IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

David Camilo Muñoz
30/03/2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Launching rockets to space are a very expensive activity
- The first stage has the major impact on total cost
- So, recovering the first stage for reuse is the key point

- Is it possible to predict if a Falcon 9 first stage will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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 was collected using two methodologies
  - Making a request to the SpaceX API
  - Using webscrapping

- More information, look at [Data_Collection_API.ipynb] y [Data Collection with Webscrapping.ipynb]

Data Collection - SpaceX API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

Check the content of the response

```
In [8]: print(response.content)
```


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images
2.imgbox.com/3c/0e/T8iJcSN3_o.png","large":"https://images2.imgbox.com/40/e3/GypSkayF_o.png"},"reddit":{"campaign":null,"launc
h":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/wa
tch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-spacex-inaugural-falcon-1-rocket-lost-launc
h.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z","static_fire_dat
```

Data Collection - Scraping

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:
```
# use requests.get() method with the provided static_url
requests.get(static_url)

# assign the response to a object
data = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- Data was processed through:
  - extraction from a URL
  - organization into a dataframe by parsing HTML tables
  - removing unnecessary columns
  - keeping only Falcon 9 launches records
  - dealing with missing values
  - creating dummy variables to categorical columns

- More information, look at [EDA.ipynb]

EDA with Data Visualization

- In this section, we explored data to find patterns
- More information, look at EDA_with_Visualization.ipynb

EDA with SQL

- In this section, we explored data to:
  - Find the launch sites used by SpaceX
  - What booster versions were used
  - When occurred the first successful land
  - Range of payload
- More information, look at EDA_with_SQL.ipynb

Build an Interactive Map with Folium

- In this section, we plotted into a map:
  - circles centered on launch sites
  - marks indicating success / failed launches on each site
  - lines and distance from sites to points of interest, like the coast, railways, cities, etc
- More information, look at Interactive_Visual_Analytics_with_Folium.ipynb

Build a Dashboard with Plotly Dash

- In this section, we built a dashboard to:
  - Discover which site has the best performance
  - Understand the relation among payload mass, booster version and the success / fail land
- More information, look at Spacex_dash_app.py

Predictive Analysis (Classification)

- In this section, we built four machine learning models:
  - K Nearest Neighbours
  - Decision Tree
  - Logistic Regression
  - SVM
- For each model:
  - Data was standardized and splitted into training and testing sets
  - Models were trained using GridSearchCV to find the best hyperparameters
- More information, look at Machine_Learning_Prediction.ipynb

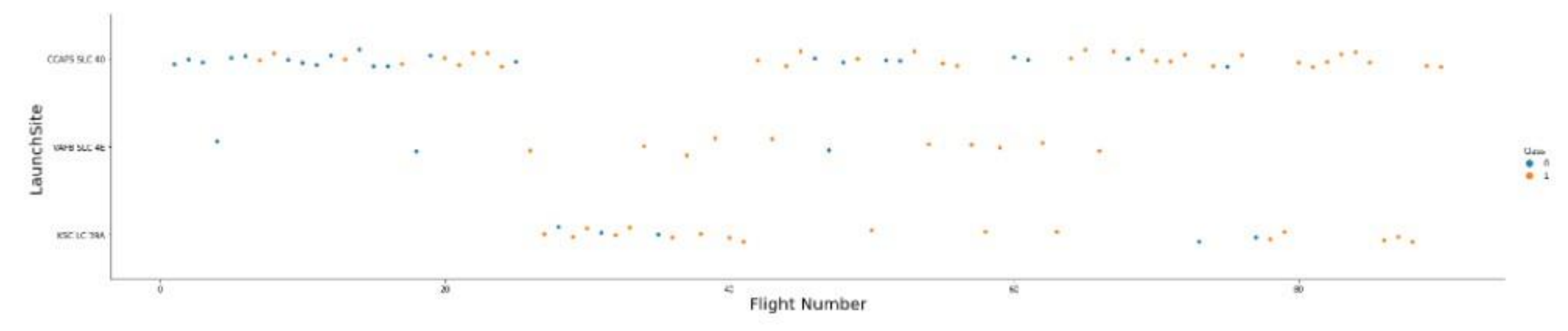
Results

- The best ML model was Decision Tree, with the following best parameters:
  - Criterion = entropy
  - max_depth = 18
  - max_features = sqrt
  - min_samples_leaf = 4
  - nim_samples_split = 2
  - splitter = random

