IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dzakiy Farid F
20/10/2023

Outline

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

Introduction

- Project background and context

For the past 5 years, companies are competing to make a space travel affordable. One of the successful company is SpaceX which can reduce the cost of rocket launch by a lot by reusing the first stage of the booster. We would like to study what factor affecting the success of the landing of the first stage and hence determine the price of each launch and whether SpaceX will reuse the first stage.

- Problems you want to find answers
  - Determine factor affecting the successful landing of the first stage
  - Determine which set of rocket features suitable for each condition

Section 1

Methodology

Methodology

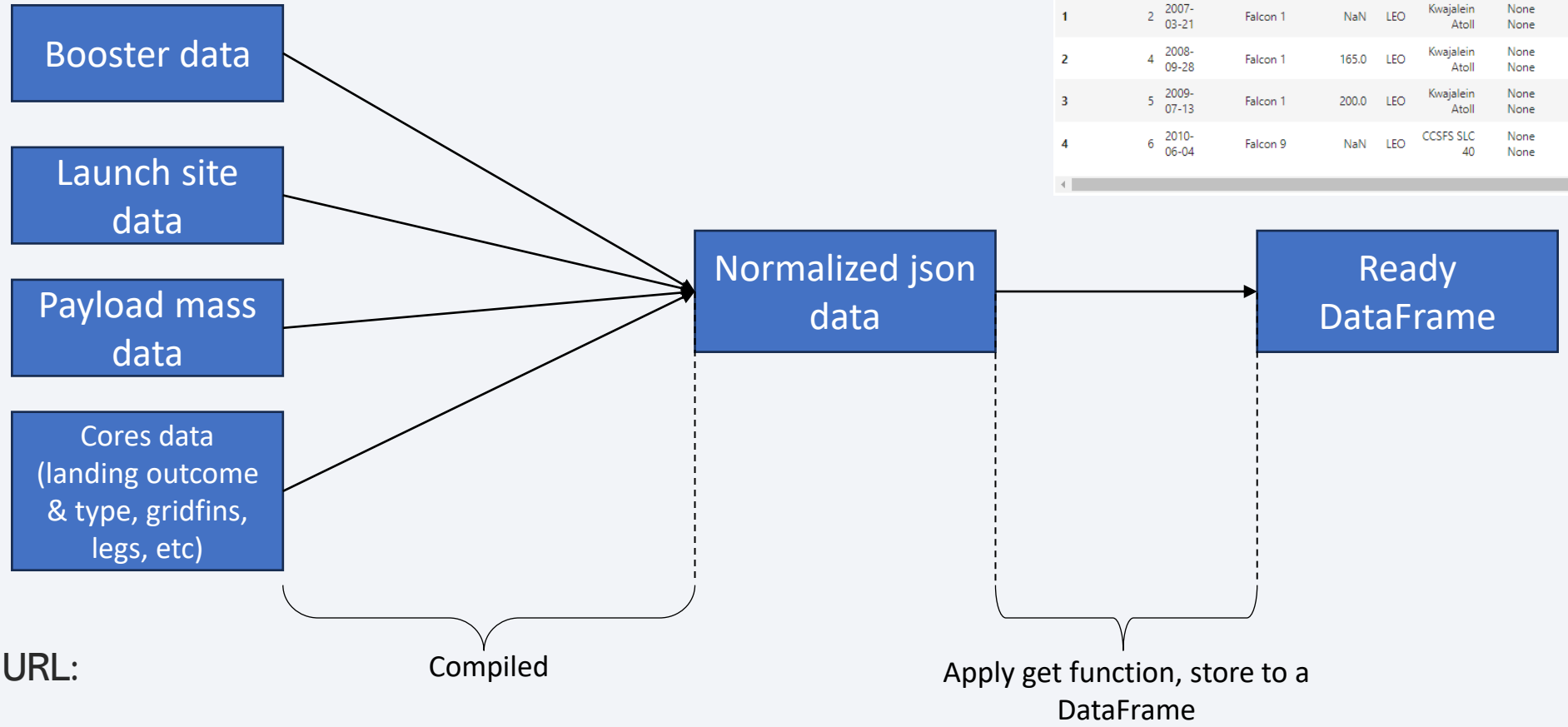
Executive Summary

- Data collection methodology:
  - Using the data by making a get request from SpaceX API and clean the requested data
- Perform data wrangling
  - Clean, process, and explore the 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
  - Make a machine learning model using scikit learn module and use gridsearch to find the best parameter

Data Collection

- Data is collected by making arequest to the SpaceX API with URL = https://api.spacexdata.com/v4/launches/past
- Several get function is define to collect information using identification numbers in the launch data
- Data is then cleaned (eliminating the NaN value) and organized (filter the data which only include necessary features)

Data Collection – SpaceX API



	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0

- Github URL:

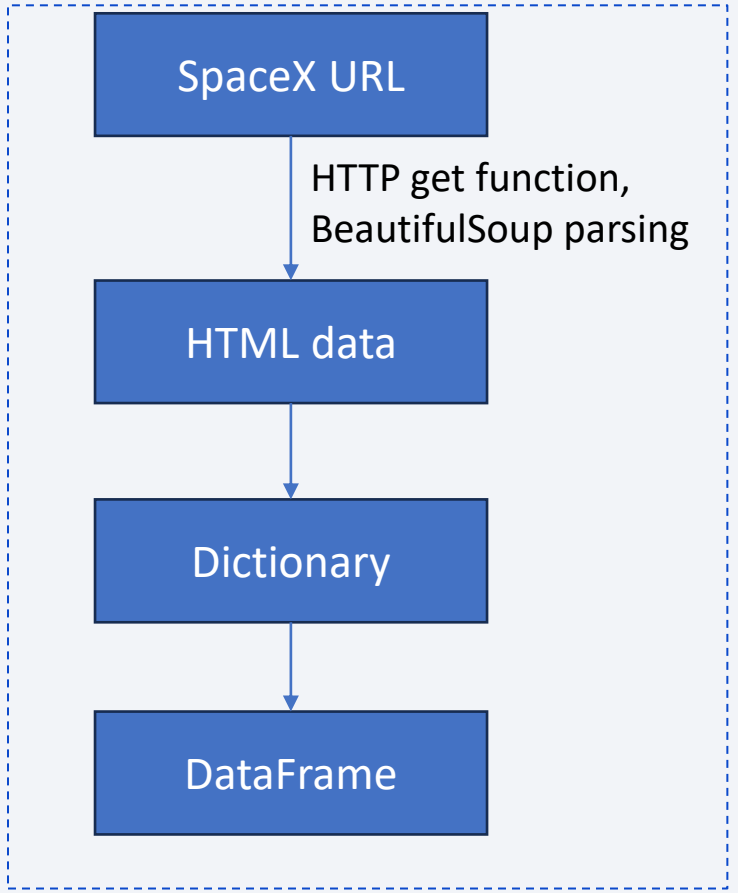
https://github.com/Dzakiy-csg/learning_data/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1. Use Python BeautifulSoup package to web scrape HTML
2. Parse the data from the table
3. Create loop function to obtain information about booster, payload mass, date, etc. and store it into the dictionary
4. Create a Pandas DataFrame from the dictionary

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Wrangling

- Use the data from web scraping
- Eliminate NaN object
- Calculate number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the landing outcome
- Assign label: 1 for successful landing, 0 for unsuccessful landing

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	

→

FlightNumber	0.000000
Date	0.000000
BoosterVersion	0.000000
PayloadMass	0.000000
Orbit	0.000000
LaunchSite	0.000000
Outcome	0.000000
Flights	0.000000
GridFins	0.000000
Reused	0.000000
Legs	0.000000
LandingPad	28.888889
Block	0.000000
ReusedCount	0.000000
Serial	0.000000
Longitude	0.000000
Latitude	0.000000
dtype: float64	

NaN object eliminated

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

https://github.com/Dzakiy-csg/learning_data/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13
Name: LaunchSite, dtype: int64	

