Winning Space Race with Data Science

Sugeng Wahyudi
25th July 2022

Outline

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

Executive Summary

After data have been wrangled and make some explanatory analysis, we can make a machine learning model to predict spaceX success landing with accuracy, and algorithm as follow

Method/Algorithm	Accuracy
Logistics Regression	: 1
Support Vector Machine	: 0.9444444444444444
Decision tree	: 0.9444444444444444
K nearest neighbors	: 1

Introduction

Space-related industries are so expensive because the cost of the rockets.

Indeed, SpaceX can make the cost cheaper than competitors because SpaceX can reuse its first stage(rocket).

So, to push the cost of operation, it is very important that we analyze the spaceX launch data and get insights for optimum spaceX launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected using two methods that is using API and Web Scraping
- Perform data wrangling
  - Data cleaned, and processed to output outcome label that classifies the first stage land successfully or not
- 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

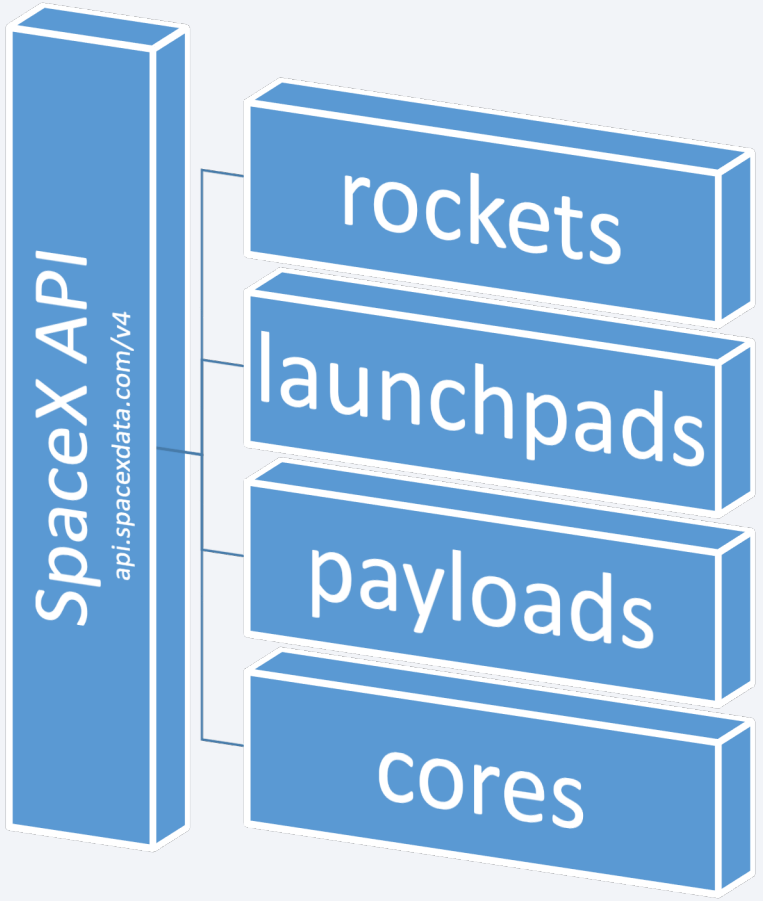
Data obtained using two methods that are using API from spacexdata website and Webscraping from wikipages as follow:

API	Webscraping
- Using request to get data from spacexAPI	- Using request to get data from wikipedia
- Decode json response and turn to dataframe using .json_normalize()	- Parse the response using BeautifulSoup to make easy to understand html and to navigate to the table data
- Wrangled data to make the dataframe clean as expected	- Extract data from html table to pandas dataframe using BeautifullSoup

Data Collection – SpaceX API

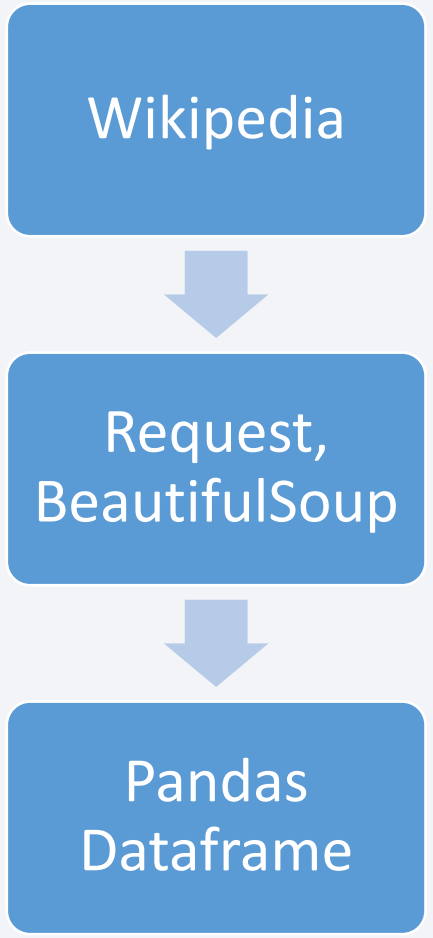
- We used request to get data from the spaceX API to collect data that contain rockets, launchpads, payloads, cores.
- Data then wrangled to make clean pandas dataframe

- Here is the notebook I used to get data from SpaceX API https://github.com/Gengsu07/LEARNING_REPO/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data is fetched from Wikipedia using request and then parse with beautifulsoup to make HTML response more readable.
- Bs4 is also used to navigate to the table needed, then the data extracted to pandas dataframe
- Here is the notebook https://github.com/Gengsu07/LEARNING_REPO/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Before wrangled, we make basic EDA to understand the data to check null, missing value, and value_counts
- Based on outcome column, make labels column ‘Class’ to classify the outcome to 0 if failed and 1 if success
- Here is complete jupyter notebook code I used:https://github.com/Gengsu07/LEARNING_REPO/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

In [9]:
```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

In [18]:
```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