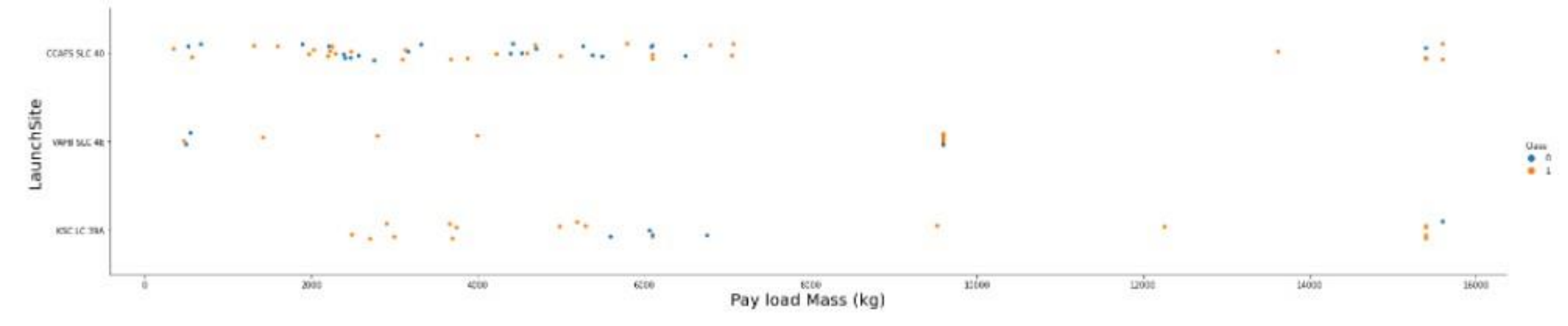
Section 2

Insights drawn from EDA

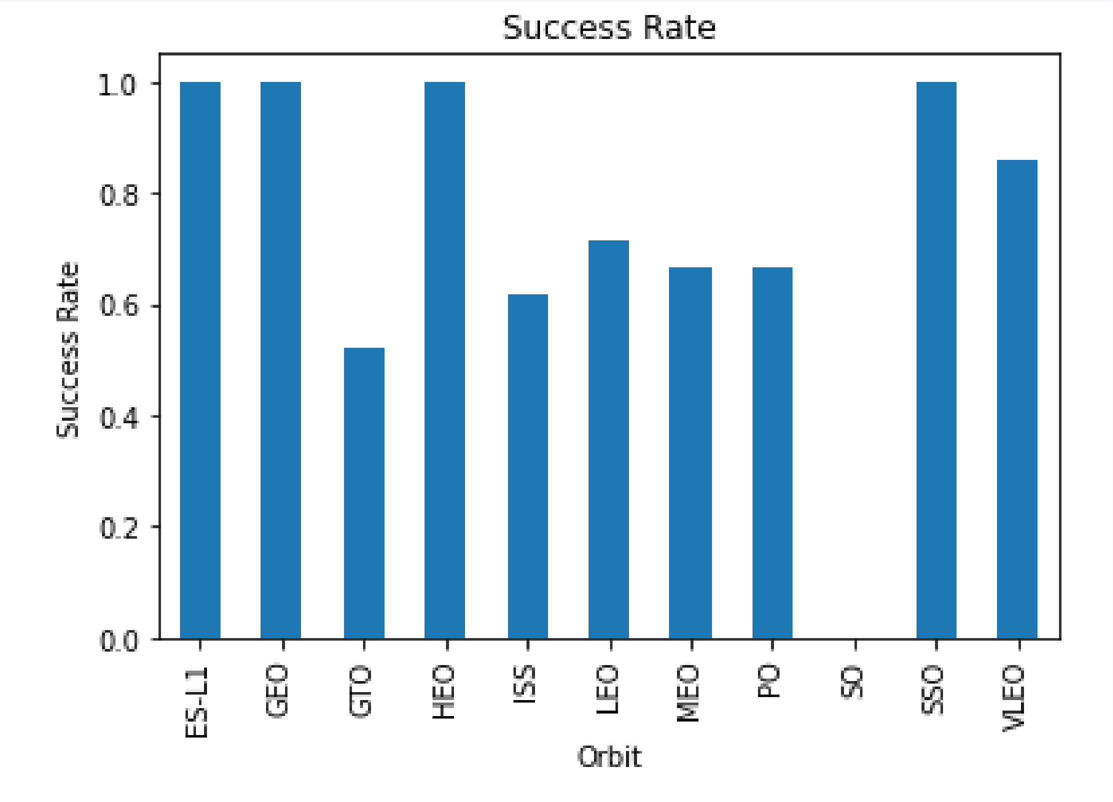
Flight Number vs. Launch Site



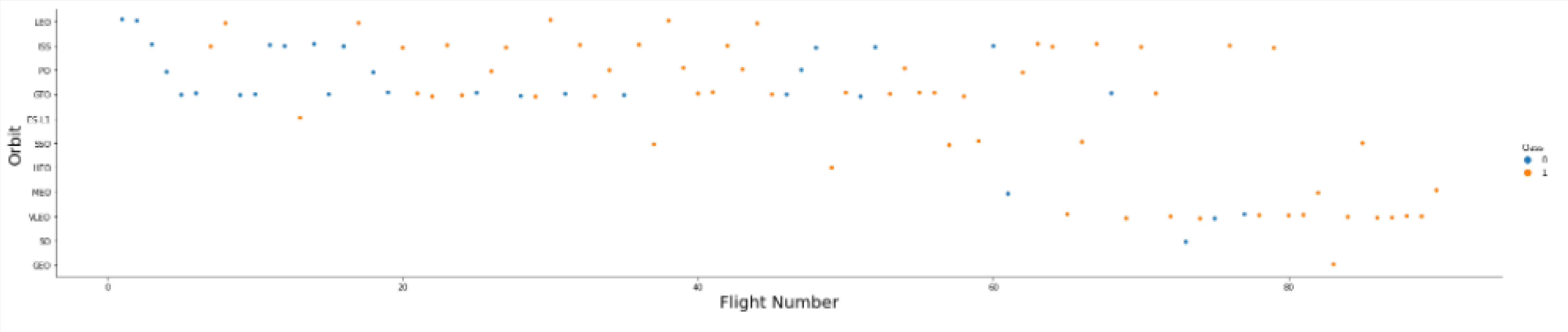
Payload vs. Launch Site



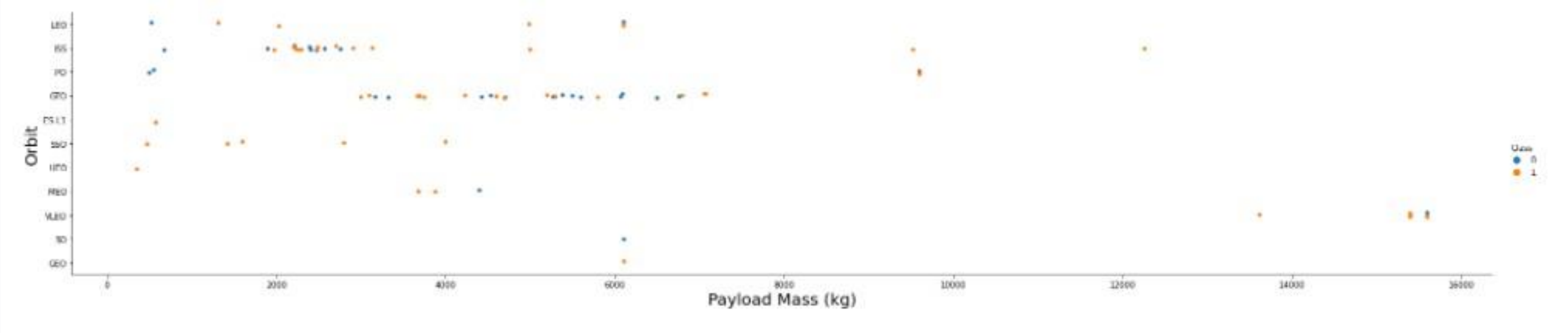
Success Rate vs. Orbit Type



