Winning Space Race with Data Science

Hiro

10 March, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Fetching data by Web Scraping and SpaceX API
  - Exploratory Data Analysis (EDA) with data Wrangling and data Visualization
  - Interactive Map with Folium
  - Dashboards with Plotly
  - Prediction by ML algorithm
- Summary of all results
  - Experiencing different ways to collect data by public resources
  - The process of EDA would help a developer find valuable data by analyzing public data
  - Testing different ML algorithms is one of good ways to efficiently predict target value with higher percentage.

Introduction

- Project background and context
  - Goal: Predicting if the Falcon 9 first stage will successfully land.
  - Background: SpaceX 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 SpaceX 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 SpaceX for a rocket launch.
- Problems you want to find answers
  - What factors lead the result of successful or failed landing?
  - What conditions affect that SpaceX achieves the best landing success rate ?

Section 1

Methodology

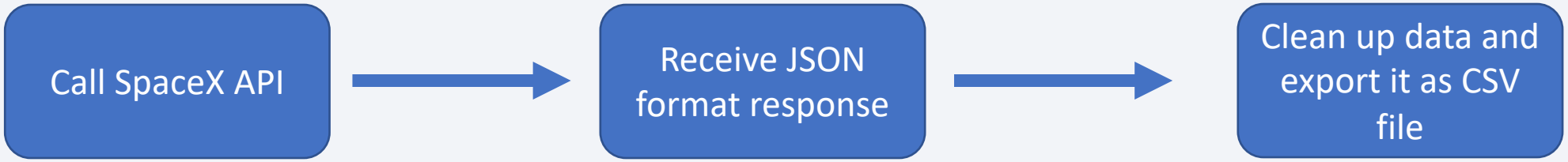
Methodology

Executive Summary

- Data collection methodology:
  - Web Scraping on Wikipedia and SpaceX API
- Perform data wrangling
  - Drop unnecessary columns
  - One Hot Encoding
- 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

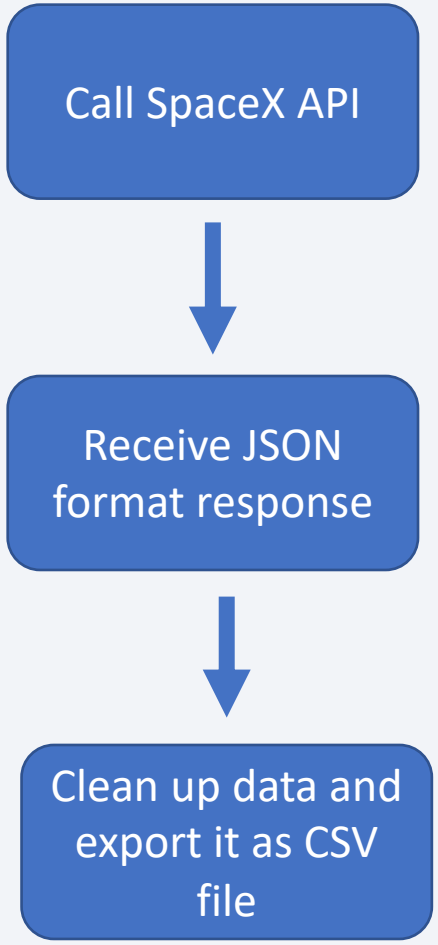
- Data sets are collected by Web Scraping on Wikipedia and SpaceX API
- SpaceX API



- Web Scraping on Wikipedia

Data Collection – SpaceX API

- After calling SpaceX API, receive JSON format data. Then, the data is reorganized by provided functions. The data is modified to Dataframe format, and it was exported as CSV format.

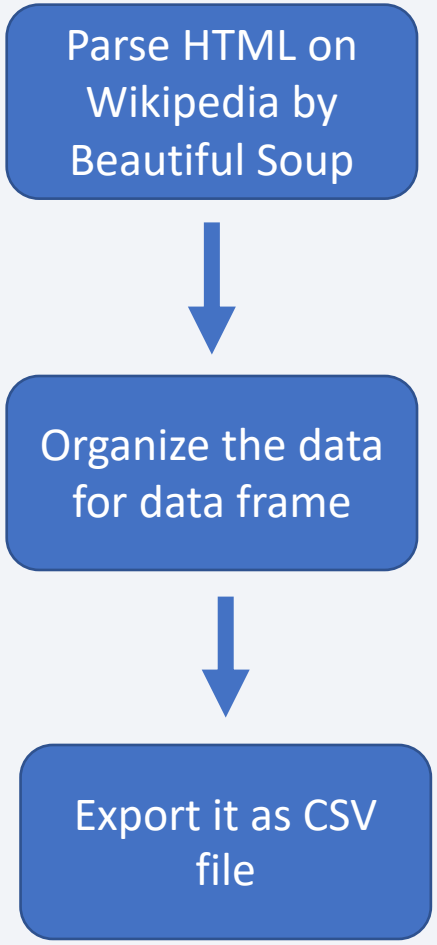


https://github.com/test-hiro42/final_data_science/blob/main/week1_1_jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Beautiful Soup fetch HTML structures from one URL. The data contains table data. Therefore, Beautifl Soup detect table header <th> tag on the HTML. Finally, create data frame after organizing to dictionary data type.

https://github.com/test-hiro42/final_data_science/blob/main/week1_2_jupyter-labs-webscraping.ipynb

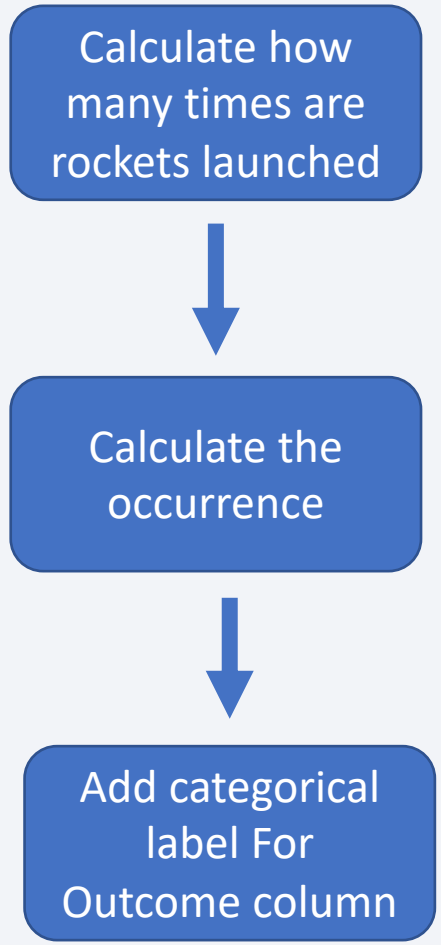


Data Wrangling

- Need to Transform ‘Outcome ‘ from a string variable to a categorical variable.
- For example, 1 means successful landing and 0 means is failure one.
- Before creating categorical data, we had to investigate which string values in ‘Outcome’ column mean success or failure.

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad.`True ASDS` means the mission outcome was successfully landed to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

https://github.com/test-hiro42/final_data_science/blob/main/week1_3_labs-jupyter-spacex-Data%20wrangling.ipynb



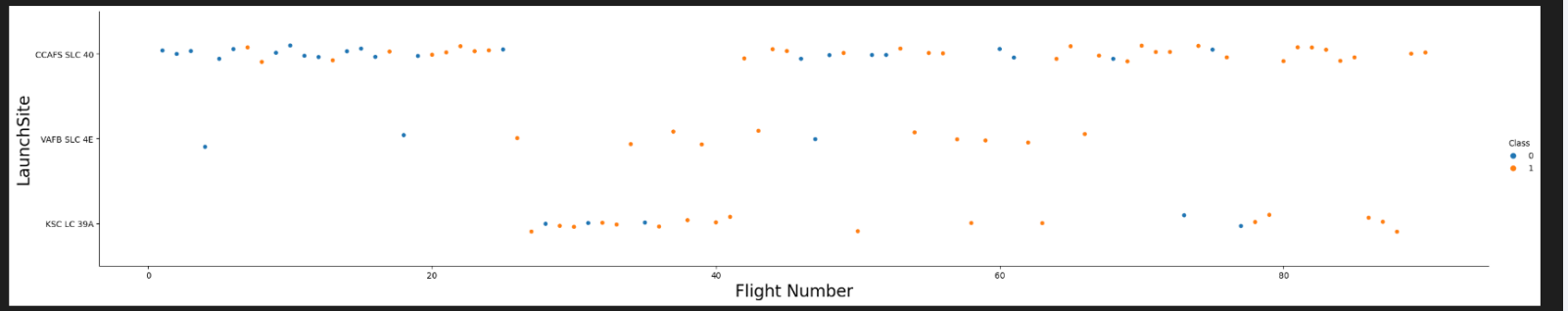
EDA with Data Visualization

- Scatter plot and Bar plot were used to visualize for analyzing relationship between Flight Number and Launch Site.