Out[18]:
```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

In [14]:
```
landing_outcomes.keys()
```

Out[14]:
```
Index(['True ASDS', 'None None', 'True RTLS', 'False ASDS', 'True Ocean',
       'False Ocean', 'None ASDS', 'False RTLS'],
      dtype='object')
```

In [26]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for outcome in df['Outcome']:
    if outcome in(bad_outcomes):
        landing_class.append(0)
    else :
        landing_class.append(1)
#anding_outcomes[-bad_outcomes].values = 1
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

In [27]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[27]:

	Class
0	0
1	0
2	0

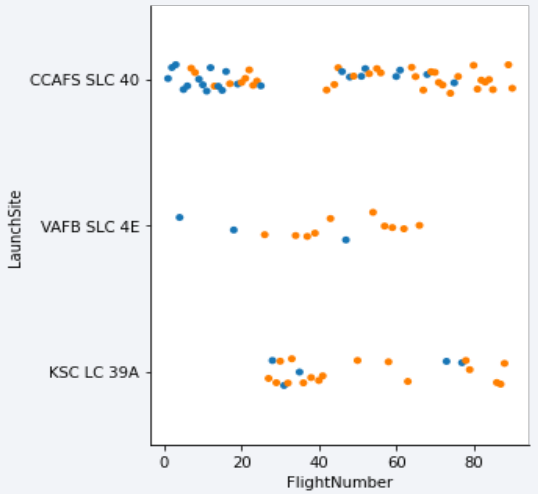
EDA with Data Visualization

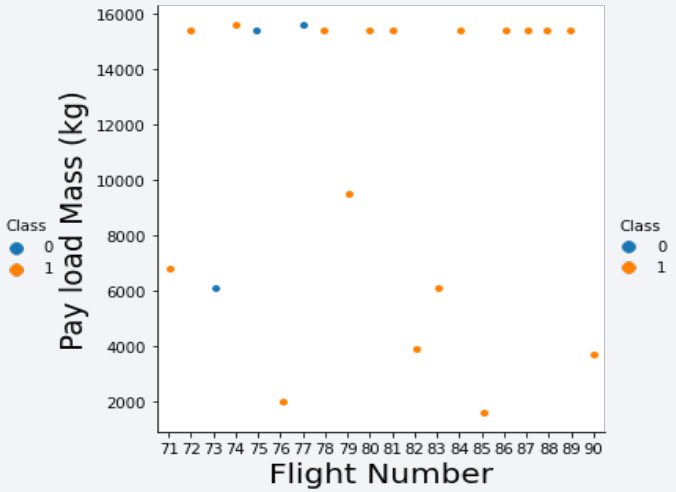
Categorical Plot

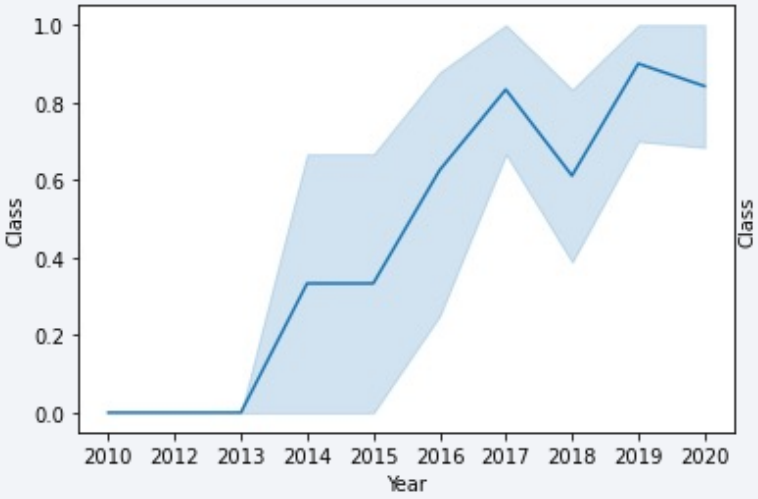
Using catplot to inspect relationship between flight number and payload mass, flight number and launch site affect to the outcome

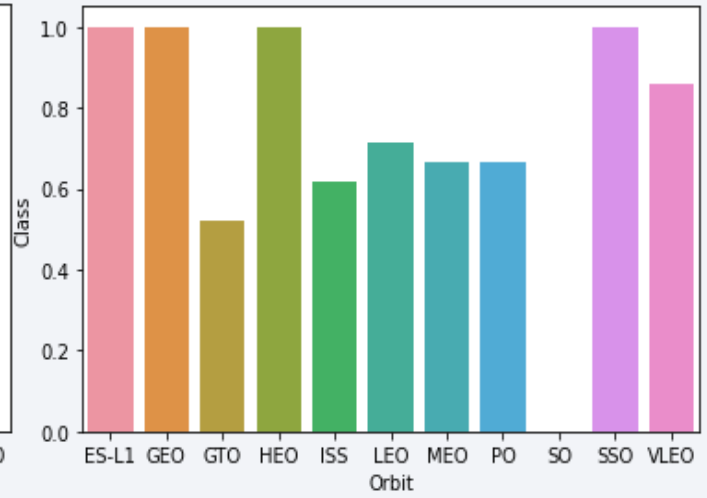
Scatterplot

Using scatterplot to inspect relationship payload mass and launch site, flight number and orbit type affect to the outcome









- **Bar Chart**

  Using bar chart to view success rate across orbit type

- **Line Chart**

  Using lineplot to visualize yearly trend average success rate

- Here is the notebook
  https://github.com/Gengsu07/LEARNING_REPO/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Data loaded to IBM Db2 database instance on IBM cloud, so we can query and analyze the data from Jupyter Notebook using IBM Db2 connector. Then we analyze the data as follow:

- The names of unique SpaceX mission launch site
- 5 launch site names that begin with 'CCA'
- Total payload mass carried by booster NASA CRS
- Average payload mass carried by booster F9 v1.1
