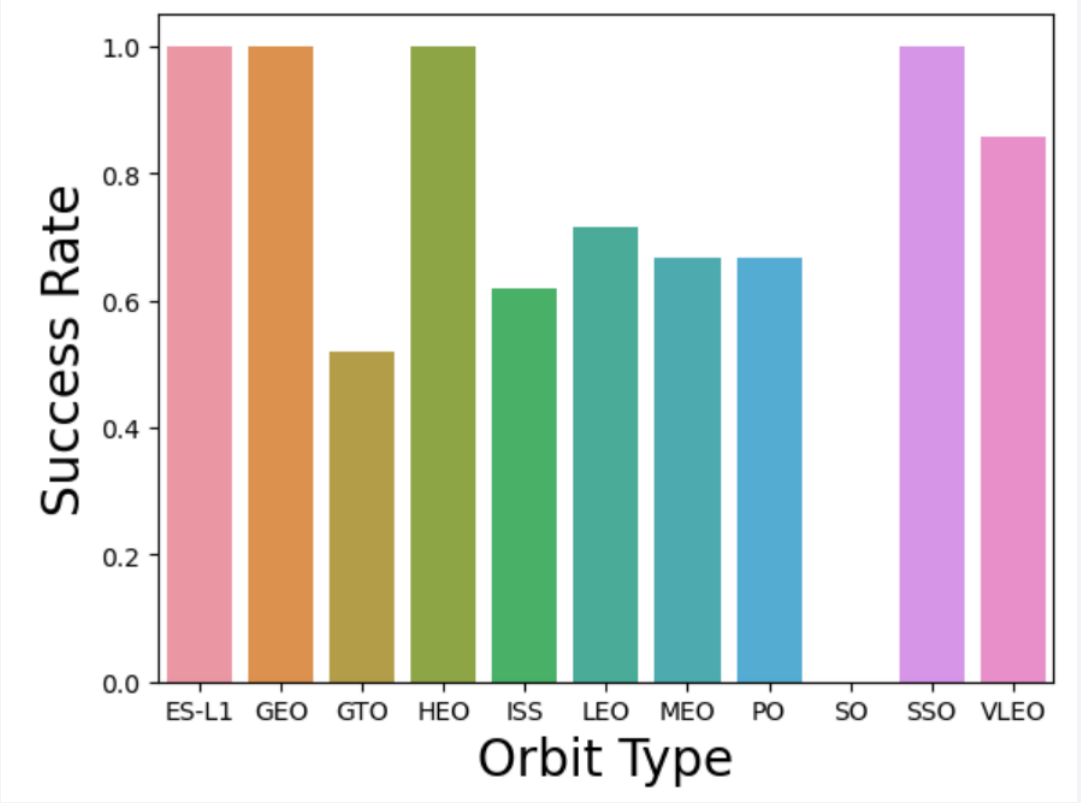
Payload vs. Launch Site



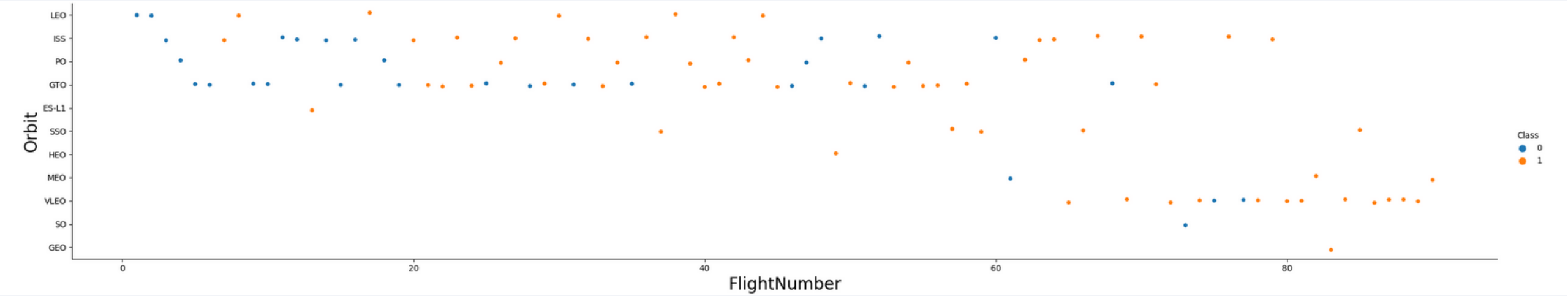
- Majority of launches are below payload mass 10000
- There is no launches from VAFB-SLC for payload mass above 10000

Success Rate vs. Orbit Type

- Orbit with high success rates: ES-L1, GEO, HEO, SSO
- Orbit with lowest success rates: GTO