https://github.com/test-hiro42/final_data_science/blob/main/week2_1_jupyter-labs-eda-sql-coursera_sqllite.ipynb

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be t
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

Python



+ Code + Markdown

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1. •
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and 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 records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015.
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

https://github.com/test-hiro42/final_data_science/blob/main/week2_2_jupyter-labs-eda-dataviz.ipynb

Build an Interactive Map with Folium

- Red circle on NASA Johnson Space Center's coordinate with label displays its name.
- Red circles on each launch site coordinates with label display launch site name.
- Green pin means successful landing, and Red one is for unsuccessful landing.
- Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them.
- These objects are created to get better understanding relationships among those data. Developers can easily analyze all launch sites, their surroundings and the number of successful and unsuccessful landings on map view.

https://github.com/test-hiro42/final_data_science/blob/main/week3_1_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

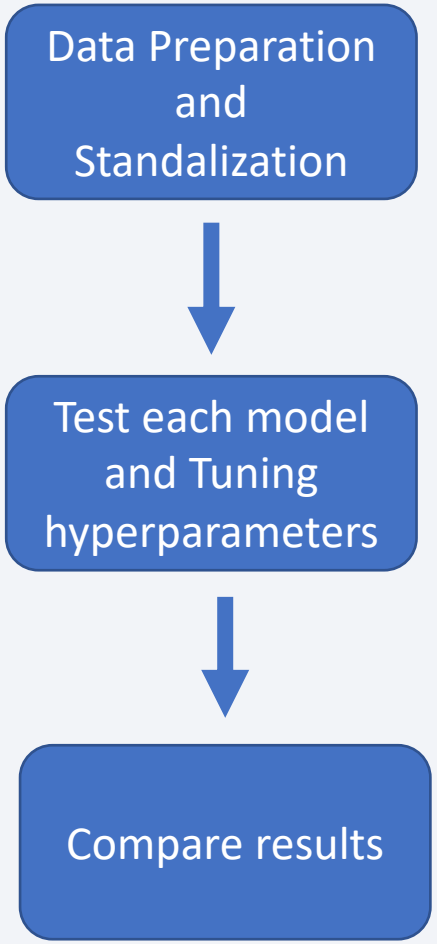
- Success launch rate on each site
- Plot payload data on scatter graph

- I thought I could come up with idea by visualizing percentage about success rate on each site
- In my guess, payload mass might affect success launch, so I plot the data on scatter graph

https://github.com/test-hiro42/final_data_science/blob/main/week4_1_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

- Data preparation
  - Scandalize data
  - Split data into train and test data.
- Model preparation
  - Selection of machine learning algorithms
  - Put parameters for each algorithm to GridSearch Cross Validation
- Model evaluation
  - Check Confusion Matrix
  - Tuning hyperparameters for each type of model
- Model comparison
  - Comparison of models according to their accuracy

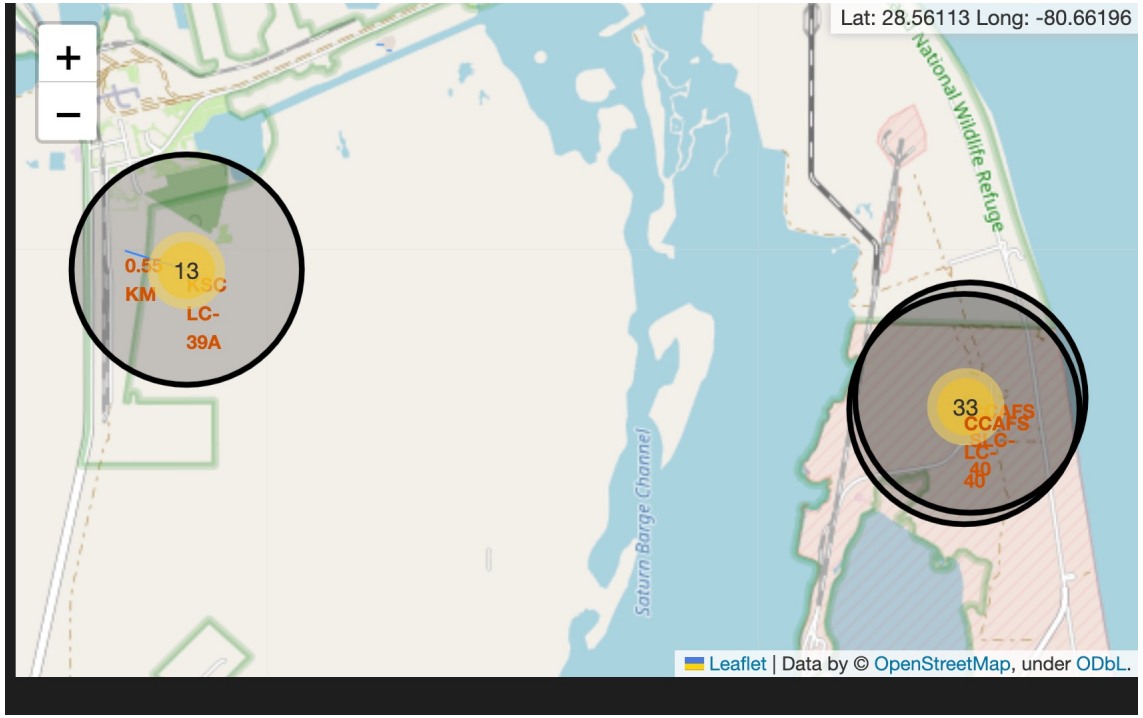


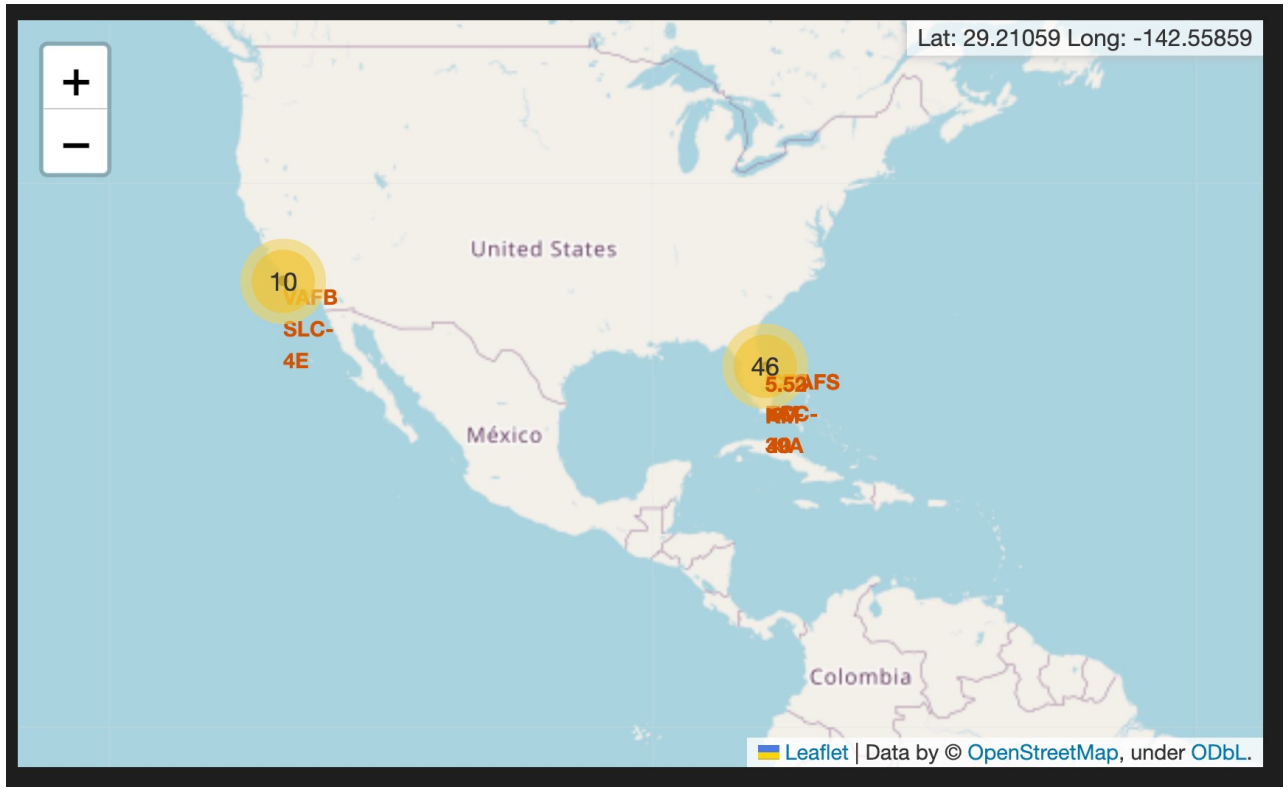
https://github.com/test-hiro42/final_data_science/blob/main/week4_1_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results:

- Exploratory data analysis results
  - Space X uses 4 different launch sites
  - The first launches were done to Space X itself and NASA
  - The first success landing was in 2015 fiver year after the first launch
  - Many Falcon 9 booster versions successfully
  - Almost 100% of mission outcomes were successful

Results

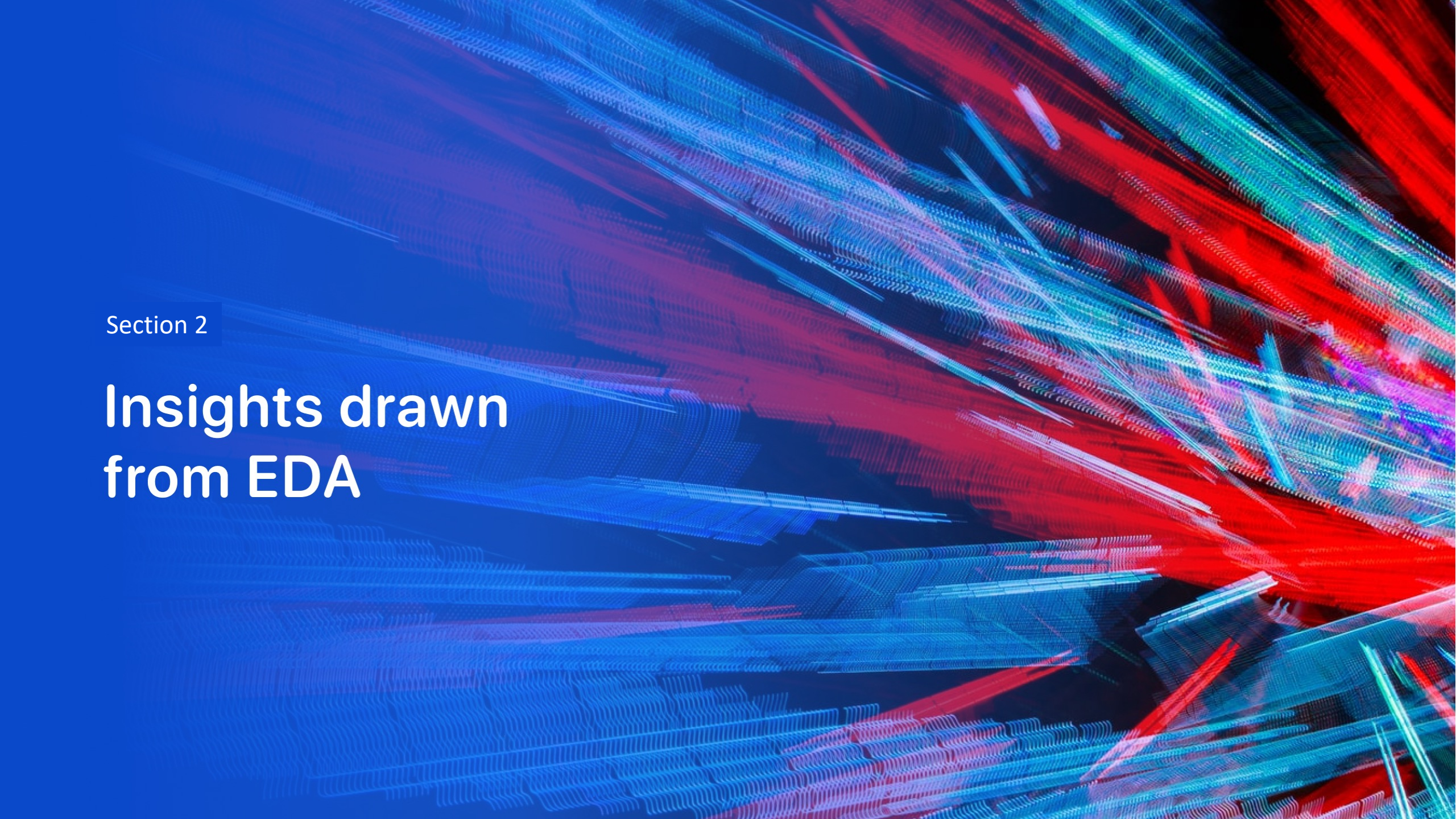

Lat: 28.56113 Long: -80.66196
13
0.55 KM
KSC LC-39A
33
CCAFS LC-40
National Wildlife Refuge
Saturn Barge Channel
Leaflet | Data by © OpenStreetMap, under ODbL.


Lat: 29.21059 Long: -142.55859
United States
10
VAFB SLC-4E
46
México
Colombia
Leaflet | Data by © OpenStreetMap, under ODbL.

Results

- Decision Tree Classifier would be the best model in the following list.

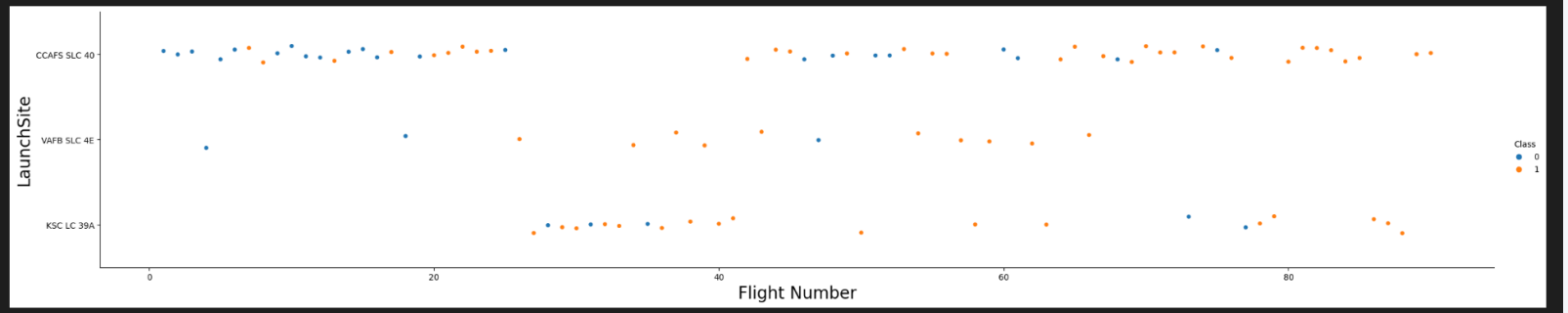
Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.875	0.83333
KNN	0.84821	0.83333


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

```python
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

Python

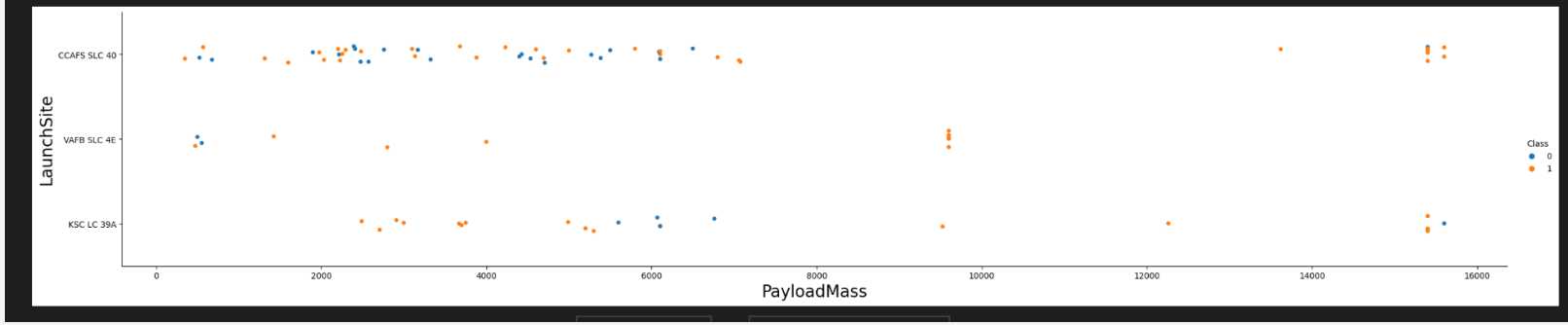


- CCAF5 SLC 40 successfully launched at many times
- Successful outcomes were increased by increasing flight number

Payload vs. Launch Site