No. of launches on each site

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	

No. and occurrence of each orbit

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	

occurence of mission outcome

EDA with Data Visualization

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

- Data visualization is used in order to make the analysis easier
- Use several library such as matplotlib and seaborn
- Scatter plot is used to show the relationship between numerical and categorical variable:
  - FlightNumber vs PayloadMass
  - FlightNumber vs LaunchSite
  - LaunchSite vs PayloadMass
  - FlightNumber vs Orbit
  - PayloadMass vs Orbit
- Bar plot is used to compare between categorical or discreet variable:
  - Orbit and their success rate
- Line plot is used to emphasize changes in values for one variable (plotted on the vertical axis) for continuous values of a second variable (plotted on the horizontal):
  - Launch success yearly trend

EDA with SQL

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/jupyter-labs-eda-sql-edx_sqllite.ipynb

- SQL queries is performed to obtain data from dataset in IBM cloud and gain several insight
- From the data we show:

1. Names of the unique launch sites in the space mission
2. 5 records where launch sites begin with the string 'KSC'
3. total payload mass carried by boosters launched by NASA (CRS)
4. average payload mass carried by booster version F9 v1.1
5. List the date where the succesful landing outcome in drone ship was achieved
6. Names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. names of the booster_versions which have carried the maximum payload mass
9. List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
10. Count of Failure (drone ship) landing outcomes between the date 2010-06-04 and 2017-03-20

Build an Interactive Map with Folium

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

- Marker is added to indicate several location in the map using their longitude and latitude such as:
  - NASA Johnson Space Center as a start location
  - All the launch sites (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E)
- Each marker is contained with all of their information:
  - Label on the map and popup text
  - Successful and unsuccessful landing marked with green and red marker in each launch site and simplified using mark cluster
  - Lines to show the distance between launch sites (VAFB SLC-4E for example) to their proximities (coastline, closest city, railway, highway)

Build a Dashboard with Plotly Dash

Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/spacex_dash_app.py

- Add a launch site drop-down for all launch site
  - User can select the launch site
- Add a call back function to render a pie chart of a success rate
  - Pie chart show the percentage of success rate from all site if ‘all site’ is selected from the drop-down and show the percentage of success and failed rate if one of the launch site is selected
- Add a range slider for the payload mass
  - User can select the range of the payload mass
- Add a call back function to render a scatter plot between class vs payload mass by the booster version

Predictive Analysis (Classification)

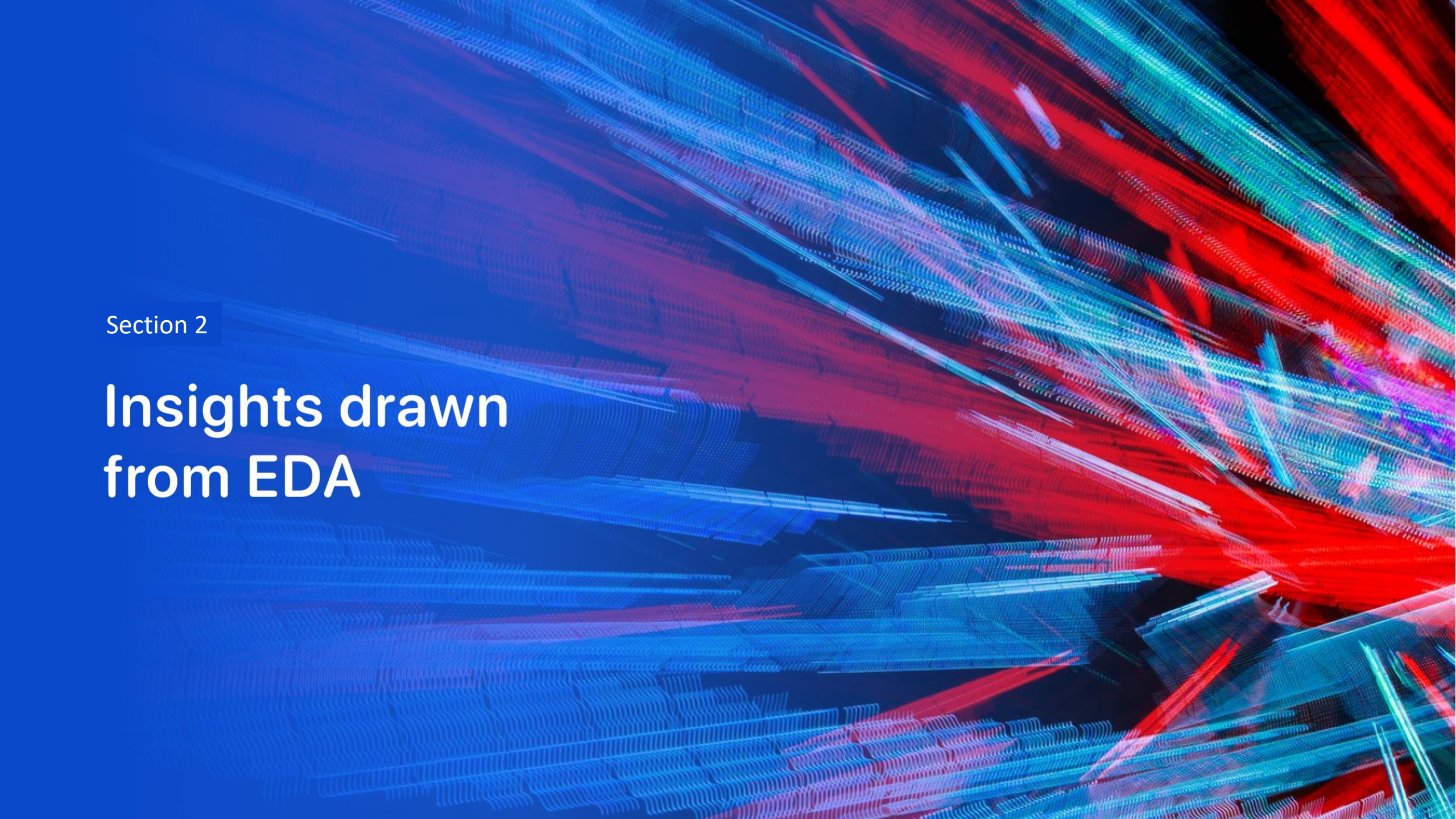
Github URL:

https://github.com/Dzakiy-csg/learning_data/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

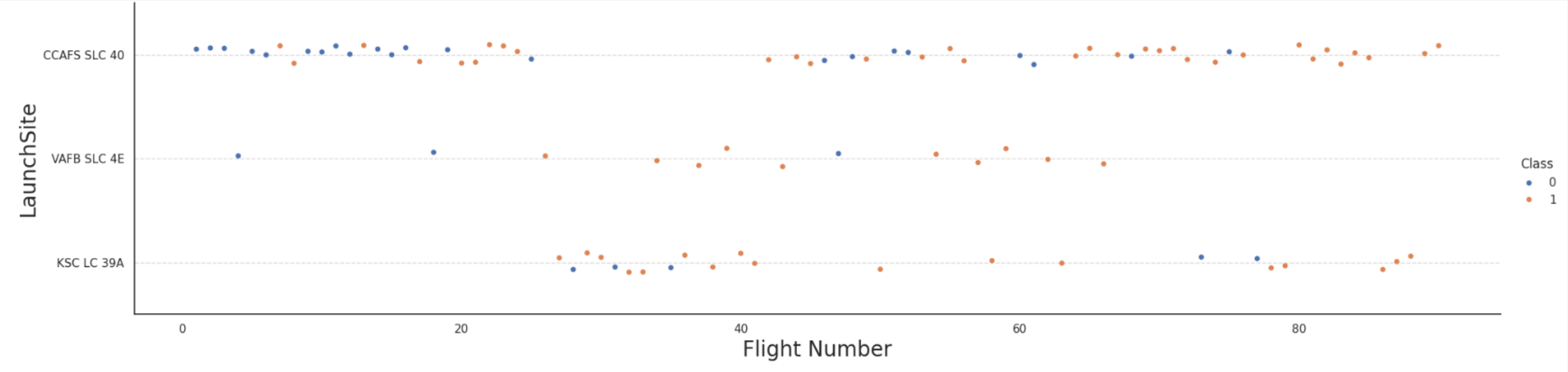
- Data preparation
    1. Load the dataset. Apply OneHotEncoder to categorical column (orbit, launch site, landing pad, serial)
    2. Separate the data set into independent (feature of the rocket) and dependent (class) variable as X and Y. Apply a preprocessing.StandardScaler() to the X variable in order to standardize the features of a dataset
    3. Split the data into training and test data with test size 20%
- Model training, testing, and evaluation
    1. Create GridSearchCV() to search for the best combination of hyperparameters for a model
    2. Apply GridSearchCV on ML models (logistic regression, support vector machine, decision tree classifier, k nearest neighbors)
    3. Calculate the accuracy of the model using test data using .score() method
    4. Plot the confusion matrix