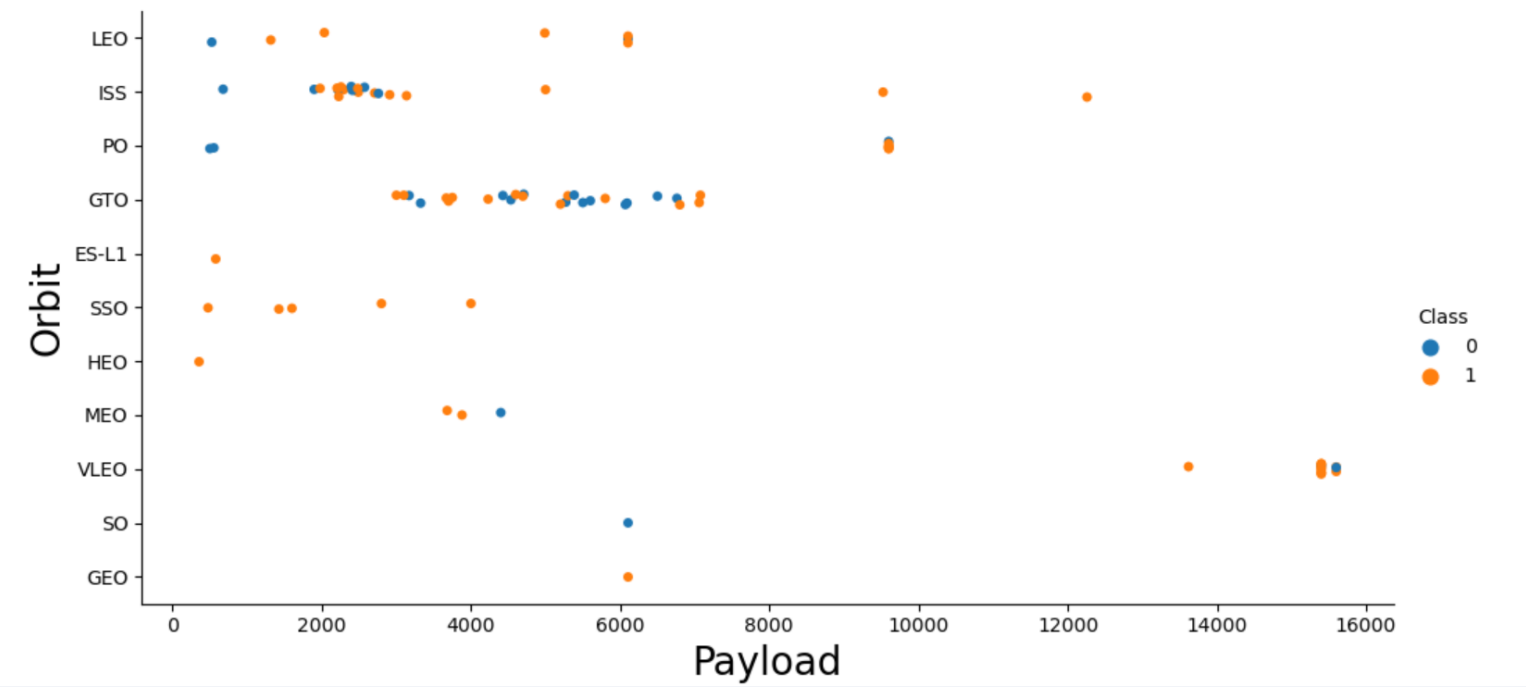


Flight Number vs. Orbit Type



- There is no clear relationship between flight number and orbit type

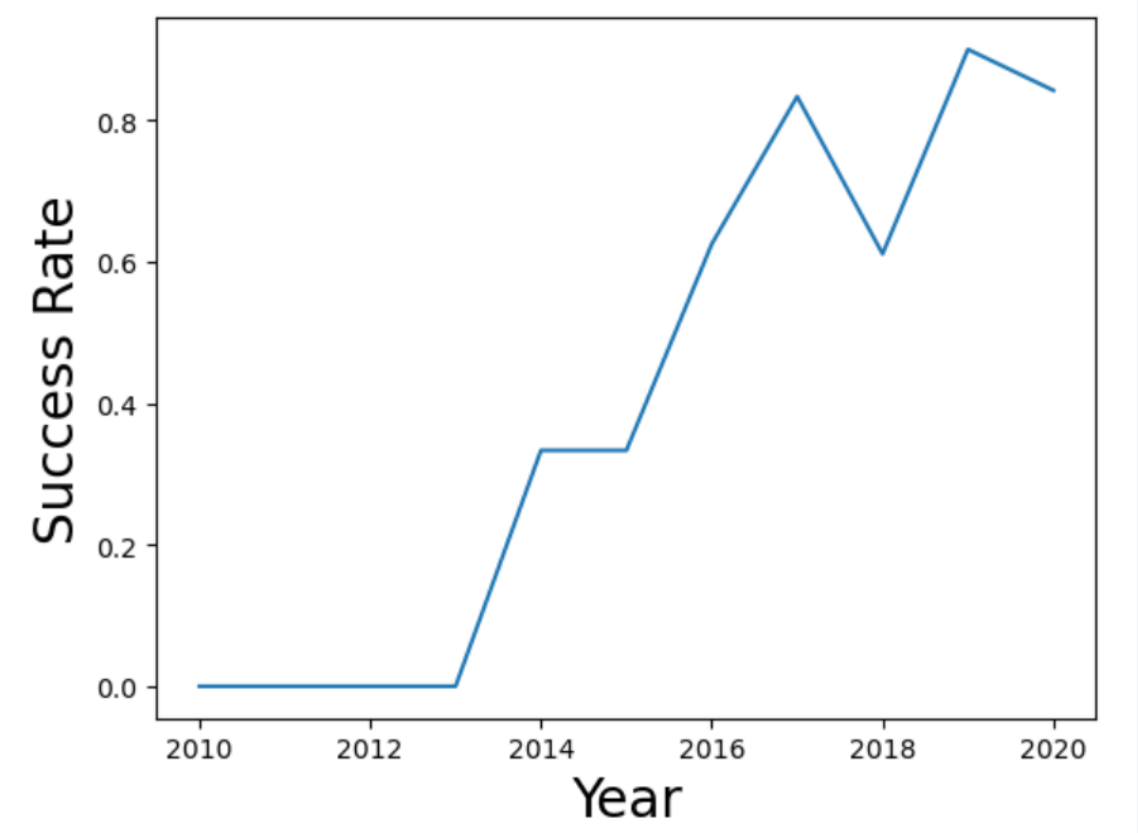
Payload vs. Orbit Type



- Success rate for GTO doesn't seem to be related to payload mass

Launch Success Yearly Trend

- Clearly success rate is increasing since 2013



All Launch Site Names

- Use keyword ‘distinct’ to retrieve unique launch site name

```
%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

Launch Site Names Begin with 'CCA'

```
%sql Select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- Keyword LIKE is used to filter for certain launch site with partial name
- 'Limit 5' is to limit only 5 entries are retrieved

Total Payload Mass

```
%sql Select SUM("PAYLOAD_MASS__KG_") from SPACEXTBL where "Customer" LIKE 'NASA (CRS)%'
```

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

SUM("PAYLOAD_MASS__KG_")
48213

- Keyword like is used to filter on column ‘Customer’ NASA CRS
- SUM will total up all the payload mass

Average Payload Mass by 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

- Function AVG will provide average values from all the filtered entries

First Successful Ground Landing Date

```
%sql SELECT min(Date) from SPACEXTBL where "Landing _Outcome" = 'Success (ground pad)'
```

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

min(Date)
01-05-2017

- Function min also can be used to find earliest date of any entries, in this case, successful ground landing

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version, "Landing _Outcome", "PAYLOAD_MASS__KG_" from SPACEXTBL
where "Landing _Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" between 4000 and 6000
```

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

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- It's possible to further limit selection with multiple filter e.g. Landing Outcome and payload mass

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT Mission_Outcome, count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome
```

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

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

- Function Count is used to count number of entries by Mission Outcome

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT 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

- Here a subquery is used to provide the maximum payload to find out booster version & its max payload

2015 Launch Records

```
%sql SELECT substr(Date, 4, 2) as month,"Landing _Outcome",BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL
where substr(Date,7,4)='2015' and "Landing _Outcome" = 'Failure (drone ship)'
```

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

month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Month, Year value is extracted from Date column via subtr function, which enable filter data by year 2015

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

```
%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
```