- List date when first successful landing outcome in ground pad was achieved
- List names of boosters that have success landing in drone ship have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcome between 2010-06-04 and 2017-03-20

Here is the notebook
https://github.com/Gengsu07/LEARNING_REPO/blob/main/EDA%20with%20SQL.ipynb

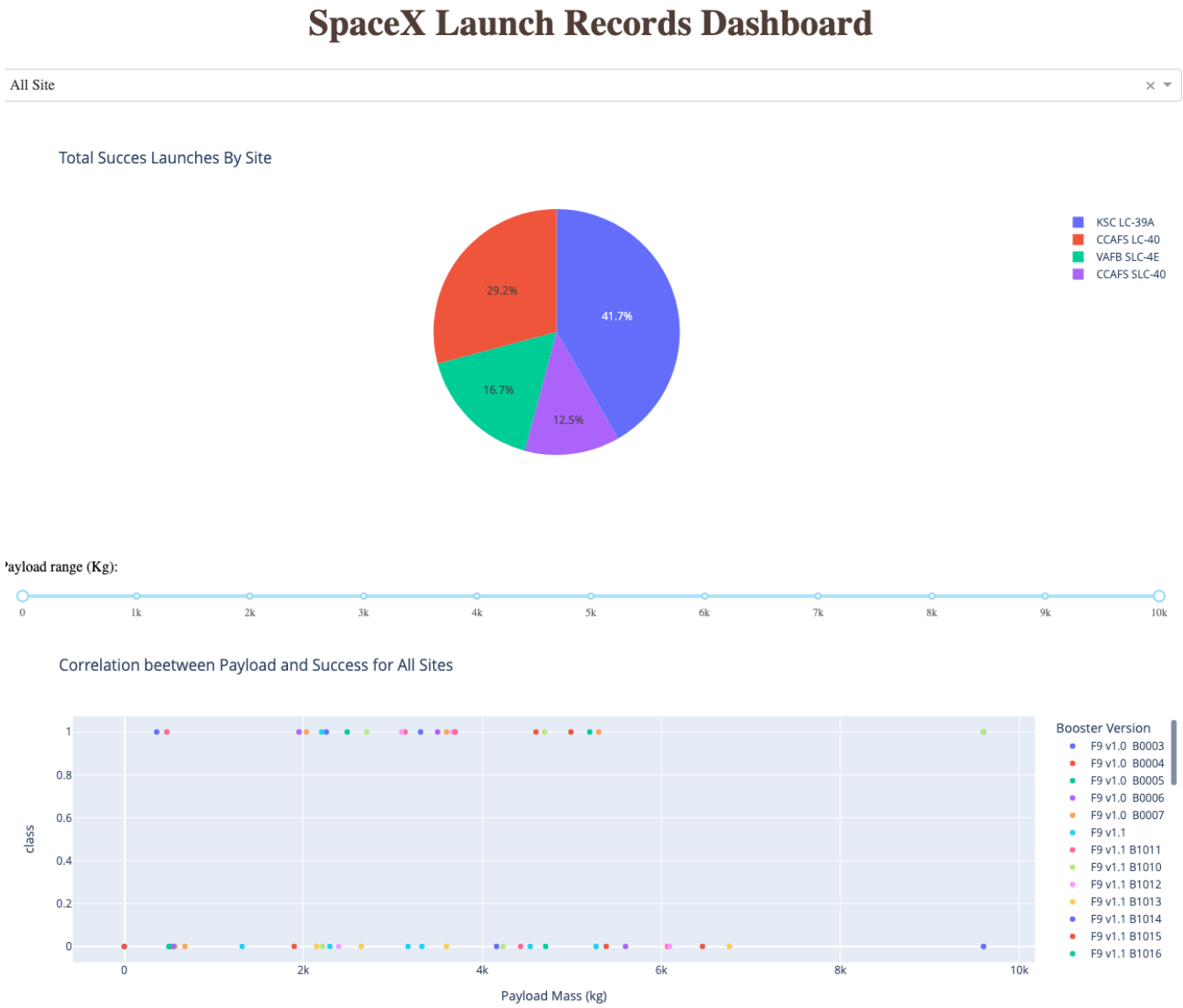
Build an Interactive Map with Folium

- We marked all launch sites with the launch site name, circle, and marker to show the launch site
- For easy map interpretation, map object colored with red for failure(0) and green for succeed(1) to indicate the landing outcome of the site
- Calculate distance between site and its proximity to answer some questions as follow:
  - How far launch site from railways, highways, coastline
  - Do launch sites keep certain distance from cities?

- Here is my notebook https://github.com/Gengsu07/LEARNING_REPO/blob/main/Folium_Jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- To see relationship between launch sites, payload mass and launch outcome interactively, we use pie chart and scatter plot
- To inspect interaction between variables, we use dropdown to choose launch sites and range slider to set the payload mass
- We add the plots and interactions to see interaction between different launch site and payload mass that can affect the launch outcome
- Here my plotly dash app python code https://github.com/Gengsu07/LEARNING_REPO/blob/main/spacex_dash_app.py

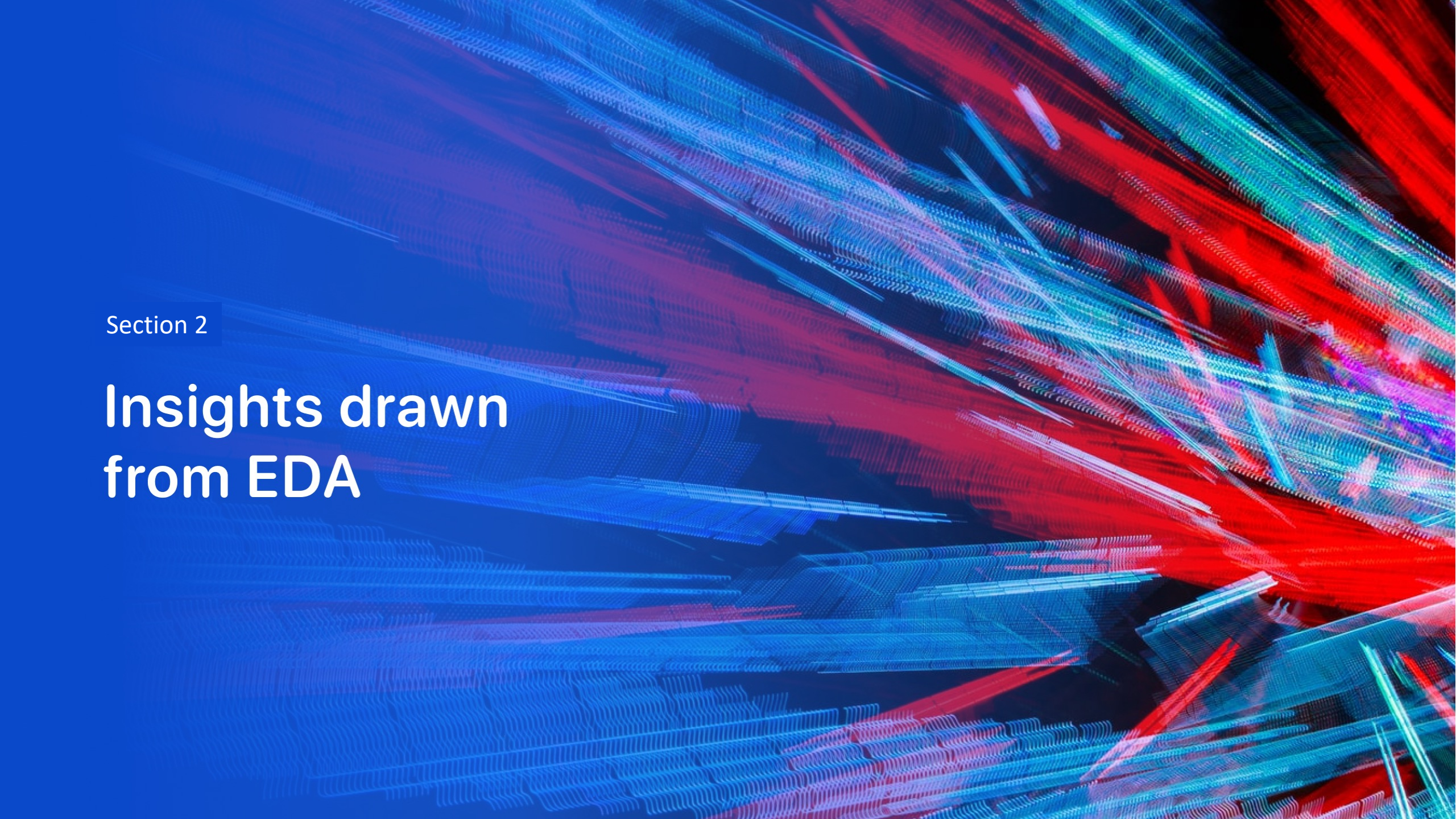


Predictive Analysis (Classification)

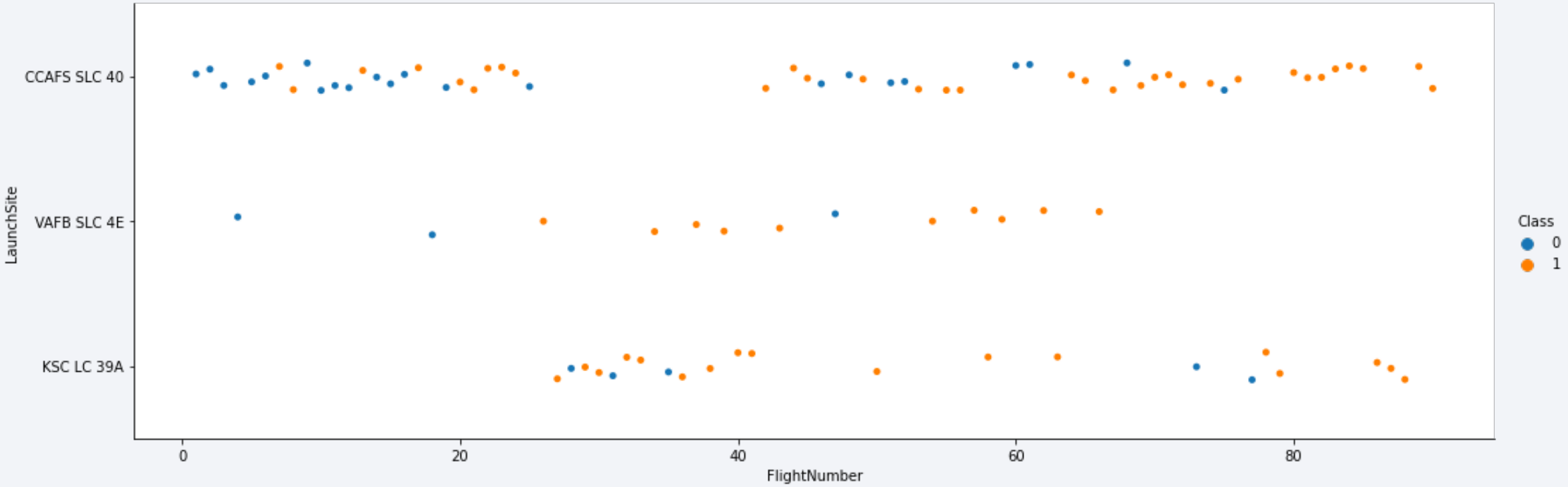
- We loaded data to pandas dataframe, then transform class label that are categorical to numpy array
- Features or X standardized using preprocessing standardscaler from sklearn