Results

- Exploratory data analysis results
  - Data is obtained, cleaned, processed, and plotted
- Interactive analytics demo in screenshots
  - Folium map
  - Ploty dash app
- Predictive analysis results
  - 4 classifiers were used

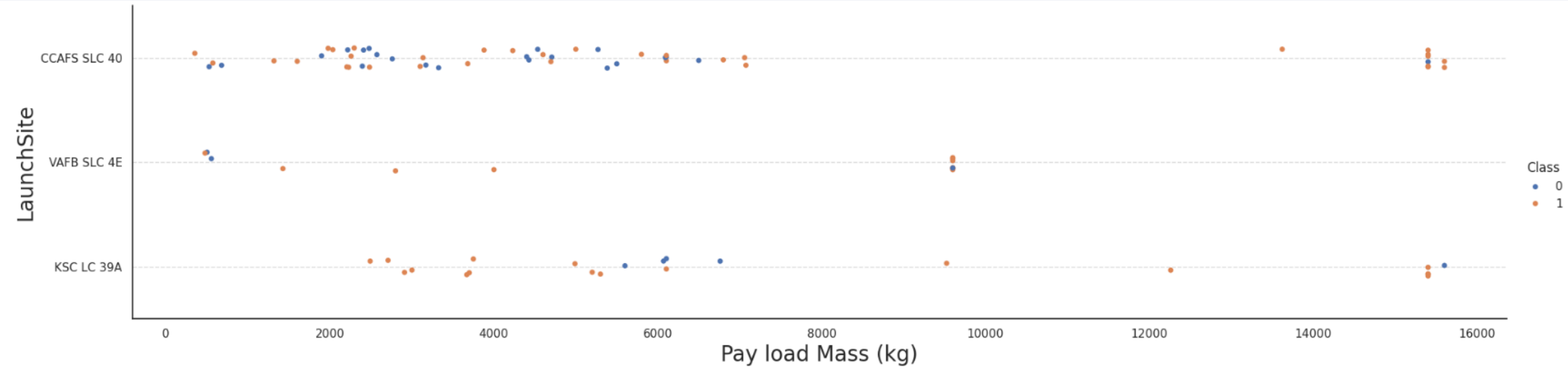

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- Task 1: (blue = failed, yellow = success)
  - Most of the early flight before flight number 20 is failed
  - Most of the launches are done in CCAFS SLC 40
  - CCAFS SLC 40 has the lowest success rate
  - From all sites, the success rate of the launch increases over time

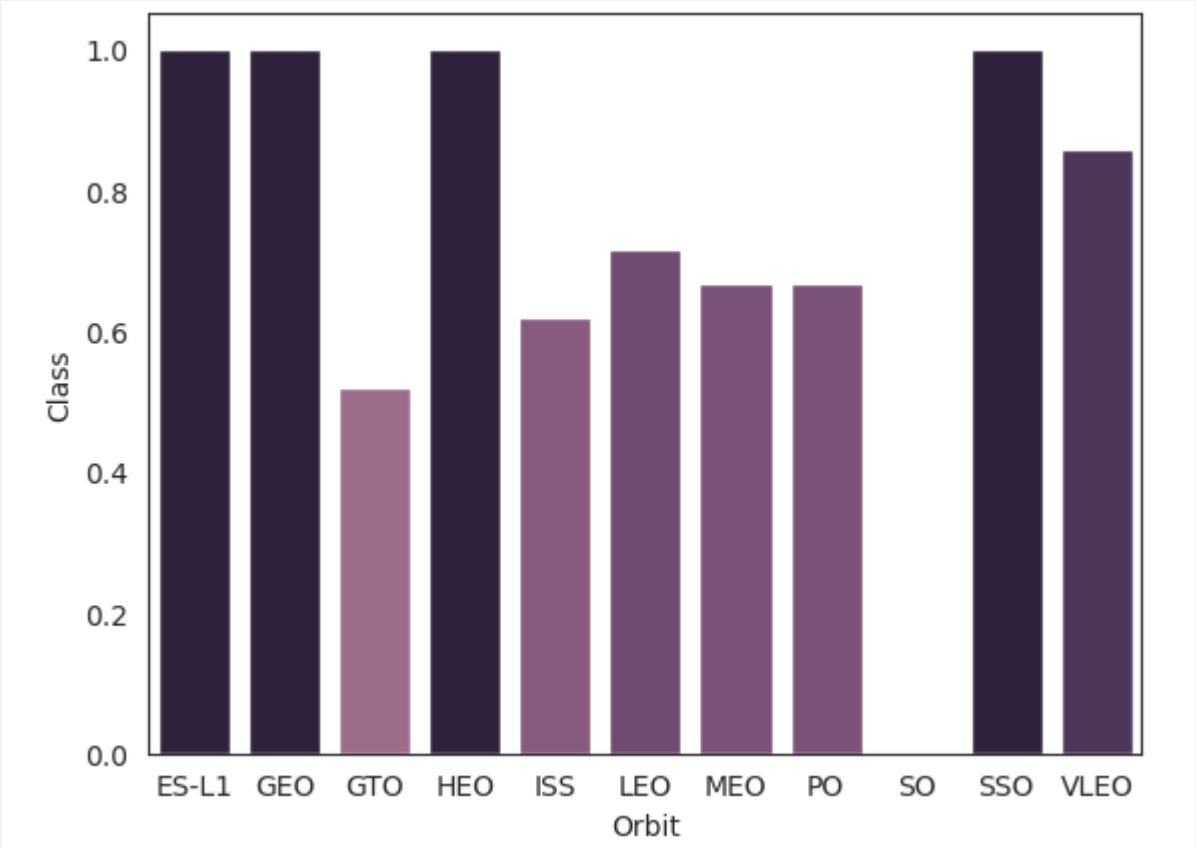
Payload vs. Launch Site



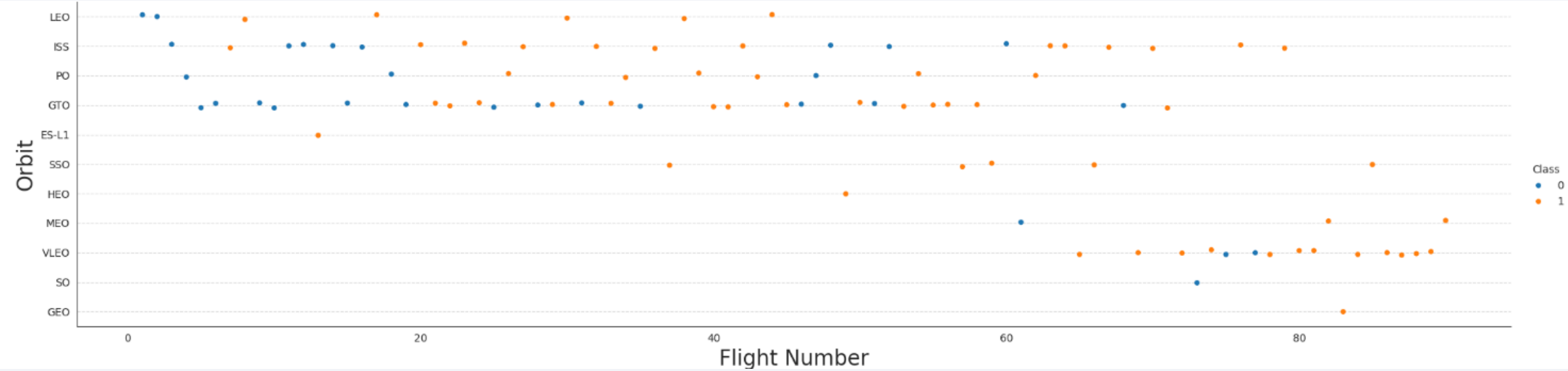
- Task 2: (blue = failed, yellow = success)
  - Most of data point with payload more than 7000 kg were successsful
  - No payload more than 10000 kg was used in VAFB SLC 4E
  - CCAFS SLC 40 and KSC LC 39A were attempted to use maximum payload

