IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Heng Rui Kang
August 5th, 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- This project aims to predict the success of Falcon 9's first stage landing, a critical factor in SpaceX's ability to offer competitive launch costs at $62 million, compared to over $165 million from other providers.
- The data was collected and prepared from the SpaceX API, focusing on key features such as payload mass and launch site. The data was cleaned and machine learning models, including Logistic Regression and Random Forest, was applied to predict landing success.
- The best-performing model, achieved an accuracy of 83.3%. These predictions can help alternative companies estimate launch costs and enhance their competitive bidding strategies against SpaceX.

Introduction

- SpaceX's ability to reuse the Falcon 9's first stage has drastically reduced launch costs, making it a leader in the commercial space industry.
- This project aims to predict the success of Falcon 9 landings, identify key factors influencing outcomes, and explore how these predictions can help estimate launch costs and enhance competitive strategies.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web Scraping: Falcon 9 historical launch records were collected by scraping data from a Wikipedia page listing Falcon 9 and Falcon Heavy launches.
  - API Collection: Additional data was collected and formatted by accessing and retrieving information through an API provided by SpaceX.
- Perform data wrangling
  - The raw data was processed and labeled, converting various landing outcomes into binary training labels (1 for successful landings and 0 for unsuccessful landings) to prepare the data for supervised machine learning models.

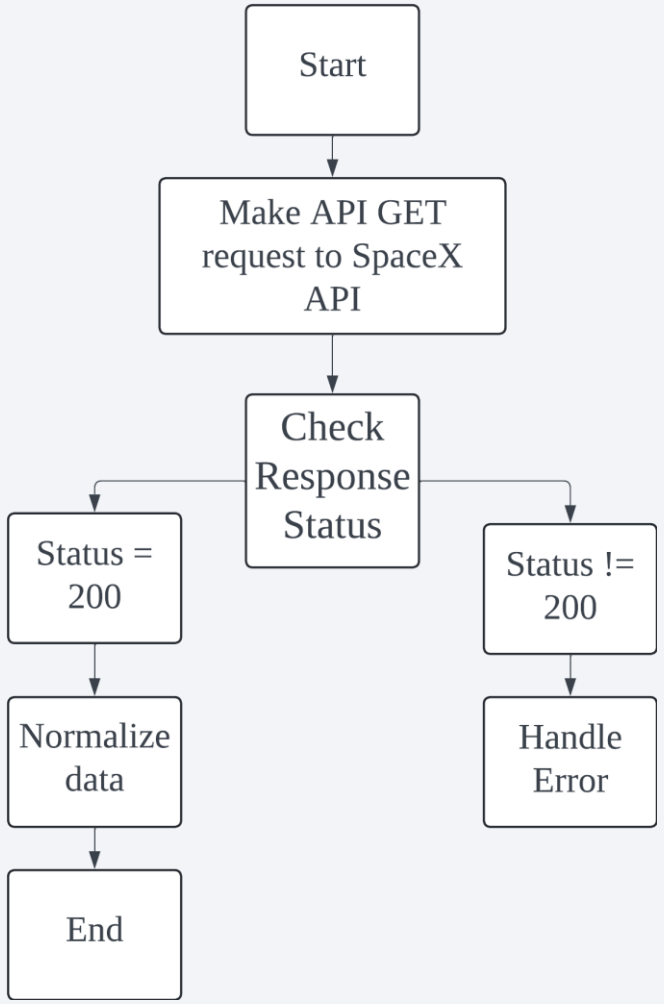
Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - The models this project are built, tuned, and evaluated using a machine learning pipeline that processes the data, applies hyperparameter tuning with grid search, and assesses performance using cross-validation and using accuracy as the metric.

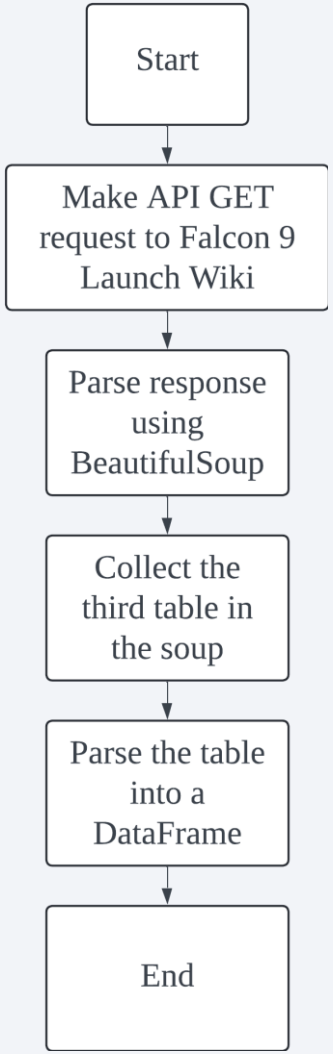
Data Collection – SpaceX API

- Make API GET request to SpaceX API using Python's requests library.
- The data is then normalized using pandas '.json_normalize()' to convert the response into a DataFrame.
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- A GET request is made to Falcon9 Launch Wikipedia page.
- The response is then parsed using BeautifulSoup.
- From the soup, the third table is collected and parsed to a DataFrame.
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- The column ‘Payload Mass’ had missing values and was replaced with the mean of the entire column.
- The categorical columns, 'Orbit', 'LaunchSite', 'LandingPad', 'Serial’ are encoded to binary for the machine learning process.
- A ‘Class’ feature in the form of binary was added that represents whether the launch was successful (1) or failure (0) derived from the ‘Outcome’ feature.
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Charts used:
  - ScatterPlot: To visualize the relationship between,
    - Number of flights on various launch sites.
    - Payload mass on various launch sites.
    - Number of flights on various orbits.
    - Payload mass on various orbits.
  - BarChart: To visualize the relationship between success rate of each orbit type.
  - LineChart: To visualize the trend of yearly success rate.
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/edadataviz.ipynb

EDA with SQL

- SQL queries performed:
  - Collecting all the names of the launch sites
  - Displayed 5 records from the launch site starting with ‘CCA’
  - Displayed total payload mass carried by NASA
  - Displayed average payload mass carried by booster version F9 v1.1
  - Displayed the date of the first successful landing outcome.
  - Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listed total number of successful and failure mission outcomes
  - Listed the names of the booster versions which have carried the maximum payload mass
  - Listed the records of failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015
  - Ranked 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
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