- Dataset then splitted to train and test data using train test split from sklearn
- Model trained using Decision Tree algorithm and using GridSearchCV to find the best parameters combination that achieve best accuracy.
- Here is my notebook https://github.com/Gengsu07/LEARNING_REPO/blob/main/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

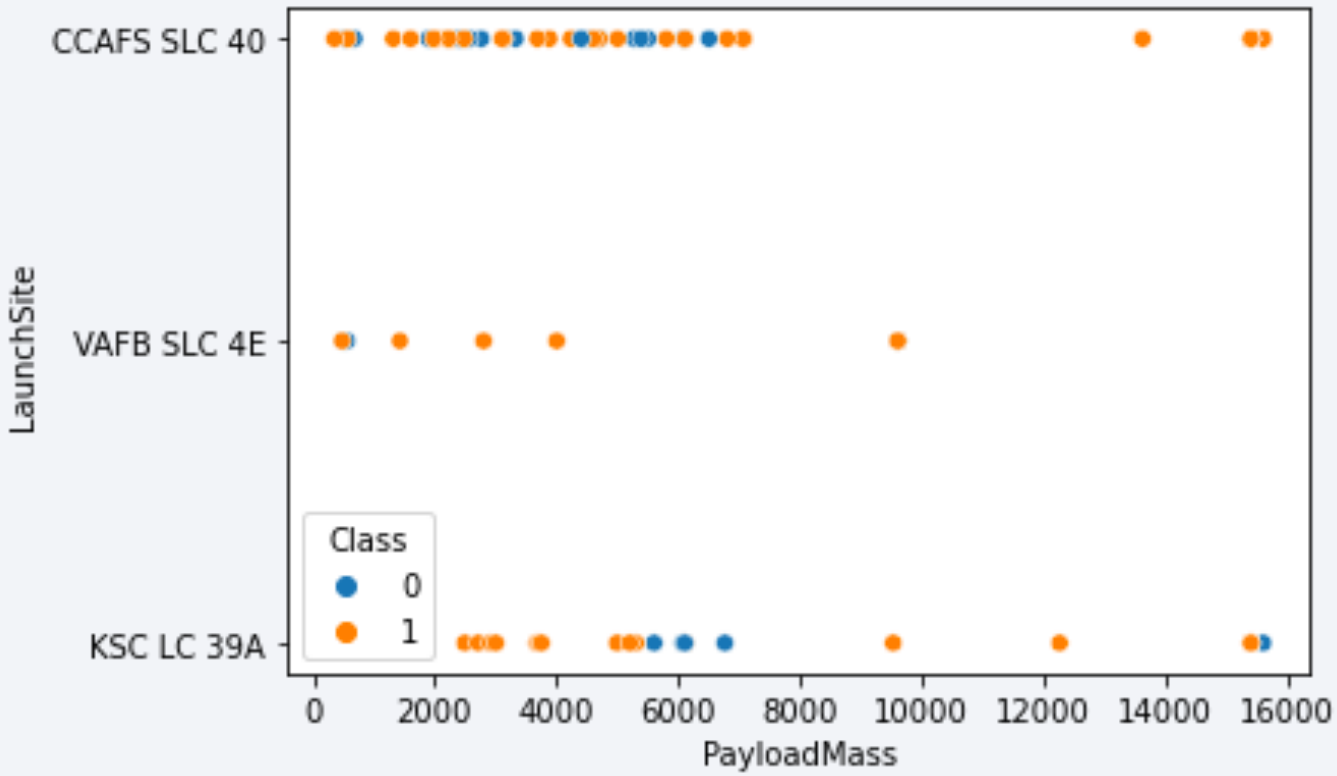


- As we see, greater or recent flight numbers have more success than earlier flights.
- Besides, we can rank the amount of flights at each launch site descending from CCAFS SLC 40, KSC LC 39A, and VAFB SLC 4E

Payload vs. Launch Site

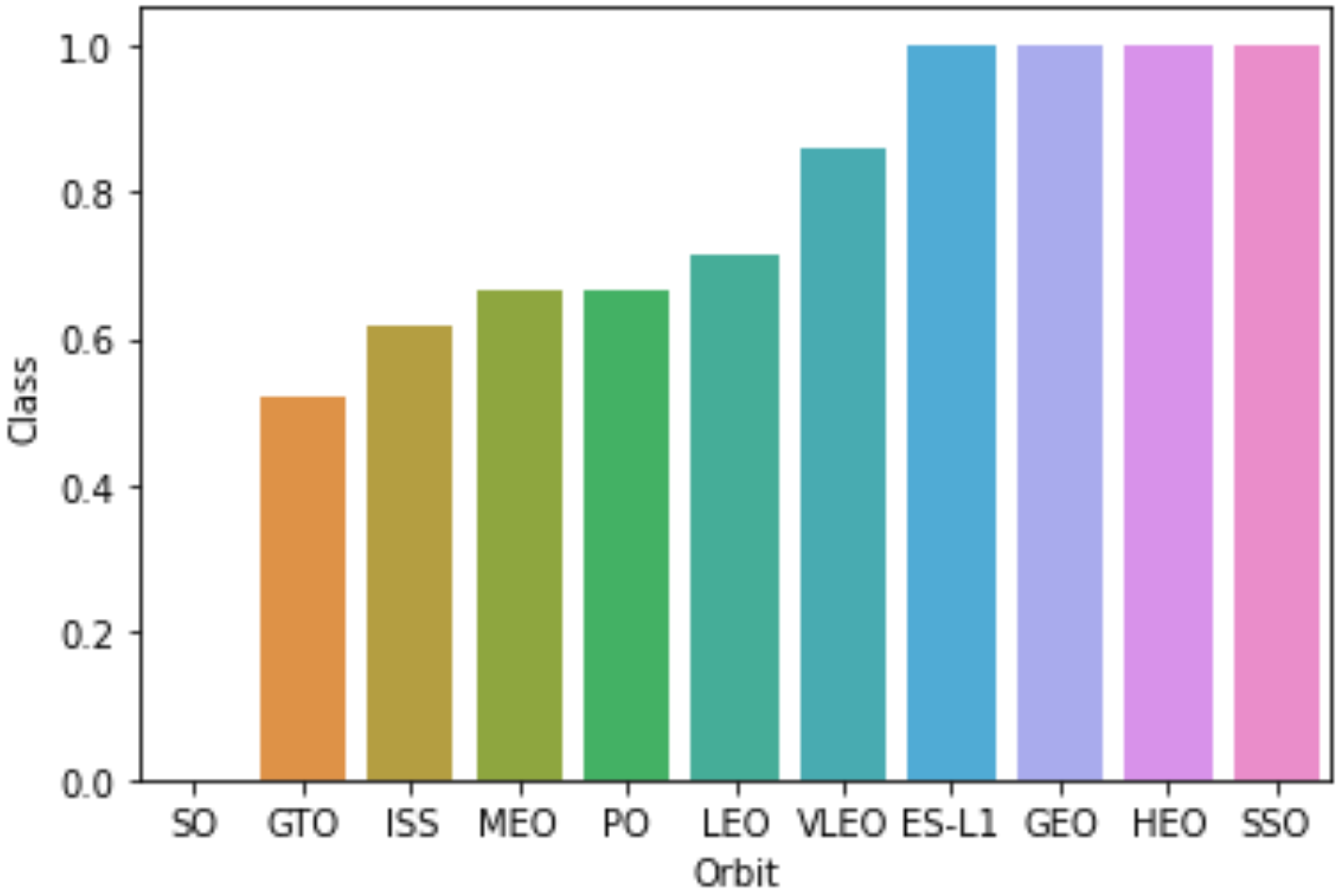
VAFB SLC 4E only used to small-sized payloads likely less than 10.000 Kg.

It seems that the more payload mass have more success landing outcome, especially for VAFB SLC 4E

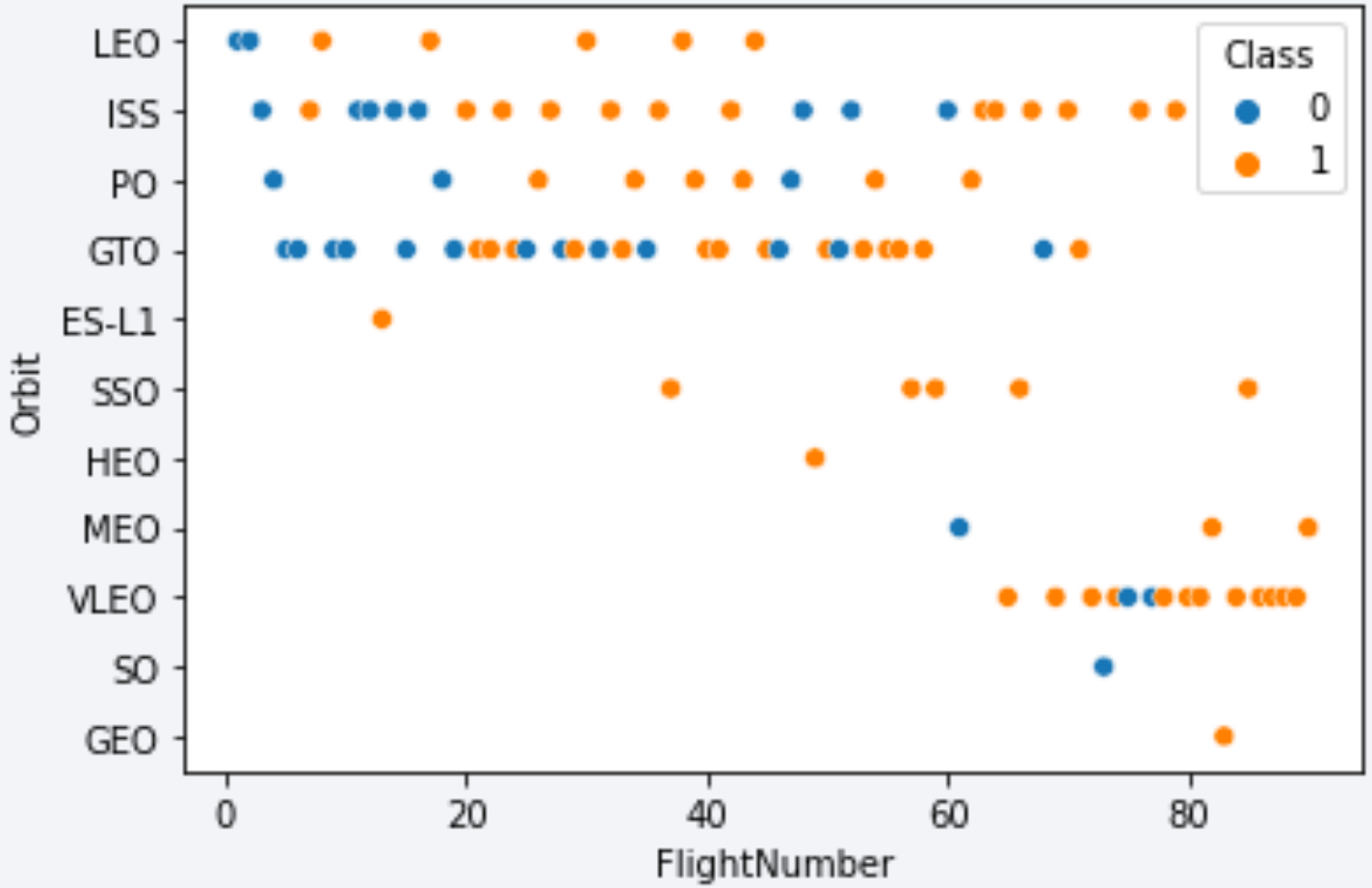


Success Rate vs. Orbit Type

Based on orbit type, we can obtain rank of success rate each orbit type ascending from SO to SSO



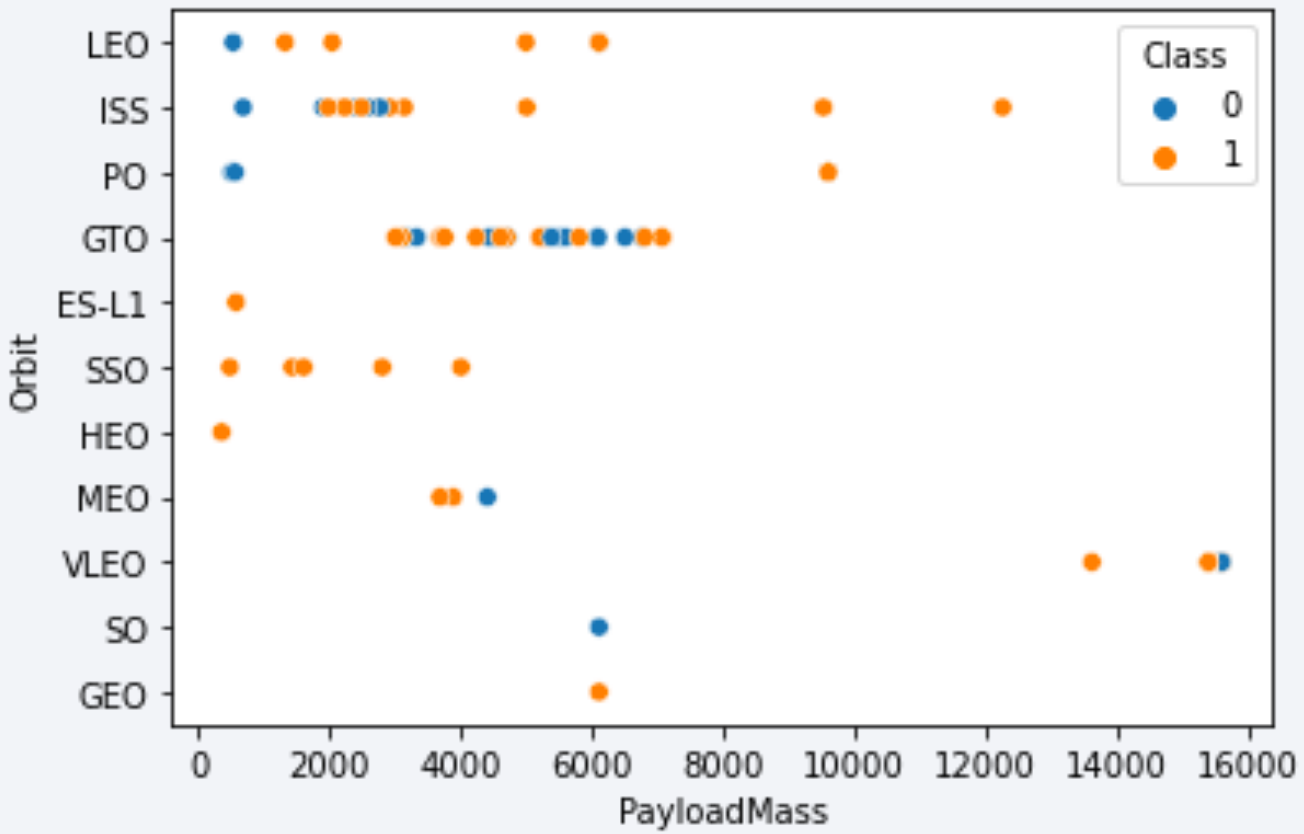
Flight Number vs. Orbit Type



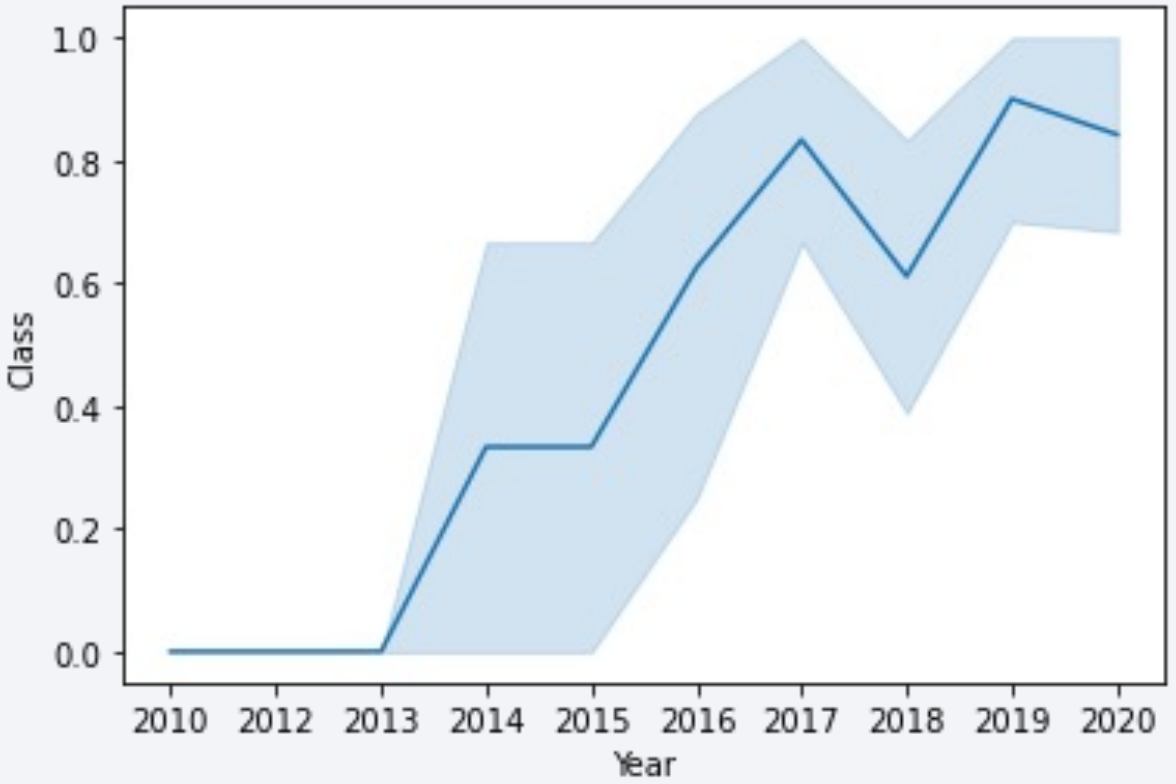
- In the early flights number about less than 40 the flights target LEO, ISS, PO and GTO orbit type, ESL-L1 and SSO just1 flight
- LEO orbit type only used less than 50 flight number
- Other orbit type become more targeted on recent flight numbers especially on VLEO orbit

Payload vs. Orbit Type

- SSO is consistently successful but only used to small-sized payload
- GTO used for payload that have mass in range 3000 kg to 8000 kg
- ES-L1, HEO, and GTO only have 1 flight and its succed, SO also have 1 flight but failed
- LEO and ISS have more succed on hight payload mass



Launch Success Yearly Trend



Based on the plot, success rate trend increased yearly from 2013 to 2020

All Launch Site Names