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

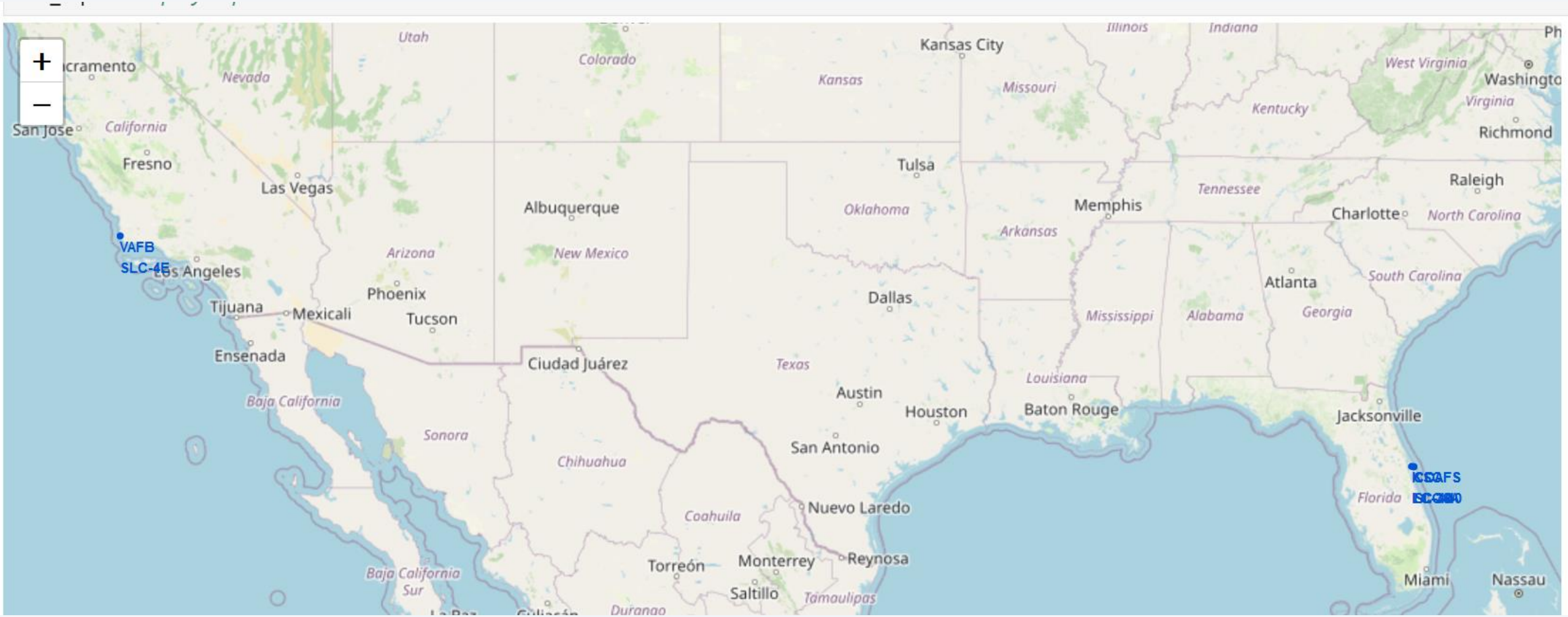
Landing _Outcome	count("Landing _Outcome")
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

- Used of Group By is need for any aggregation function such as Count, Avg, etc..
- Results can also be sorted by aggregation column

