APPLIED DATA SCIENCE CAPSTONE PROJECT

ROCKET BOOSTER LANDING PREDICTION

Yutong He

2024/10/28

IBM Developer SKILLS NETWORK

OUTLINE

- Executive summary
- Part 0: Introduction
- Part 1: Methodology — Data collection, data wrangling, exploratory data analysis (EDA), interactive visual analytics, predictive analysis.
- Part 2: Results — EDA, launch site analysis, dashboard, predictive analysis.
- Part 3: Conclusion
- Part 4: Appendix

EXECUTIVE SUMMARY

- This project made use a number of data science methods, such as data collection via application programming interface (API) and web scrapping, preprocessing to handle missing data, exploratory data analysis, and regression modeling for predictive analysis.

- Results are summarized in various visual forms, i.e., charts and graphs, using data visualization libraries in Python and web-based dashboards.

PART 0: INTRODUCTION

- Project background
  - SpaceX's Falcon 9 rocket significantly reduces the cost per launch, at an advertized 65 million USD instead of over 165 million USD by other providers. Therefore, the cost estimates heavily rely on accuratley predicting whether Falcon 9 boosters would land successfullly.
- Problem definition
  - Given the historical data of Falcon 9, can we predict whether the next launch will be successful? If so, how well can our predict so?

PART 1: METHODOLOGY

- Data collection
- Data wrangling
- Exploratory data analysis (EDA): data visualization and SQL
- Interactive visual analytics: maps via Folium, dashboards via Plotly Dash
- Predictive analysis: classification models via sklearn

PART 1: METHODOLOGY - DATA COLLECTION

- Method 1: Application programming interface (API)
- Process:
    - Make HTTP request to SpaceX API.
    - Parse the requested json data. Turn it into a pandas dataframe. Clean it.
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.1_API_spacex.ipynb.

PART 1: METHODOLOGY - DATA COLLECTION

- Method 2: Web scraping
- Process:
  - Extract Falcon 9 launch records from HTML tables on Wikipedia.
  - Parse and turn data into pandas dataframe. Clean the data.
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.2_webscraping_spacex.ipynb.

PART 1: METHODOLOGY - DATA WRANGLING

- Process:
    - Identify missing values in each attribute.
    - Identify which columns are categorical / numerical.
    - Determine the number of launches per launch site / orbit type / landing outcomes (success, fail, ocean, ground, etc.).
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.3_data_wrangling_spacex.ipynb.

PART 1: METHODOLOGY - EDA & DATA VISUALIZATION

- Data visualization using matplotlib and seaborn. Charts plotted were:
  - Flight number vs. Payload mass (scatter plot)
  - Flight number vs. Launch site (scatter plot)
  - Payload mass vs. Payload mass (scatter plot)
  - Success rate per orbit type (bar chart)
  - Flight number vs. orbit type (scatter plot)
  - Payload mass vs. orbit type (scatter plot)
  - Yearly trend of launch success rate (line plot)
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.5_EDA_dataviz_spacex.ipynb.

PART 1: METHODOLOGY - EDA & SQL

- Performed 10 SQL queries:
  - %sql select distinct(launch_site) from SPACEXTBL
  - %sql select * from SPACEXTBL where launch_site LIKE 'CCA%' LIMIT 5
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)’
  - %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1’
  - %sql select min(Date) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)’
  - %sql select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
  - %sql select count(*) from SPACEXTBL where Mission_Outcome like ‘Success%'
  - %sql select count(*) from SPACEXTBL where Mission_Outcome like ‘Failure%'
  - %sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
  - %sql select substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome like '%(drone ship)' and substr(Date, 0, 5) = ‘2015'
  - %sql select Landing_Outcome, count(*) as outcome_count from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by outcome_count desc
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.4_EDA_SQL_spacex.ipynb.

PART 1: METHODOLOGY - FOLIUM MAP

- Created and added the following objects to Folium map:
  - Markers for all launch site coordinates
  - Markers for all successful/failed launches for each site
  - Distances between and lines connecting a launch site and its nearest coastline position, city, railway, and highway.
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.6_folium_spacex.ipynb.

PART 1: METHODOLOGY - PLOTLY DASH

- Graphs added to the dashboard:
  - Pie chart showing the number of successful launches per site.
  - Dropdown menu to select launch sites.
  - Scatter plot of payload mass and successful launches, with a slider to choose payload mass range.
- GitHub link to the plotly dash lab: https://github.com/yutonghe96/DS_material/blob/main/proj6_capstone_dash.py.