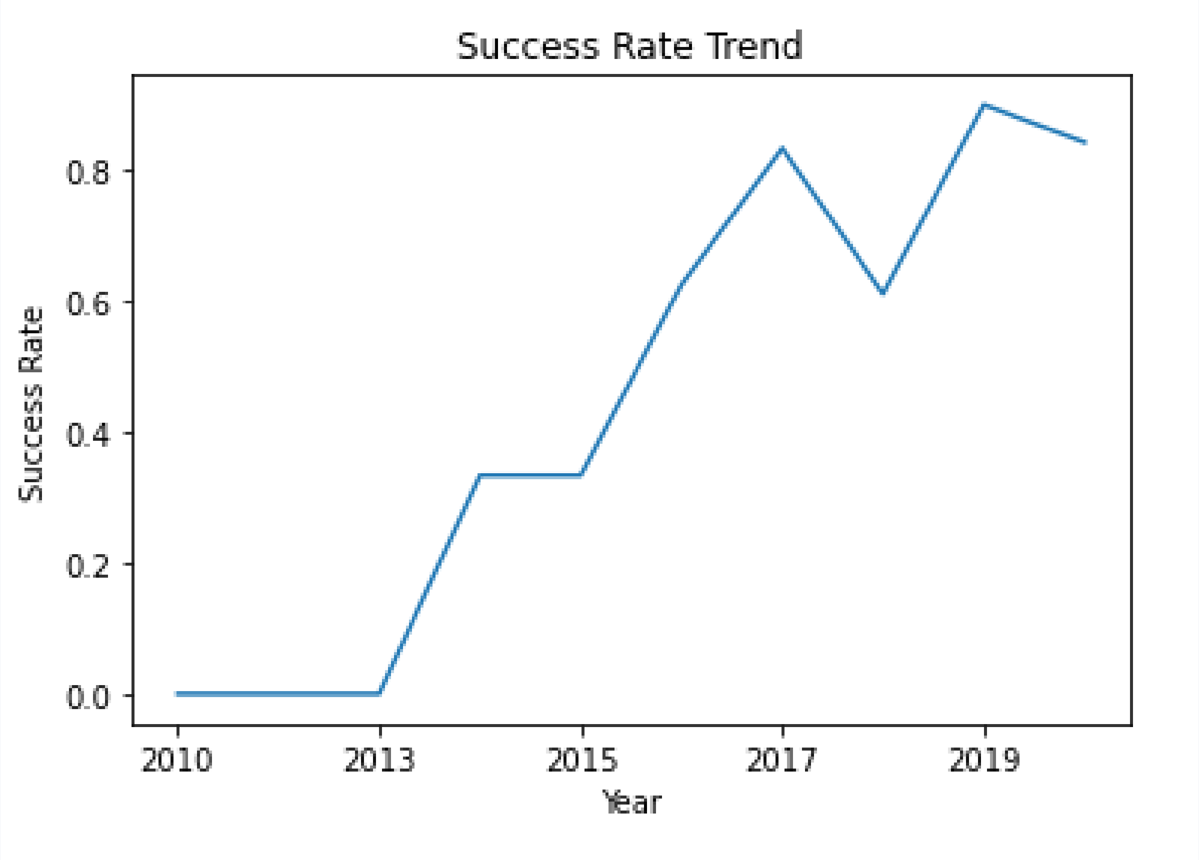
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

In [5]:
```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBLNEW
```

```
 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

In [9]:
```
%sql SELECT * FROM SPACEXTBLNEW WHERE LAUNCH_SITE LIKE 'CCA%'LIMIT 5
```