Section 3

Launch Sites Proximities Analysis

SpaceX Launch Site Locations

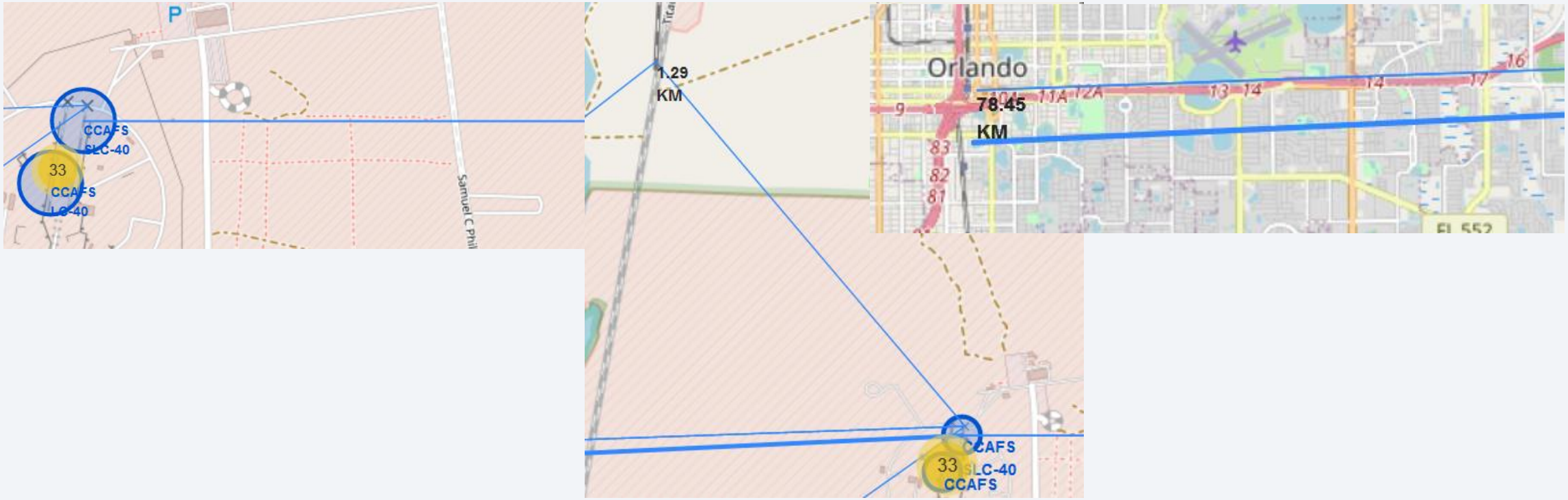


- All SpaceX launch locations are mark on map

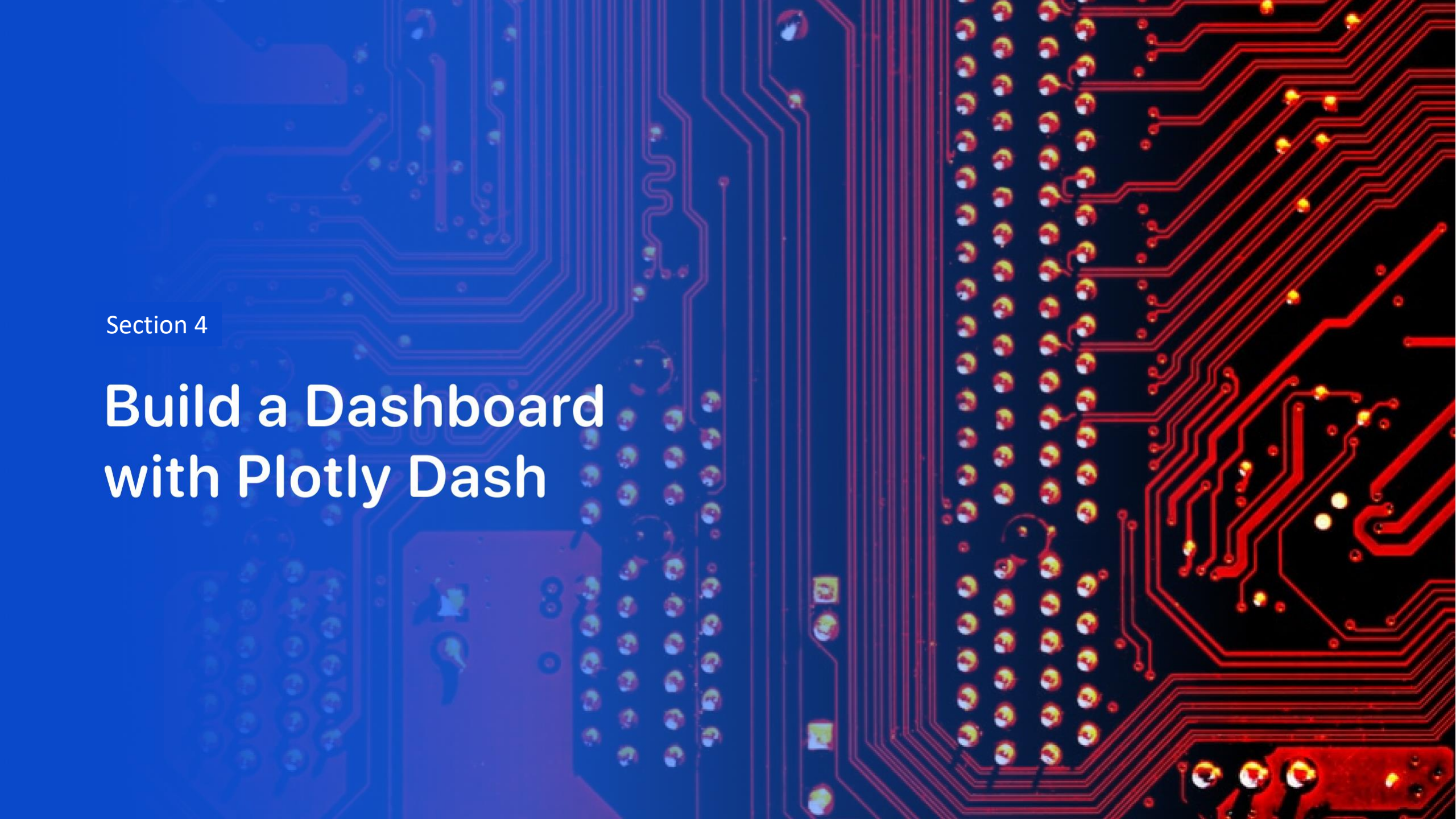
Launch site with color markers

- Color coded markers shows successful launches in green while red represents failed launches.