Success Rate vs. Orbit Type

- Task 3
  - Orbit that has high sucess rate are ES-L1, GEO, HEO, SSO
  - The lowest success rate is SO

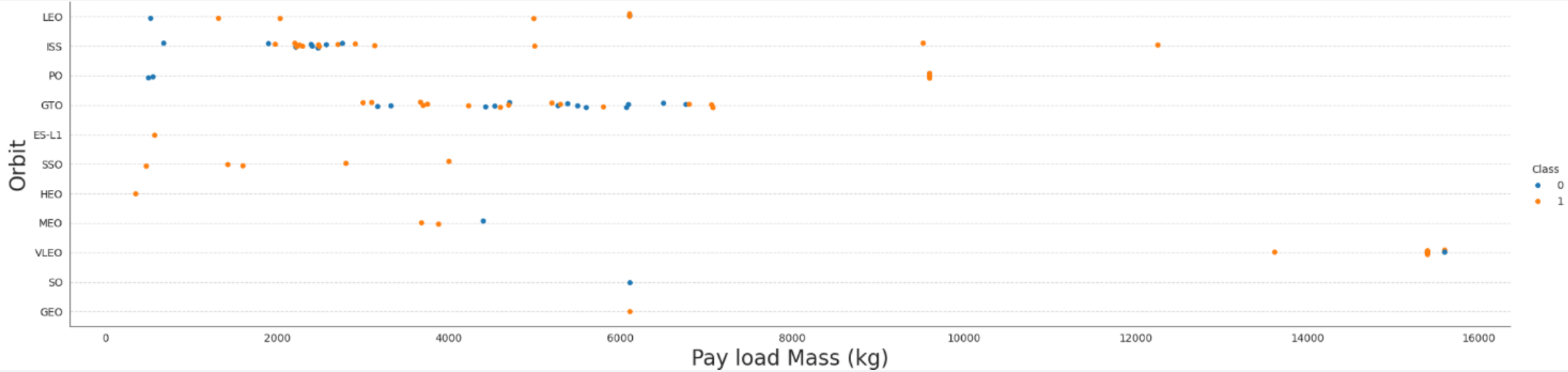


Flight Number vs. Orbit Type



- Task 4: (blue = failed, yellow = success)
  - There is a positive correlation between flight number and the success in LEO
  - There is no correlation between flight number and the success in GTO
  - VLEO type orbit only has high flight number and has high success rate

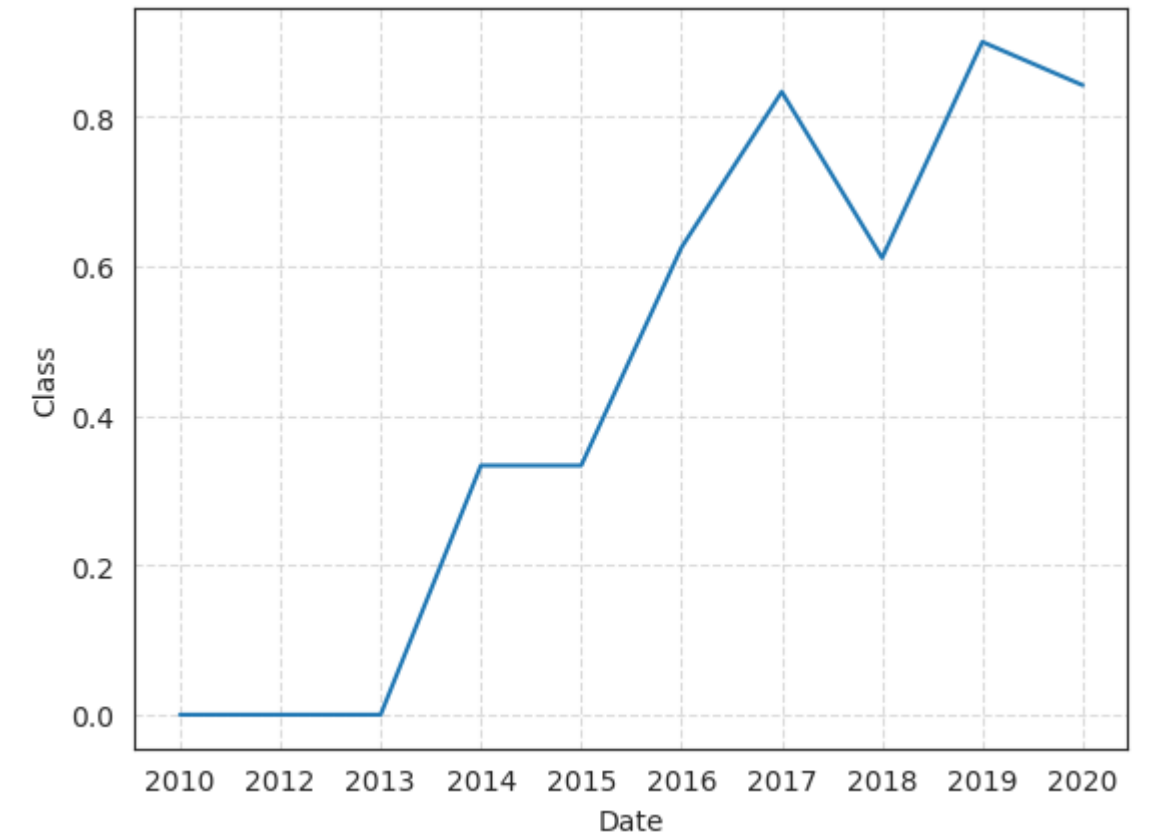
Payload vs. Orbit Type



- Task 5: (blue = failed, yellow = success)
  - In GTO type orbit, we cannot see the effect of payload mass to the successful landing or negative landing
  - Orbit type ISS, PO, and VLEO have high successful rate at with heavy load (>9000 kg)
  - Orbit type LEO, ES-L1, SSO, HEO, and MEO have high successful rate at with low load (<7500 kg)

Launch Success Yearly Trend

- Task 6
  - Overall, for all orbit, the success rate is increase over time from 2013 to 2017
  - At the first 3 years, there is no increase in success rate
  - The success rate decrease from 2017 to 2018 and from 2019 to 2020



All Launch Site Names

- SQL query and result:

```
In [9]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

Out[9]:

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

Display all the launch site name used in SpaceX dataset

Launch Site Names Begin with 'KSC'

- SQL query and result:

```
In [11]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'KSC%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Display 5 launch site names begin with ‘KSC’

Total Payload Mass

- SQL query and result:

```
In [13]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
Out[13]: SUM(PAYLOAD_MASS__KG_)
         45596
```

Shows the total payload masscarried by booster launched by NASA (CRS)

Average Payload Mass by F9 v1.1

- SQL query and result:

```
In [14]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
Out[14]: AVG(PAYLOAD_MASS__KG_)
         2928.4
```

Shows the average payload mass carried by booster version F9 v1.1

Date where the succesful landing outcome in drone ship was acheived

- SQL query and result:

List the date where the succesful landing outcome in drone ship was acheived

```
In [15]: %sql SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)'
 * sqlite:///my_data1.db
Done.
```

Out[15]:

Date
2016-08-04
2016-06-05
2016-05-27
2016-08-14
2017-01-14
2017-03-30
2017-06-23
2017-06-25
2017-08-24
2017-09-10
2017-11-10
2017-10-30
2018-04-18
2018-11-05

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL query and result:

```
[17]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
 * sqlite:///my_data1.db
Done.
```

[17]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Shows the list of boosters which have success in ground pad and have payload mass between 4000 and 6000. It appears only three booster succeed

Total Number of Successful and Failure Mission Outcomes

- SQL query and result:

```
In [19]: %sql SELECT Mission_Outcome, COUNT(1) FROM SPACEXTABLE GROUP BY Mission_Outcome
 * sqlite:///my_data1.db
Done.
```

Out[19]:

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

Shows number of successful and failure mission outcomes

Boosters Carried Maximum Payload

- SQL query and result:

The list shows the booster version which carry the maximum payload

```
In [20]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

Out[20]:

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

2017 Launch Records

- SQL query and result:

```
[21]: %sql SELECT substr(Date,6,2) AS 'Month', Customer, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' \
      AND Date BETWEEN '2017-01-01' AND '2017-12-31'

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

[21]:

Month	Customer	Landing_Outcome	Booster_Version	Launch_Site
02	NASA (CRS)	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
01	NRO	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
03	NASA (CRS)	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	NASA (CRS)	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
07	U.S. Air Force	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	NASA (CRS)	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

List of records that displays months, successful landing outcome in ground pad, booster version and launch site in year 2017

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

- SQL query and result:

```
[22]: %sql SELECT RANK() OVER(ORDER BY counter DESC) AS rank, Landing_Outcome, counter FROM (SELECT Landing_Outcome, COUNT(1) AS 'counter' \
FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ).
 * sqlite:///my_data1.db
Done.
```

[22]:

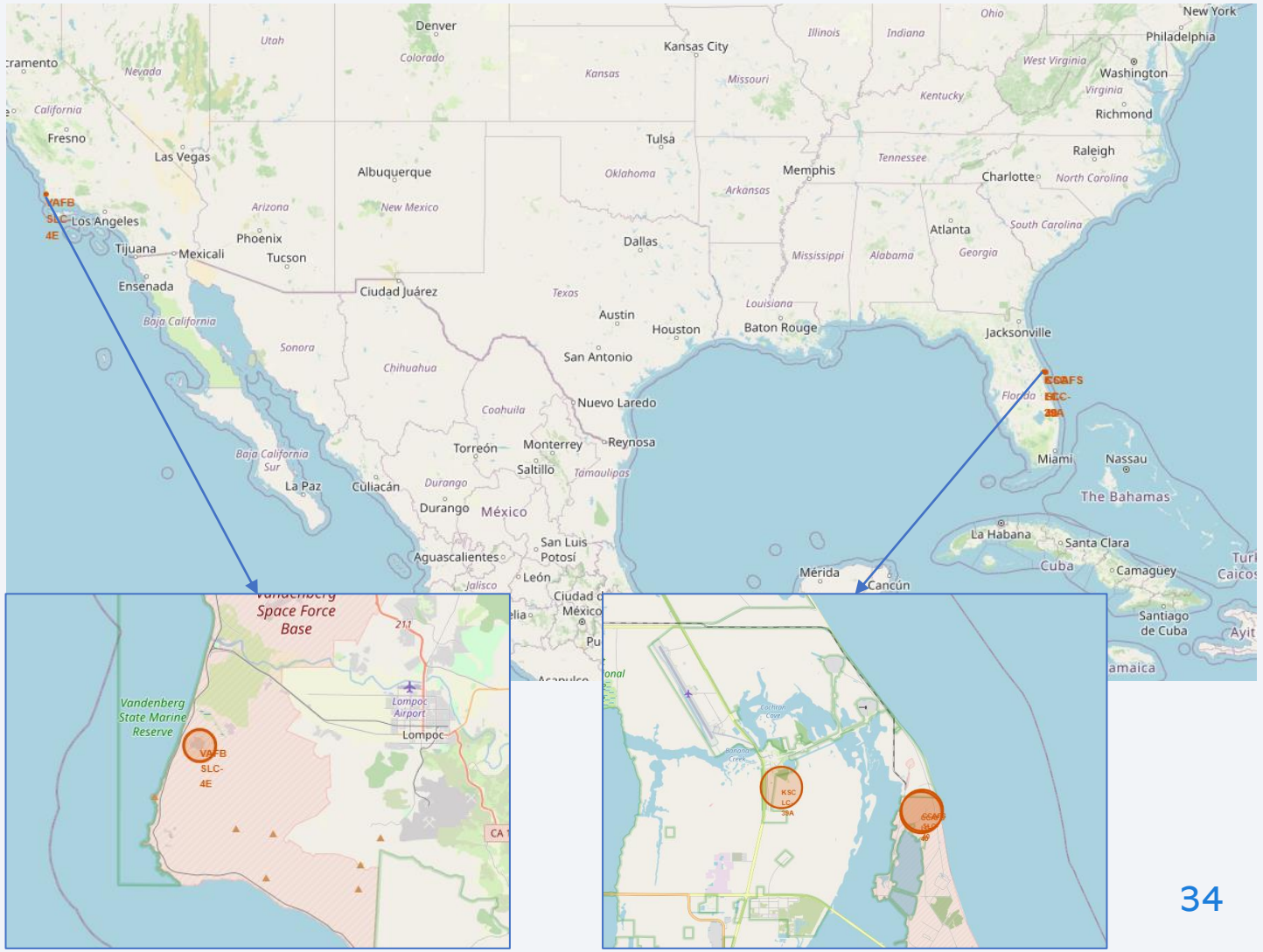
rank	Landing_Outcome	counter
1	No attempt	10
2	Failure (drone ship)	5
2	Success (drone ship)	5
2	Success (ground pad)	5
5	Controlled (ocean)	3
6	Uncontrolled (ocean)	2
7	Failure (parachute)	1
7	Precluded (drone ship)	1

List of the count of Failure (drone ship) landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order