```
 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[9]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC/40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC/40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC/40	SpaceX CRS/1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC/40	SpaceX CRS/2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
In [11]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBLNEW WHERE CUSTOMER LIKE UCASE('%NASA (CRS)%')

 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[11]:

total_payload_mass
48213

Average Payload Mass by F9 v1.1

```
In [13]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXTBLNEW WHERE BOOSTER_VERSION LIKE '%F9 v1.1%'

 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.

Out[13]:
```

avg_payload_mass
2534

First Successful Ground Landing Date

```
In [23]: %sql SELECT MIN(DATE) FIRST_DATE FROM SPACEXTBLNEW WHERE LANDING__OUTCOME LIKE '%ground pad%' and LANDING__OUTCOME LIKE '%Succes
s%'

 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[23]:

first_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [24]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBLNEW WHERE LANDING__OUTCOME LIKE '%Success%' and LANDING__OUTCOME LIKE '%drone ship%' a
         nd PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

          * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
         Done.

Out[24]:
```

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

Total Number of Successful and Failure Mission Outcomes

```
In [29]: %sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBLNEW GROUP BY MISSION_OUTCOME

 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[29]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
In [30]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBLNEW WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBLNEW)

 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.

Out[30]:
```

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

In [31]:
```
%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBLNEW WHERE LANDING__OUTCOME = 'Failure (drone ship)' and
YEAR(DATE) = 2015
```

```
 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[31]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC/40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC/40

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

```
In [34]: %sql SELECT LANDING__OUTCOME, COUNT(*) FROM SPACEXTBLNEW WHERE GROUP BY LANDING__OUTCOME ORDER BY COUNT(*) DESC
```

```
 * ibm_db_sa://jdc79296:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

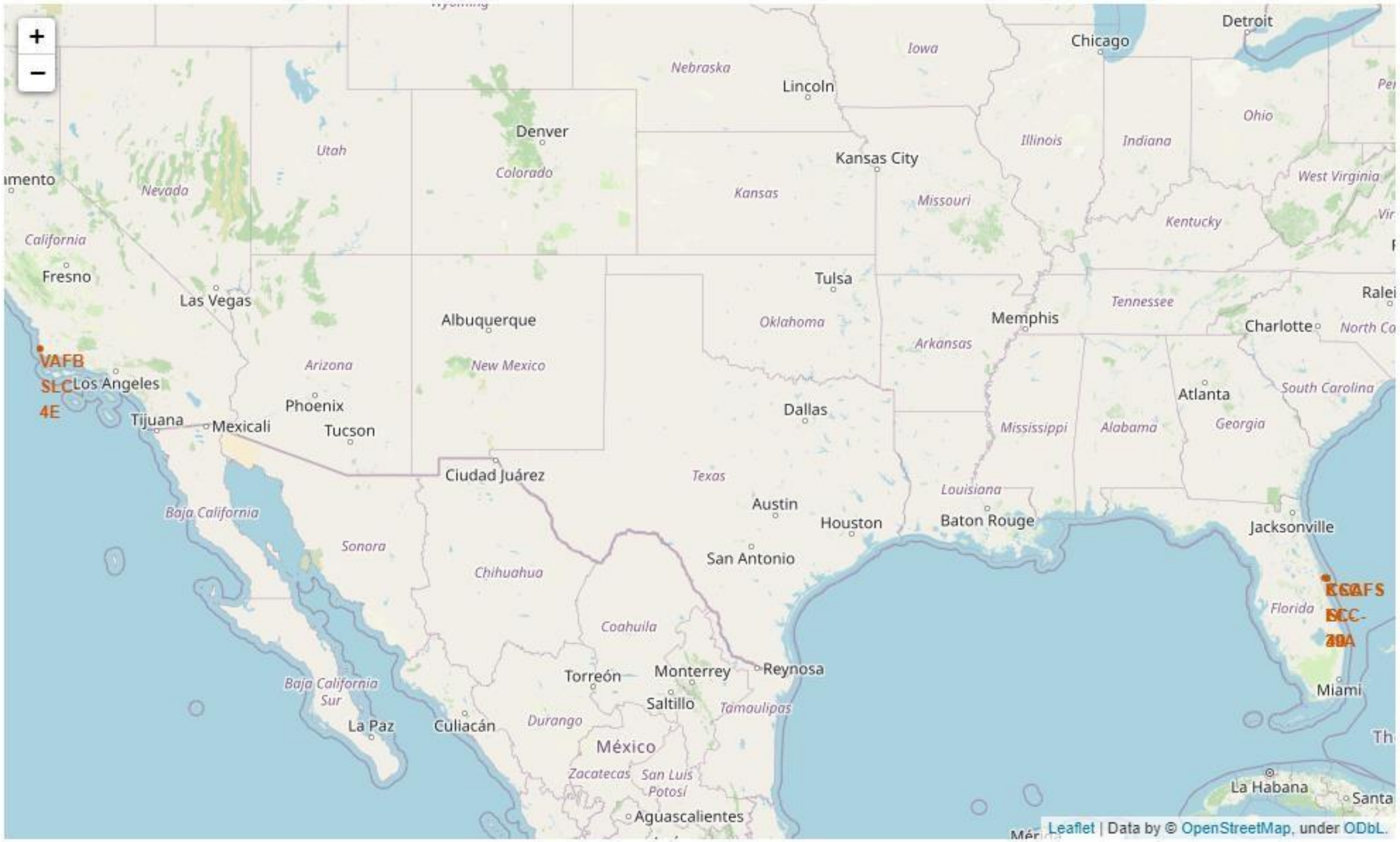
Out[34]:

landing__outcome	2
Success	38
No attempt	22
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

