IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Vincent Lynch>
<2025-01-03>

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- 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.

- If we can determine if the first stage will land, we can determine the cost of a launch.

SpaceX Falcon 9 Successful Booster Landing on Ground Pad and Drone Ship

Section 1

Methodology

Methodology

Executive Summary

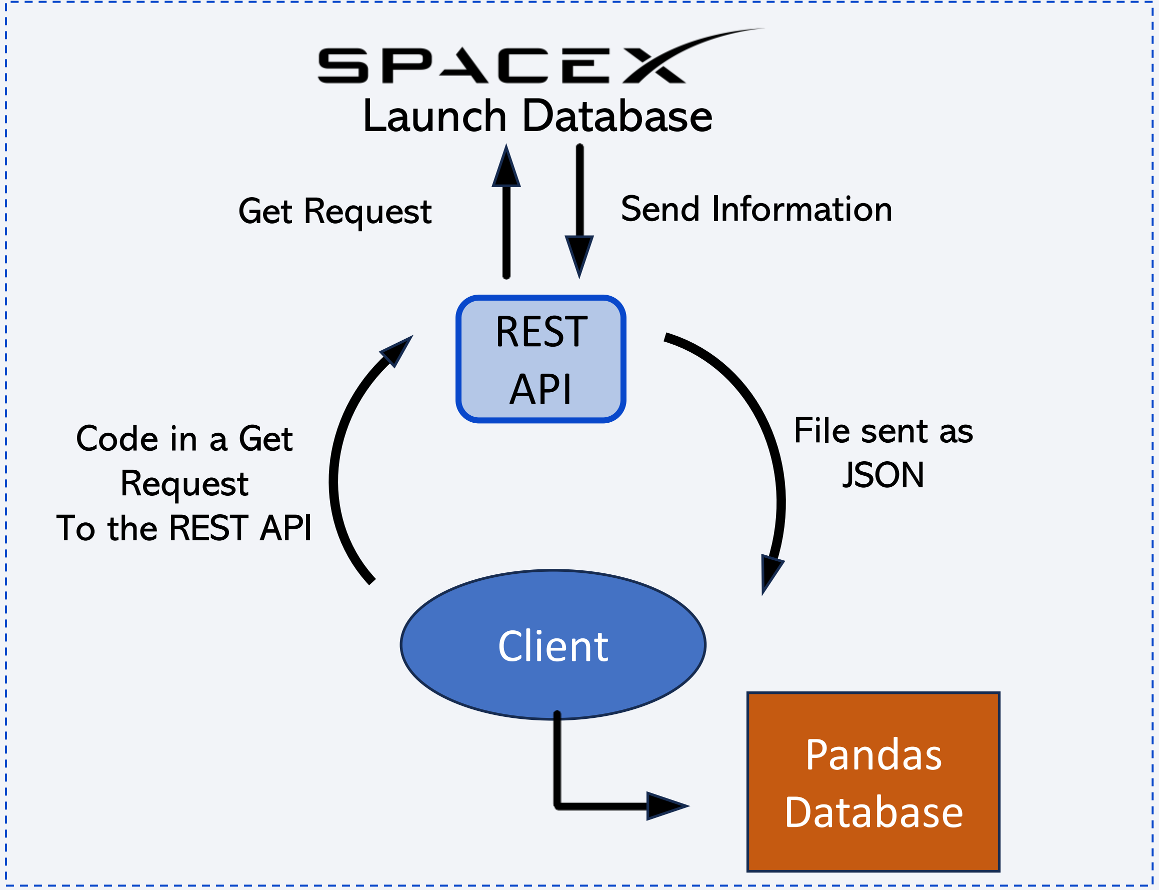
- Data collection methodology:
  - SpaceX uploads their results on their websites. Using their API, we requested their data into Python to analyze. Additionally, Wikipedia has useful information on their website in which we scraped the tables from them
- Perform data wrangling
  - Since this strictly focuses on the Falcon 9 Launch, we first cleaned and parsed the data into a Dataframe. All missing data were replaced by averages and we also added if the launch failed or succeded
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using Machine Learning in scikitlearn, we were able to gather predictions on the locations and the mass required for a successful launch to be effective

Data Collection – GET REQUESTs

- We used a get request to SpaceX API to get our information for the database. Namely:
  - Booster Version (Which booster and its version used for the mission)
  - Launch Site (Where was the mission launched from)
  - Payload Data (How large and massive was the payload the rocket was carrying)
  - Core Data (Result of the landing, gridfins used, core was reused, etc.)
- Thanks to the API and Scraping tables for the rocket information off Wikipedia, we were able to get unfiltered data to start with the analysis.
- We used SpaceX API for the primary dataset, and BeautifulSoup's Request for the Wiki page HTML to facilitate our project

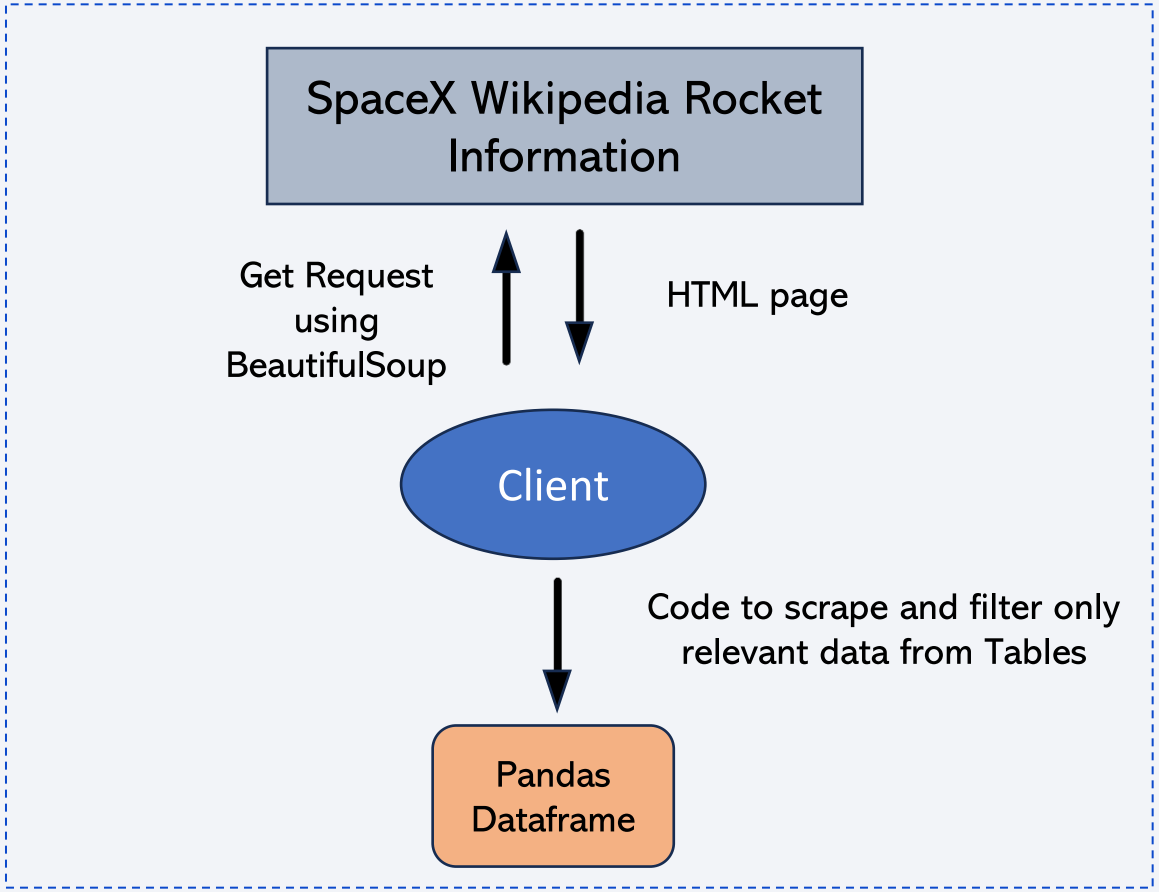
Data Collection – SpaceX API

- Data collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a Json result which was then converted into a Pandas data frame
- Here is the URL to the API notebook used (https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb)



Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using BeautifulSoup and request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.
- Here is the Github URL for the Webscraping notebook used (https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/jupyter-labs-webscraping.ipynb)



Data Wrangling

- After processing the data, we filtered the data to only the *Falcon 9* missions from the *BoosterVersion* column and added them onto our Dataframe we called *df*. We added the mean to the missing data from the columns of *PayloadMass*.
- We added an additional column called class, that shows if the mission was a success or a failure as a 1 or a 0 respectively. This will help later for visualization and classifying the data

- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

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 [22]:
```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)


landing_class
```

Out[22]:
```
[0,
 0,
 0,
 0,
 0,
 0,
 1,
 1,
 0,
 0,
 0,
 0,
 1,
 0,
 0,
 0,
```

In [13]:
```python
df.head(5)
```

Out[13]:

rsion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
con 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
con 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
con 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
con 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
con 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

We can use the following line of code to determine the success rate:

In [14]:
```python
df["Class"].mean()
```

Out[14]: 0.6666666666666666

EDA with Data Visualization

- Here we created visuals of Scatter plots, Bar chart, and a Line graph of the database using the Class column by creating a condition. This condition indicates that if the launch was a success, colour the point in orange, if unsuccessful, colour blue.
- We used the Seaborn and matplotlib libraries in Python to develop these plots.

- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/edadataviz.ipynb

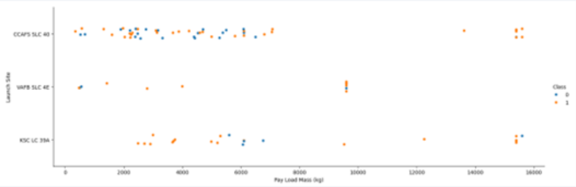


Fig 1: Result of Launch of Payload Mass vs Launch Site

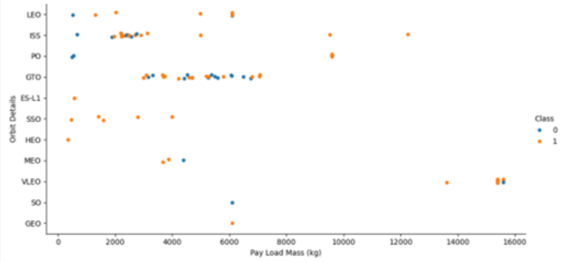


Fig 2: Result of Launch from Payload Mass vs Orbit Details

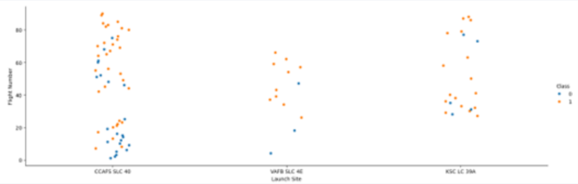


Fig 3: Result of Launch by Launch Site vs Flight Number

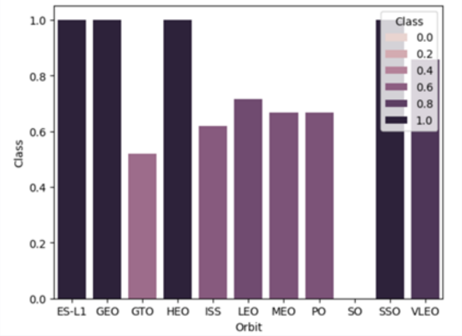


Fig 4: Success Rate of Orbit Details

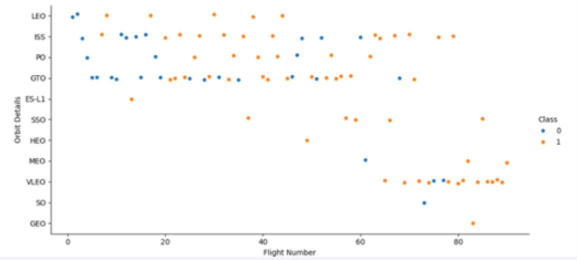


Fig 5: Result of Launch by Flight Number vs Orbit Details

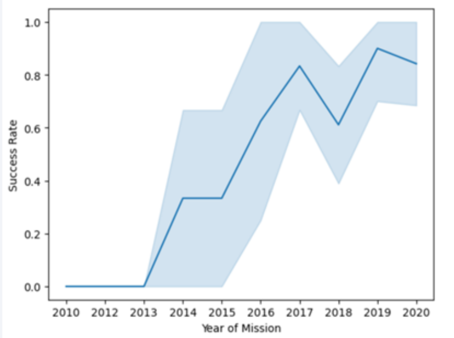


Fig 6: Success Trend over the Years

EDA with SQL

- Selects each unique launch site in the table

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE
```

- Selects the first 5 datapoints in the launch site column where the first three characters are 'CCA'

```
%sql SELECT LAUNCH_SITE FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

- Calculates the sum of the Payload from all of NASA (CRS) missions in total

```
%sql SELECT CUSTOMER, SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER == 'NASA (CRS)'
```

- Calculates the average of the Falcon 9 v1.1 Booster Payload Mass in kg

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

- Results the first successful launch date

```
%sql SELECT min(DATE) FROM SPACEXTABLE WHERE "Landing_Outcome" == 'Success (ground pad)'
```

- Shows the successful landings on a drone ship with a payload carrying between 4000 and 6000kg

```
%%sql
SELECT Booster_Version From SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
AND 4000 < "PAYLOAD_MASS__KG_" < 6000
```

- Displays a count of all the results from the Mission Outcome column

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_COUNT FROM SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

EDA with SQL – Cont.

- Displays the ranking of the highest payload mass of all the Boosters, displaying which booster carried the most to the least

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

- Shows the Date of all failure to land on the drone ship missions, with the booster version and launch site

```
%%sql
SELECT substr(DATE,0,5) AS "Year", substr(DATE,6,2) AS "Month",Landing_Outcome, Booster_Version, Launch_site FROM SPACEXTABL
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND DATE Like '2015%'
```

- Ranking of the number of all landing outcomes between the dates of 2010-06-04 and 2017-03-20

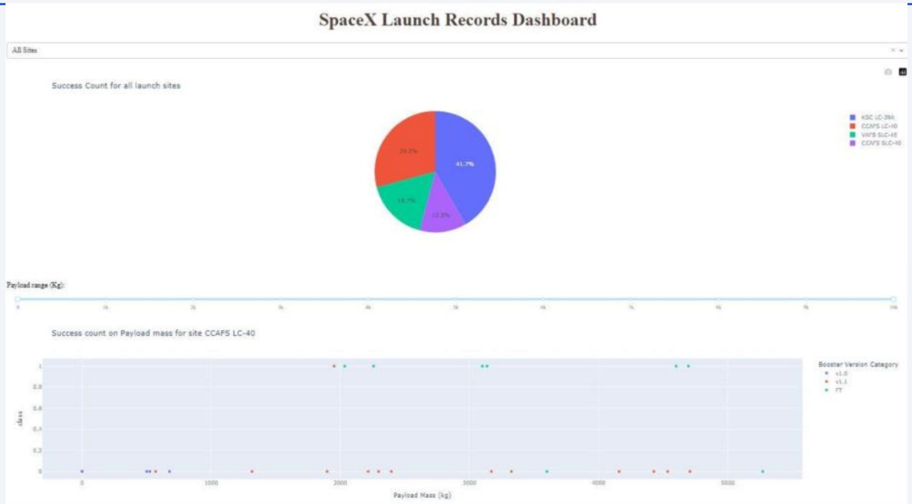
```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS TOTAL_NUMBER FROM SPACEXTABLE
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY TOTAL_NUMBER DESC
```

- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- In creating a map to display the resulting area of all launch sites, we add:
  - Markers: to display the name of the Sites
    - Marker Clusters: When displaying multiple points to see how concentrated the number is accordingly to the data
  - Circles: to show where each Site is at an overview
  - Lines: to display how far to the closest coastline is
- This was add to clarify where and how large of a distance between each site, their similarities to the coast it is, and how each unique launch site's success rate. This is to give a clear understanding of what conditions and variables coincide with the launch sites.
- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash



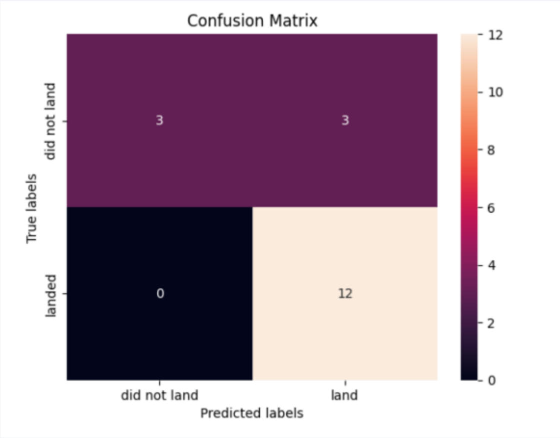
- The Pie chart is a visual tool to help explain how each location's success rate is categorized as
- Continuously, the scatter plot displays the success rate of each location based off its range of payload mass selected. This is to further understand the relation between the payload mass and the success rate of each location
- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

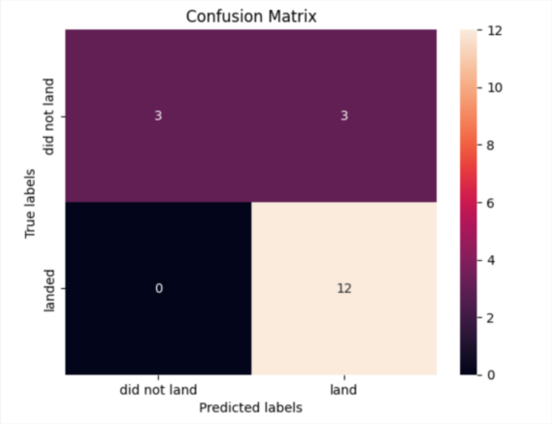
- Using sci-kit learn library, we created multiple machine learning methods to predict the data and if the boosters will land or not creating a Confusion Matrix shown.
- We used:
  - Logarithmic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K-Nearest Neighbor (KNN)

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.944444
KNN	0.833333

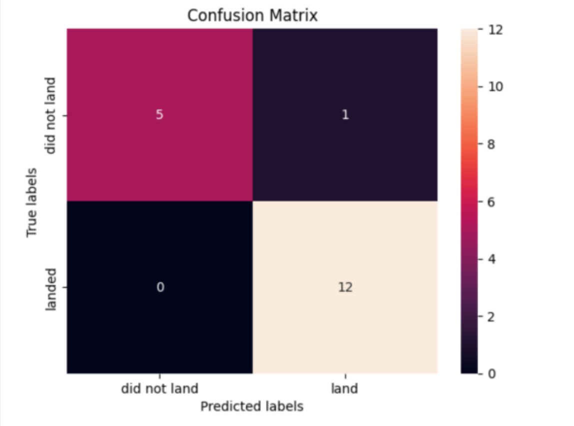
- We then calculated the score from the different methods and found which is the best to use for creating a model.
- https://github.com/Vinyodiny/Space-X-Prediction-Launch-Data-Analysis-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



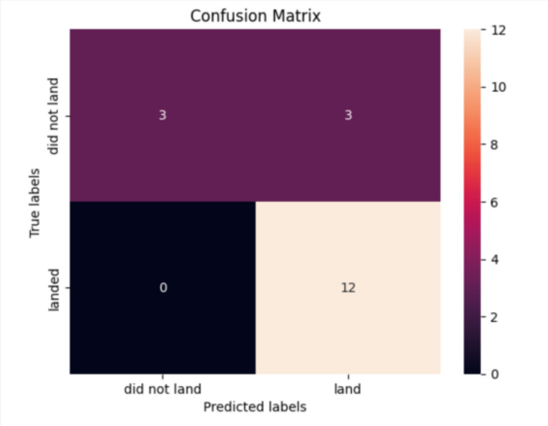
Logarithmic Regression



SVM



Decision Tree



KNN

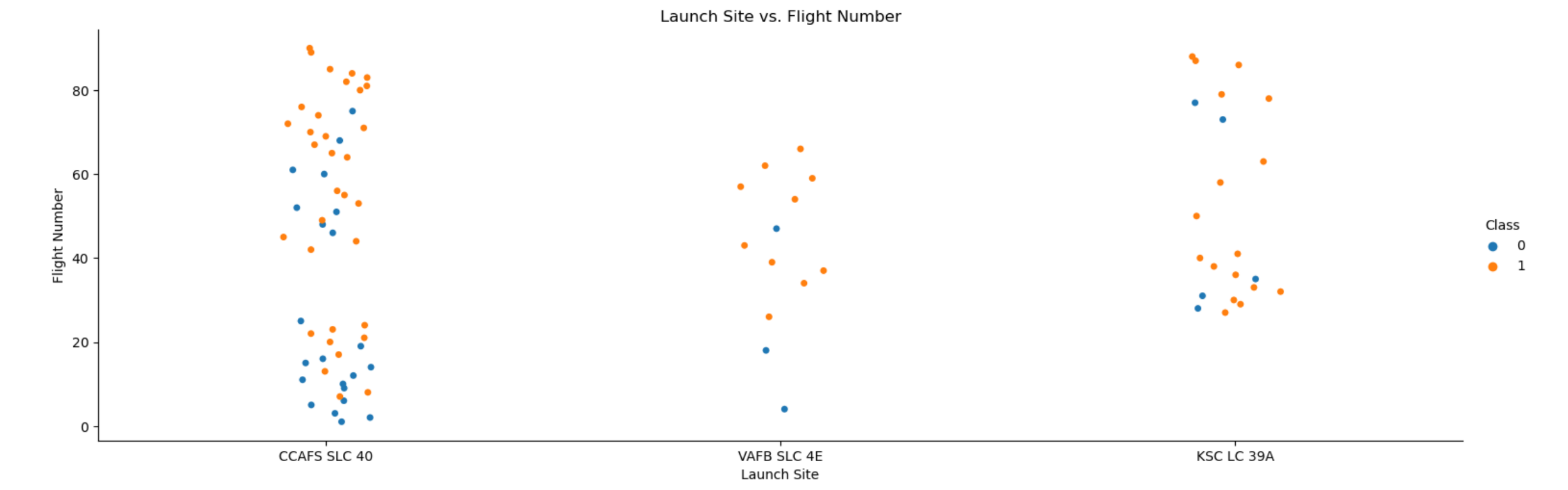
Results

- Exploratory data analysis results
  - We saw that the launch sites have differing results as the payload increases. So if we want to send a large payload mass onto our boosters, we should choose CCAPS SLC 40 as they have a higher success rate for higher payloads.
  - Additionally, depending on where we are sending the payload orbit location, it should affect our results as well.
- Interactive analytics demo in screenshots
  - This explained that all the launch sites that are lighter or heavier have different success rates. It tells us where the boosters should launch from and depending on its payload, the Falcon 9 will land successfully
- Predictive analysis results
  - For future models, we should follow a Decision Tree model as it results in the highest accuracy rate according to the data