Section 3
Launch Sites
Proximities Analysis

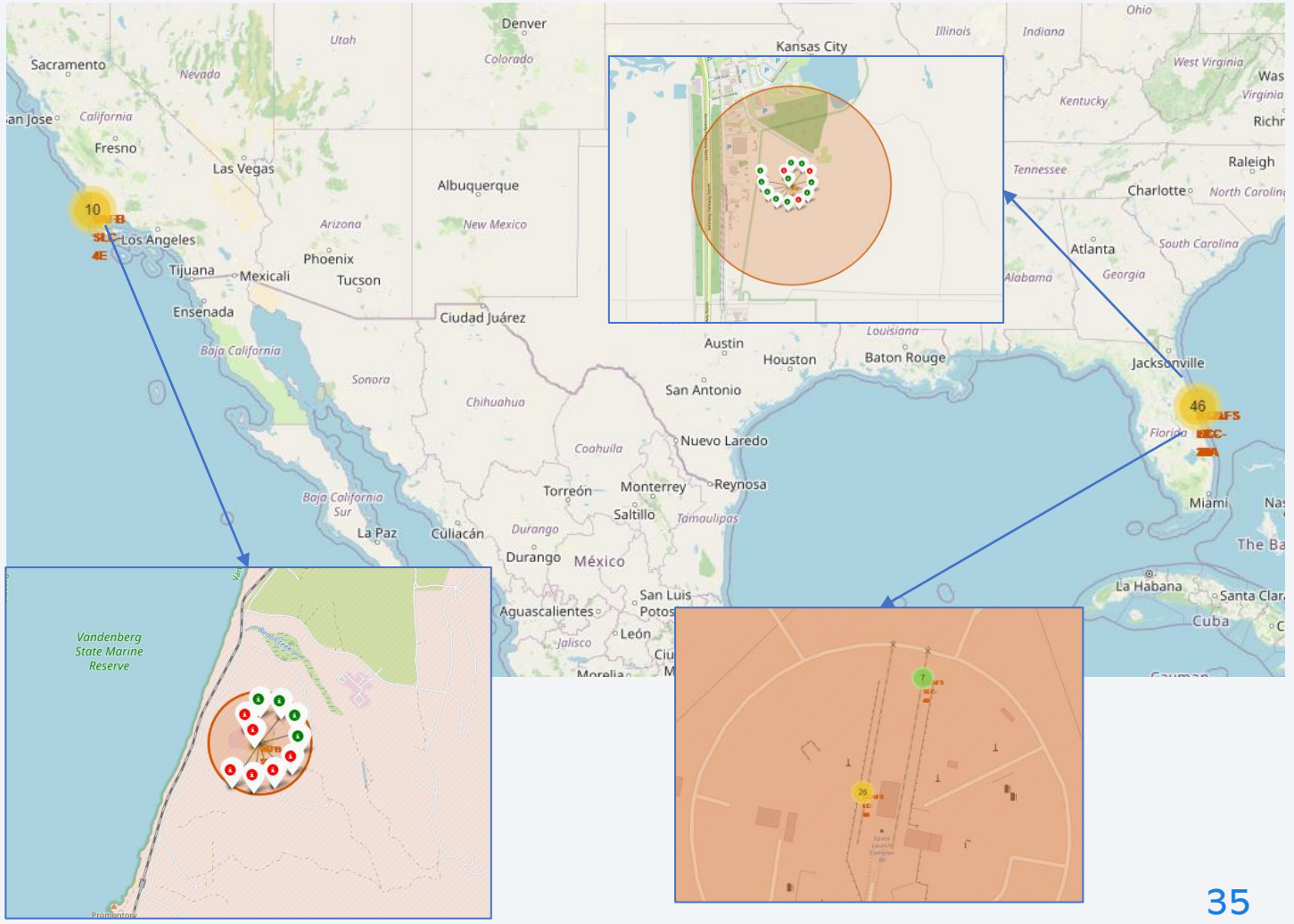
Launch sites

- Launch sites is marked with orange circle
- We can observed that all launch sites are proximity to the equator line which means they are all using the of earth rotation as an advantage in order to make additional centrifugal force which make the launch easier hence lower the use of fuel and cost
- All of the launch sites also proximity to the coast so that in case of failure, the debris will fall into water



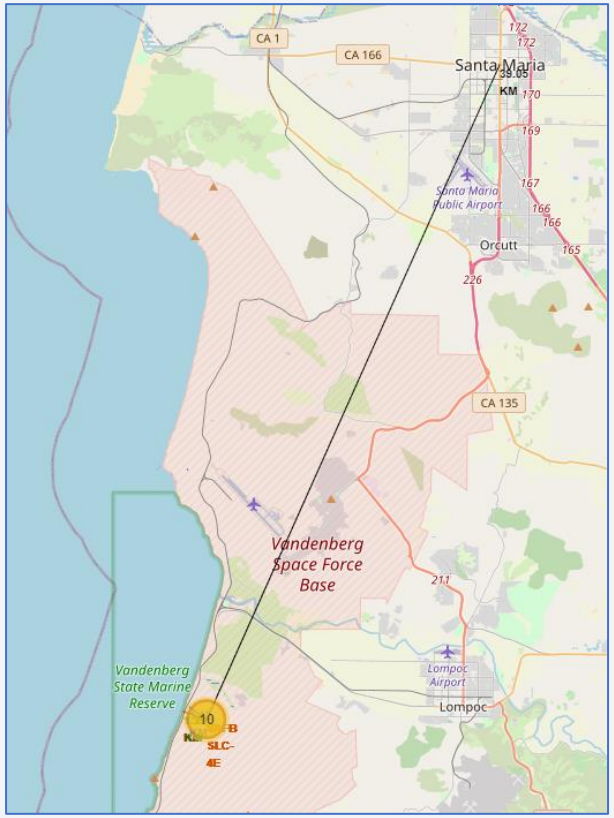
Success/failed launches marker

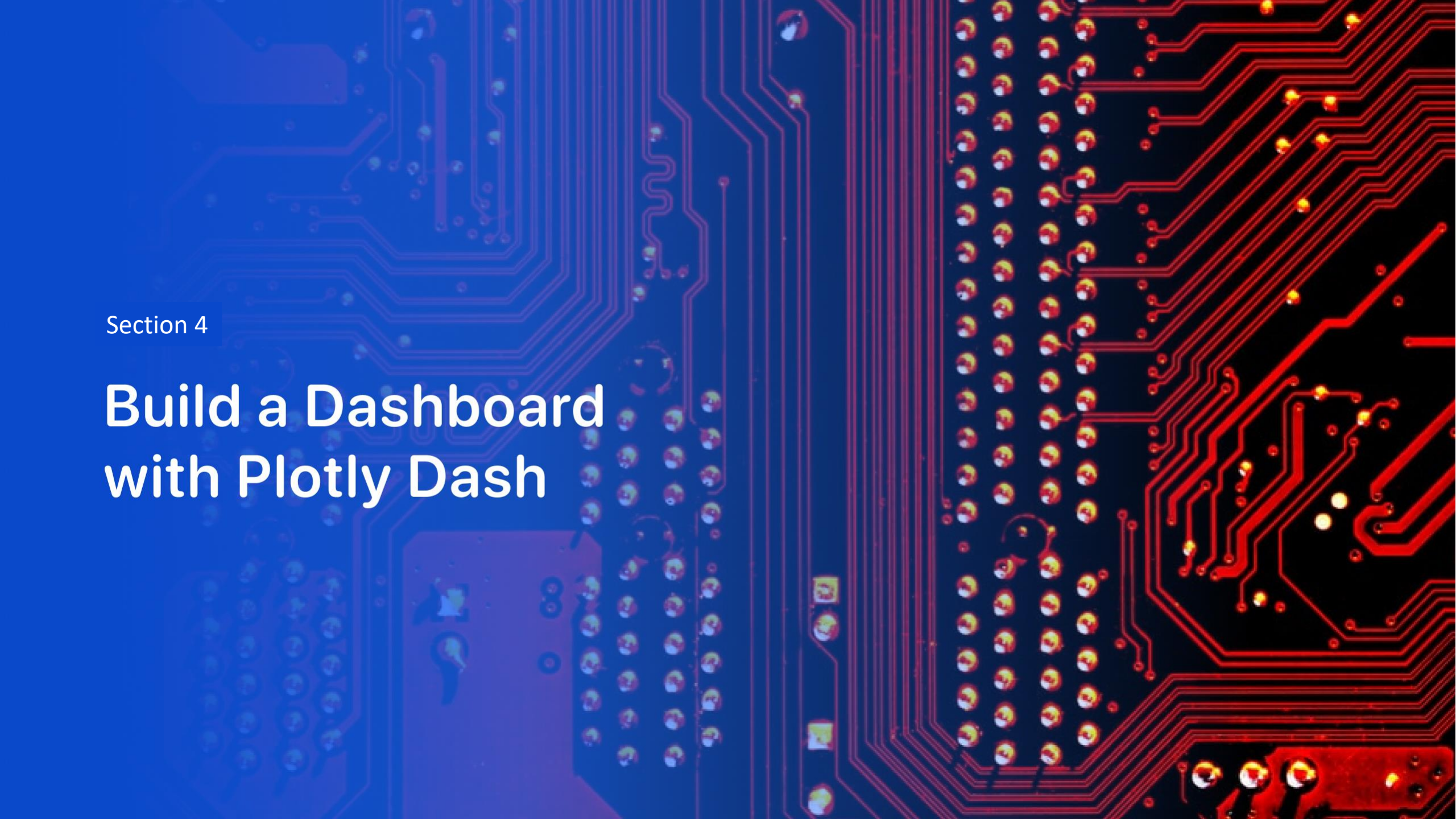
- The launch outcome is marked for each launch sites
  - Green marker = successful launches
  - Red marker = failed launches
- The marker is also clustered to simplify area that has many markers having the same coordinate