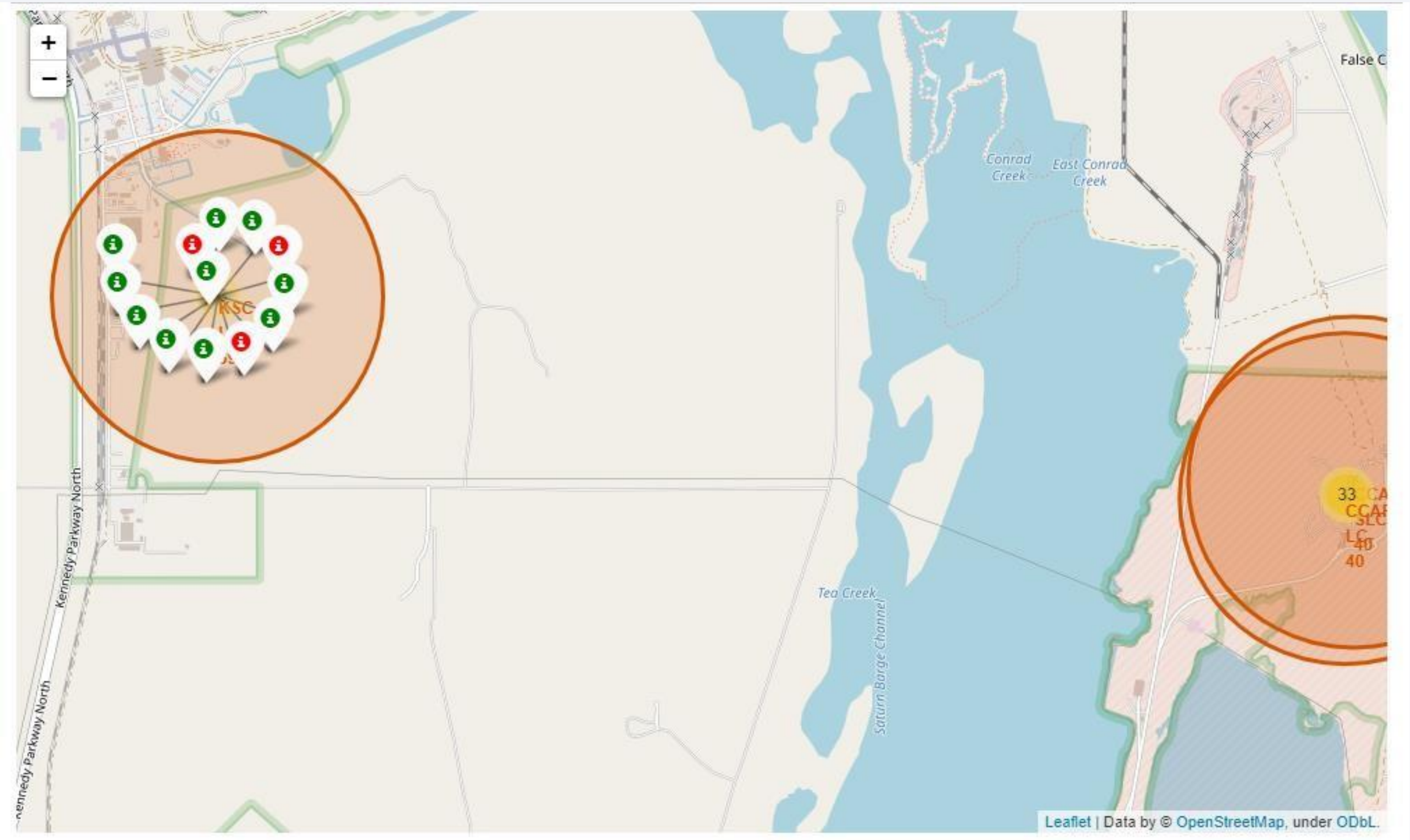
Section 3

Launch Sites Proximities Analysis

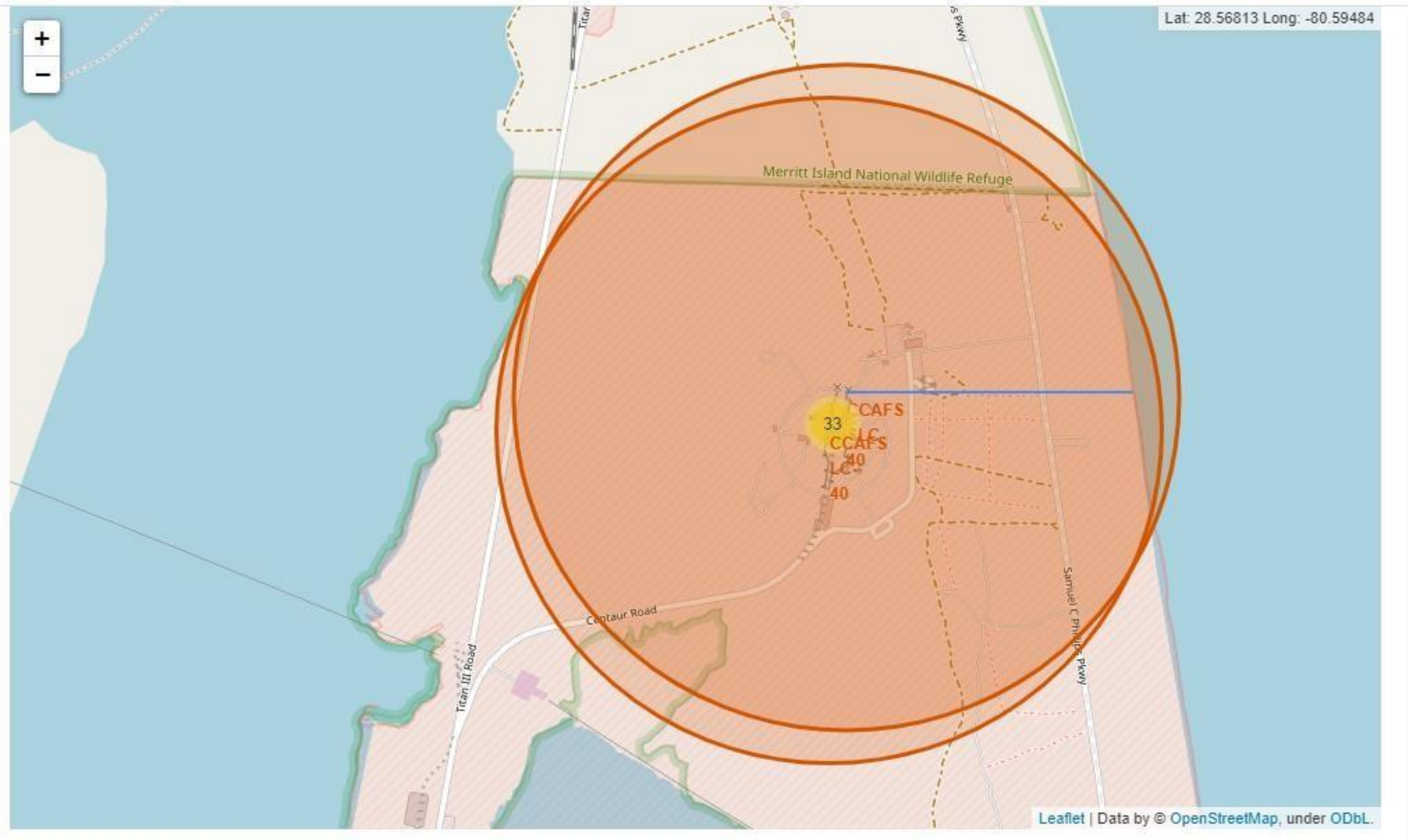
Site Locations



Launch Site - Successful / Failed Launches