Section 2

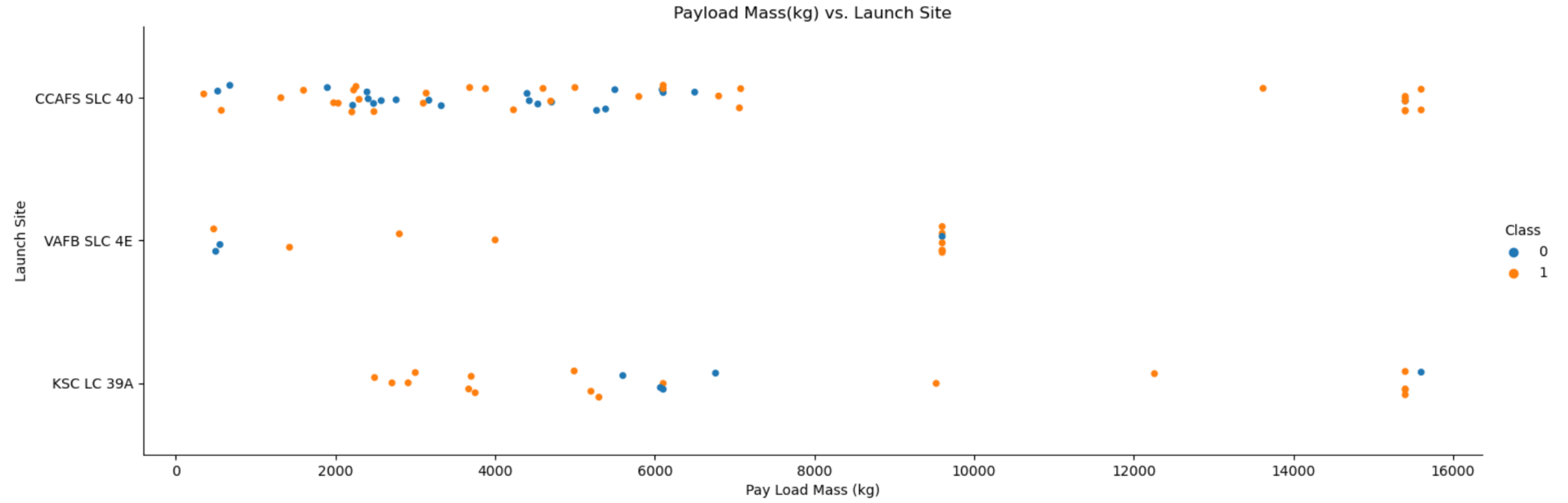
Insights drawn from EDA

Flight Number vs. Launch Site



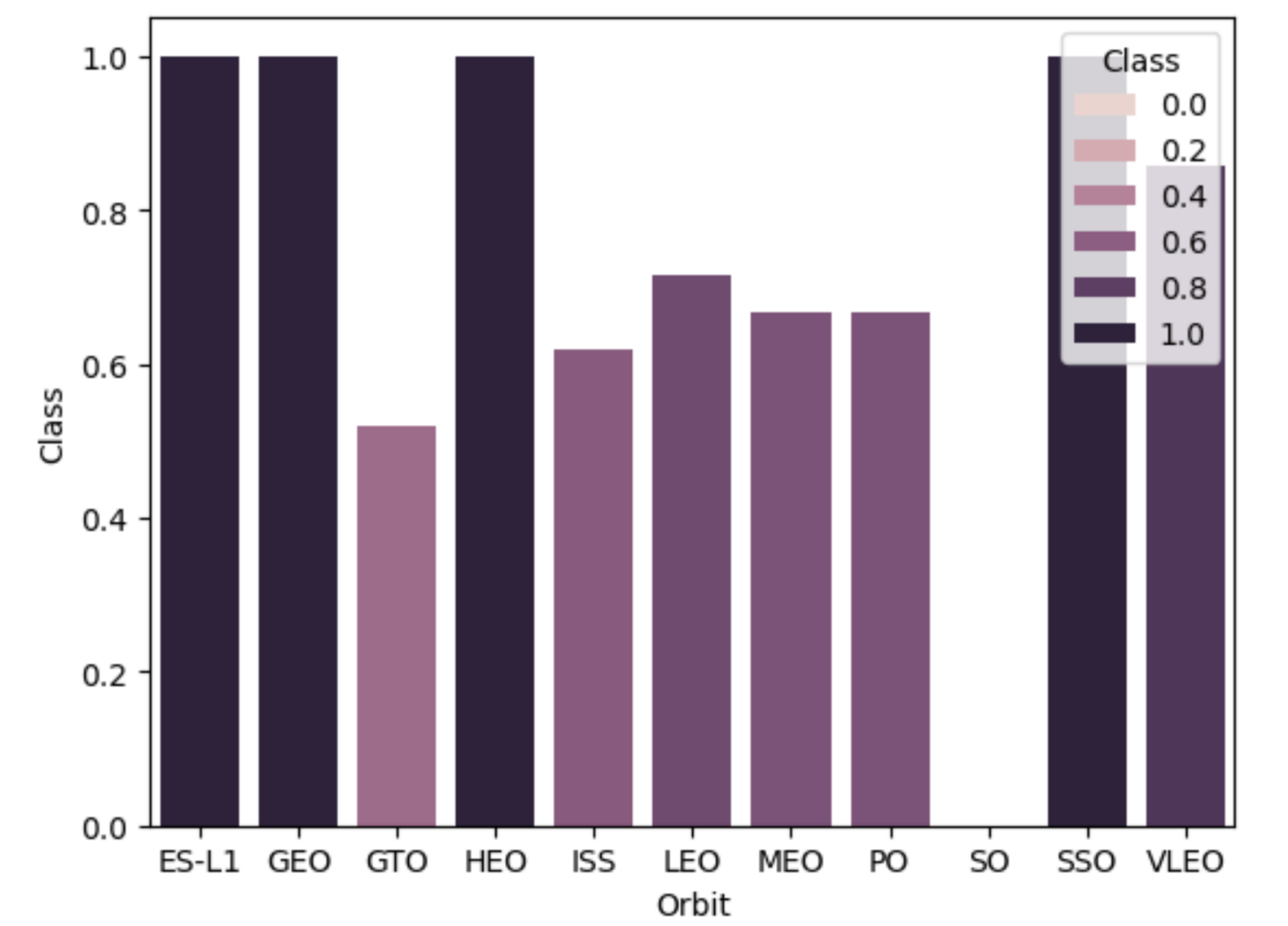
We see that the higher the launch number is, the more successful the launches are launched. This is from learning from each previous launch and anticipating the errors of each launch