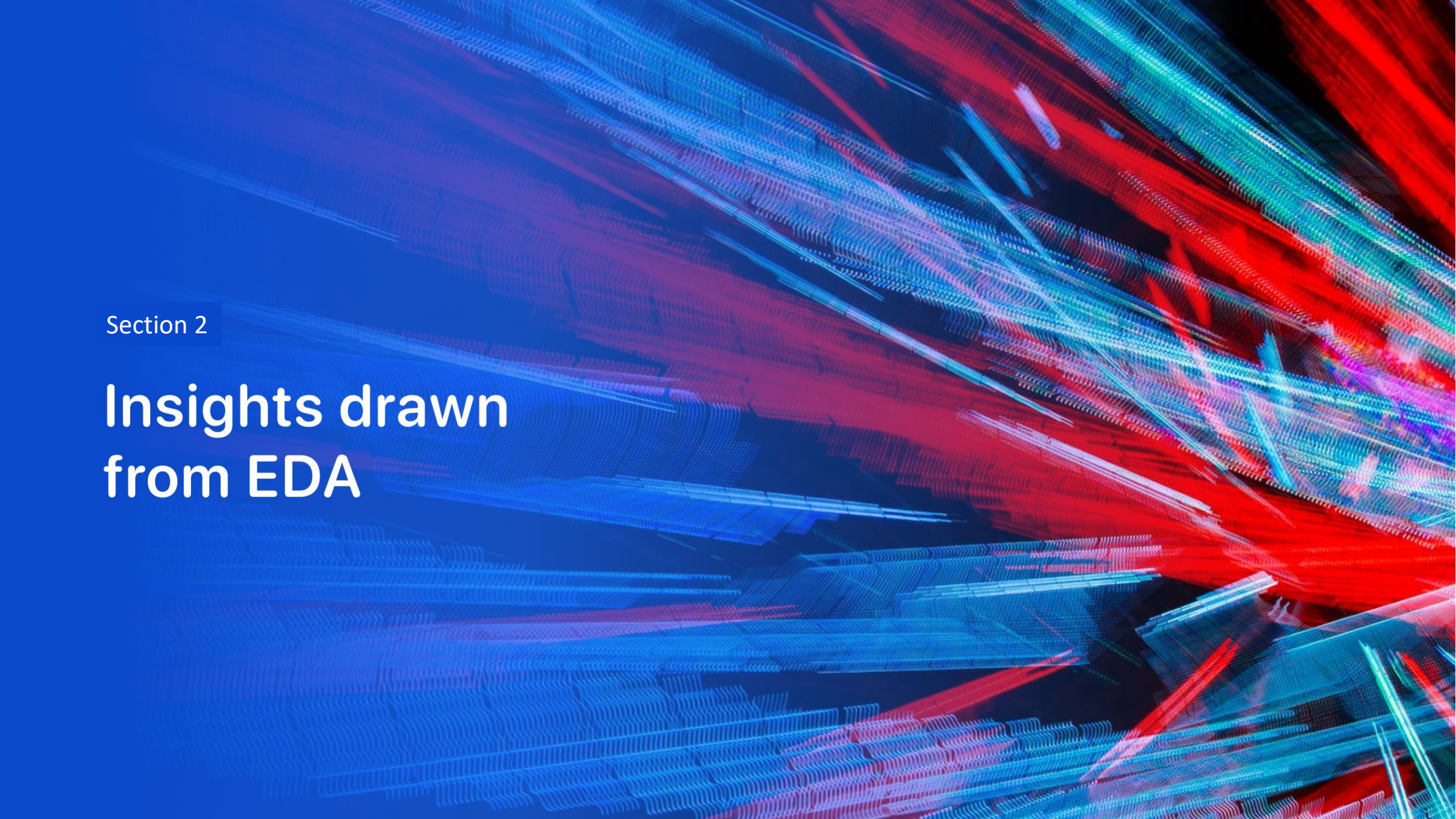
- A folium map is created with its location starting at NASA's coordinates
- Markers are added to the coordinates of all 4 launch sites
- Markers are used because the named of the launch sites can be made to show when clicked on
- MarkerClusters are also added to show the launch outcomes at the launch sites
- MarkerClusters is a good way to simplify a map containing many markers having same coordinates
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Added a Pie Chart and a Scatter Plot for visualization
- If no site is selected:
  - Pie Chart shows the total success launches of ALL sites.
  - Scatter Plot shows the correlation between payload and success of ALL site.
- If a site is selected:
  - Pie Chart shows the total success launches of SELECTED site.
  - Scatter Plot shows the correlation between payload and success of SELECTED site
- A slider bar is also added for the user to set the payload range for the Scatter plot
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/spacex_dash_app.py

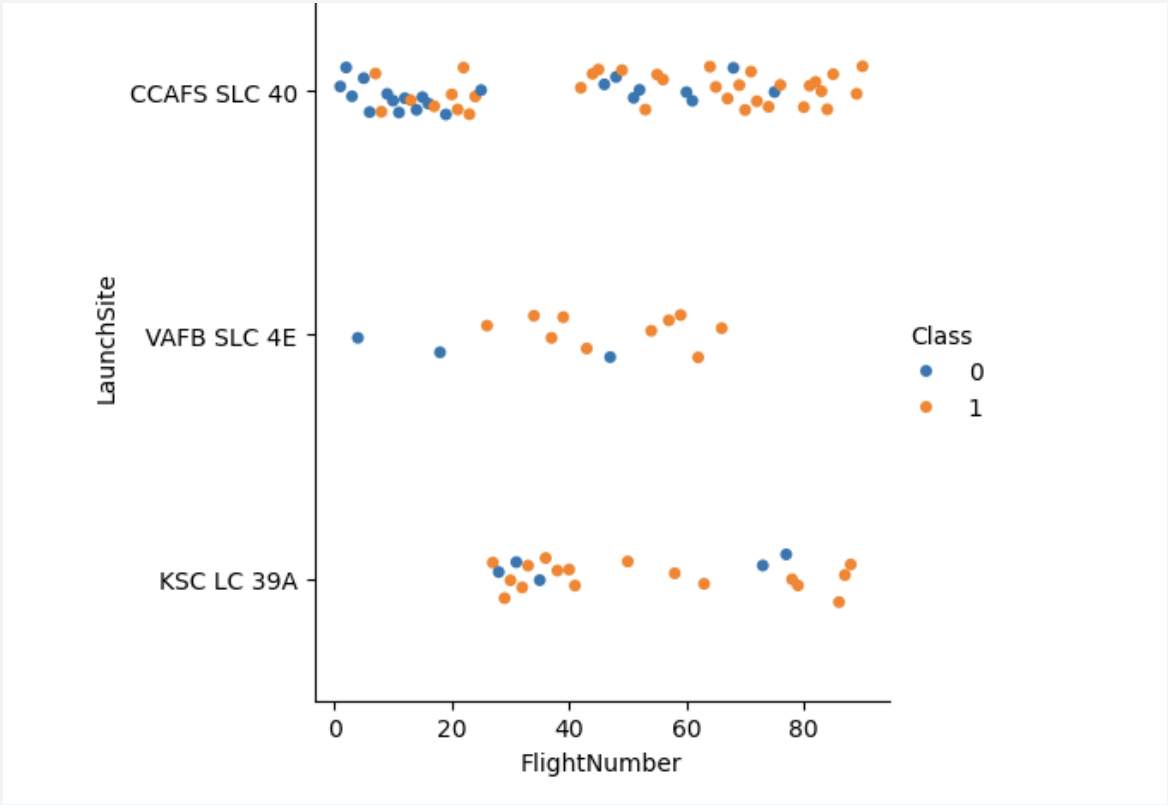
Predictive Analysis (Classification)

- The data is first scaled using StandardScaler()
- Then, the data is split into 80% training set, 20% test set
- Next, different models are created, including Logistic Regression, Support Vector Machine, Decision Tree, KNN
- Each model had its best parameters found using GridSearchCV with 10 folds
- For every model, the performance is evaluated using the accuracy metric on the testing set
- https://github.com/Ruitherli/Predicting-Rocket-Launch-Success-and-Costs/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb


Section 2
Insights drawn from EDA

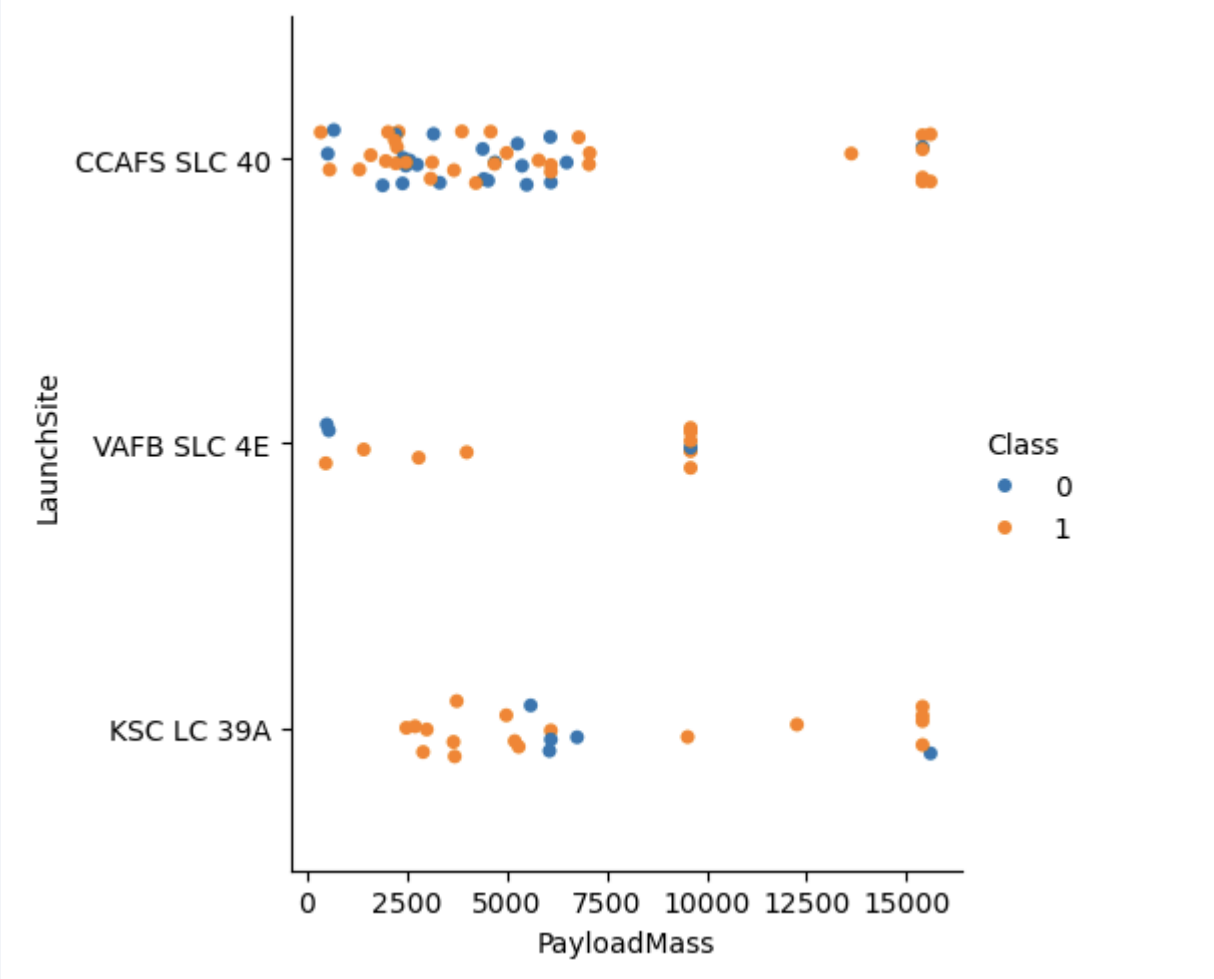
Flight Number vs. Launch Site

- For launch site ‘CCAFS SLC 40’, the number of successful launches increased when the number of flights increased



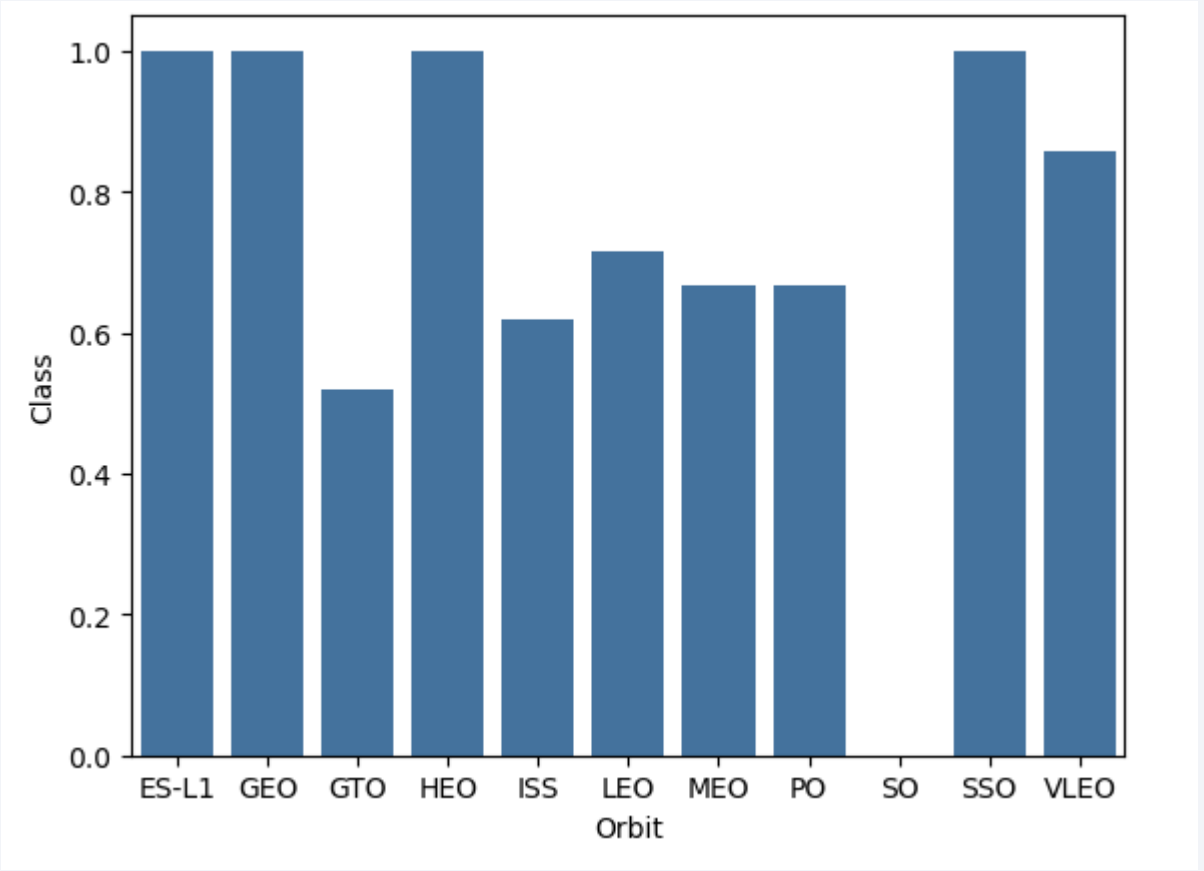
Payload vs. Launch Site

- Launch site ‘VAFB SLC 4E’ has no launches with payload mass of more than 10000kg.