Launch Site and Its Proximities

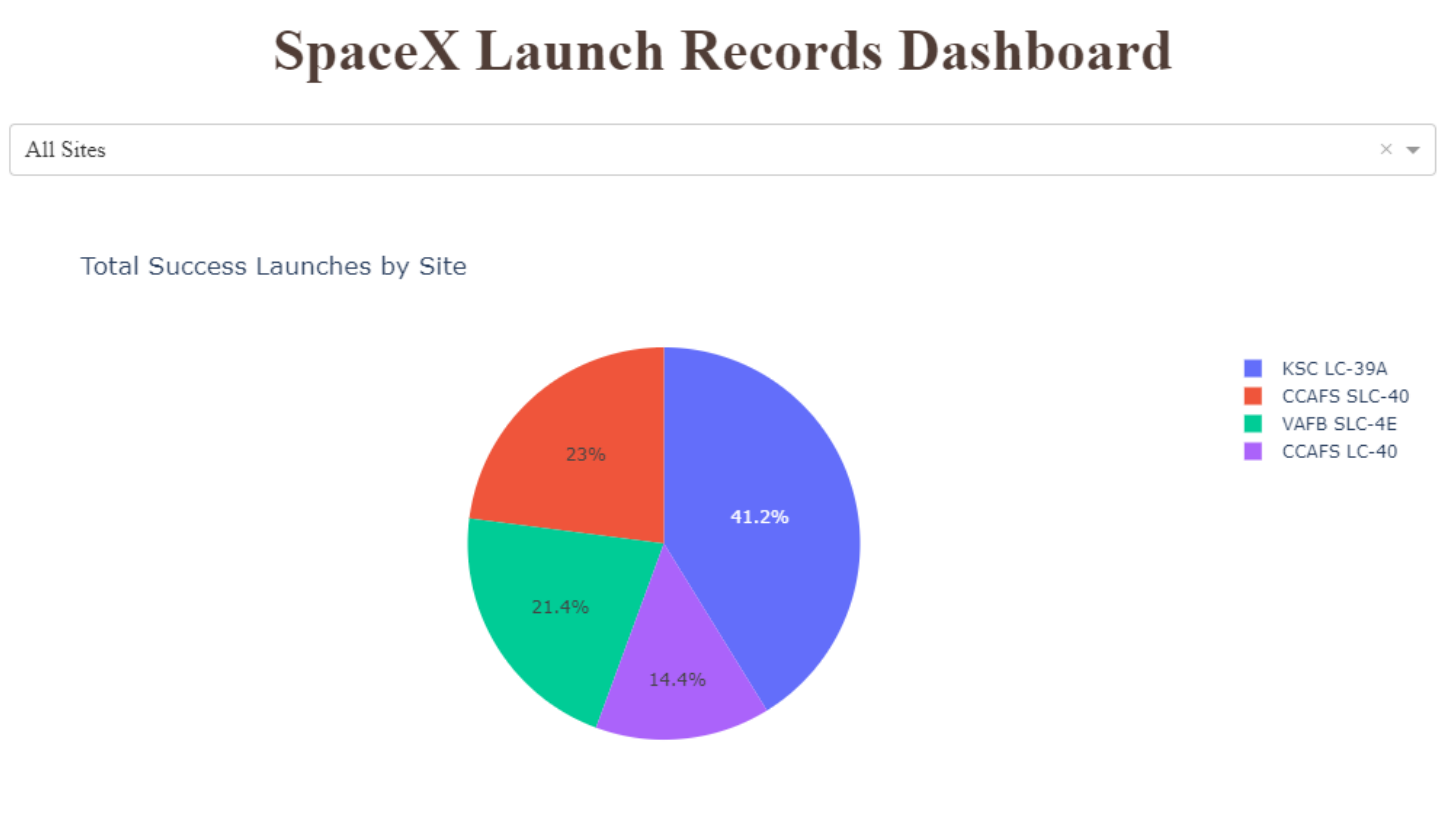
- Proximities of launch site are also marked with line with different color and their distance to the launch site (VAFB SLC-4E) is calculated:
  - Coastline (red) = 1.35 km
  - Railway (green) = 1.25 km
  - Highway (blue) = 1.12 km
  - City (black) = 39.05 km
- The launch sites are close to railways, highways, and coastline but far from the nearest city (Santa Maria)




Section 4
Build a Dashboard with Plotly Dash

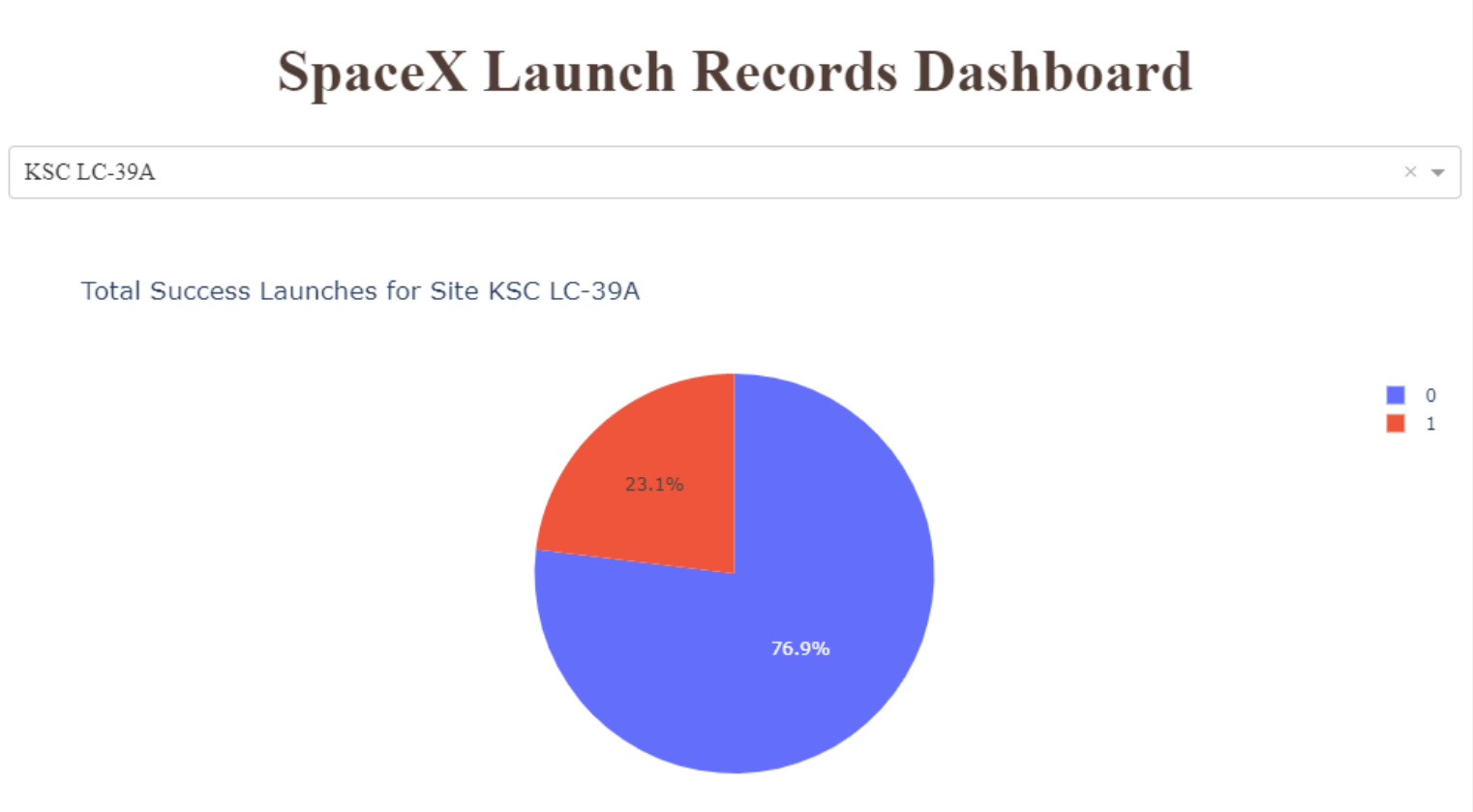
Pie Chart of All Launch Site

- From the chart, KSC LC-39A has the highest success rate (41.2%) compared to other launch site