Payload vs. Launch Site



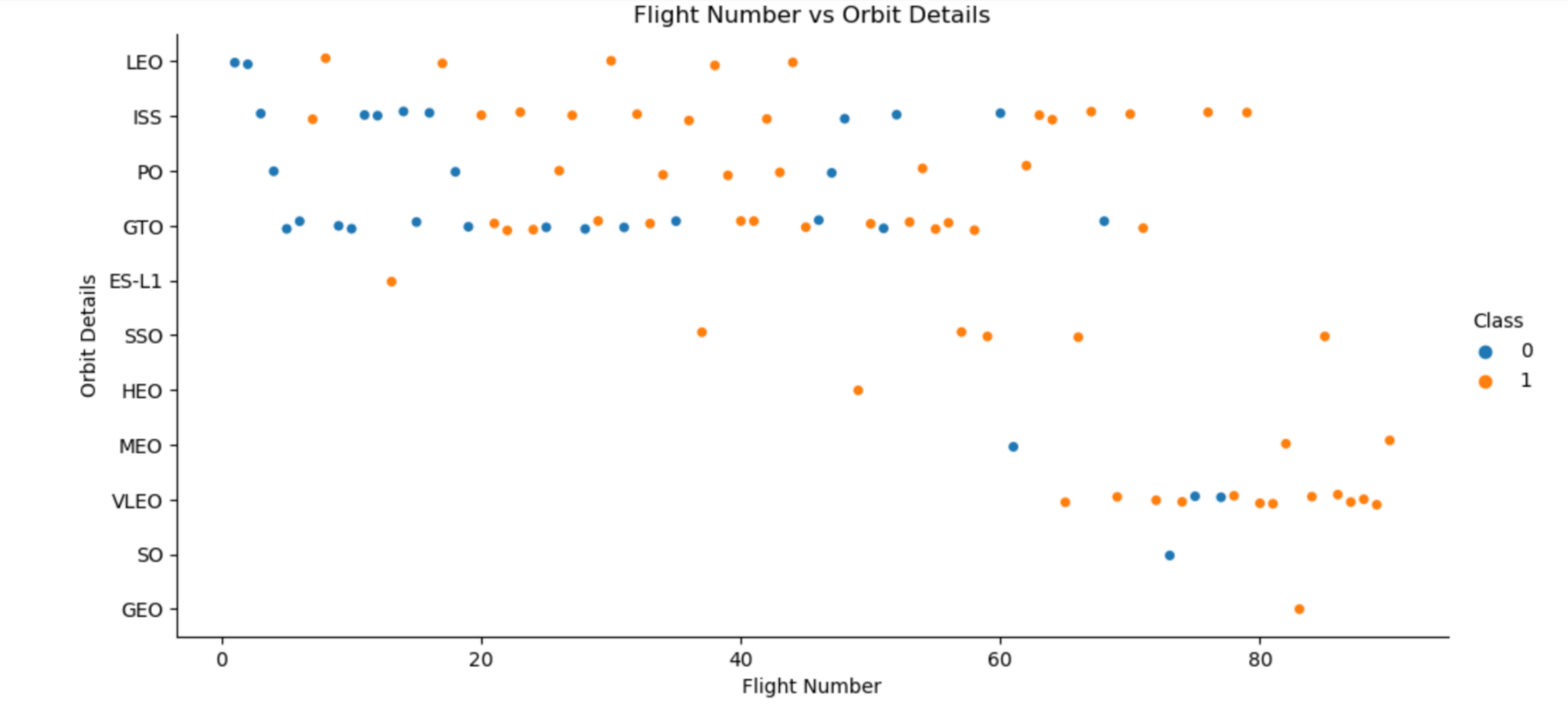
Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



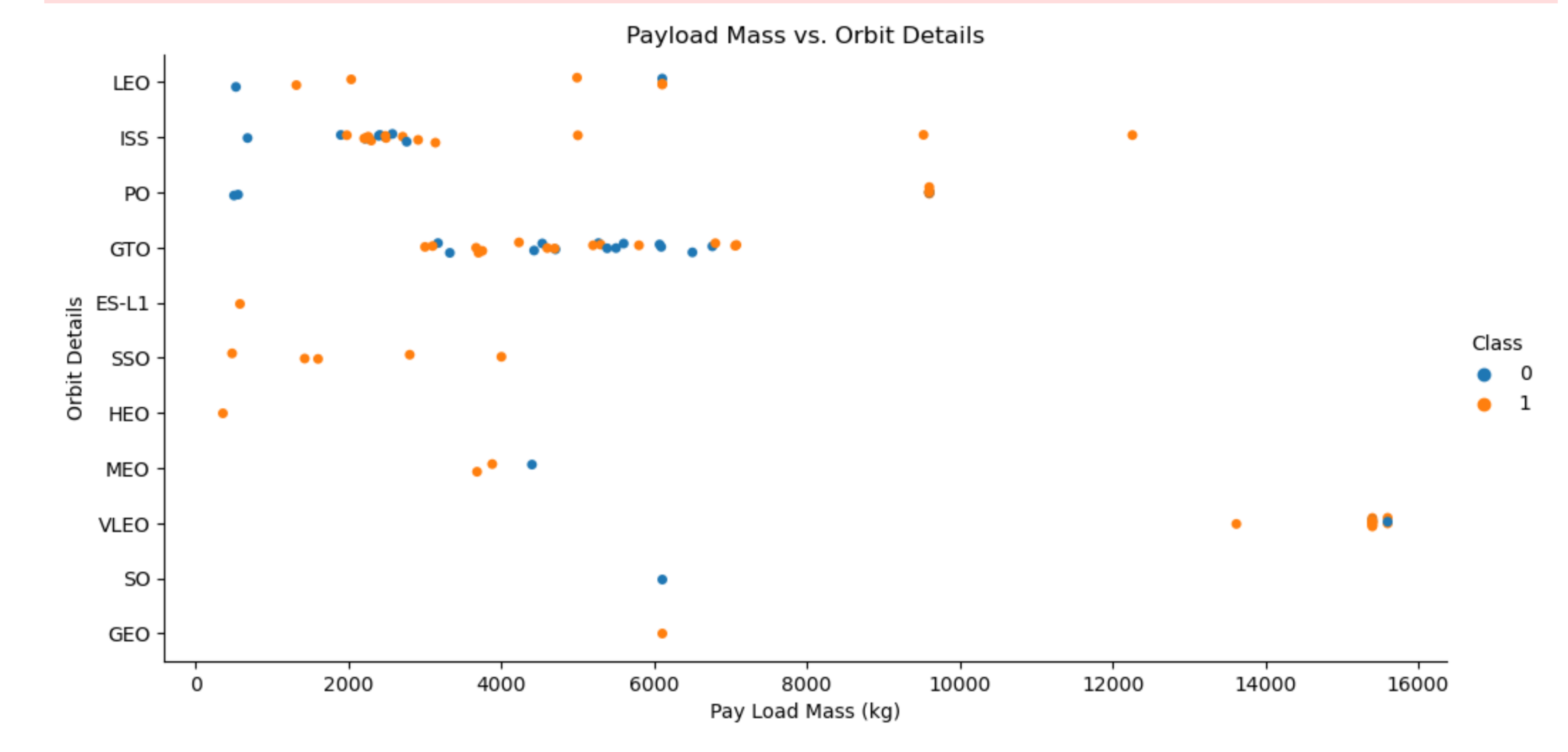
Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