```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the la
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect=5)
plt.xlabel("PayloadMass", fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

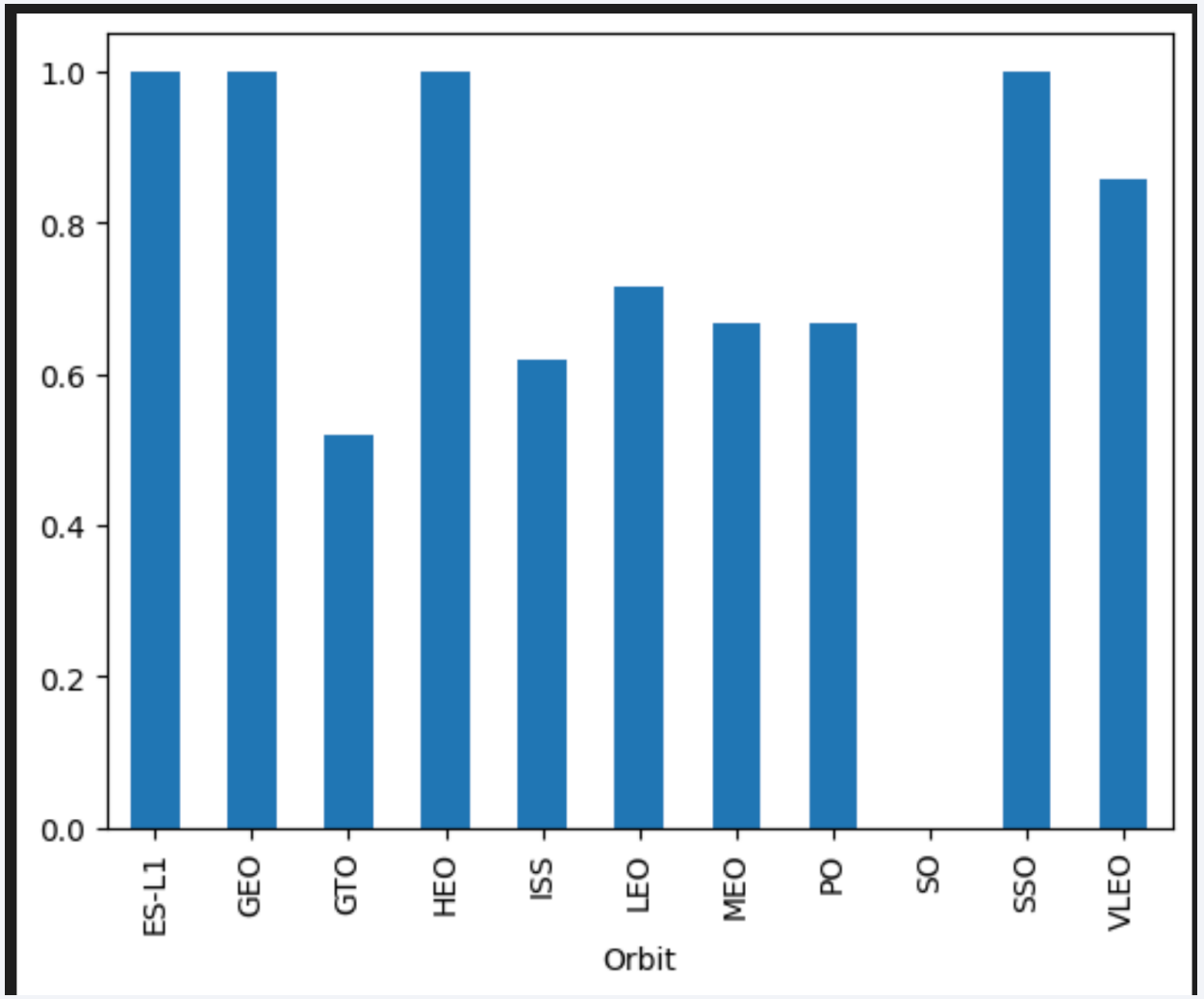
Python



- Payload over 9000 kg has many successful outcomes
- VAFB SLC 4E does not has payload over 10000 kg

Success Rate vs. Orbit Type

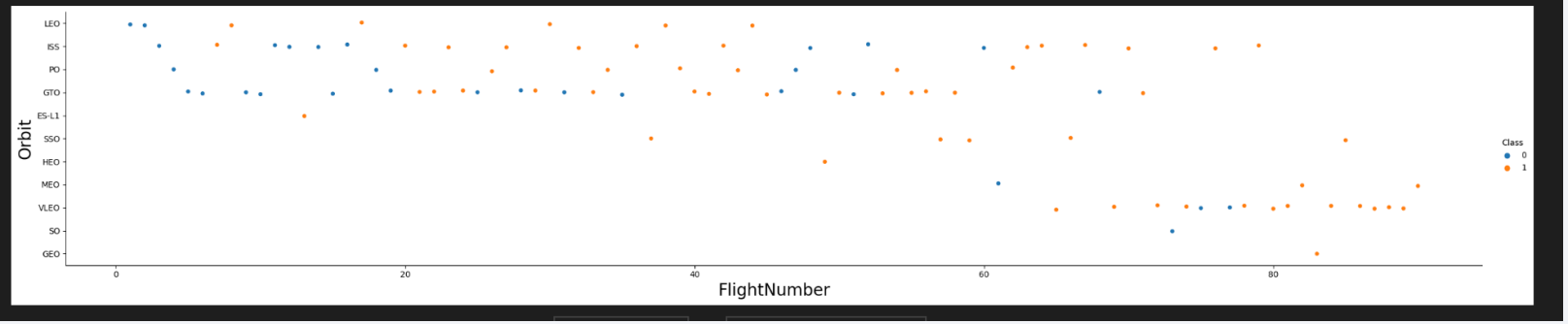
- ES-L1, GEO, HEO and SSO have higher rate almost 1.0



Flight Number vs. Orbit Type

```python
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, a
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect=5)
plt.xlabel("FlightNumber", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Python

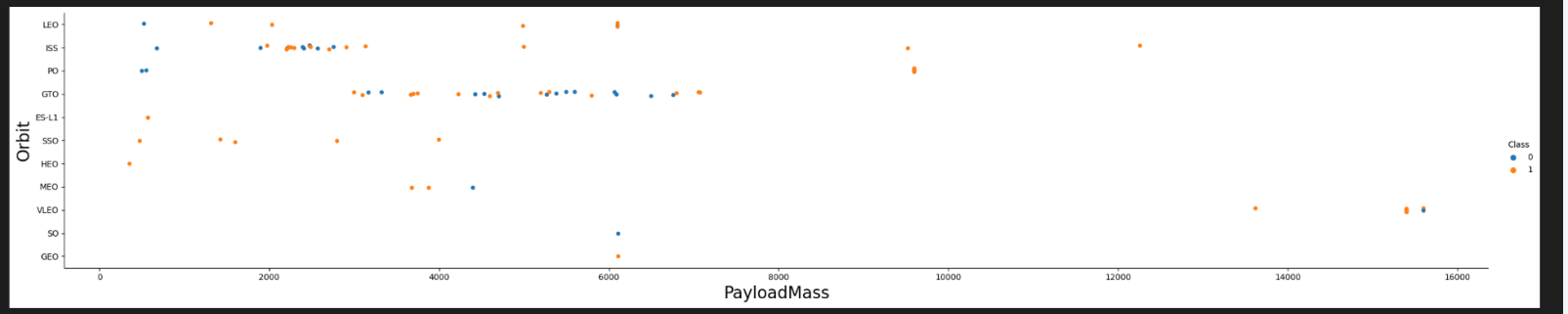


When increasing flight number, each orbit has higher successful landing

Payload vs. Orbit Type