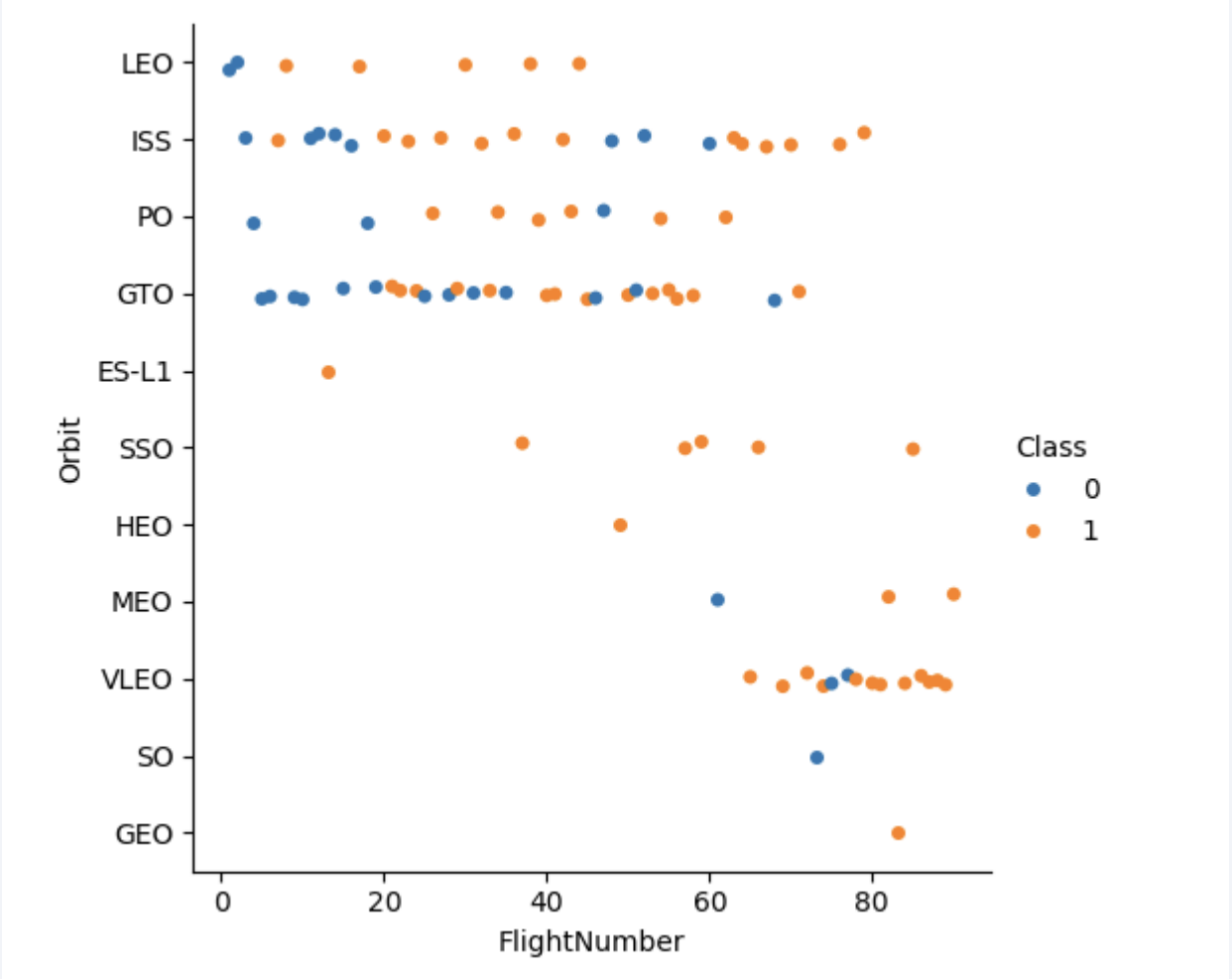
Success Rate vs. Orbit Type

- Orbit ‘ES-L1’, ‘GEO’, ‘HEO’, ‘SSO, has a success rate of 100%,
- Orbit ‘VLEO’ has a success rate of 83%
- Orbit ‘GTO’, ‘ISS’, ‘LEO’, ‘MEO’, ‘PO’ has an average success rate
- Orbit ‘SO’ has a 0% success rate



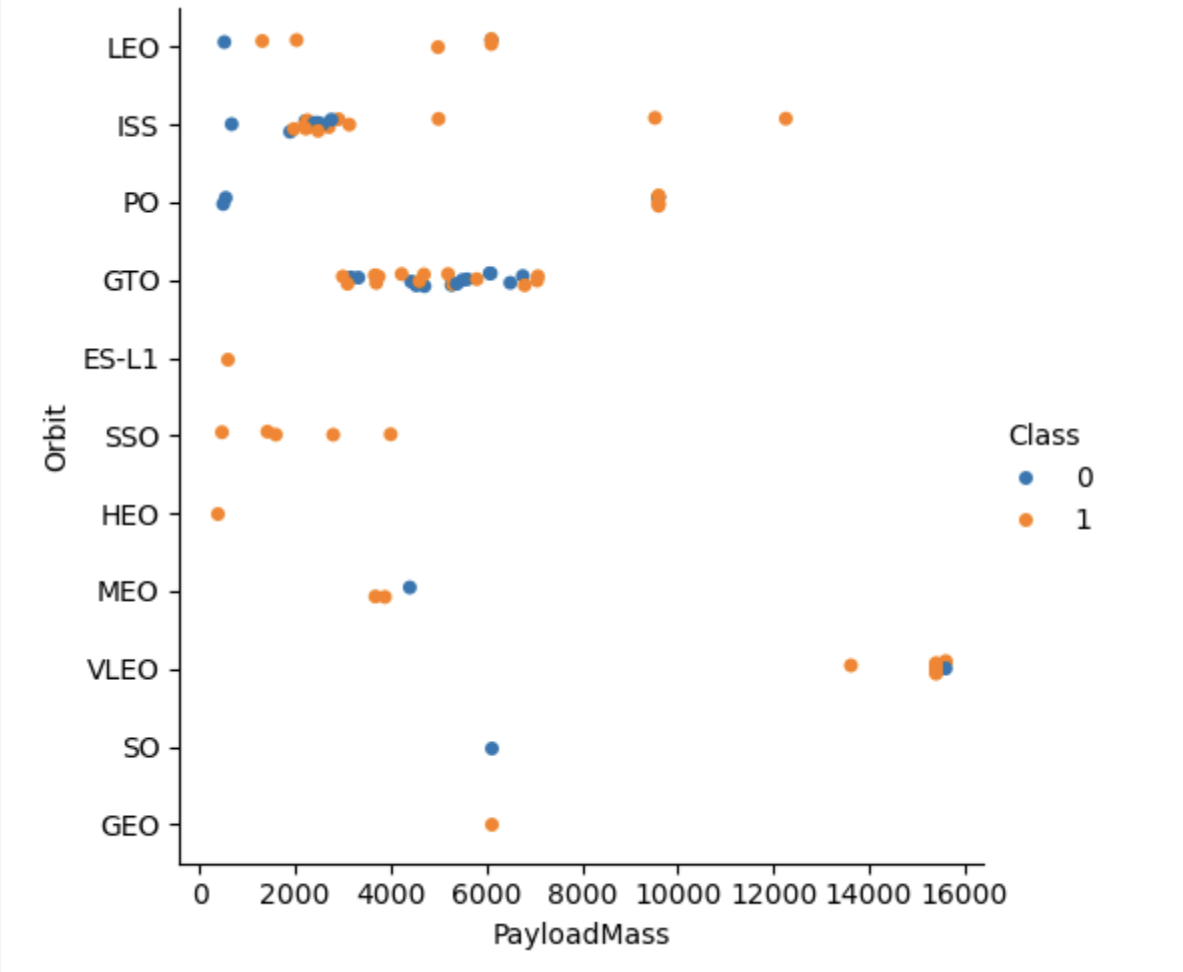
Flight Number vs. Orbit Type

- In ‘LEO’, the number of successful launches increases as the number of flights increases
- Conversely, in ‘GTO’, the number of flights appears to have no impact on the successful launches