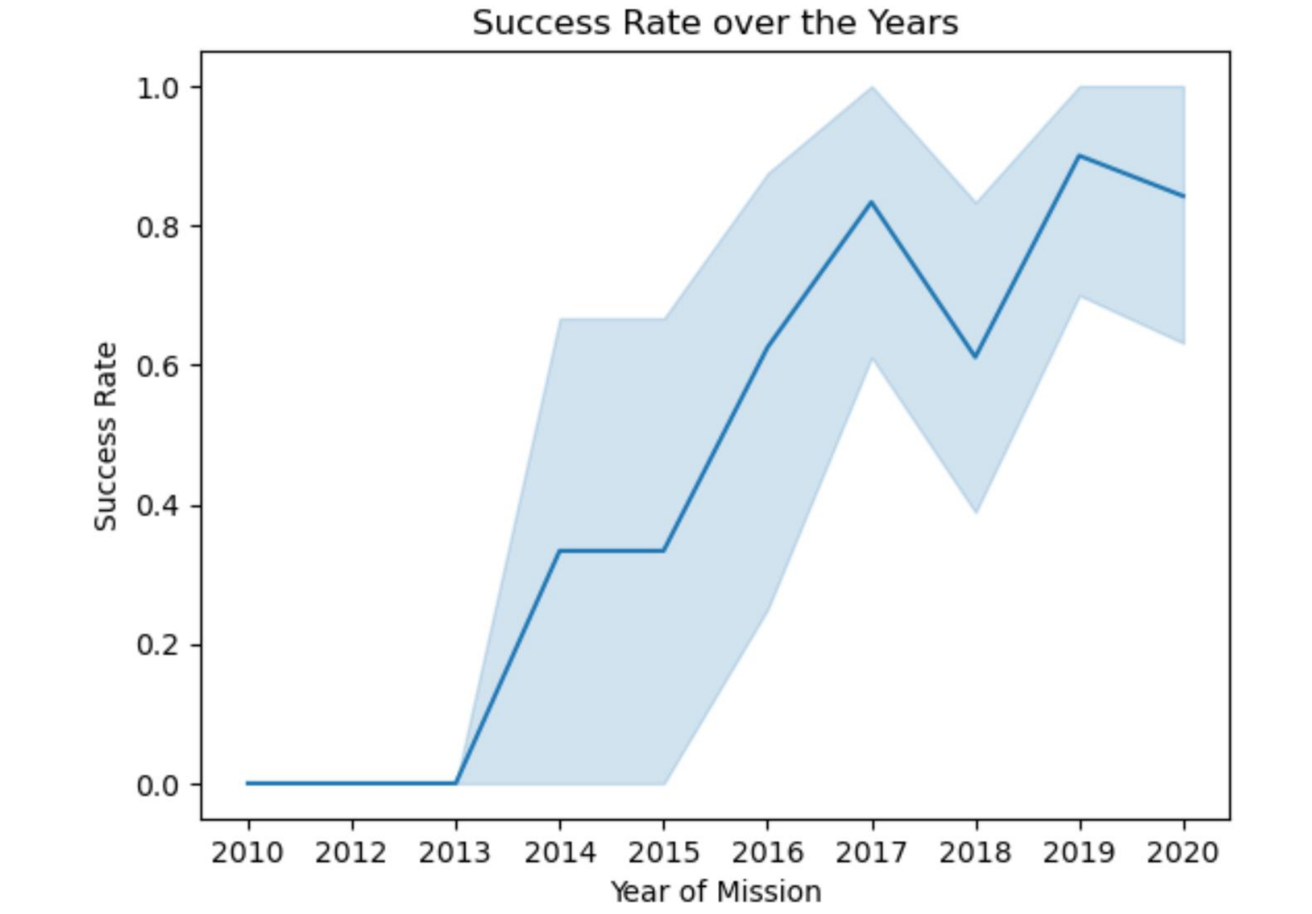
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Selects each unique launch site in the table

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Selects the first 5 datapoints in the launch site column where the first three characters are ‘CCA’

```
%sql SELECT LAUNCH_SITE FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

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

Total Payload Mass

- Calculates the sum of the Payload from all of NASA (CRS) missions in total

```
%sql SELECT CUSTOMER, SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER == 'NASA (CRS)'
```

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

Customer	SUM(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- Calculates the average of the Falcon 9 v1.1 Booster Payload Mass in kg

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Results the first successful launch date

```
%sql SELECT min(DATE) FROM SPACEXTABLE WHERE "Landing_Outcome" == 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Shows the successful landings on a drone ship with a payload carrying between 4000 and 6000kg

```
%%sql
SELECT Booster_Version From SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
AND 4000 < "PAYLOAD_MASS__KG_" < 6000
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- Displays a count of all the results from the Mission Outcome column
- Present your query result with a short explanation here

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_COUNT FROM SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

- Displays the ranking of the highest payload mass of all the Boosters, displaying which booster carried the most to the least
- Present your query result with a short explanation here

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

```
 * 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

- Shows the Date of all failure to land on the drone ship missions, with the booster version and launch site

```
%%sql
SELECT substr(DATE,0,5) AS "Year", substr(DATE,6,2) AS "Month",Landing_Outcome, Booster_Version, Launch_site FROM SPACEXTABL
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND DATE Like '2015%'
```

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

Year	Month	Landing_Outcome	Booster_Version	Launch_Site
2015	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Ranking of the number of all landing outcomes between the dates of 2010-06-04 and 2017-03-20
- Present your query result with a short explanation here