```python
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hu
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect=5)
plt.xlabel("PayloadMass", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

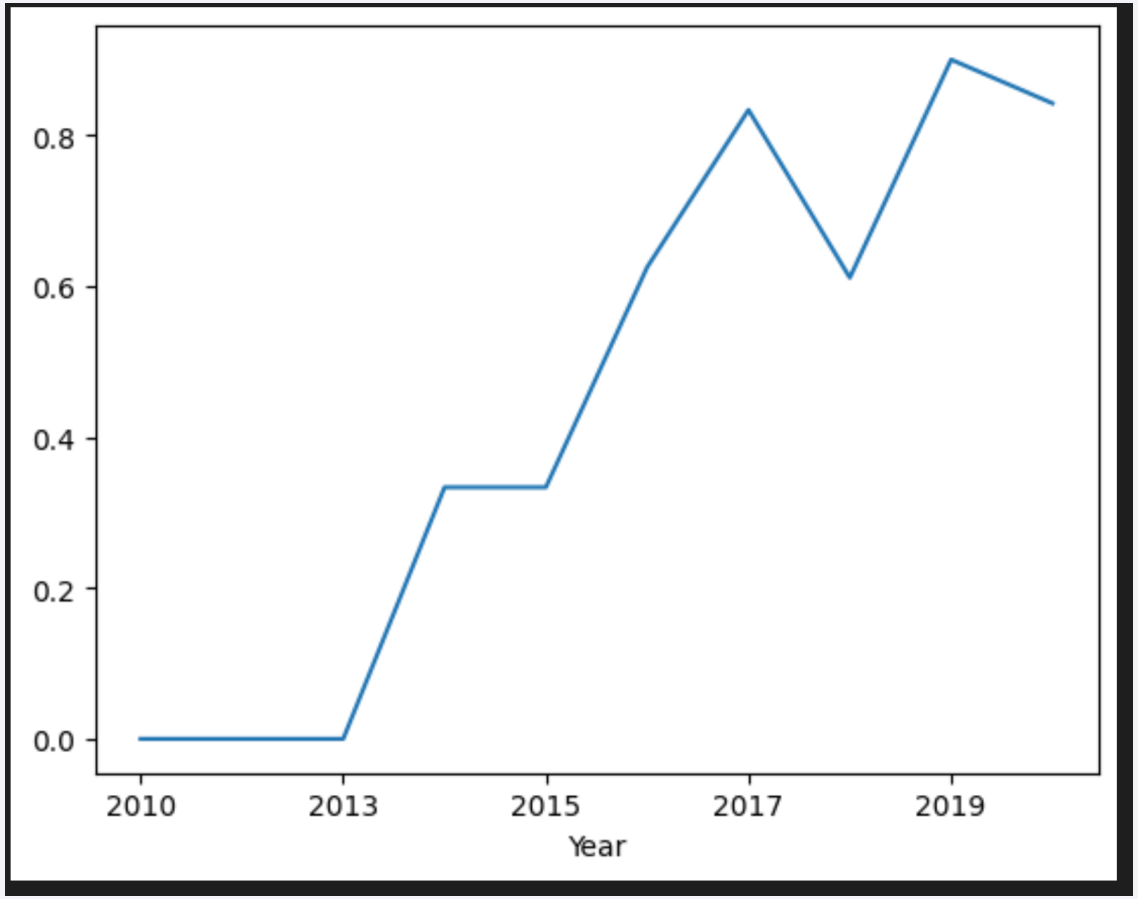
Python



- In my opinion, there is no relation between payload and orbit type
- Orbit type, ISS has similar payload mass around 2000 kg

Launch Success Yearly Trend

- Success rate will be increased time by time.



All Launch Site Names

- CCAFS LC-40, CCAFS SLC-40, KSC LC-39A and VAFB SLC-4E

Task 1

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

```
[7]: sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1
```

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

[7]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

Task 2

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

[8]:
```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

[8]:

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

- Total payload was 1112268 kg.

Task 3

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

```
[9]: sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD LIKE '%CRS%'
 * sqlite:///my_data1.db
Done.
```

[9]:

TOTAL_PAYLOAD
111268

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1

Task 4

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

```
[10]: sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
```

[10]:

AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

- 01/05, 2017 was the first Date to successfully land

Task 5

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

Hint:Use min function

```
[21]: sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
 * sqlite:///my_data1.db
Done.
[21]: FIRST_SUCCESS_GP
      01-05-2017
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Tbooster version where landing was successful and payload mass is between 4000 and 6000 kg.

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

```
[25]: STINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landing_Outcome" = 'Success (drone shi
```

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

[25]:

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

Total Number of Successful and Failure Mission Outcomes

- The result is Success 100 and Failure1

Task 7

List the total number of successful and failure mission outcomes

```
[ ]: sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME
```

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

[13]:

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass

Task 8

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

```
[ ]: SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY
```

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

[14]:

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

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

[27]:
```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

[27]:

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

[28]:
```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

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

[28]:

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

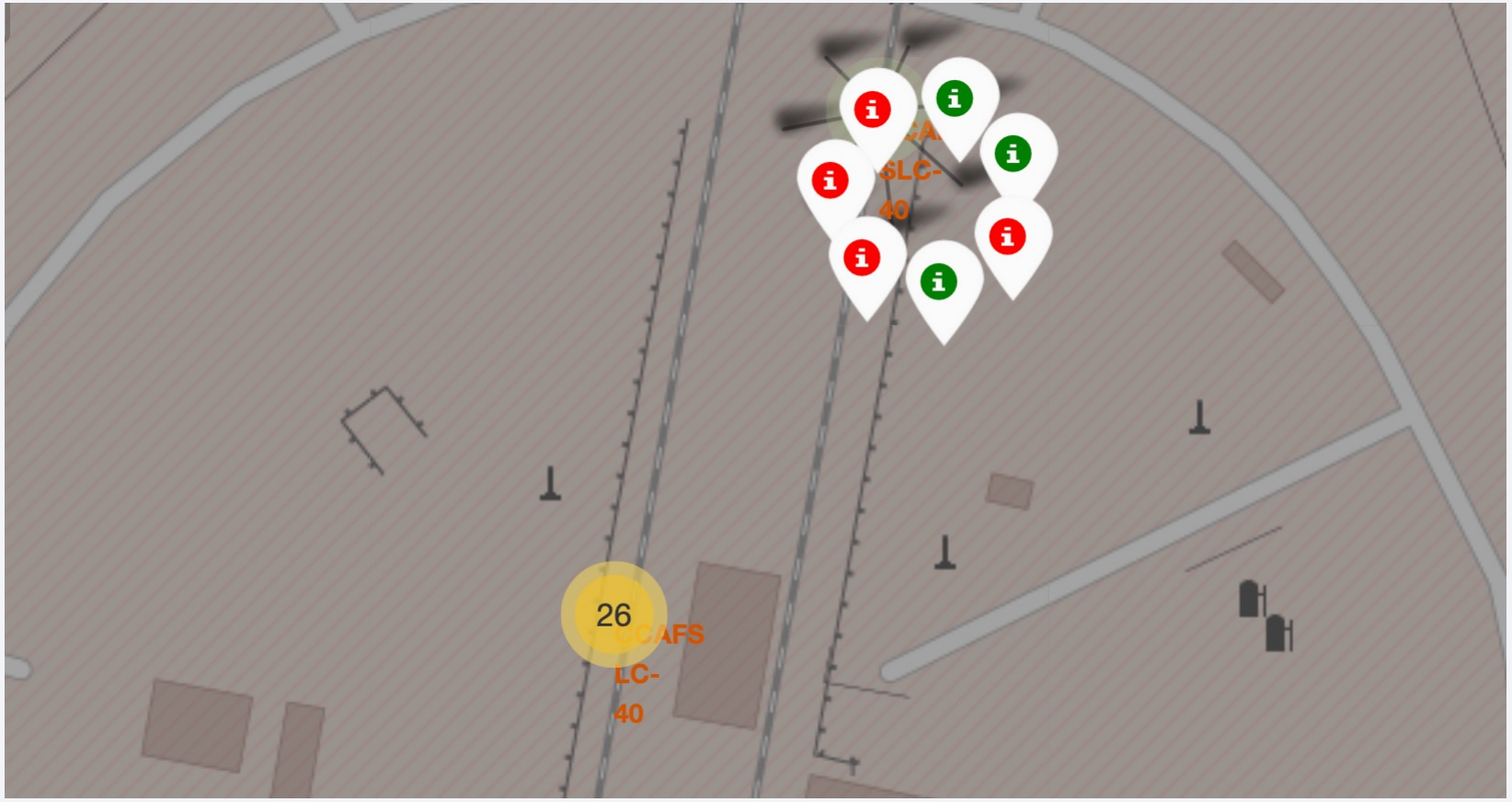

Section 3
Launch Sites
Proximities Analysis

All sites are Launched

- All launched sites are close to coast side.