PART 1: METHODOLOGY - PREDICTIVE MODEL

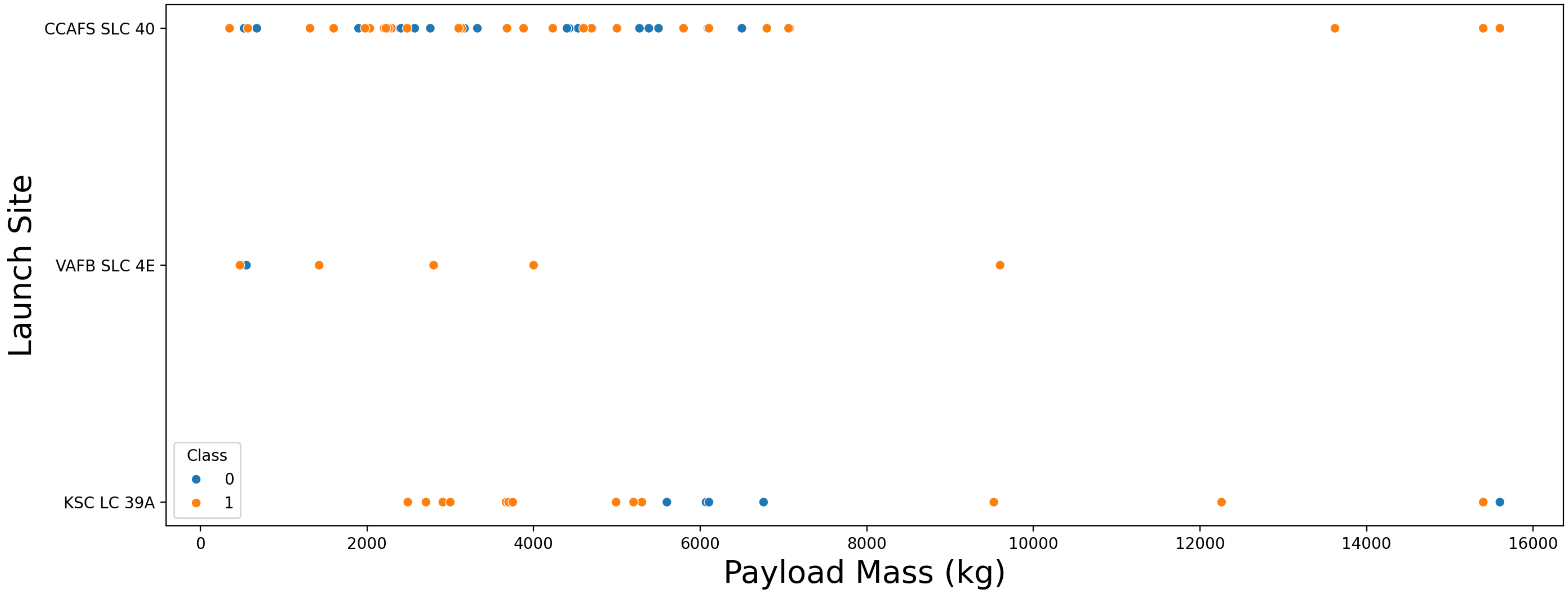
- Model development process:
  - Preparation. Transform and standardize all data.
  - Split data into training and testing subsets.
  - Built different models, i.e., logistic regression (logreg), support vector machine (svm), decision tree, k-nearest neighbor (knn).
  - Compared the accuracy of these models on the test sets and found three models (logreg, svm, knn) performed equally well, whereas decision tree performed slightly worse.
- GitHub link to the Jupyter notebook: https://github.com/yutonghe96/DS_material/blob/main/10.7_ML_prediction_spacex.ipynb.