```
In [ ]: %sql select distinct (launch_site) from SPACEXTBL

 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399
Done.
```

Out[ ]:

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

To get unique launch site, we use distinct keyword to retrieve from SpaceX Data table

Launch Site Names Begin with 'CCA'

I'm using 'like' wildcard to get launch site that begins with 'CCA'.

To restrict only 5 record, we use 'limit' keyword

```
%%sql select *
from SPACEXTBL
where launch_site like 'CCA%'
limit 5
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-
Done.
```

DATE	time__utc_	booster_version	launch_site	
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	D
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	

Total Payload Mass

```
%%sql
select sum("PAYLOAD_MASS__KG_")
from SPACEXTBL
where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-939
Done.
    1
45596
```

To get total payload mass we using ‘sum’ keyword and ’where’ to filter customer

Average Payload Mass by F9 v1.1

We using ‘avg’ to calculate average payload mass and ‘where’ to filter the booster_version

```
%%sql
select avg("PAYLOAD_MASS__KG_")
from SPACEXTBL
where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399-
Done.
```

1
2928

First Successful Ground Landing Date

```
%%sql
select min(DATE)
from SPACEXTBL
where landing__outcome='Success (ground pad)' ;
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399-
Done.
        1
2015-12-22
```

We using ‘where’ to filter landing outcome , then add ‘min’ keyword to get the first date based the criteria

Successful Drone Ship Landing with Payload between 4000 and 6000

We using 'where' to filter landing outcome and using 'between' to slice payload mass in range 4001 to 5999

```
%%sql
select booster_version
from SPACEXTBL
where landing__outcome = 'Success (drone ship)' and
payload_mass__kg_ between 4001 and 5999
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399-c8
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select distinct mission_outcome, count(mission_outcome) as count
from SPACEXTBL
group by mission_outcome
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399-c8177b21803b.c
Done.
```

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

First we retrieve distinct mission outcome that should contain success or failure.

Then we count the occurences of each mission outcome using group by

Boosters Carried Maximum Payload

We set ‘where’ condition using subquery to only return record that have maximal payload mass, after that we retrieve unique booster version using ‘distinct’ keyword

```
%%sql
select distinct booster_version
from SPACEXTBL
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0
Done.
```

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

2015 Launch Records

```
%%sql
select booster_version, launch_site
from SPACEXTBL
where landing__outcome = 'Failure (drone ship)'
and year(DATE)= 2015
```

```
 * ibm_db_sa://lgn00012:***@125f9f61-9715-46f9-93
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

We using ‘year’ to get the year of the flight to only in 2015 and filter the landing outcome to get onl ‘Failurre (drone ship)’

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

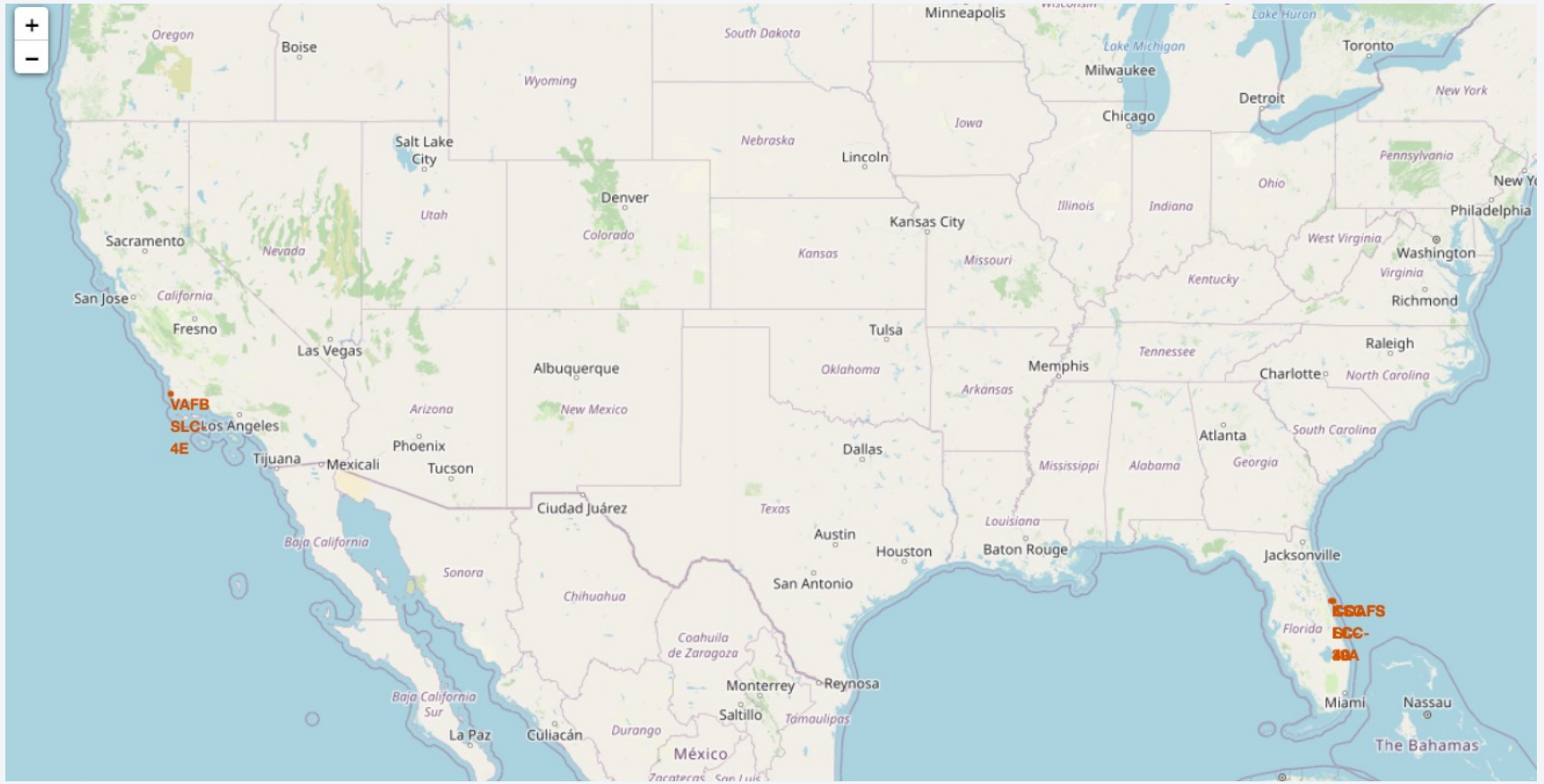
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Present your query result with a short explanation here


Section 3
Launch Sites
Proximities Analysis

SpaceX Launch Sites Map

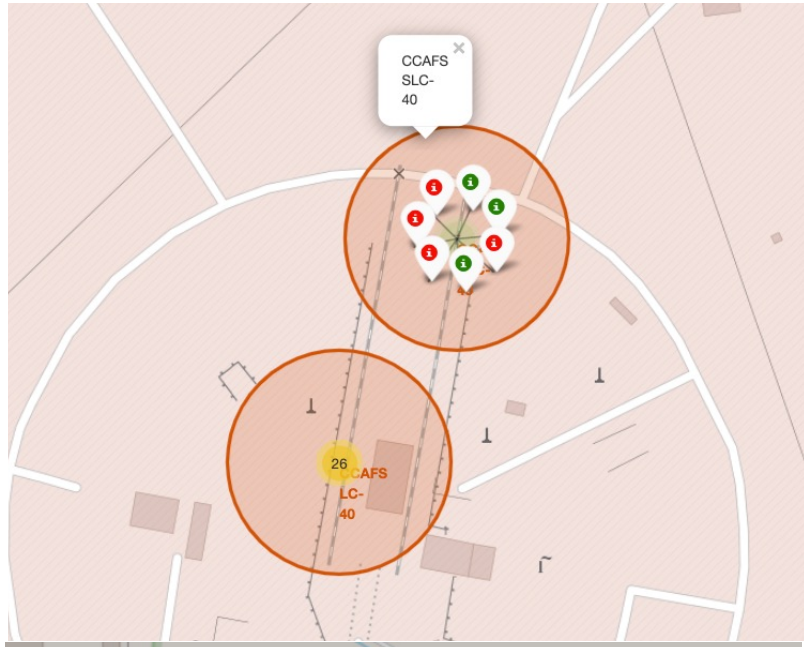
All launch sites are near from coast and have proximity with the equator line