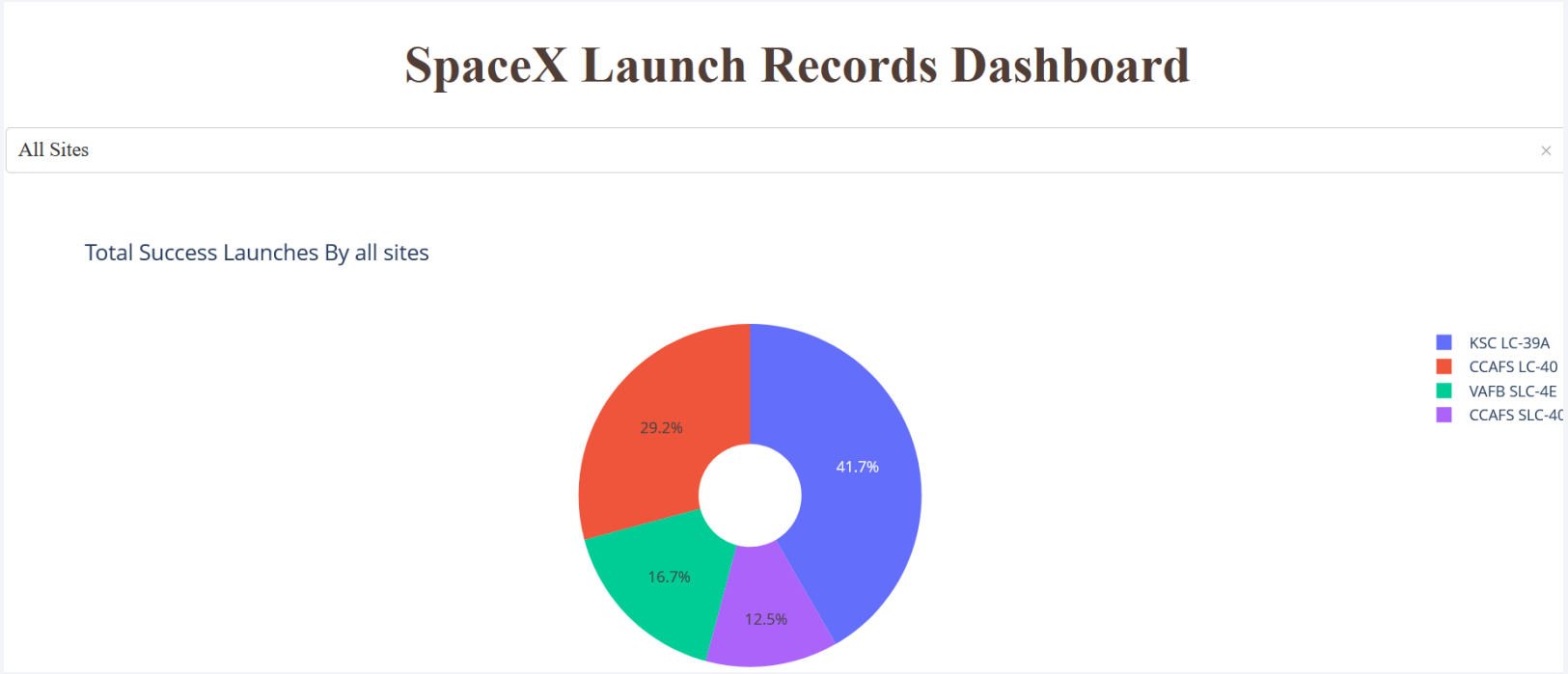
Launch site distance to landmarks



- Folium map also can display distance between launch site to certain landmarks


Section 4
Build a Dashboard with Plotly Dash