PART 2: RESULTS — INSIGHTS FROM EDA

- Flight number vs. Launch site

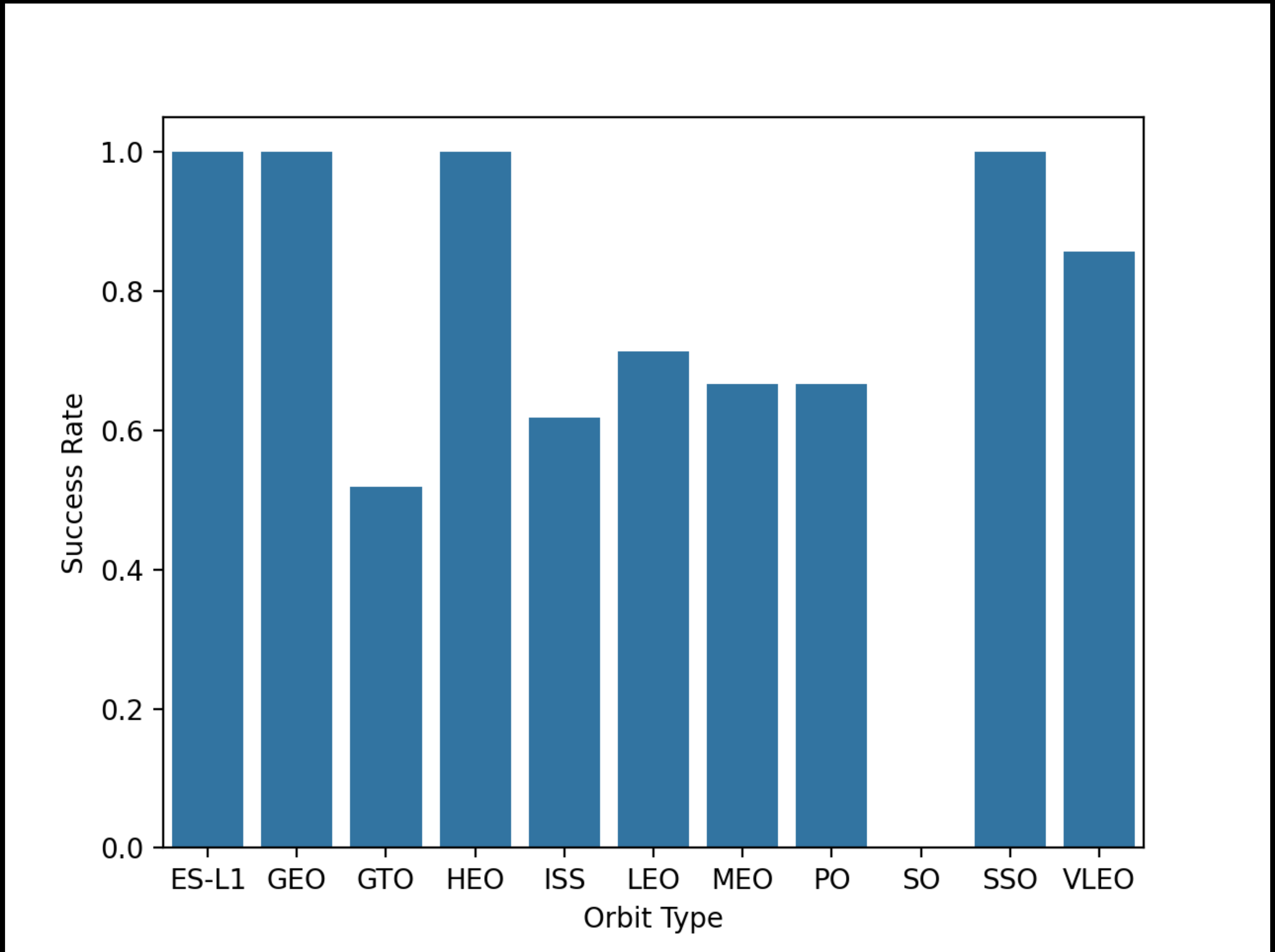
PART 2: RESULTS — INSIGHTS FROM EDA

- Payload mass vs. Launch site



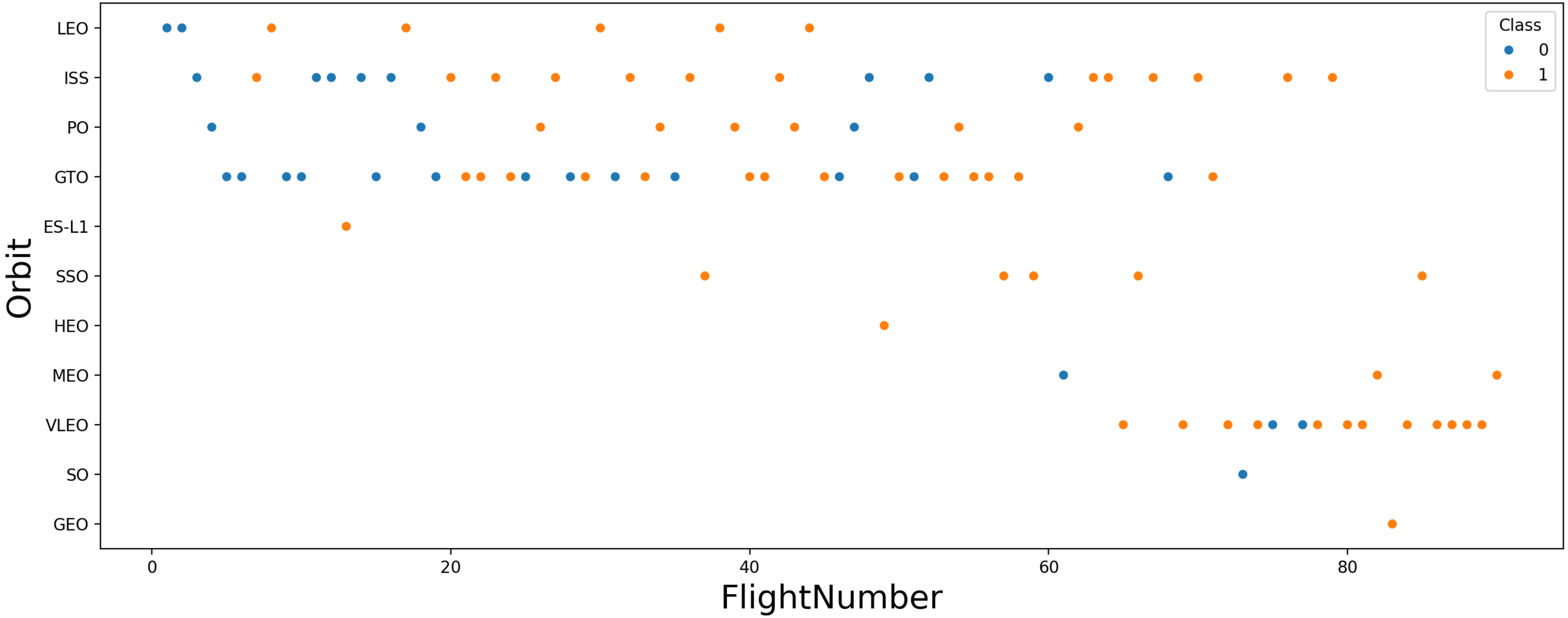
PART 2: RESULTS — INSIGHTS FROM EDA

- Success rate vs. Orbit type



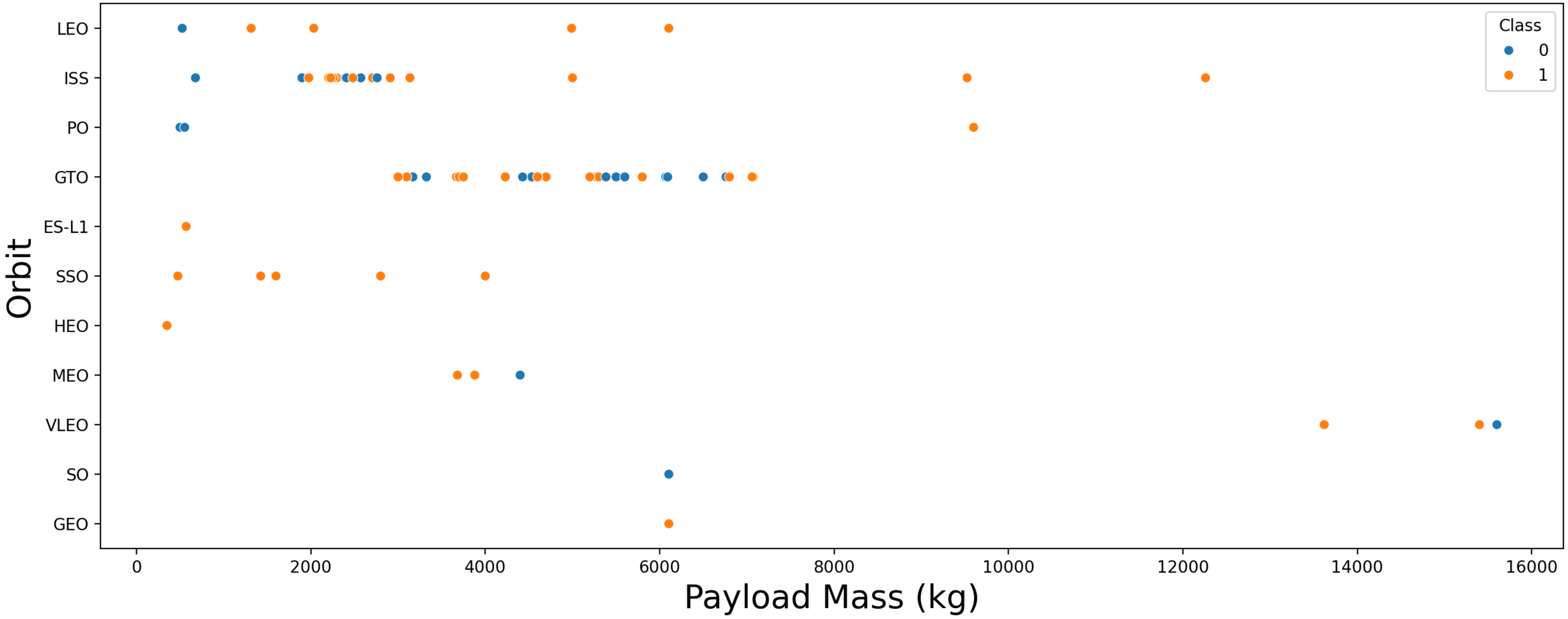
PART 2: RESULTS — INSIGHTS FROM EDA

- Flight number vs. Orbit type



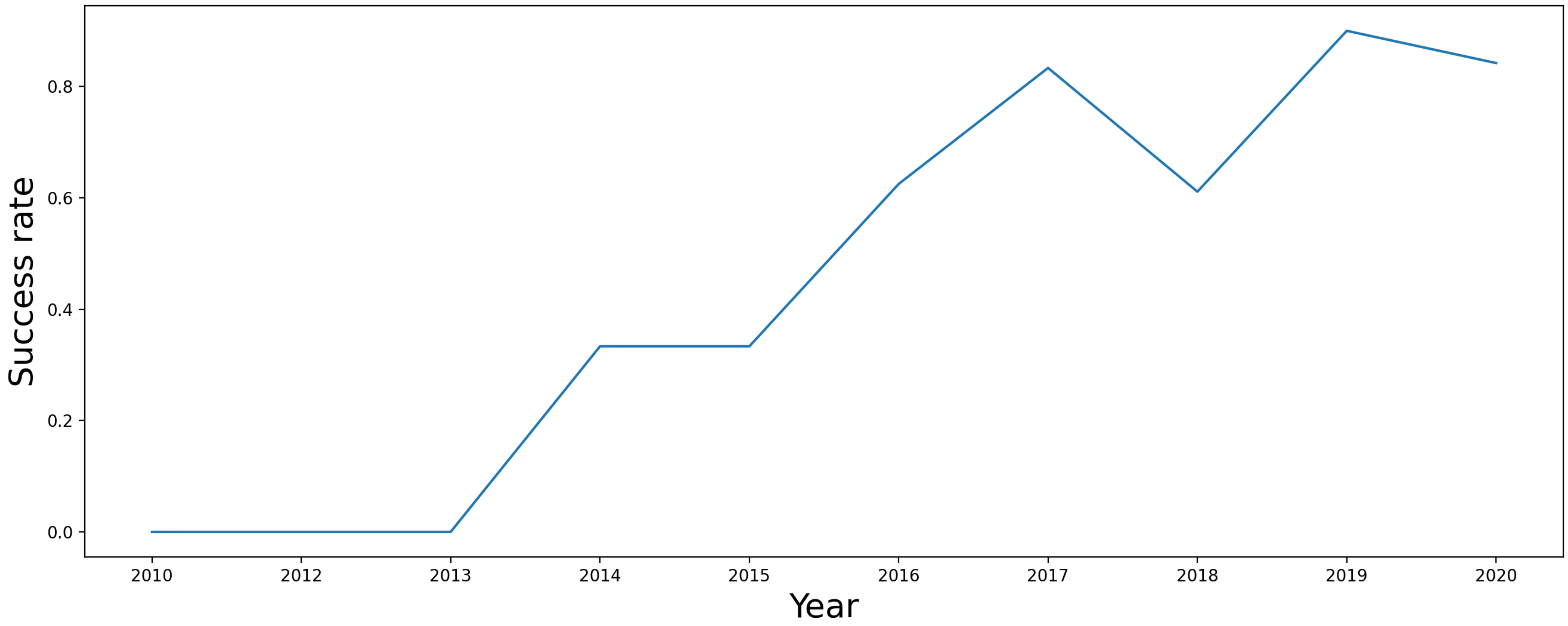
PART 2: RESULTS — INSIGHTS FROM EDA

- Payload vs. Orbit type



PART 2: RESULTS — INSIGHTS FROM EDA

- Launch success yearly trend



PART 2: RESULTS — INSIGHTS FROM EDA

- The list of 4 unique launch sites:

Task 1

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

[10]:
```
%sql select distinct(launch_site) from SPACEXTBL
```

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

[10]:

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

PART 2: RESULTS — INSIGHTS FROM EDA

- 5 launches from sites with names beginning with CCA

Task 2

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

[11]:
```
%sql select * from SPACEXTBL where launch_site LIKE 'CCA%' LIMIT 5
```

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

[11]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

PART 2: RESULTS — INSIGHTS FROM EDA

- Total payload carried by NASA boosters: 45596 kg.

Task 3

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

```