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS TOTAL_NUMBER FROM SPACEXTABLE
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY TOTAL_NUMBER DESC
```

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

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

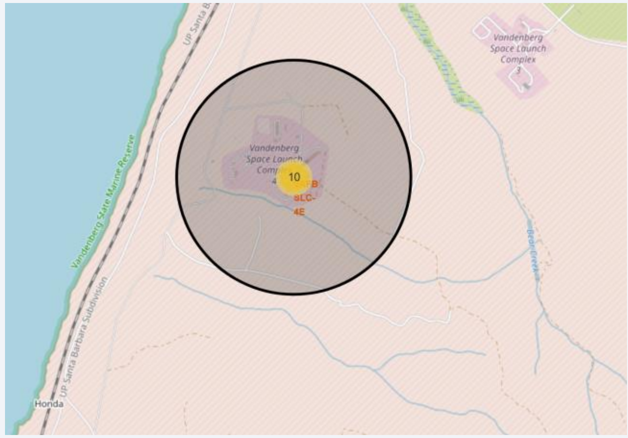
Section 3

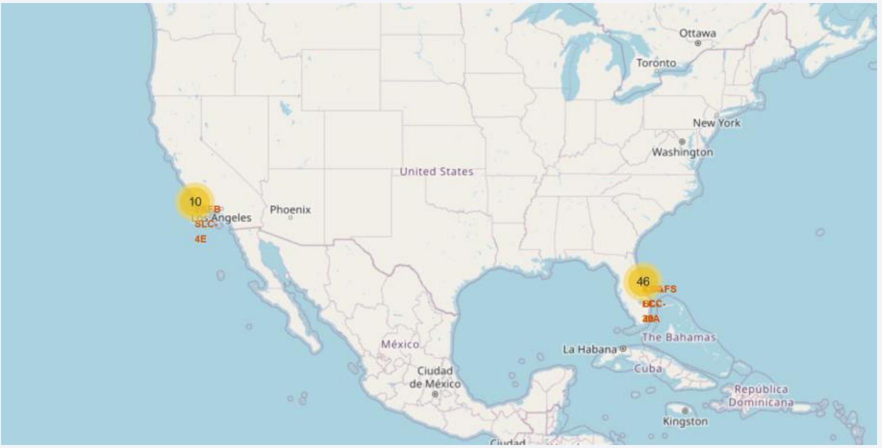
Launch Sites Proximities Analysis

Launch Sites in the US

All launch sites are in proximity to the Equator, (located southwards of the US map). Also all the launch sites are in very close proximity to the coast

Launch Site Success Clusters





- From the color-labeled markers in marker clusters, we sare be able to easily identify which launch sites have relatively high success rates.

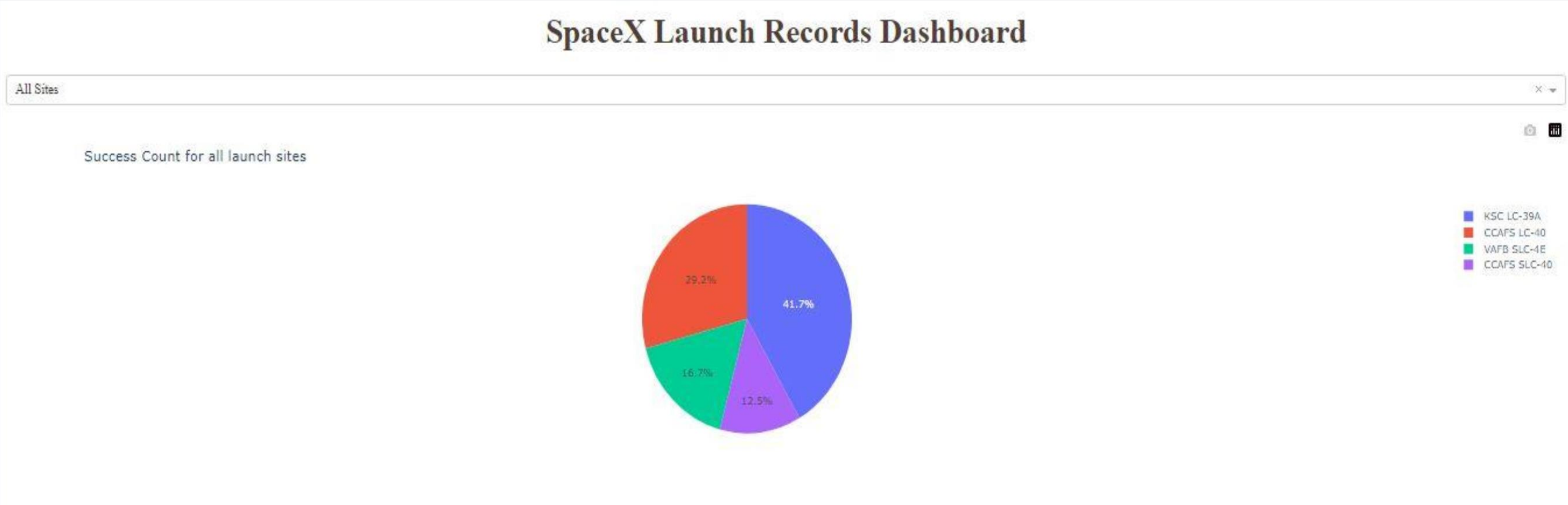
Distance Between VAFB SLC-4E and Coast Line

All Launch sites are relatively close to a coast line or a railway/highway. This is to facilitate transportation from the build site of the rocket and its launch site

Section 4

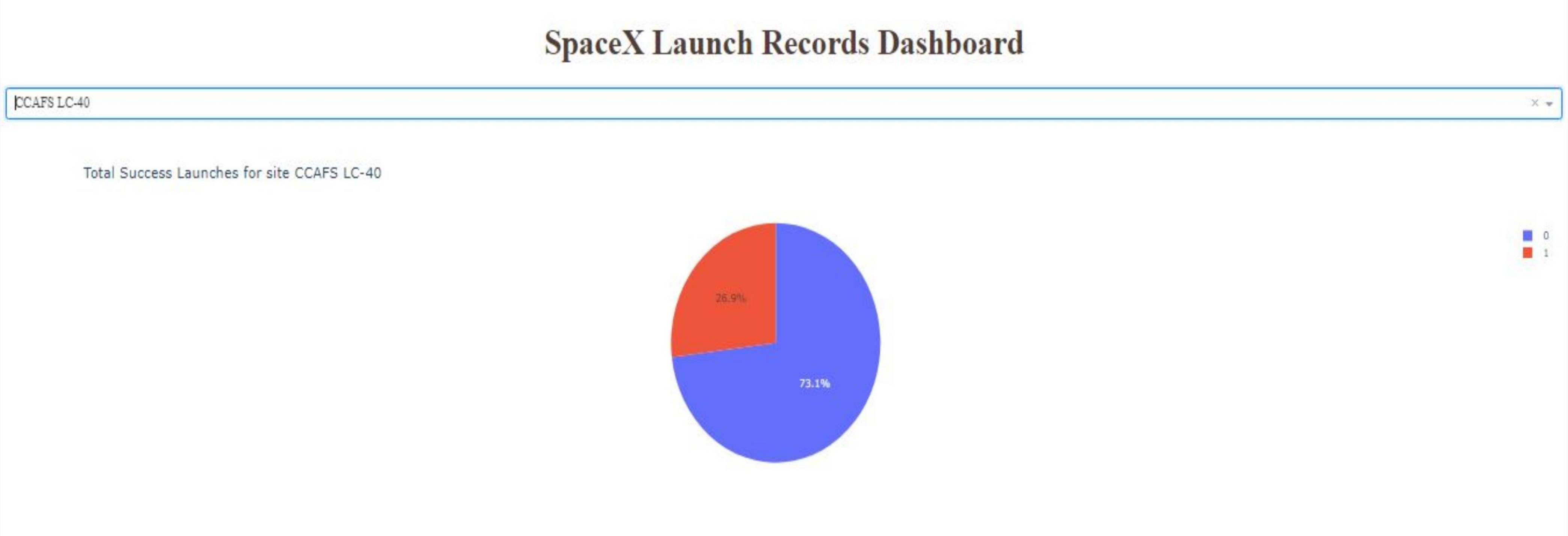
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



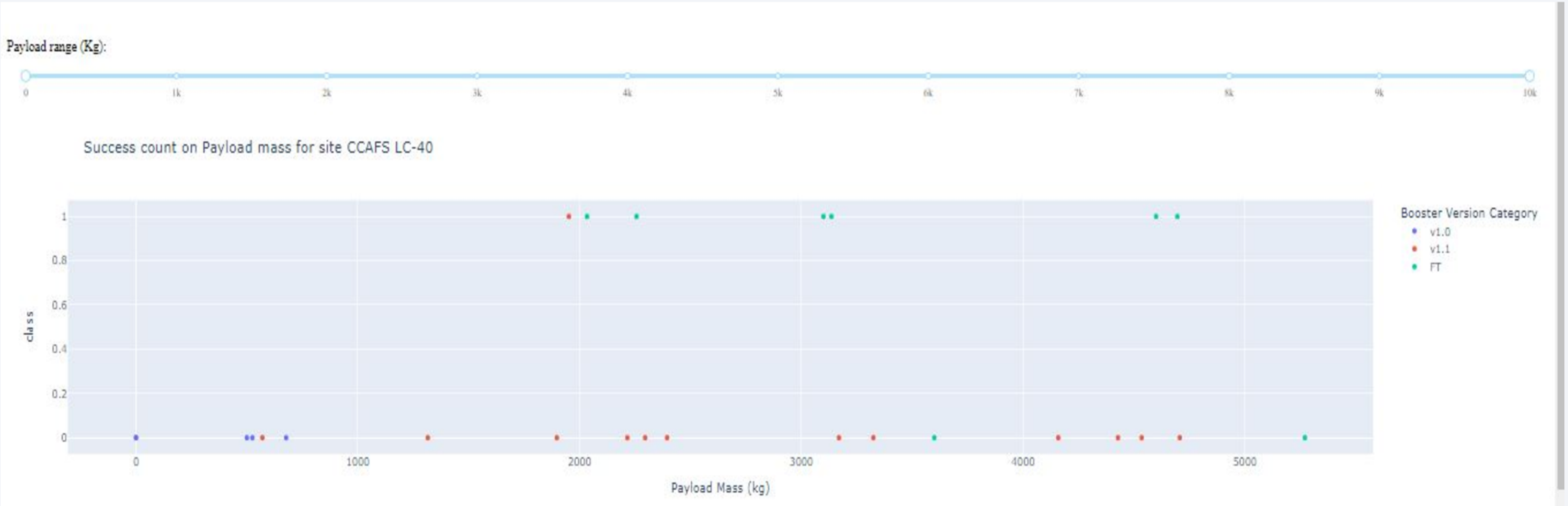
- Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAF SLC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

<Dashboard Screenshot 2>



- Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success against 27% failed launches

<Dashboard Screenshot 3>