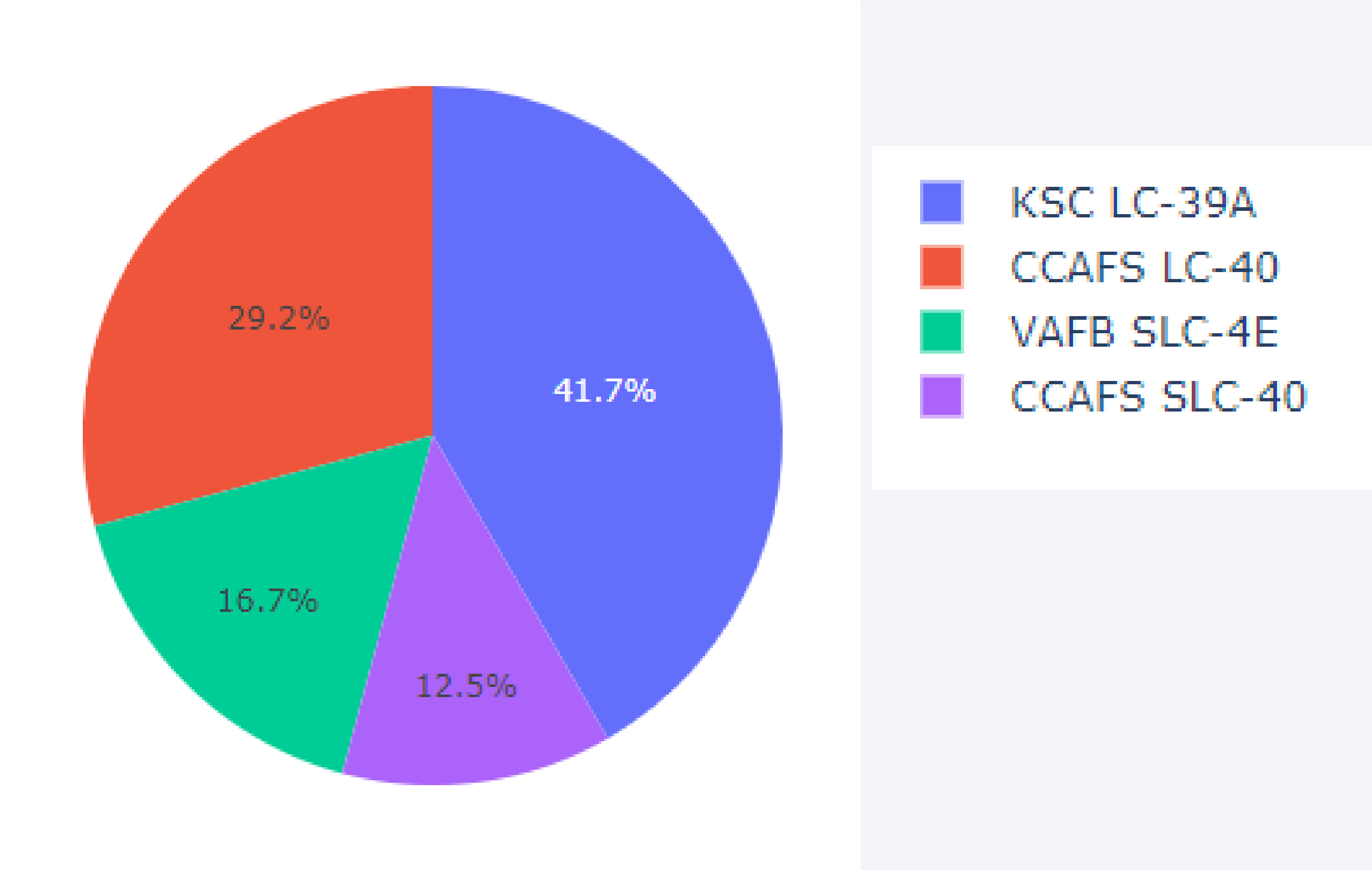
Launch Sites - Proximity



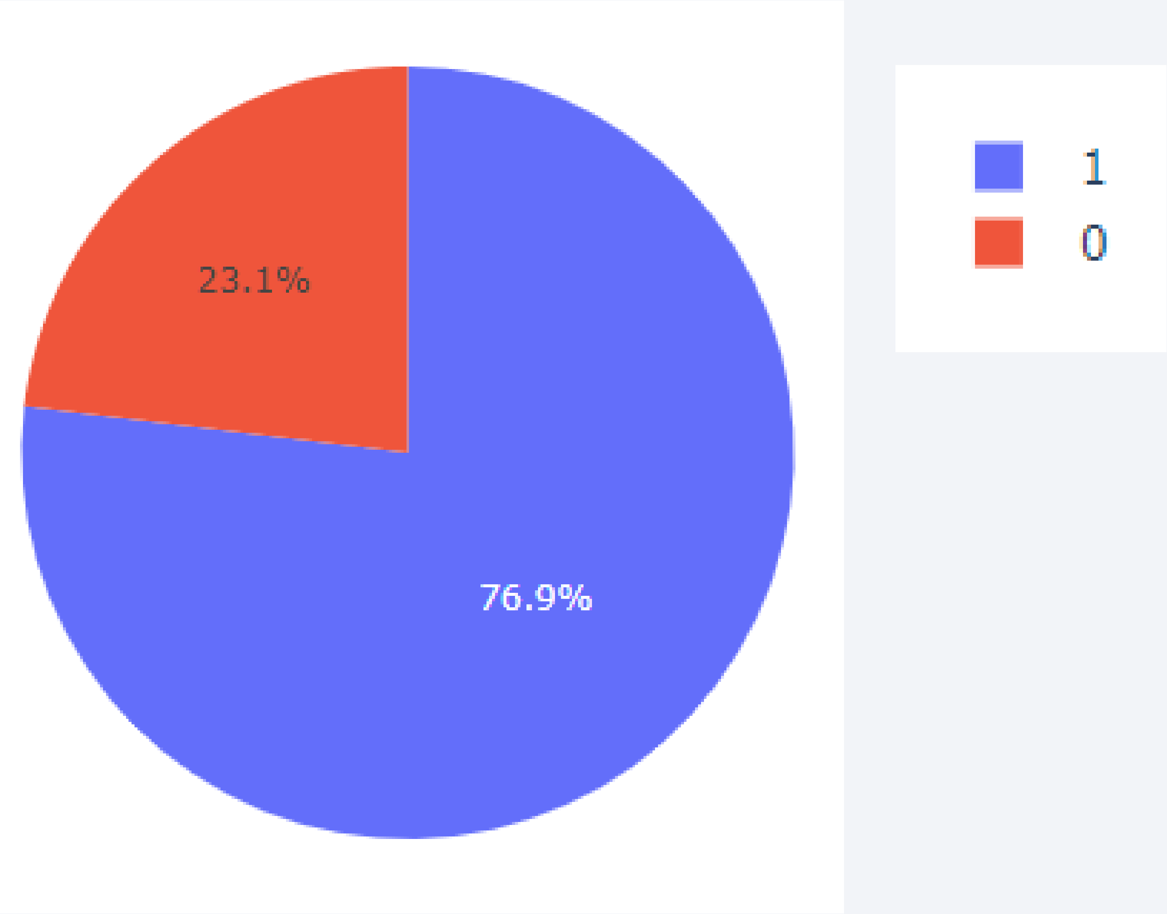
Section 4

Build a Dashboard with Plotly Dash