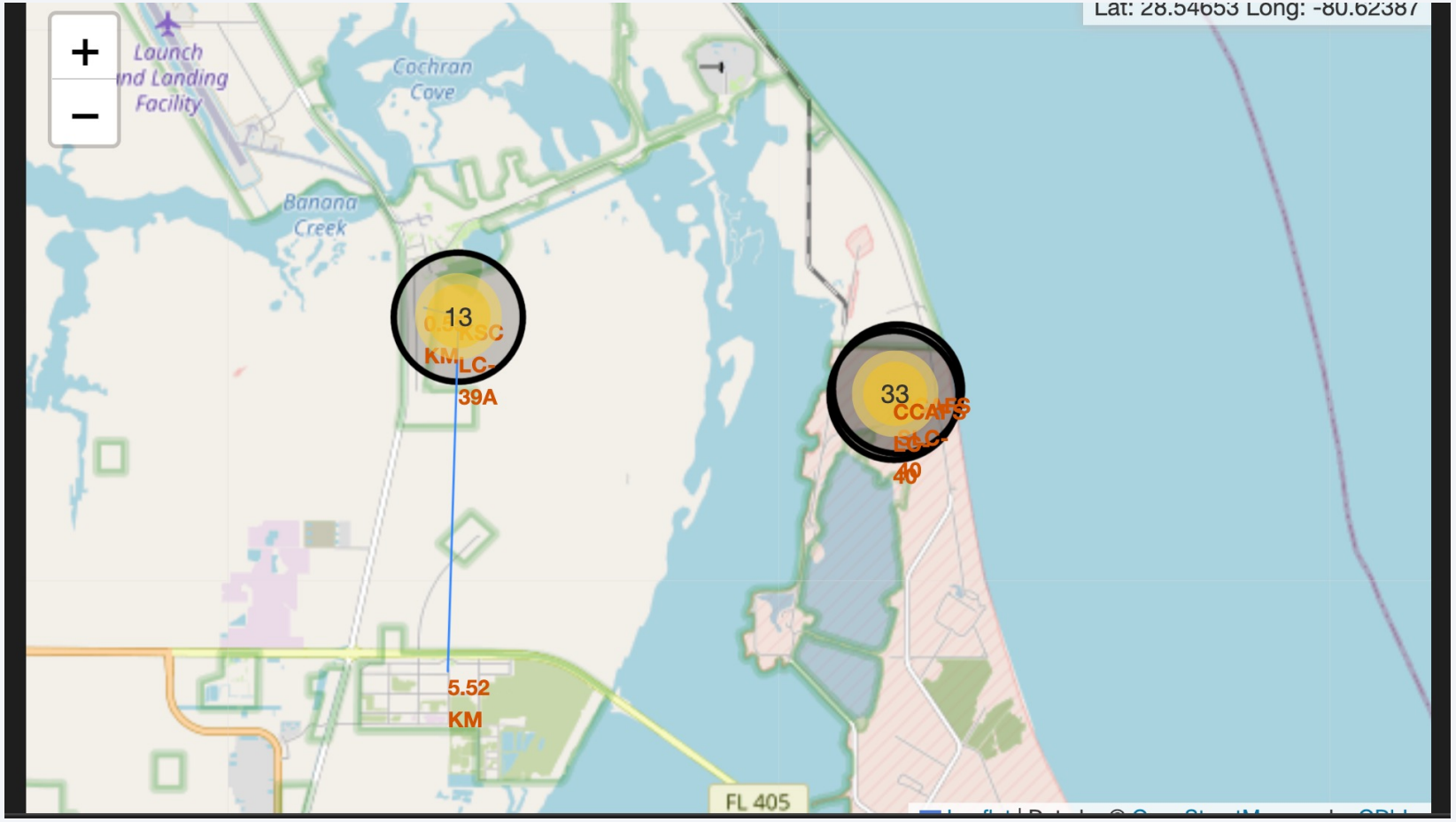
Marked the success/failed launches for each site on the map

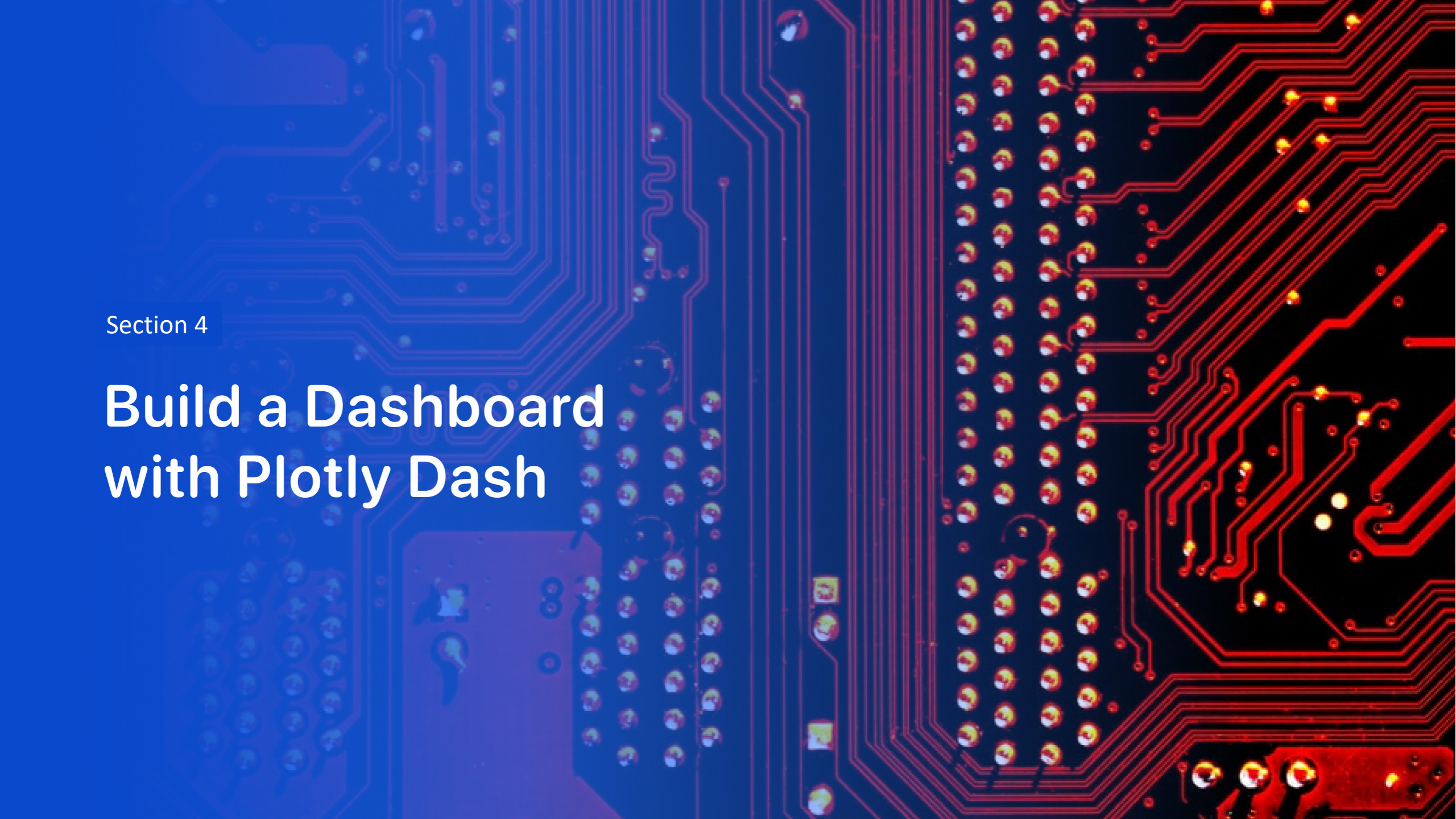
- Green mark means success landing, and red one means failure one