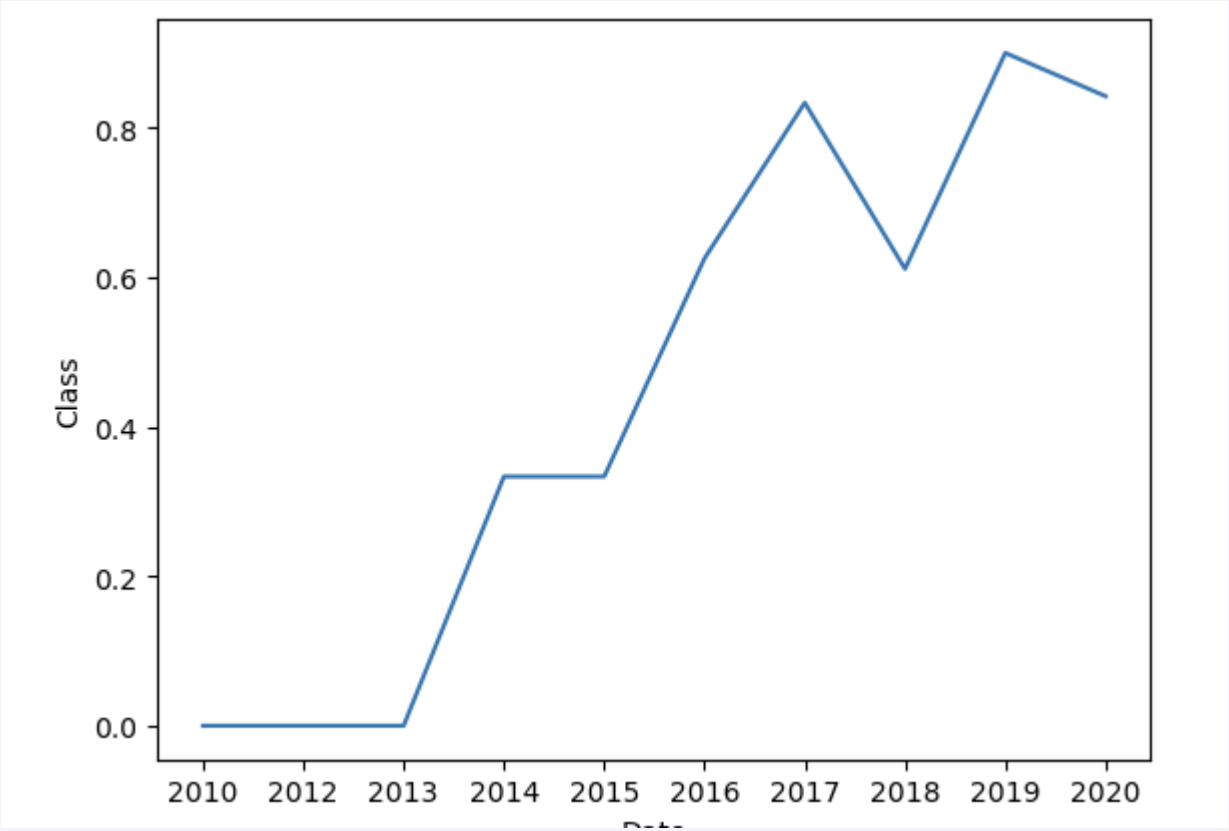
Payload vs. Orbit Type

- ‘LEO’, ‘ISS’, ‘PO’ has an increase of successful launches when the payload mass increases
- However, ‘GTO’ appears to deny this relationship as the failures and successful launches are present on similar payload mass.



Launch Success Yearly Trend

- The success rate of launches has been steadily increasing from the year 2010 to the year 2020 with a slight plummet on 2018 before recovering in 2019.



All Launch Site Names

- Find the names of the unique launch sites
- Use DISTINCT to filter out duplicates.

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

[12]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- LIKE ‘CCA%’, checks for a string that starts with CCA in the Launch_SITE column.

Task 2

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

```
[14]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

[14]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Ou
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	S
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	S
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	S
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	S
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	S

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- SUM the PAYLOAD_MASS_KG used by NASA

Task 3

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

```
[15]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

[15]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- AVG the PAYLOAD_MASS_KG of every booster version F9 v1.1

Task 4

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

```
[16]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = "F9 v1.1"
```

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

[16]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- ORDER BY Date ASC LIMIT 5 to show the first 5 successful launches

Task 5

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

Hint:Use min function

[19]:
```
%sql SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome = 'Success' ORDER BY Date ASC LIMIT 5
```

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

[19]:

Date
2018-07-22
2018-07-25
2018-08-07
2018-09-10
2018-10-08

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Use BETWEEN for the PAYLOAD_MASS_KG condition

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

```
[21]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' AND PAYLOAD_MASS__KG_
```

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

[21]:

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Group the outcomes, it appears there are 3 types of successes and one type of failure

Task 7

List the total number of successful and failure mission outcomes

```
[23]: %sql SELECT Mission_Outcome,COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

[23]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Use subquery to obtain the max payload_mass_kg and list the boosters who meet this condition

Task 8

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

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_
```

```
 * 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

2015 Launch Records

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

- Substr the Date to obtain the years and match it with 2015 as a condition

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
31]: %sql SELECT substr(Date,6,2) AS MONTH, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
```

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

31]:

MONTH	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	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

- Group by landing_outcome and apply COUNT then ORDER BY DESC

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.

```
36]: %sql SELECT Landing_Outcome, Date, COUNT(*) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '201
```

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

36]:

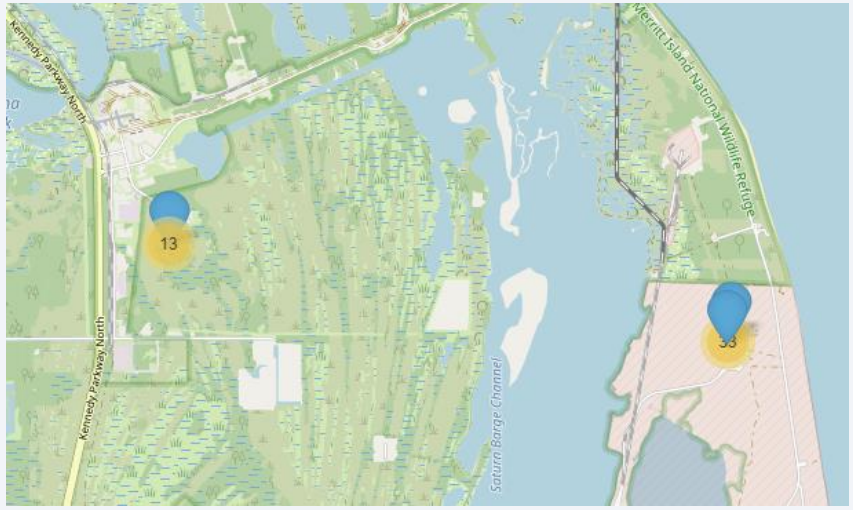
Landing_Outcome	Date	COUNT(*)
No attempt	2012-05-22	10
Success (drone ship)	2016-04-08	5
Failure (drone ship)	2015-01-10	5
Success (ground pad)	2015-12-22	3
Controlled (ocean)	2014-04-18	3
Uncontrolled (ocean)	2013-09-29	2
Failure (parachute)	2010-06-04	2
Precluded (drone ship)	2015-06-28	1