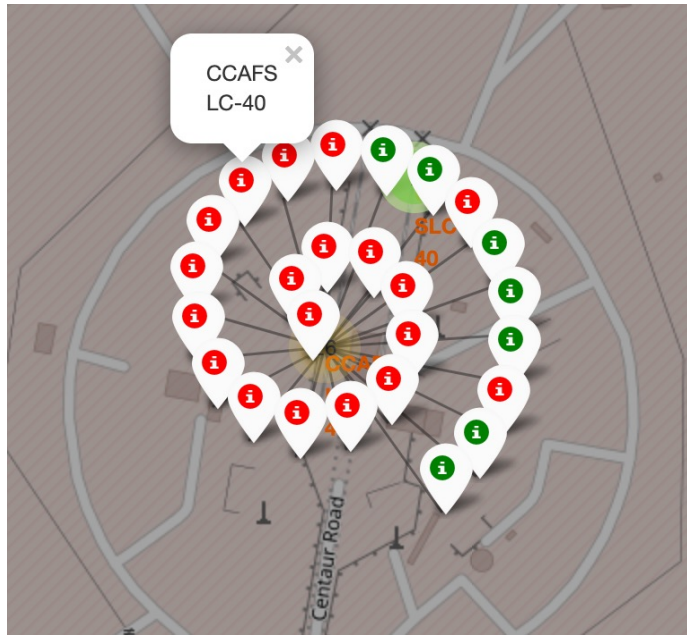


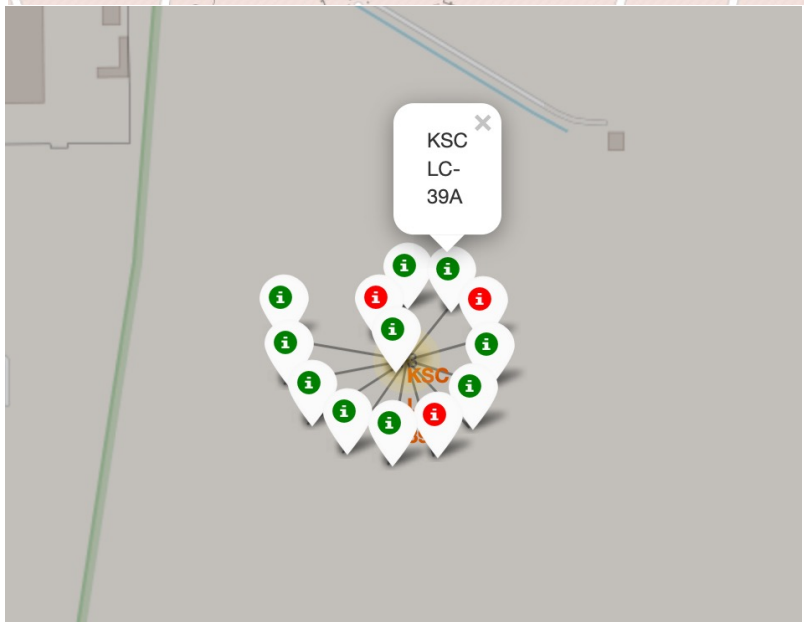
SpaceX Success Launch Map

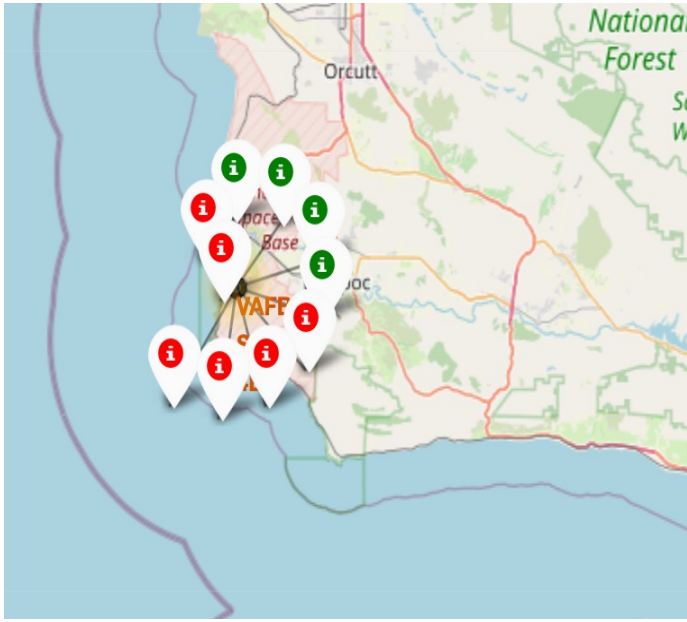
Launch sites colored based on succeed landing outcome