[18]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'

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

[18]:

sum(PAYLOAD_MASS__KG_)
45596

PART 2: RESULTS — INSIGHTS FROM EDA

- Average mass of payload carried by Falcon9 booster version F9 v1.1: 2928.4 kg.

Task 4

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

```
[19]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

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

[19]:

avg(PAYLOAD_MASS__KG_)
2928.4

PART 2: RESULTS — INSIGHTS FROM EDA

- The date of the first successful landing a ground pad: 2015-12-22.

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) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'

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

[21]:

min(Date)
2015-12-22

PART 2: RESULTS — INSIGHTS FROM EDA

- The list of boosters that successfully landed on a drone ship, and had payload mass over 4000 kg and under 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

```
[22]: %sql select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000

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

[22]:

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

PART 2: RESULTS — INSIGHTS FROM EDA

- Total number of successful and failed missions (NOT successful/failed landings).

Task 7 ¶

List the total number of successful and failure mission outcomes

```
[34]: %sql select count(*) from SPACEXTBL where Mission_Outcome like 'Success%'

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

[34]:

count(*)
100

```
[35]: %sql select count(*) from SPACEXTBL where Mission_Outcome like 'Failure%'

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

[35]:

count(*)
1

PART 2: RESULTS — INSIGHTS FROM EDA

- The list of boosters that carried max payload.

Task 8

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

[37]:
```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

[37]:

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

PART 2: RESULTS — INSIGHTS FROM EDA

- List of the failed landings in 2015 on a drone ship, their booster versions, and launch sites.

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.

[20]:
```
%sql select substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL \
where Landing_Outcome like '%(drone ship)' and substr(Date, 0, 5) = '2015'
```

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

[20]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

PART 2: RESULTS — INSIGHTS FROM EDA

- Categorize and rank the landing outcomes between 2010-06-04 and 2017-03-20.

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.

```
[21]: %sql select Landing_Outcome, count(*) as outcome_count from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' \
group by Landing_Outcome\
order by outcome_count desc
```

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

[21]:

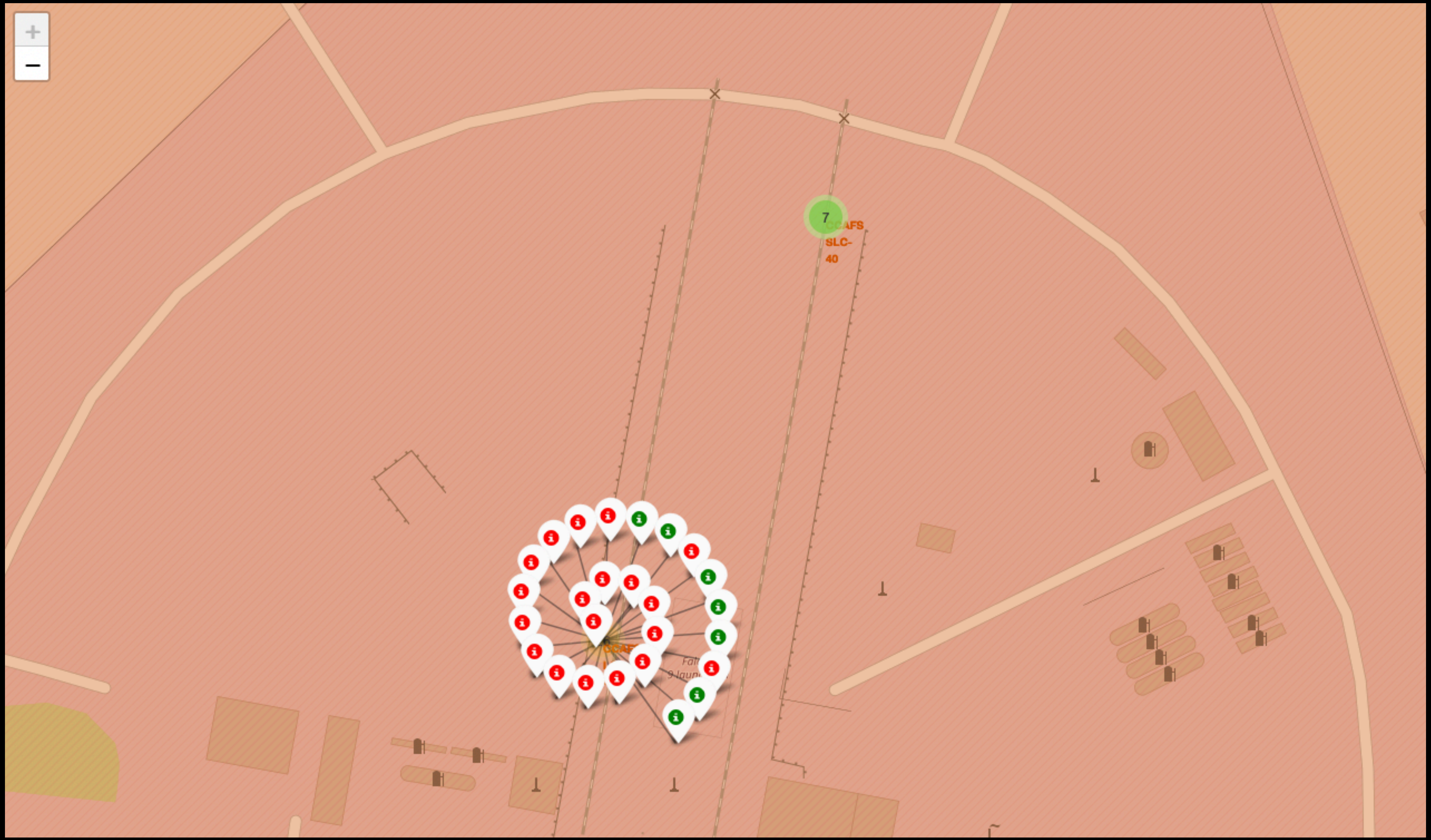
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

PART 2: RESULTS — LAUNCH SITE ANALYSIS

- A global Map showing all launch site locations:

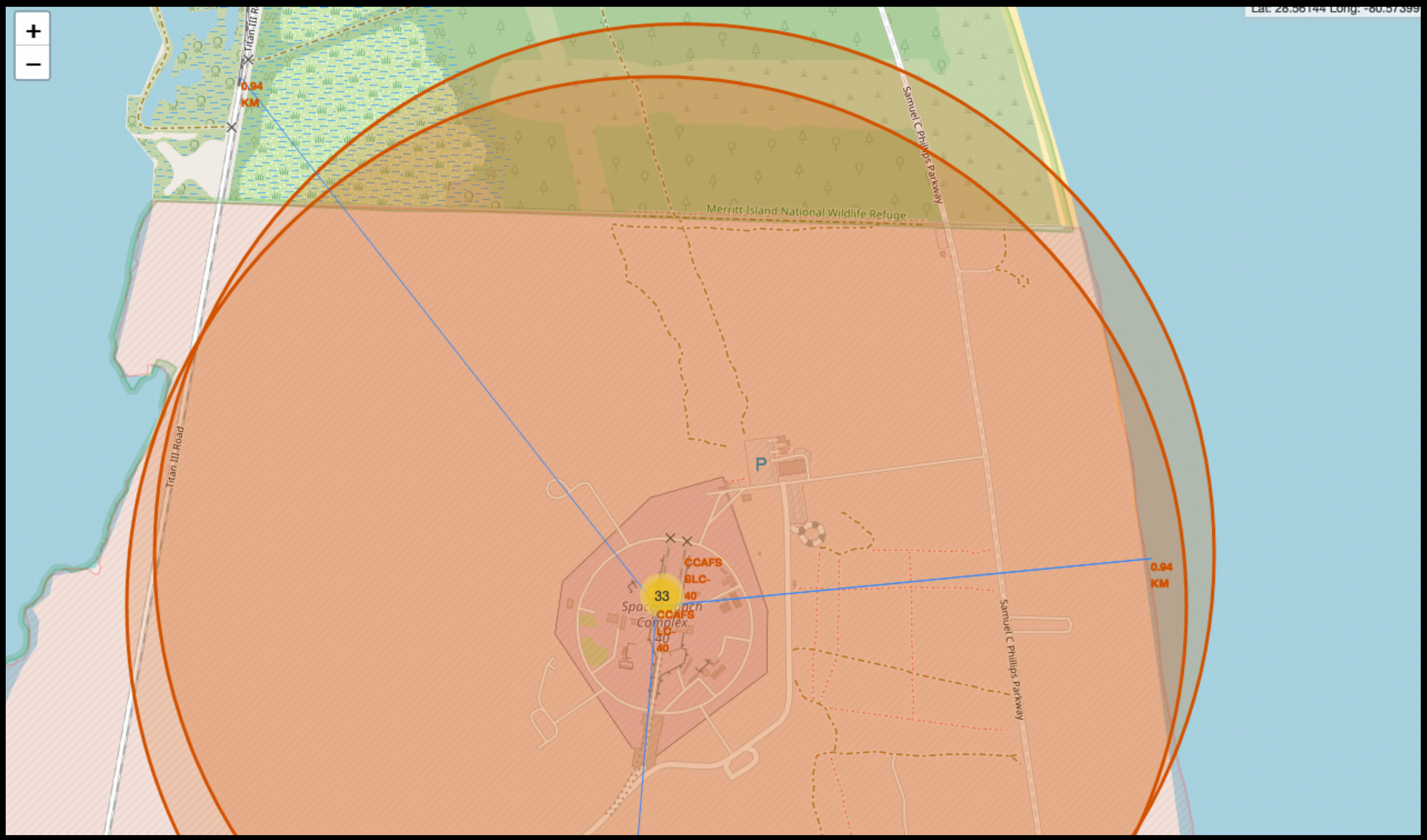
PART 2: RESULTS — LAUNCH SITE ANALYSIS

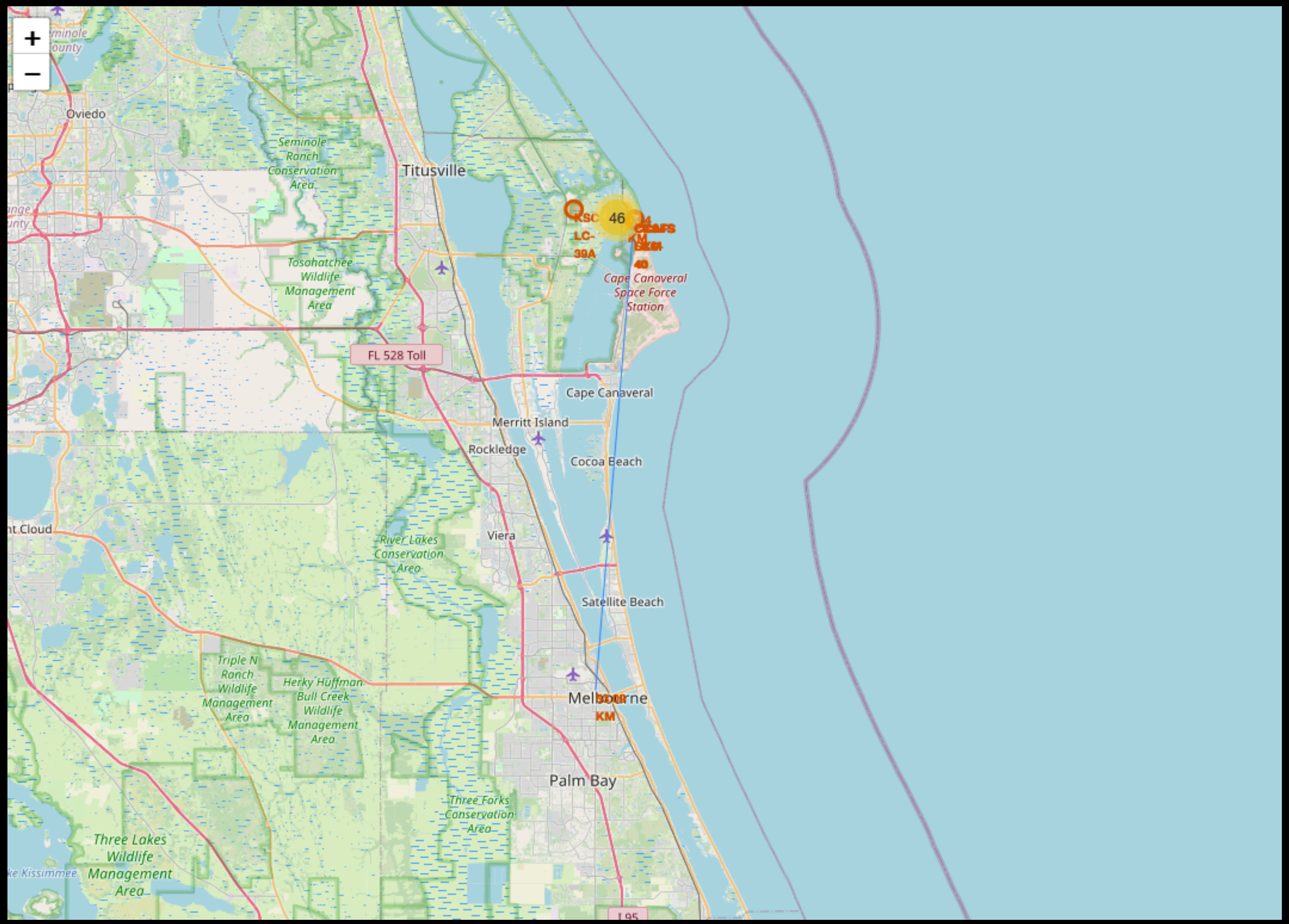
- Zoomed-in map of a launch site with outcomes labeled in green (success) and red (failed).