The distances between a launch site to its proximities

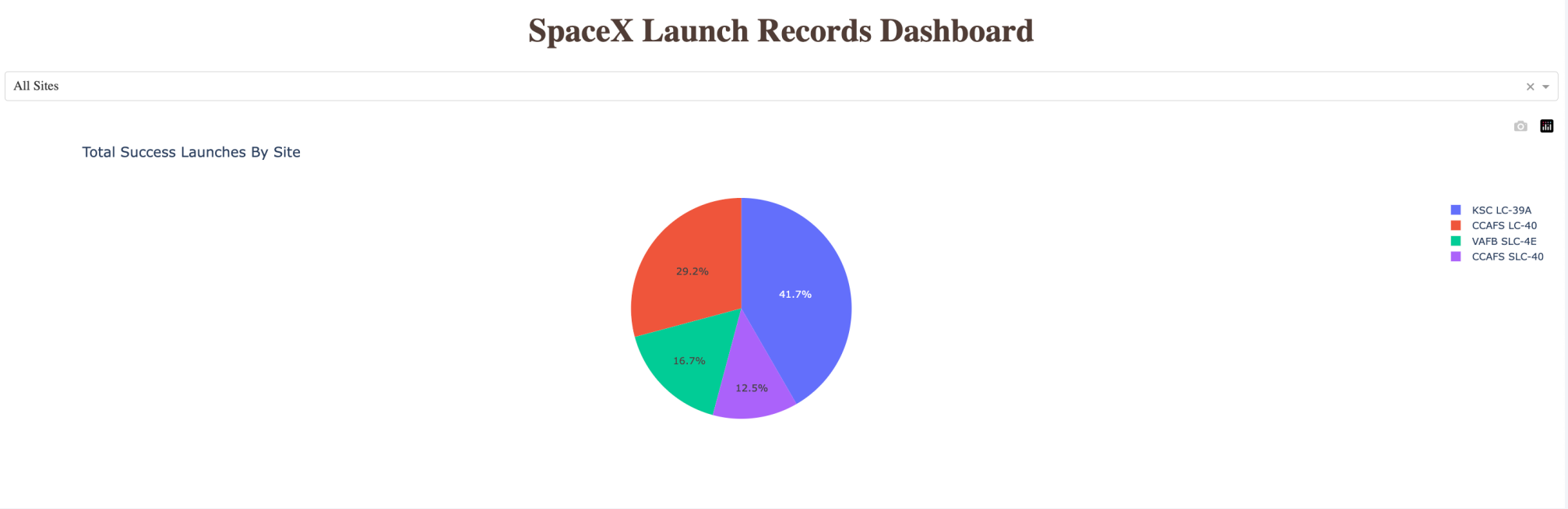
- KSC LC-39A is on good location from living area.



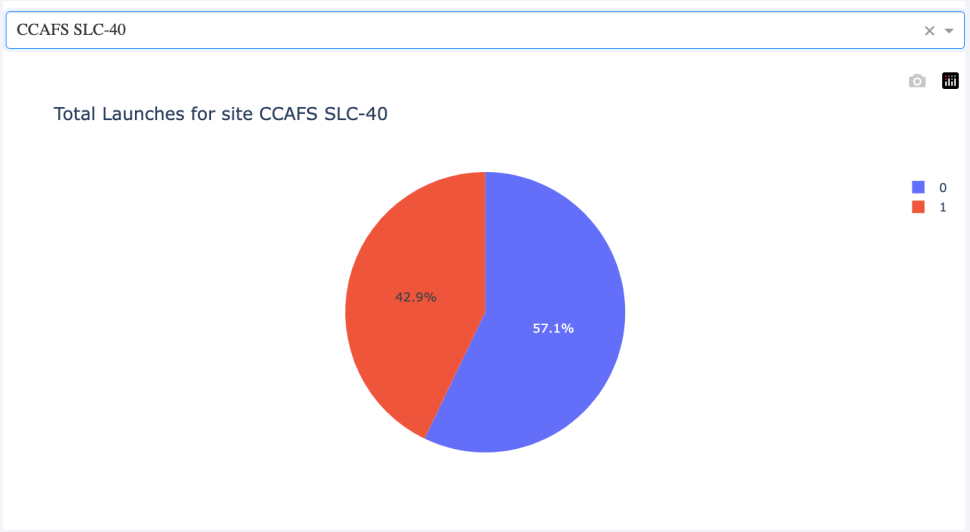

Section 4
Build a Dashboard with Plotly Dash

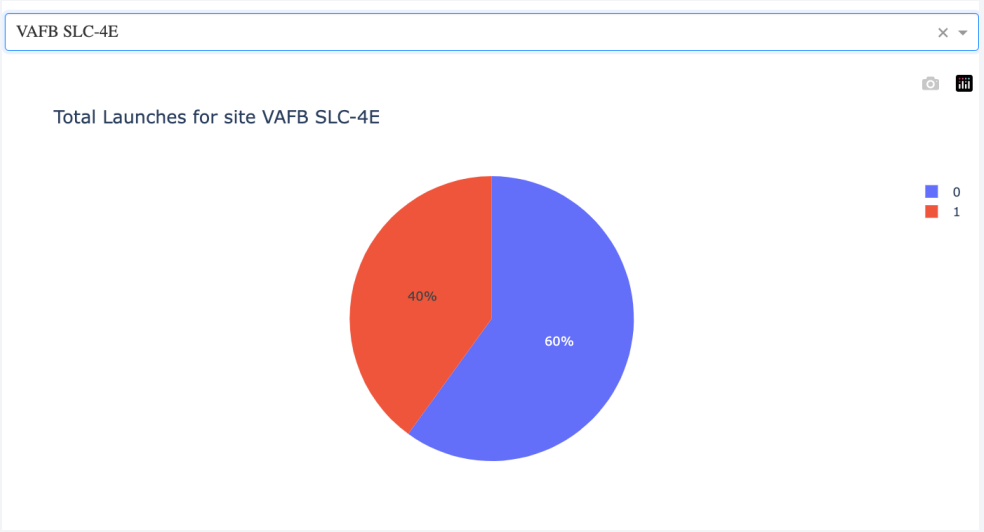
Success launch rate on all sites

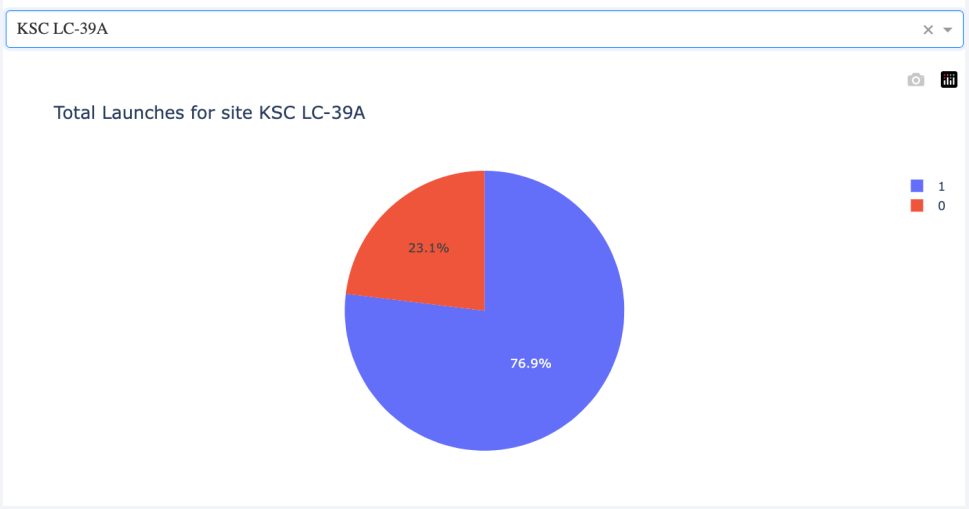
- KSC LC-39A contains 41.7% of success rate in the pie chart.