Green for success and Red for failure

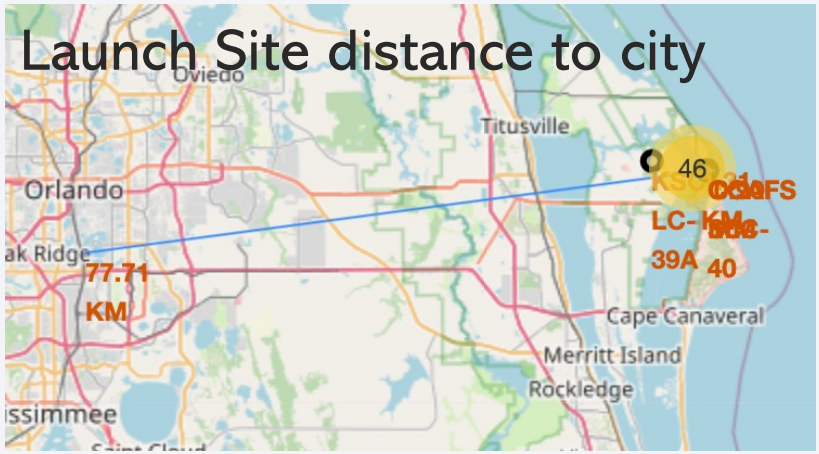


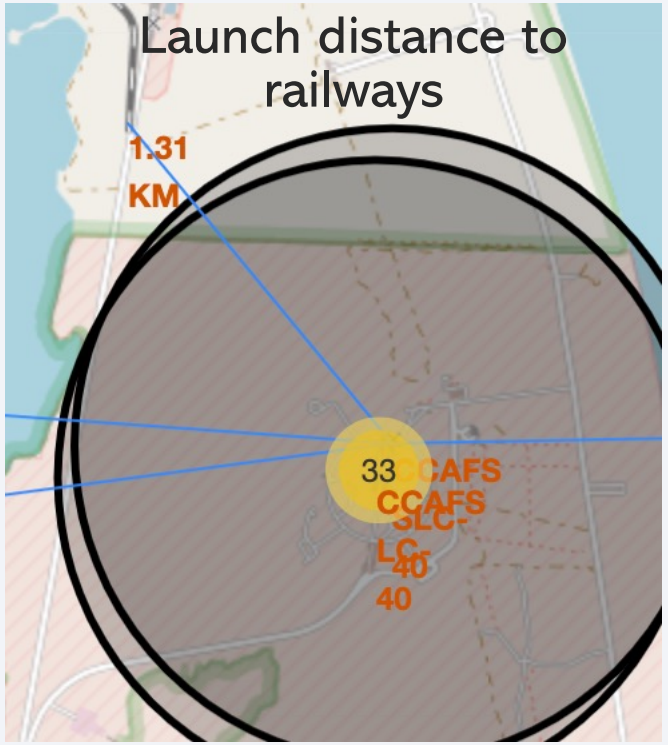


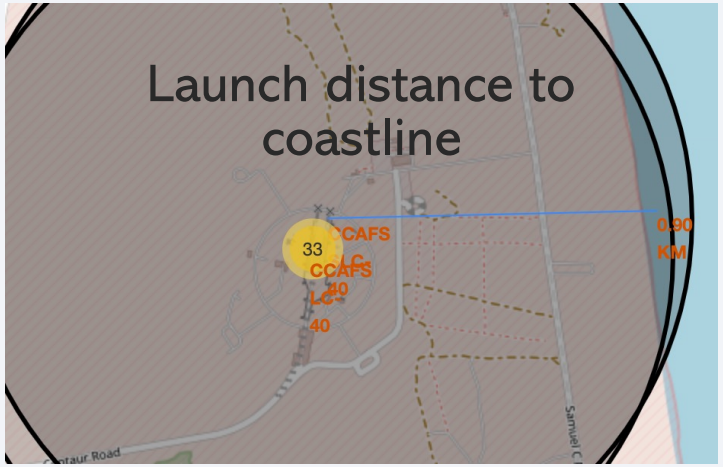


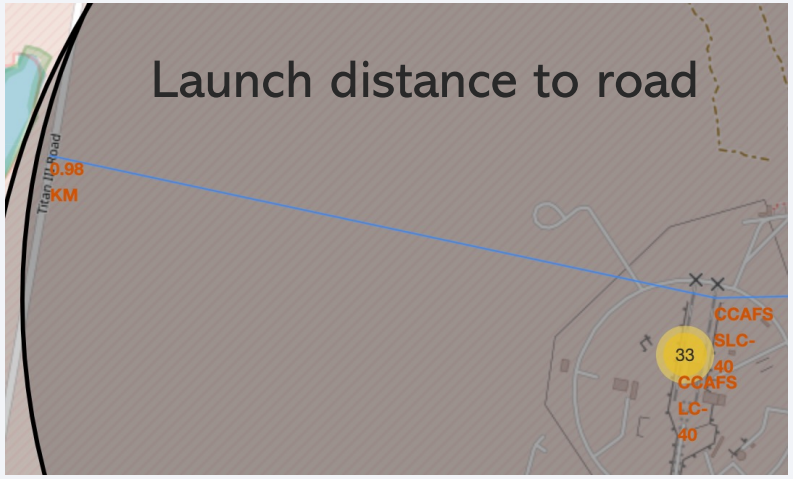


SpaceX Launch Site Proximity



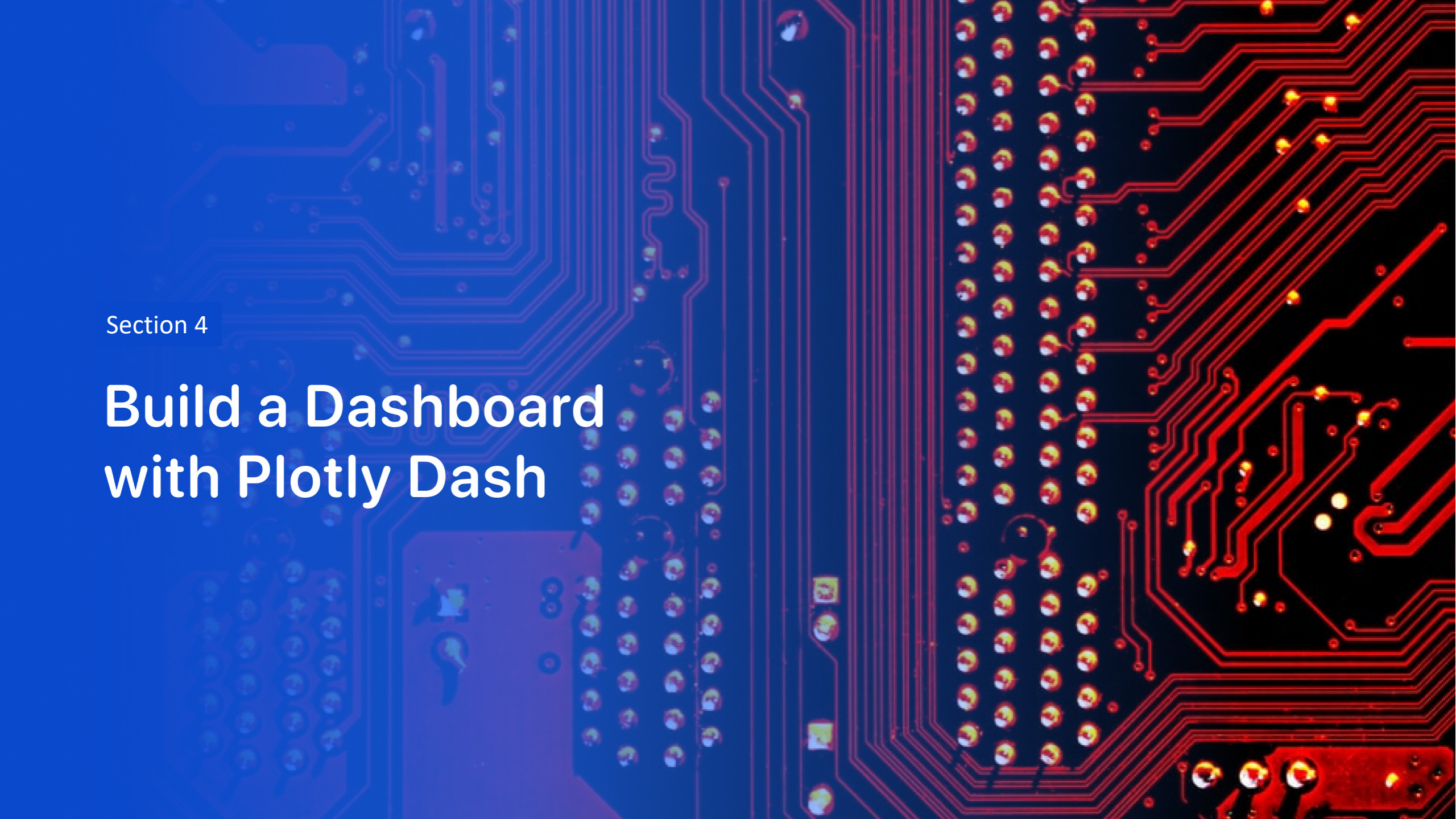




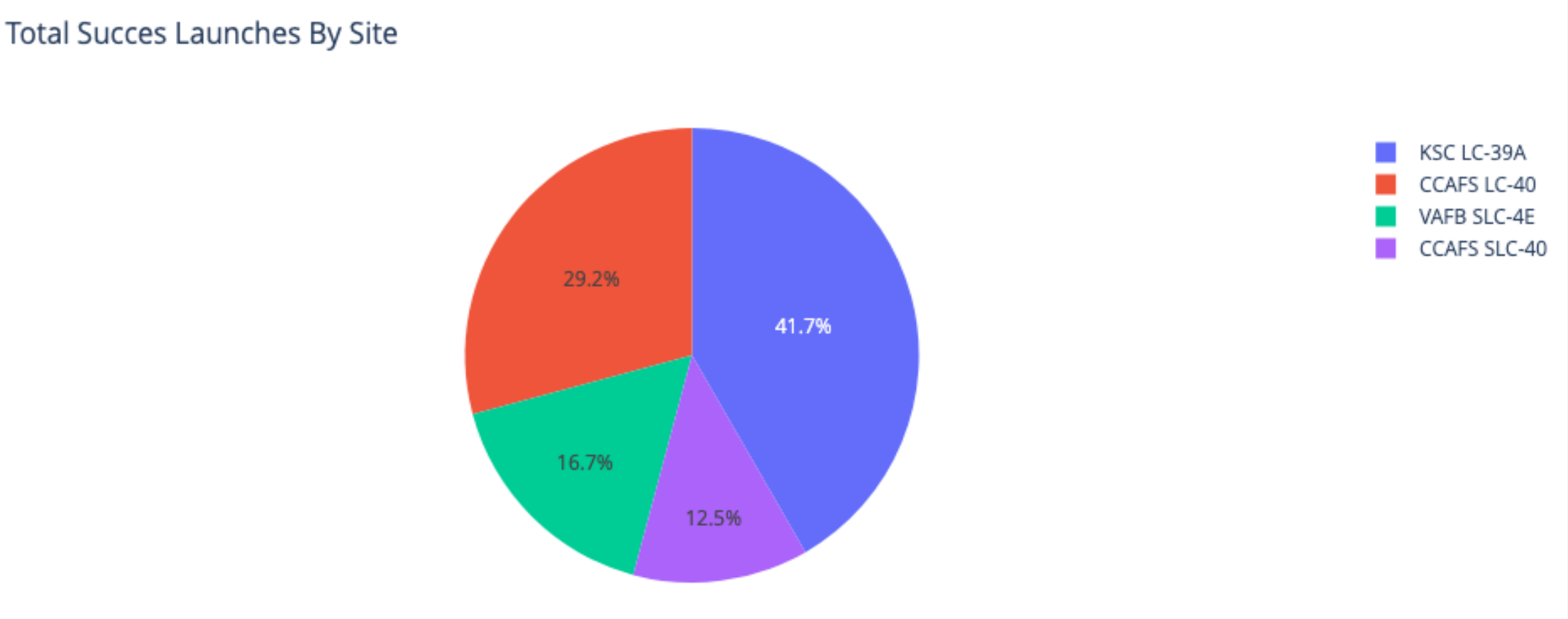


Launch site near from coast, road, and railways but far from cities

https://github.com/Gengsu07/LEARNING_REPO/blob/main/Folium_Jupyter_launch_site_location.ipynb


Section 4
Build a Dashboard with Plotly Dash

Success Rata Across Laucnh Site

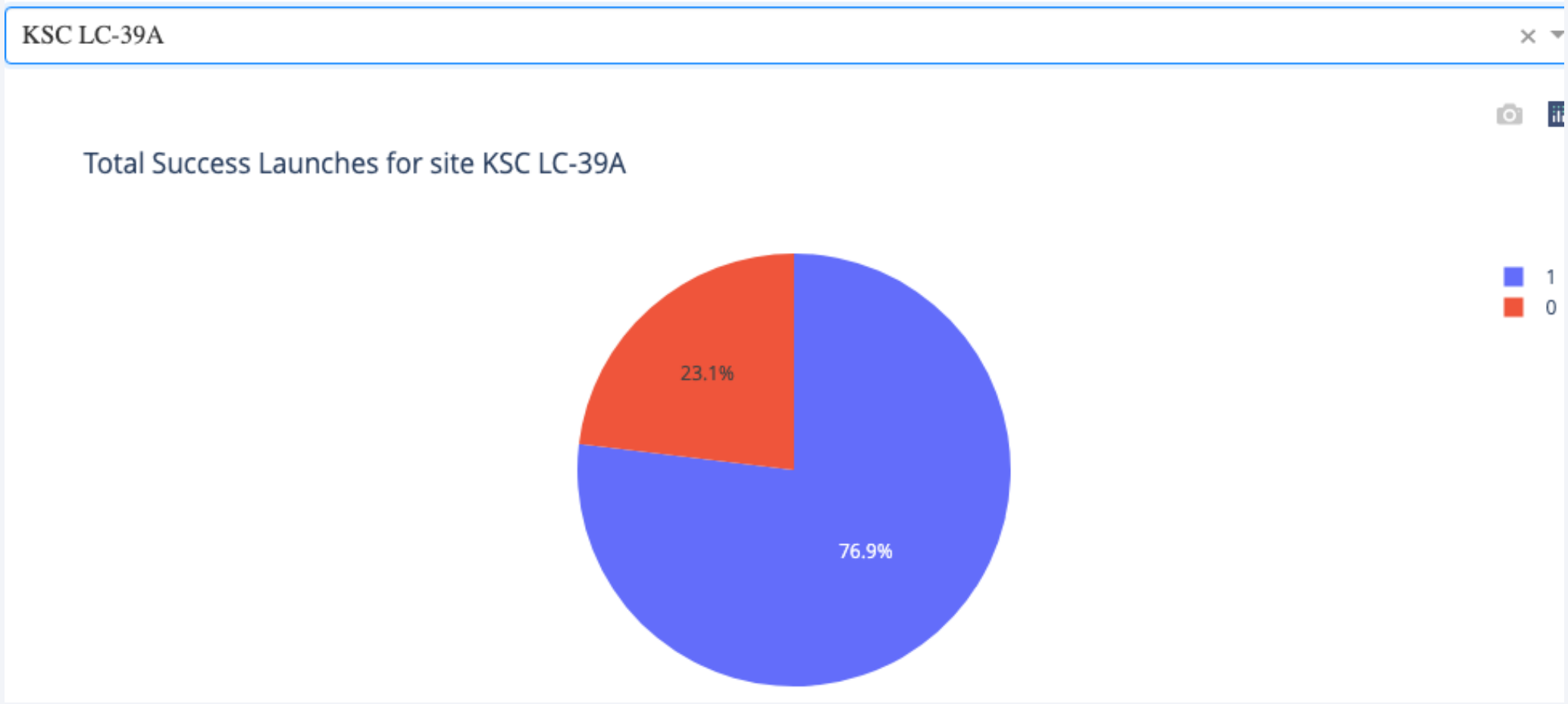


Easily understand success rate SpaceX landing outcome across launch site.