Section 3

Launch Sites Proximities Analysis

Folium Map Screenshot of Launch Sites

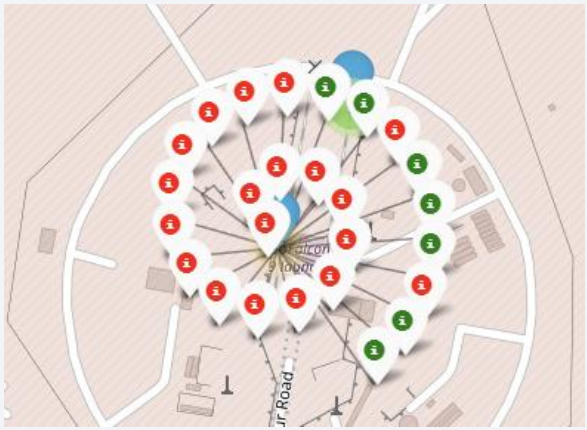
- 3 out of 4 Launch sites (CCAFS SLC-40, CCAFS LC-40, KSC LC-39A) are in close proximity to each other, nearby a placed called Titusville
- VAFB SLC-4E is at Los Angeles



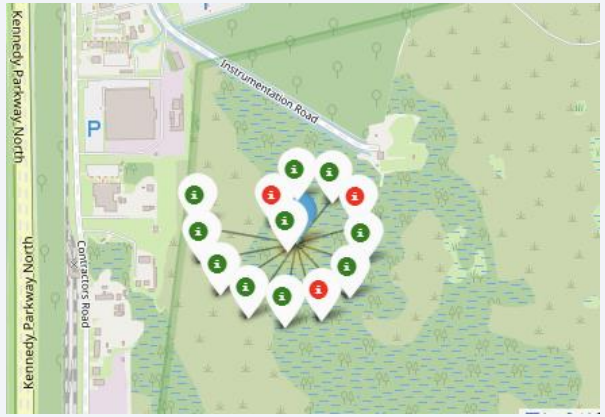
Folium Map Screenshot of Launch Outcomes

- From the folium map, KSC LC-39A launch site appears to have the highest success rate