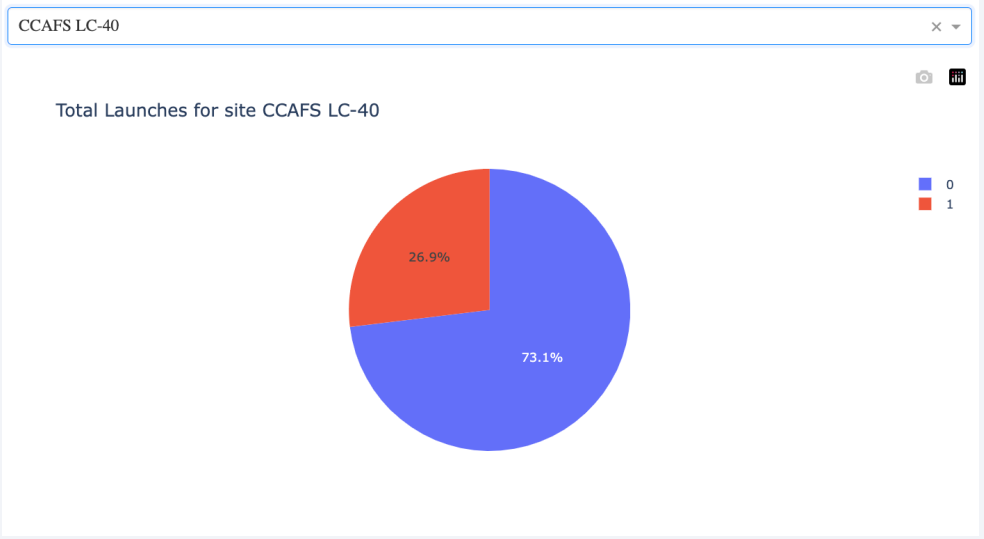


The highest Success launch is KSC LC-39A







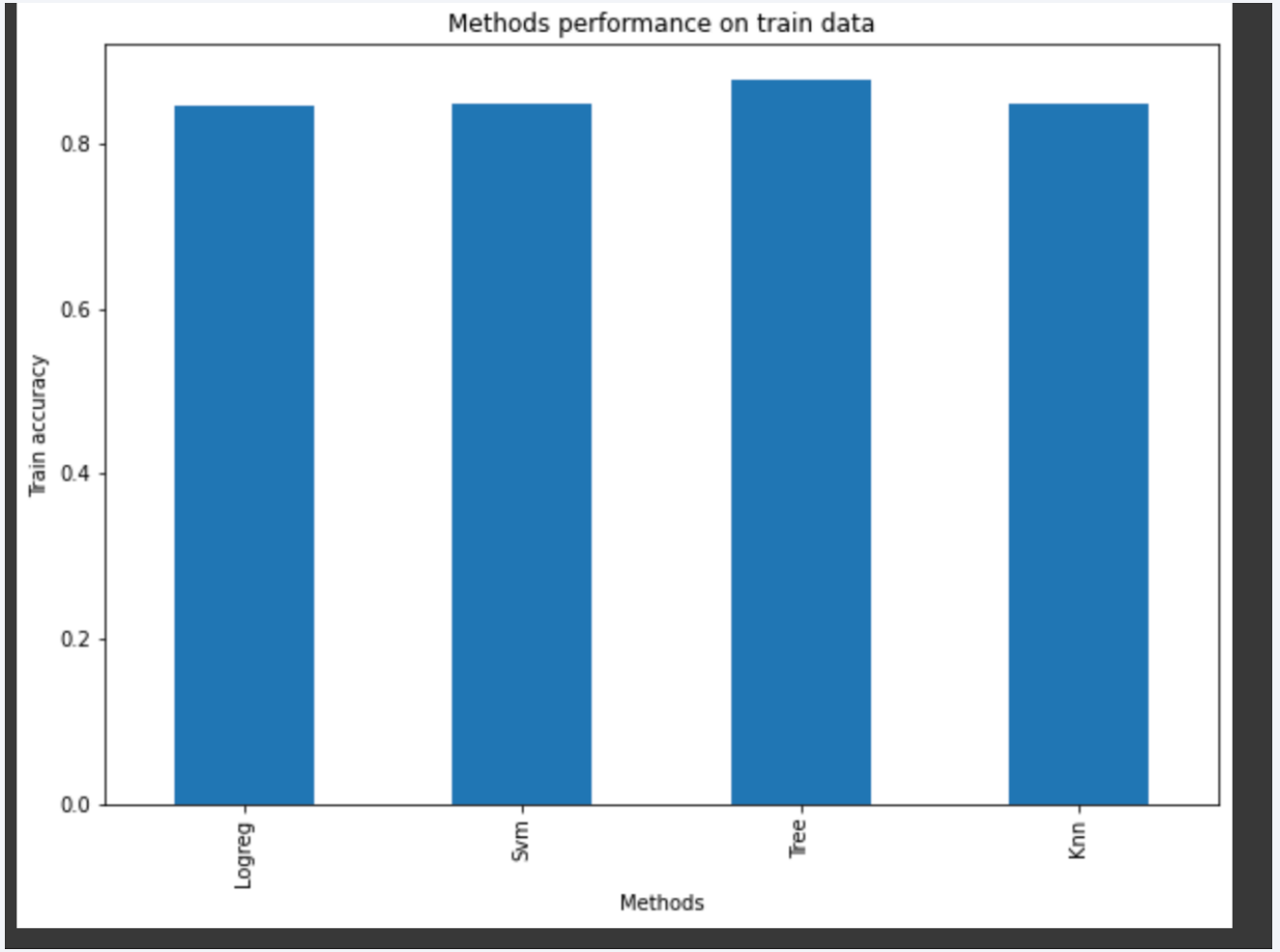


Section 5

Predictive Analysis (Classification)

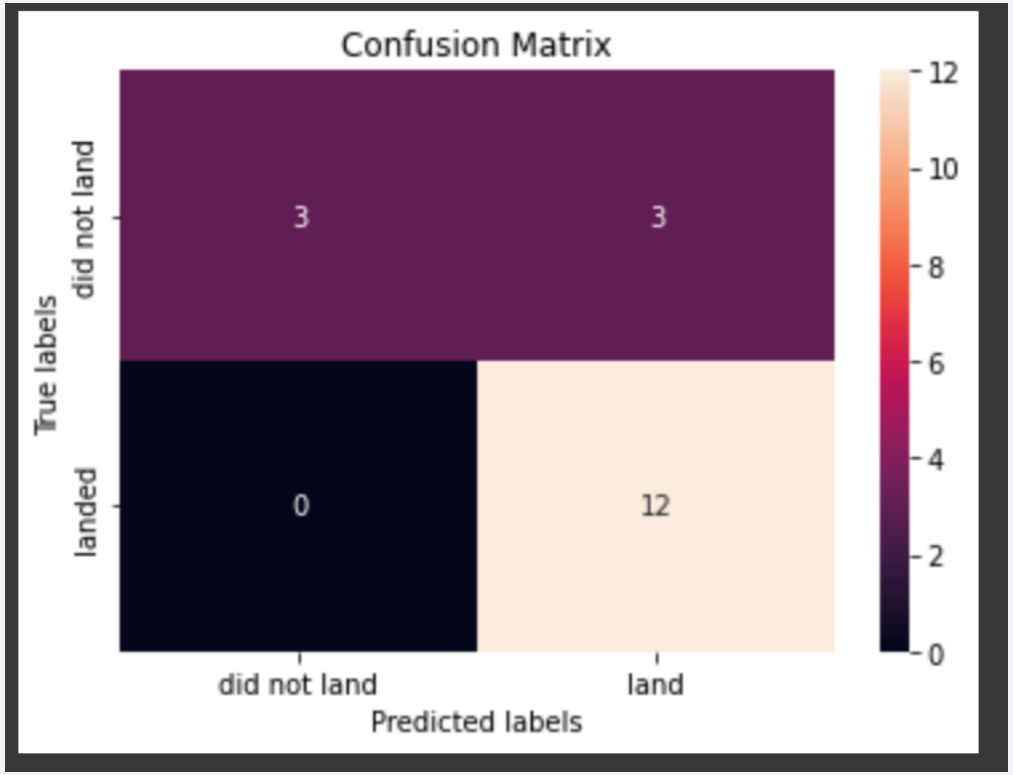
Classification Accuracy

- Decision Tree Classifier is the highest performance model.



Confusion Matrix

- FP (false positives), which is Predicted Label is ‘land’ and True label is ‘did not land’, would be a new improvement points. As the reason, If people believe the result of this model, some rockets might be failed to land. It would be a huge problem in the real world.



Conclusions

- Collecting data from public resources would be sometimes useful
- Success rate would be increased by challenging flight numbers, but we can find the problems by analyzing failure and success cases.
- Predicting serious problems by Machine Learning like landing rockets or deciding healthy status would be more important because people think they want to save failure cost and they want to save someone's life as soon as possible.
- Confusion matrix is a good clue how ML model would be improved

Thank you!