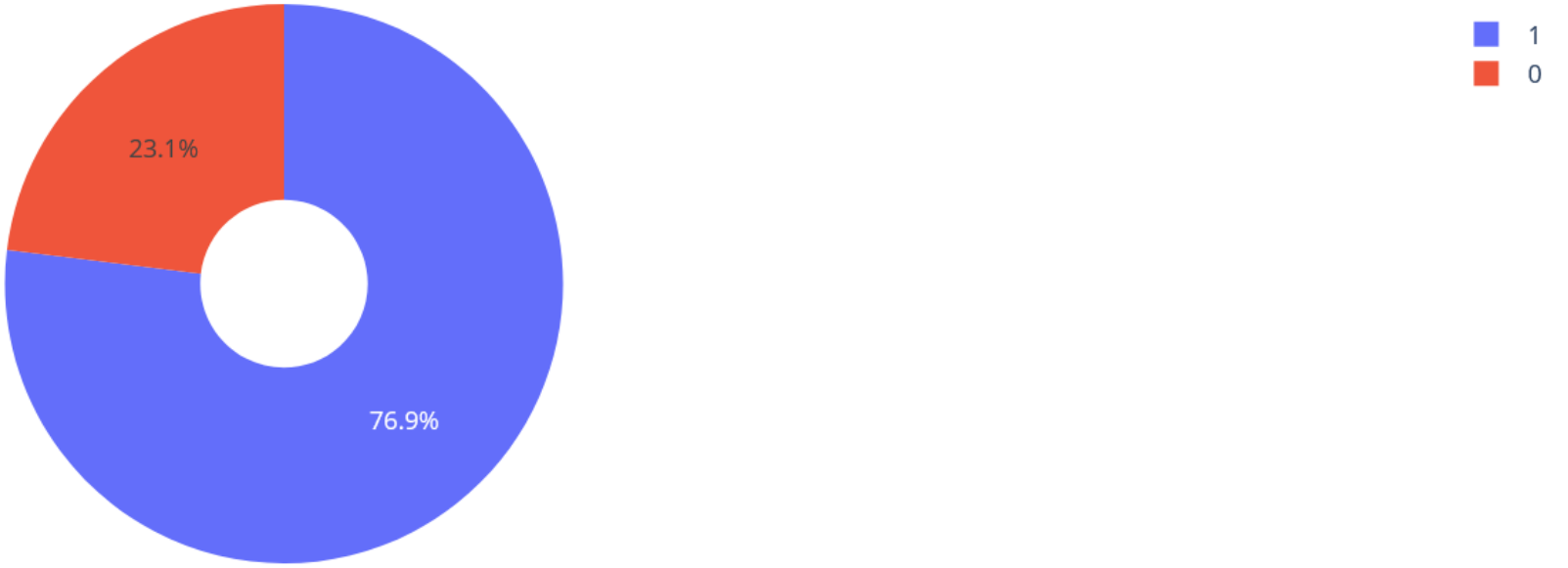
Launch success rate by sites



- KSC LC-39A has highest success launches
- CCAF SLC-40 has lowest success launches

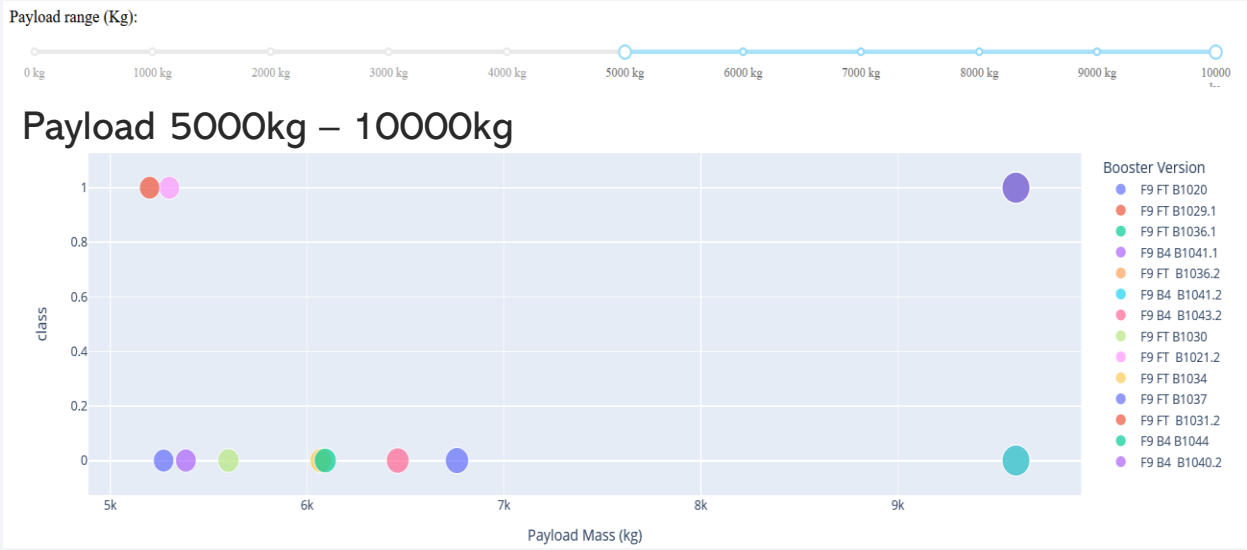
Highest Success Rate Launch Site: KSC LC-39A