VAFB SLC-4E

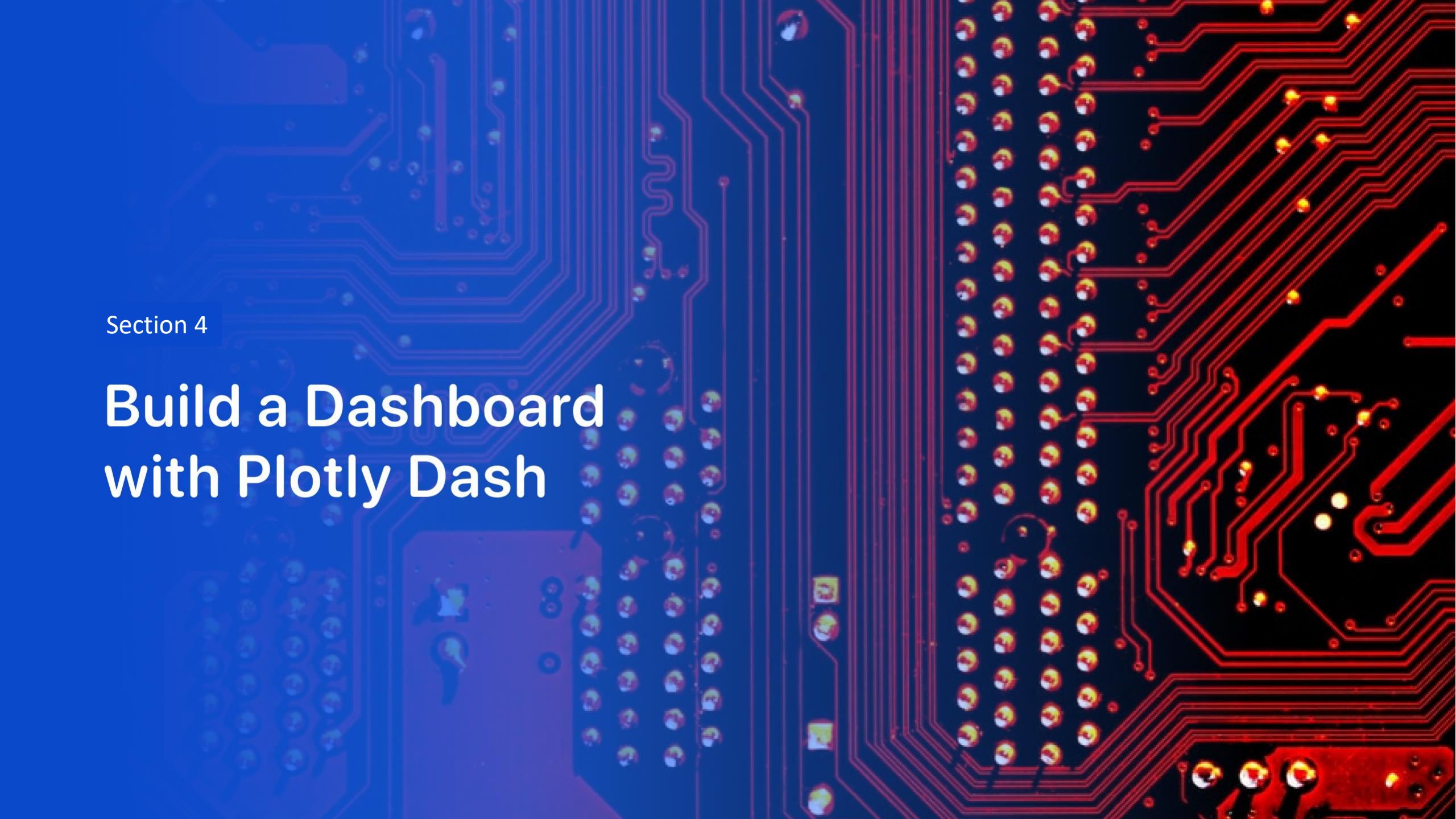


CCAFS LC-40



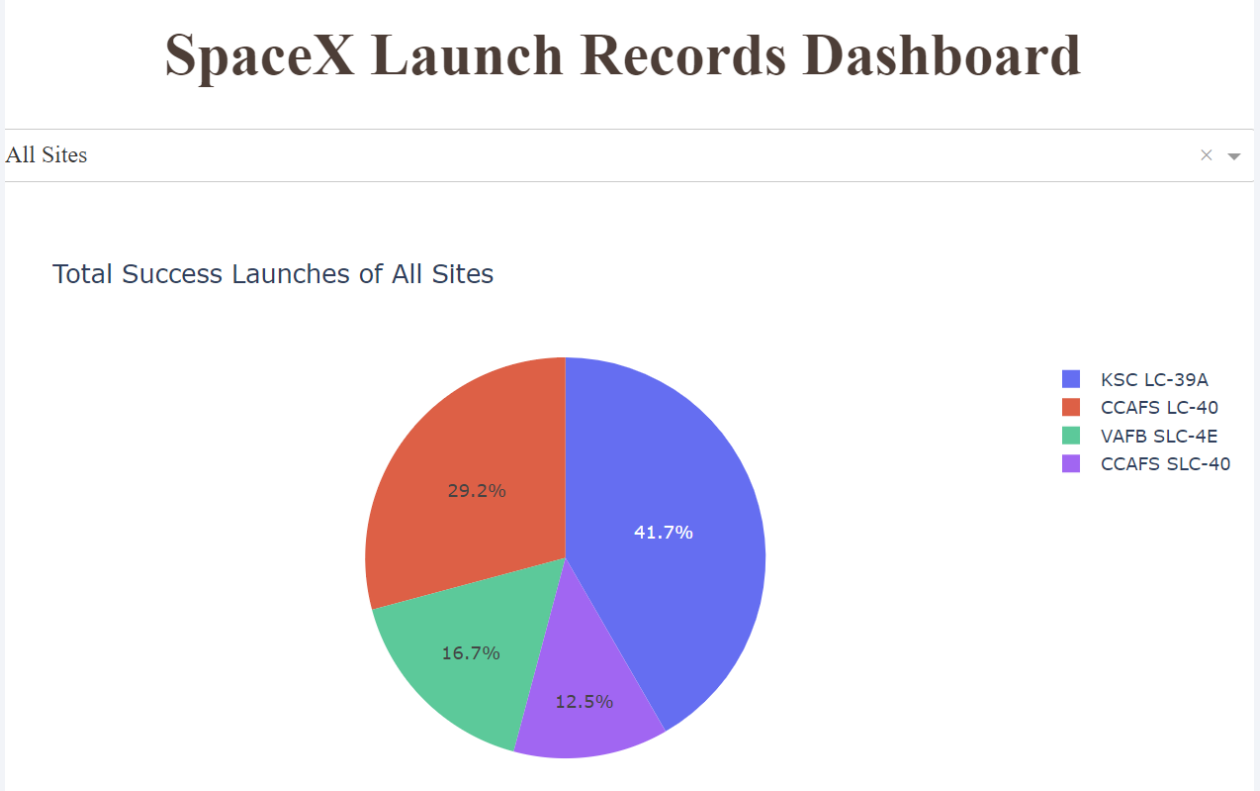
KSC LC-39A

CCAFS SLC-40


Section 4
Build a Dashboard with Plotly Dash

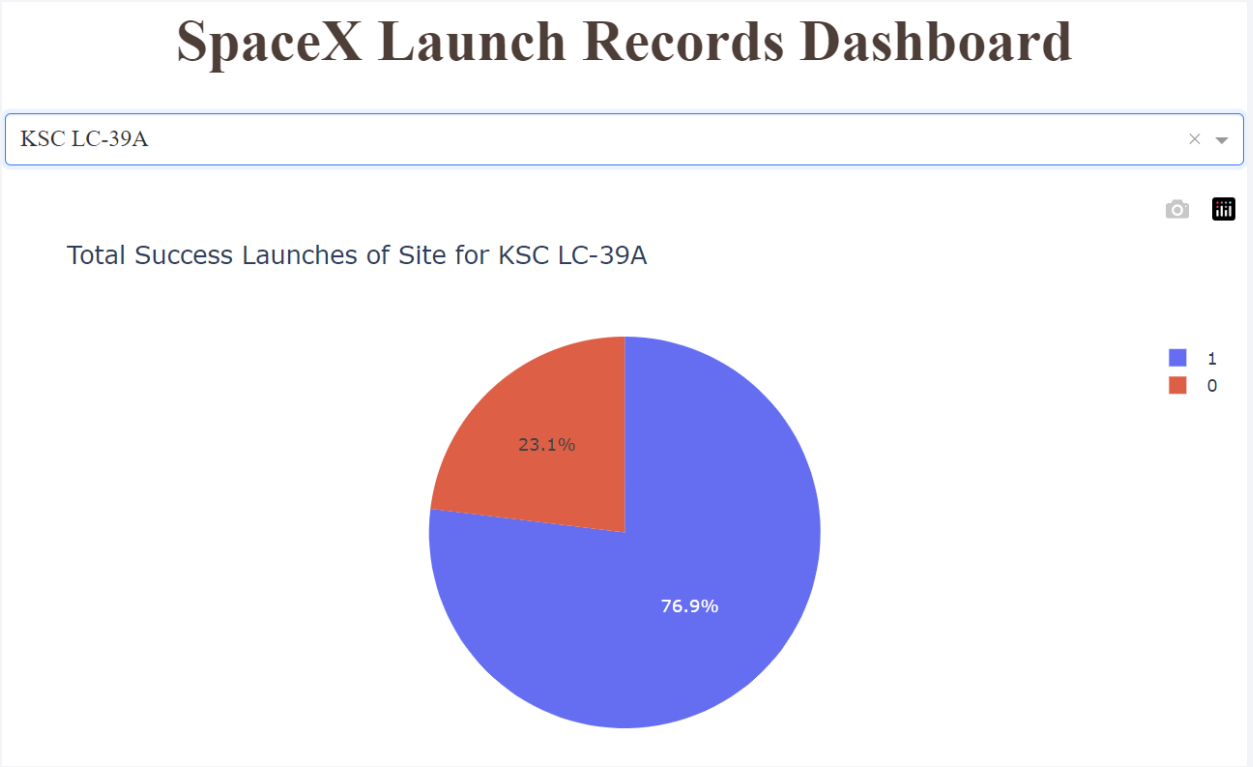
SpaceX Launch Records Dashboard

- KSC LC-39A launch site has the highest total success launches out of every site with 41.7%



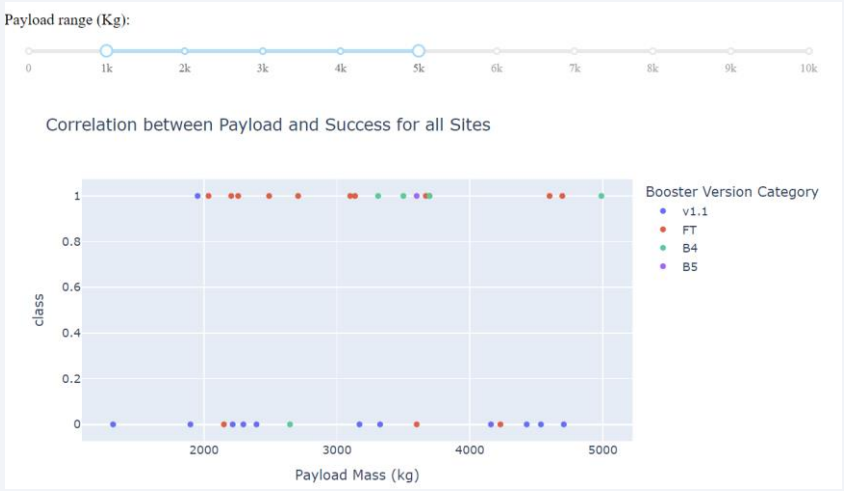
KSC LC-39A Pie Chart

- From the Pie Chart, we can see that 76.9% of total launches ended in a success



<Dashboard Screenshot 3>

- From the Scatter Plot, we can see that the Booster ‘FT’ has more successful launches at the payload range of 2000kg to 6000kg
- ‘FT’ also have more successful launches than other booster versions