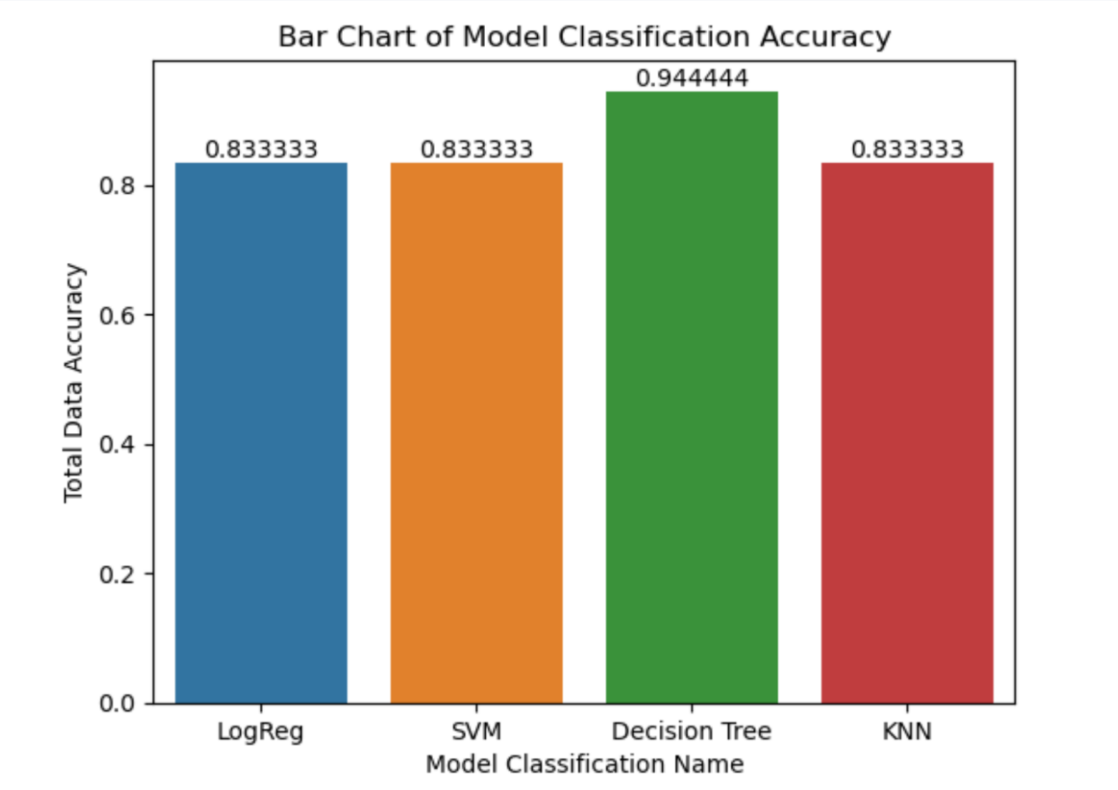
- For Launch site CCAFS LC-40 the booster version FT has the largest success rate from a payload mass of >2000kg

Section 5

Predictive Analysis (Classification)

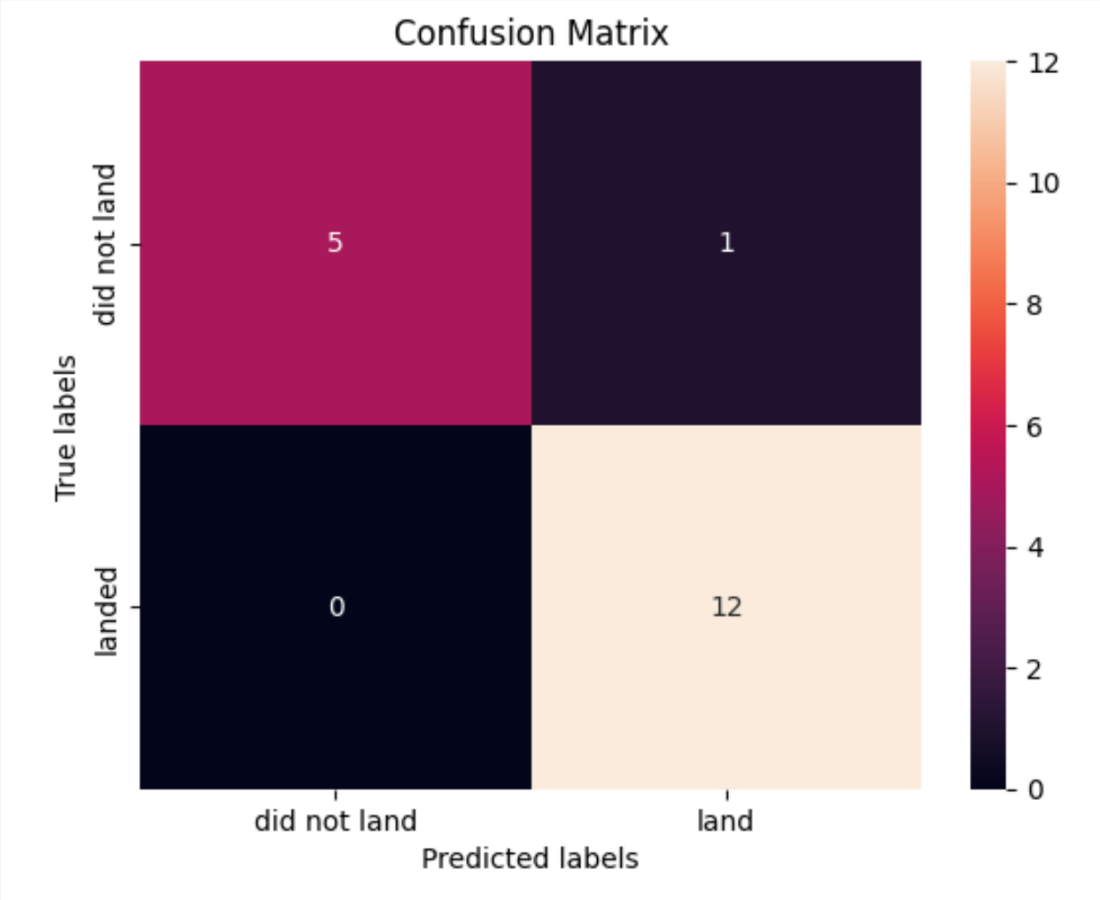
Classification Accuracy

- The model with the highest accuracy is the Decision tree model with a value of 94%. This shows the highest value while the other models show a high confidence, not as much as the Decision Tree



Confusion Matrix

- The Confusion Matrix for the Decision Tree is shown. This accurately shows that the model predicted that when it would land, it did, and when it shouldn't, it shouldn't.
- It had only 1 error when it predicted that would land when in reality, it didn't.



Conclusions

- From our analysis, we can see that it we should carefully choose where we should launch our *Falcon 9* rocket.
- If our required payload is on the lower side, we could choose any launch site really. But as our payload grows, we should towards choosing CCAFS SLC-40 as they exceed in launching payloads <10,000kg
- Depending on where our Payload Orbit desired location is also is important. If our payload exceeds 10,000kg, we will stay in the ISS to VLEO range orbit.
    - If exceeding 13,000kg, then that range reduces to just the VLEO as there have not been a lot of mission experience going below that with a large payload.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!