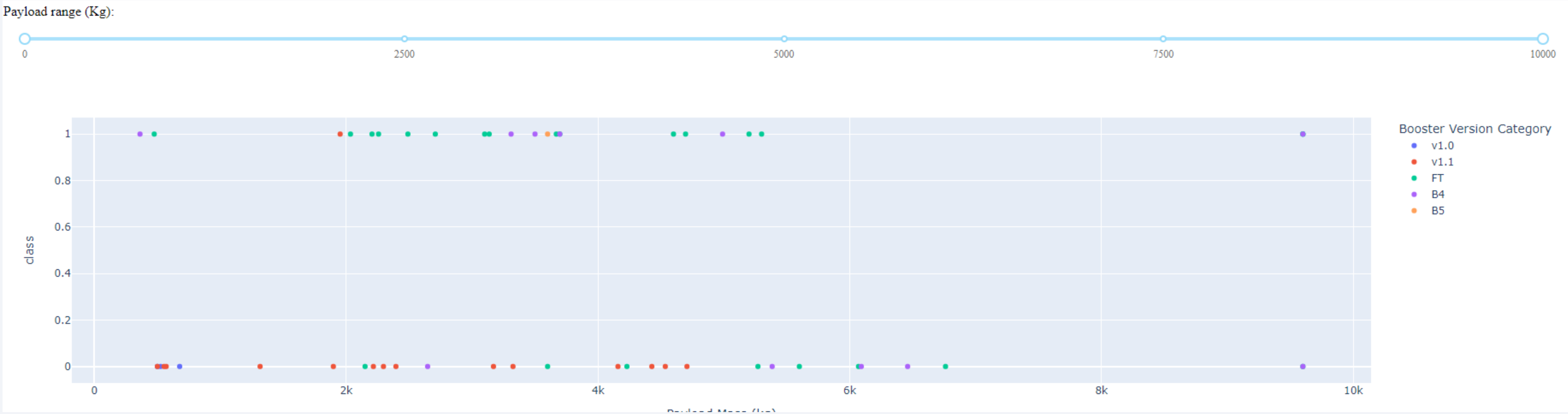
Total Success Launches



Total Success Launches for site KSC LC-39A



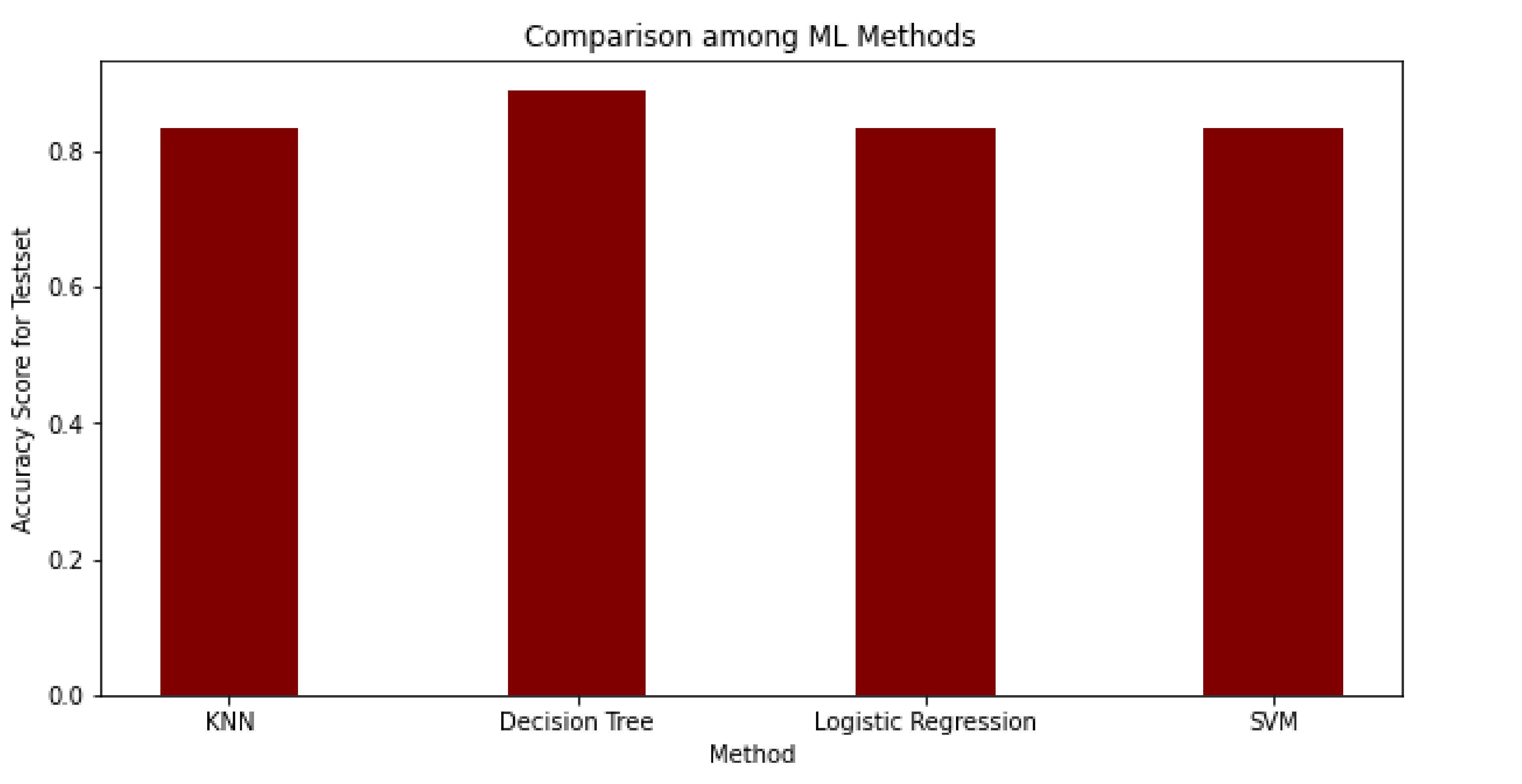
Payload x Success/Failed Launches by Site



Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix

Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

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!