KSC LC-39A has the biggest success rate

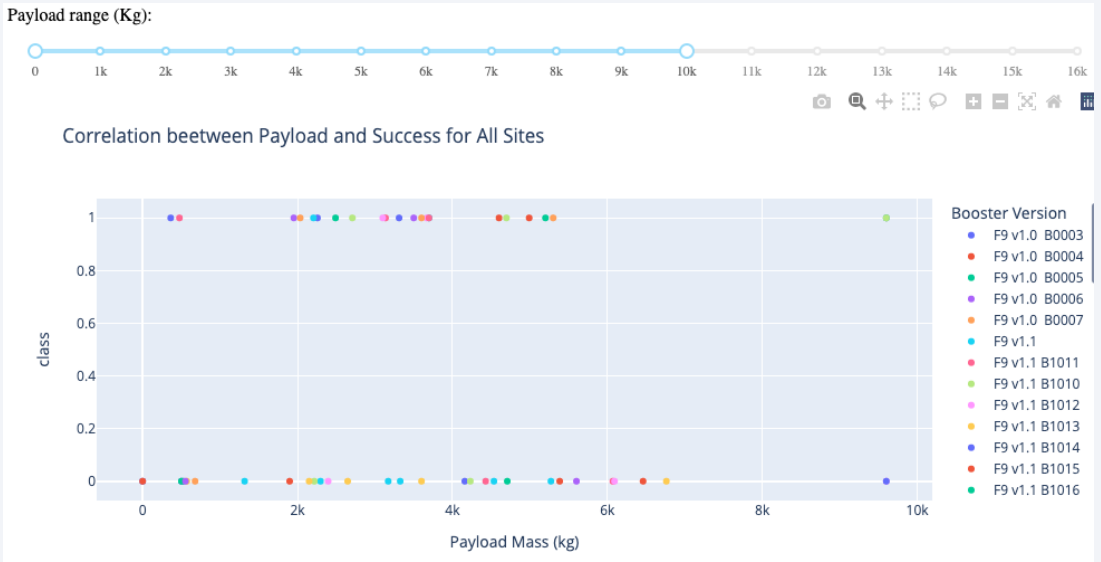
Drilldown Succes Rates each Launch Site

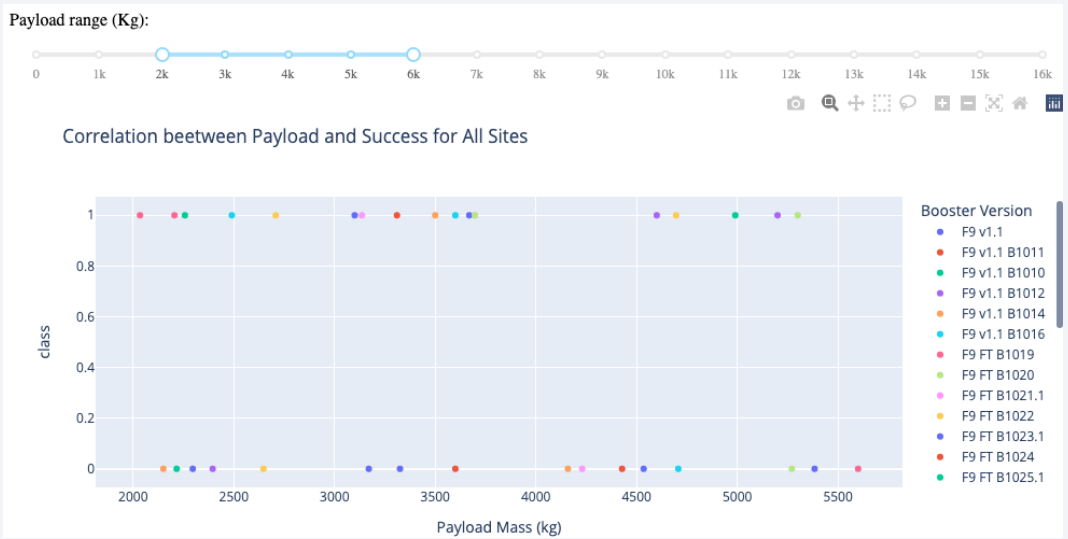
Drilldown to see landing outcome percentages in each launch site.
KSC LC-39A have 76,9% success landing and 23,1% failure



Payload Mass and Booster to Success

Insights relationship payload mass, booster version, and its effect on landing outcome easily found

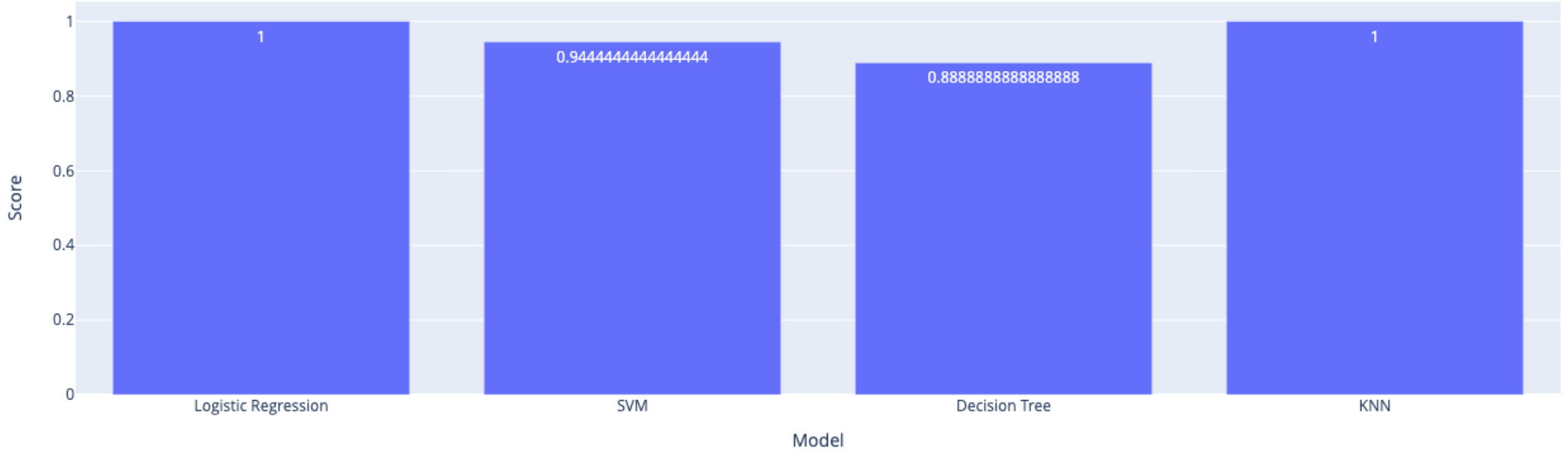




Most of the successful booster have payload mass between 2.000Kg and 5.500 Kg

Section 5

Predictive Analysis (Classification)

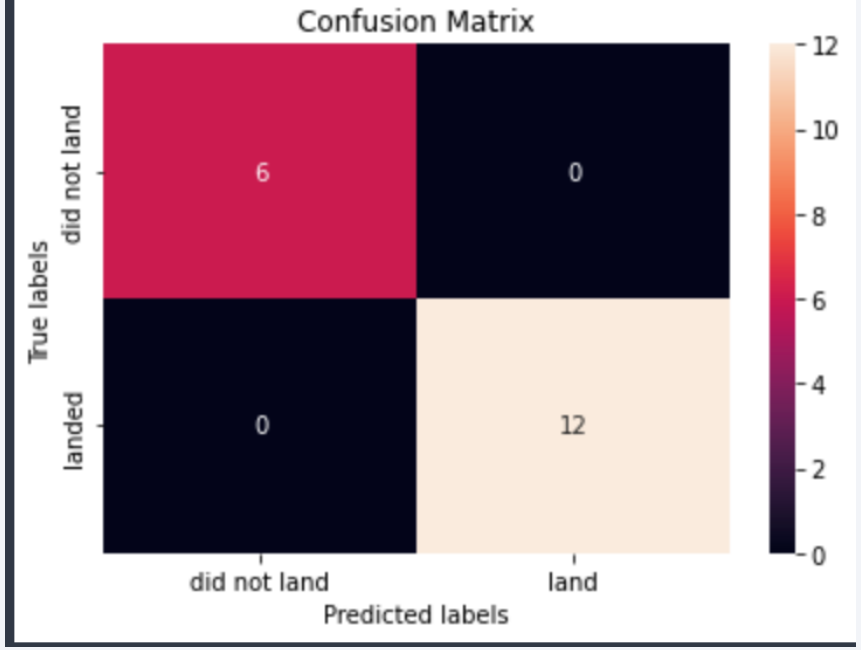


Classification Accuracy

Logistic Regression and KNN has the highest accuracy score

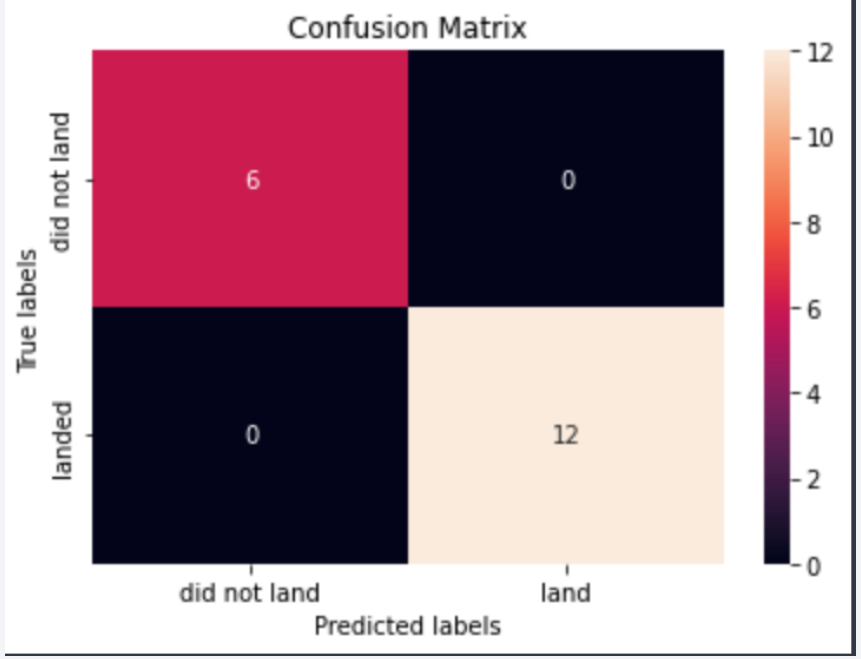
Confusion Matrix

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```
✓ 0.4s



KNN model perfectly predict true did not land 6 with 0 false, and predict 12 landed with zero false

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```
✓ 1.1s



Logistic Regression model perfectly predict true did not land 6 with 0 false, and predict 12 landed with zero false

Conclusions

- KSC LC-39A have the highest success rate across launch sites
- Payload mass between 2k Kg and 5.5k Kg seems have more succeed than others
- ES-L1, GEO, HEO, and SSO orbit type tend to have high success landing outcome
- Logistic Regression and KNN are suited model to predict the landing outcome
- 66,67% SpaceX Flights are Succeed
- SpaceX can reuse their rockets about 66,67%
- SpaceX rockets cost can be cheaper because 66,67% their first stage can be saved

Appendix

- All the codes can be found in my github repositories here https://github.com/Gengsu07/LEARNING_REPO

Thank you!