Pie Chart of KSC LC-39A

- KSC LC-39A also has the highest total success launches (76.9%) with 10 success launches out of 13



All Launch Site Scatter Plot

- At payload range 0-5000 kg, FT and B4 have the best performance compared to other booster
- The success rate at payload range 5000-10000 kg is not high with 3 success out of 11
- Overall, low payload mass has higher success rate compared to high payload mass

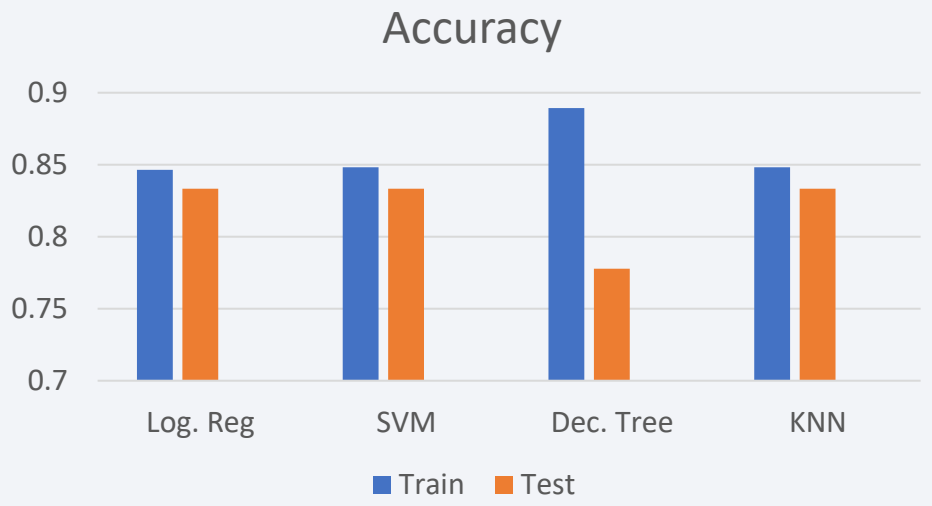
Section 5

Predictive Analysis (Classification)

Classification Accuracy

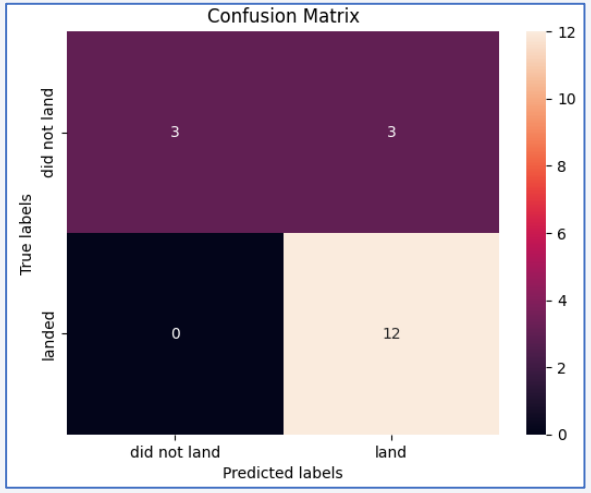
- All machine learning models give the same test accuracy except decision tree which lower than the others

Classifier	Train accuracy	Test accuracy
Log. Reg	0.8464	0.8334
SVM	0.8482	0.8334
Dec. Tree	0.8893	0.7778
KNN	0.8482	0.8334

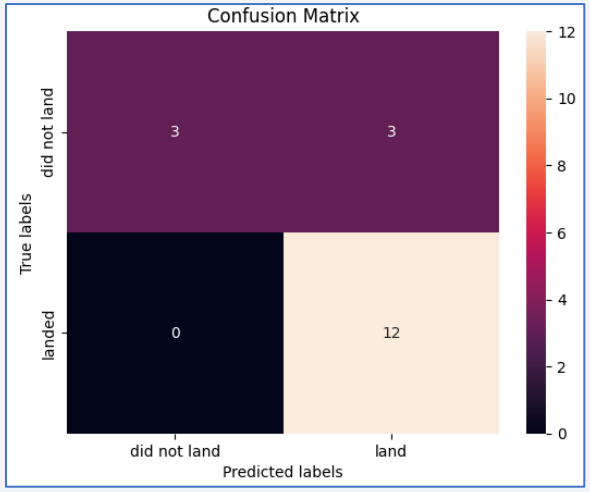


Confusion Matrix

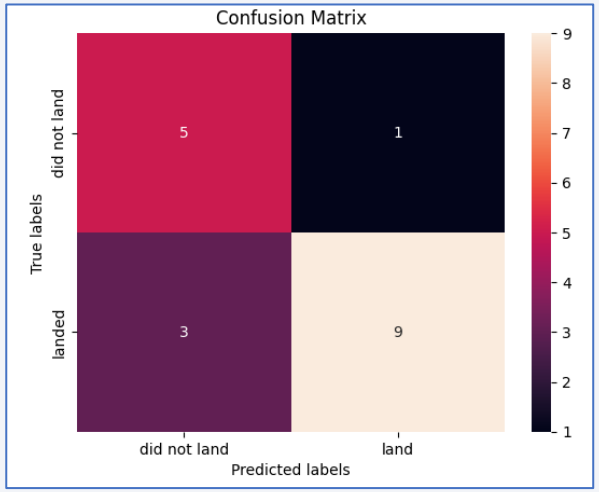
- From all modes logistic regression, support vector machine, and K-nearest neighbors have similar confusion matrix with 12 true positive, 3 true negative, and 3 false negative



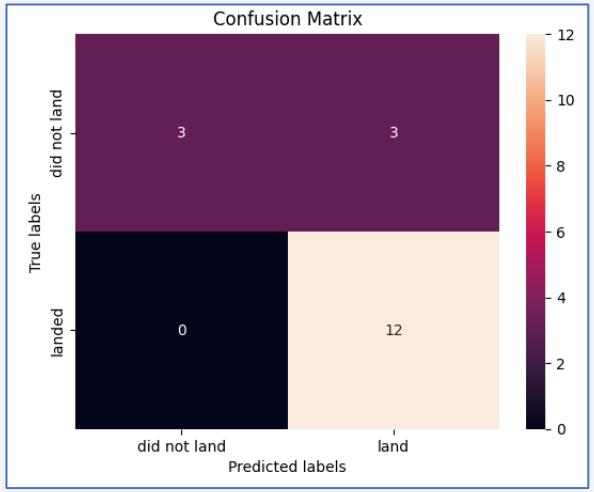
Logistic regression



Support vector machine



Decision tree classifier



K-nearest neighbors

Future works

- Other machine learning models may be considered to be used such as MLP classifier, xgboost, random forest, and ensemble classifier
- Use F1_score to assess the model's ability to balance precision and recall. F1-score is particularly useful in situations where there is an imbalance between the classes in the dataset
- Use a larger dataset so that it can eliminates the possibilities of a models having the same accuracy. In this dataset, there are only 18 data points on test dataset.

Conclusions

- Data is collected by making a get request from SpaceX API, cleaned and processed
- Over the year, the successful rate increased
- Ploty dash is useful to see the proximity of launch site and determine whether it is fulfilled the required condition
- Low payload mass has higher success rate compared to high payload mass
- Classification tools such as Logistic regression, support vector machine, and K-nearest neighbors perform equally best for predicting whether the launch successful based on the feature

Thank you!