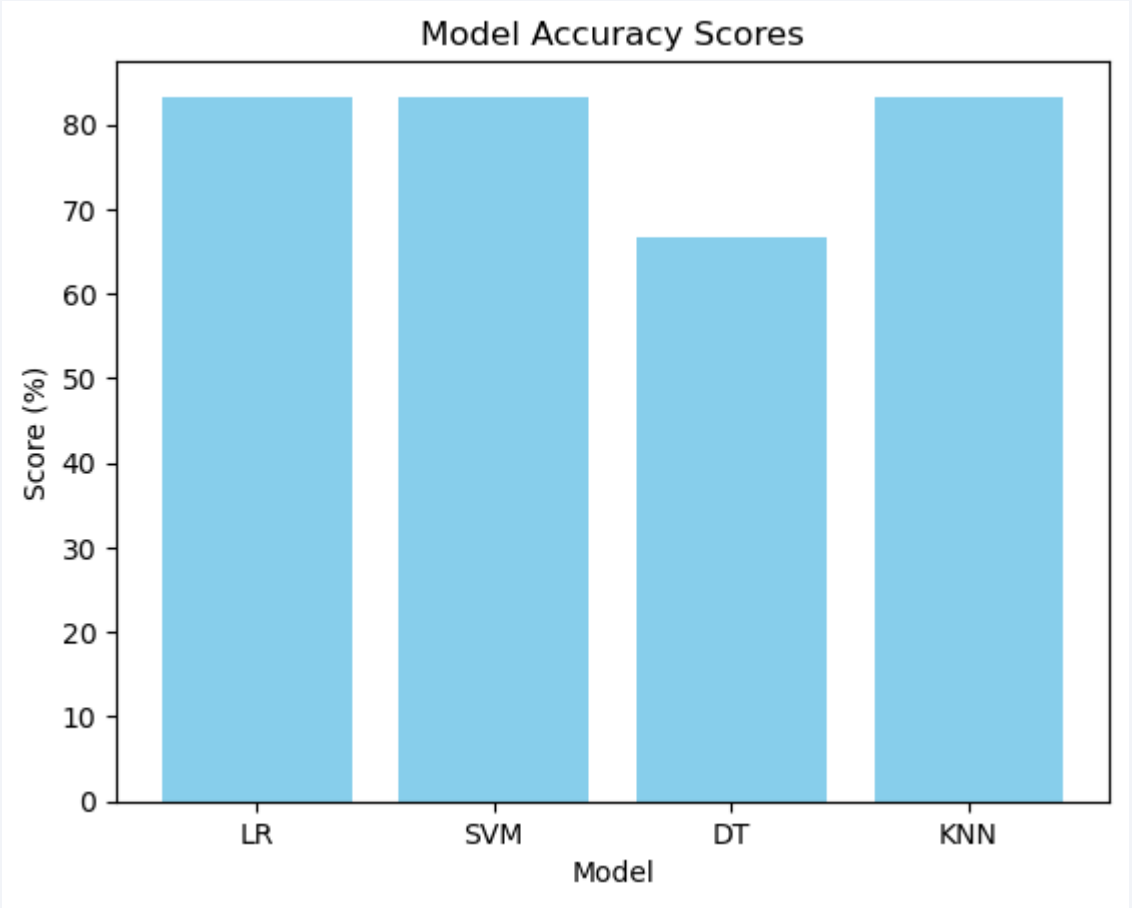


Section 5

Predictive Analysis (Classification)

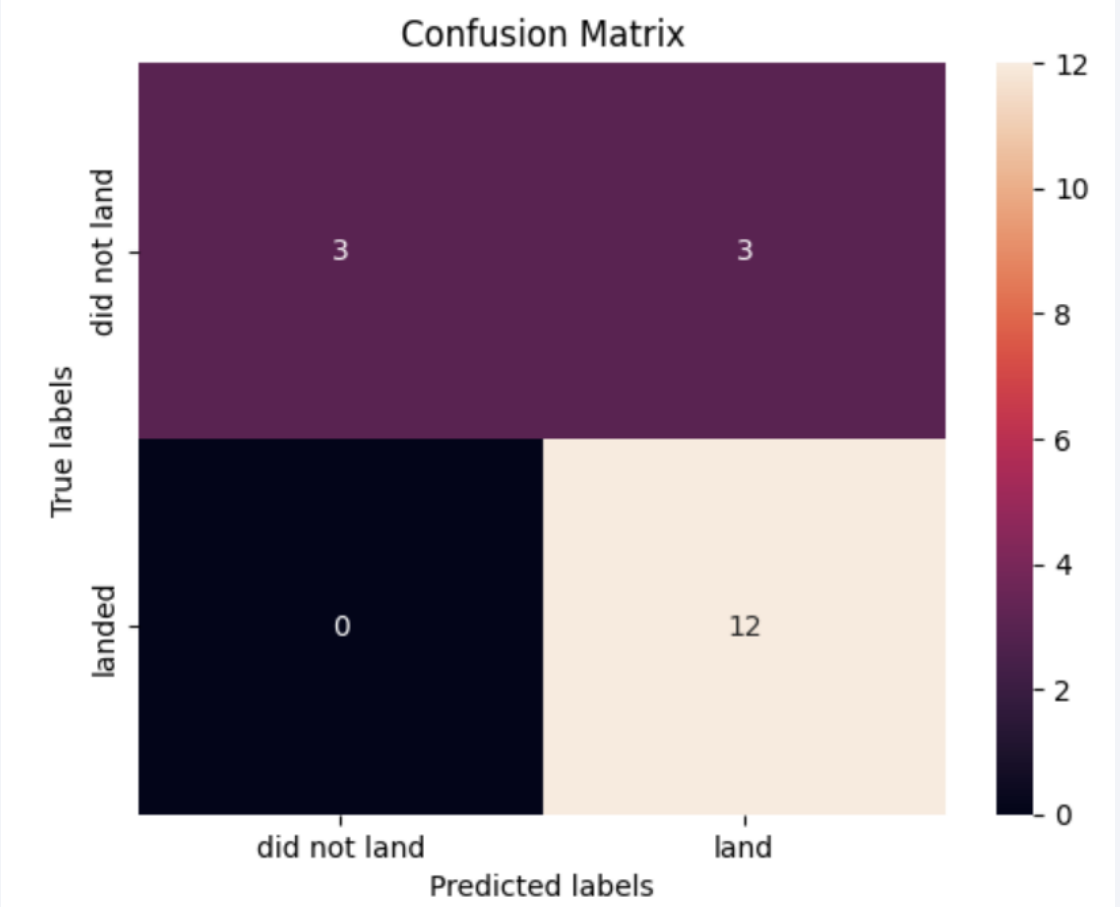
Classification Accuracy

- Logistic Regression, Support Vector Machine and K-Nearest Neighbors models have the same accuracy of 83.3%



Confusion Matrix

- The KNN model had some (3) False Positive labels but none (0) False Negative labels.



Conclusions

- KSC LC-39A launch site appears to have the highest success rate, 76.9% for launches
- There are more successful launches at the payload mass range of 2000kg to 6000kg
- The booster version FT has more successful launches than other versions
- Logistic Regression, Support Vector Machine and K-Nearest Neighbors model performed the same with an accuracy of 83.3%

Thank you!