Total Success Launches for site KSC LC-39A



- KSC LC-39A has success launch rate of 76.9%

Launch Outcome Status vs Payload



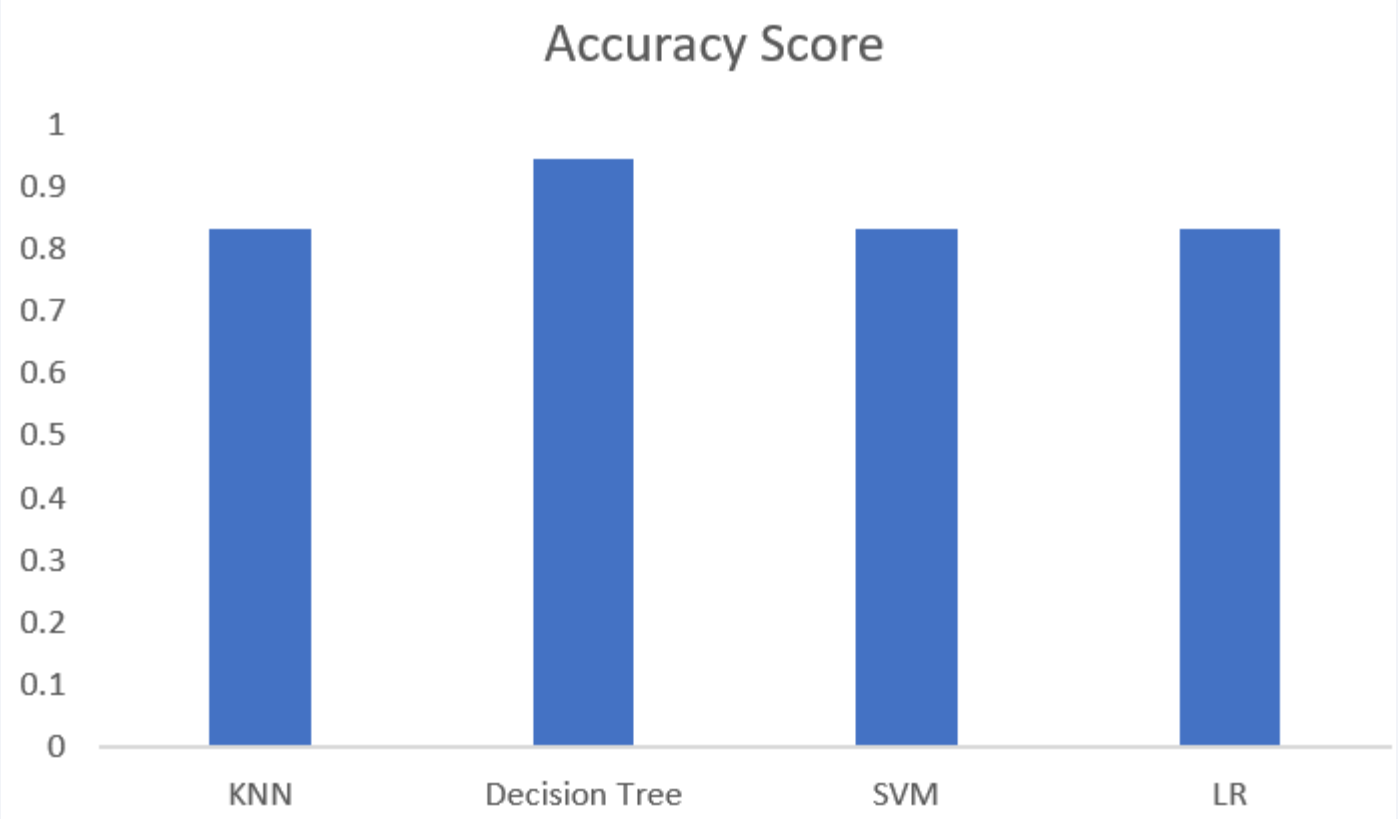
Payload <5000kg has higher success rate than payloads >5000kg

Section 5

Predictive Analysis (Classification)

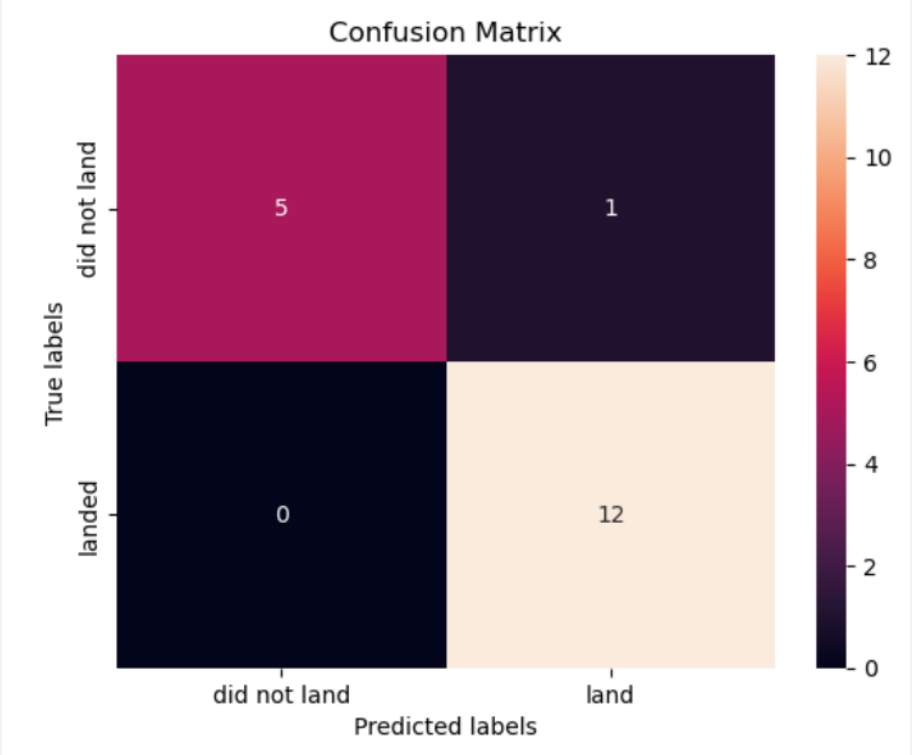
Classification Accuracy

- Decision Tree has highest accuracy with 0.94



Confusion Matrix

- Confusion matrix for Decision Tree show the classifier can clearly separate different classes.



Conclusions

- Launch site KSC LC-39A have the most successful launches of any sites, 76.9%
- Lower payload mass launches have higher success rates
- SpaceX success launches have been increasing since 2013
- Decision Tree classifier is most suitable Machine Learning algorithm for this dataset

Thank you!