PART 2: RESULTS — LAUNCH SITE ANALYSIS

- Zoomed-in map showing a launch site to its proximities, i.e., railway (1.3 km), highway (0.9 km), coastline (0.9 km), and city (53 km).





PART 2: RESULTS — DASHBOARD

- Dashboard pie chart showing launch success count for all sites.

PART 2: RESULTS — DASHBOARD

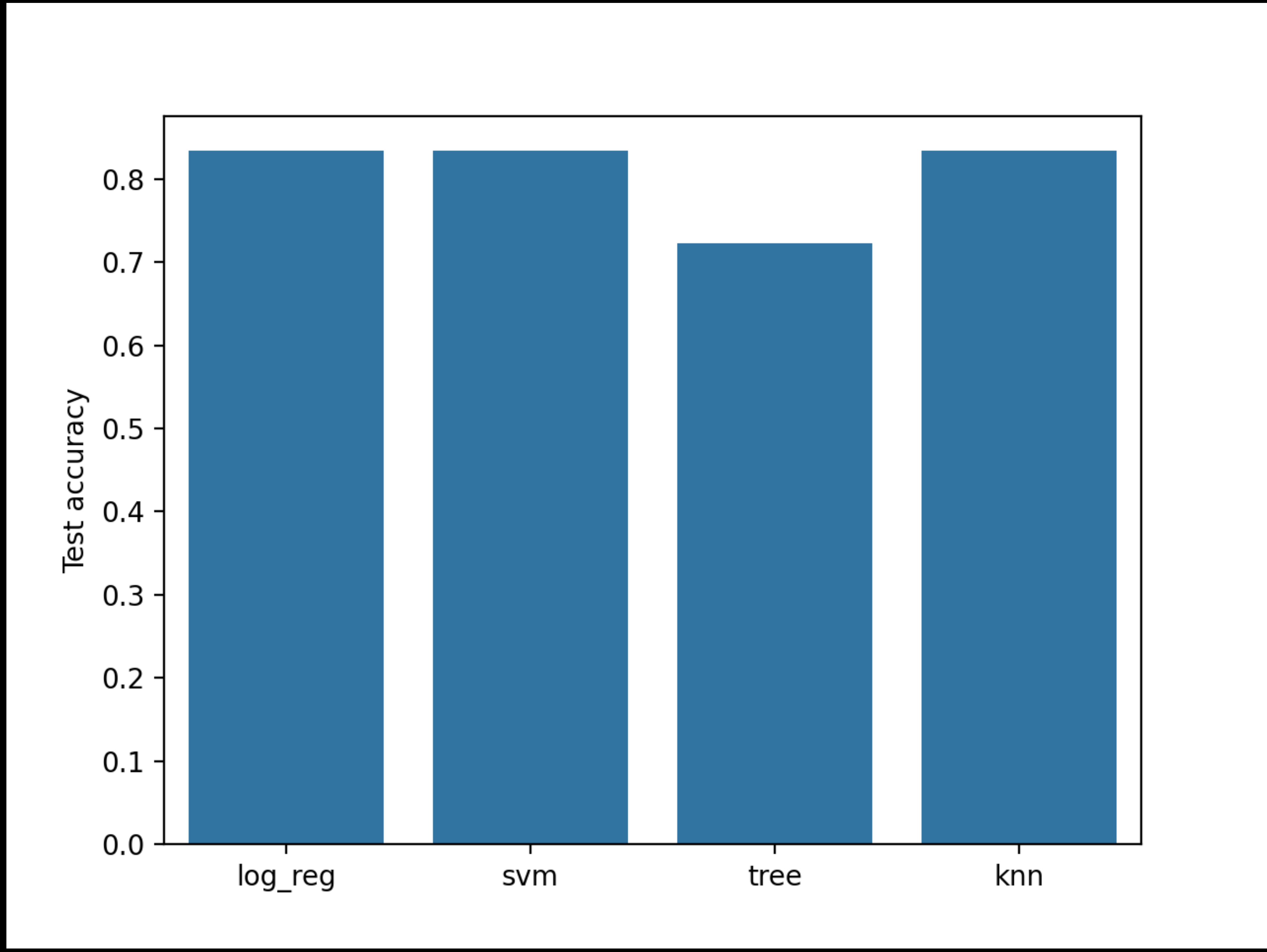
- Dashboard pie chart showing launch site with highest success rate.

PART 2: RESULTS — DASHBOARD

- Dashboard scatter chart and payload mass slider showing payload mass vs. launch outcome for all sites.

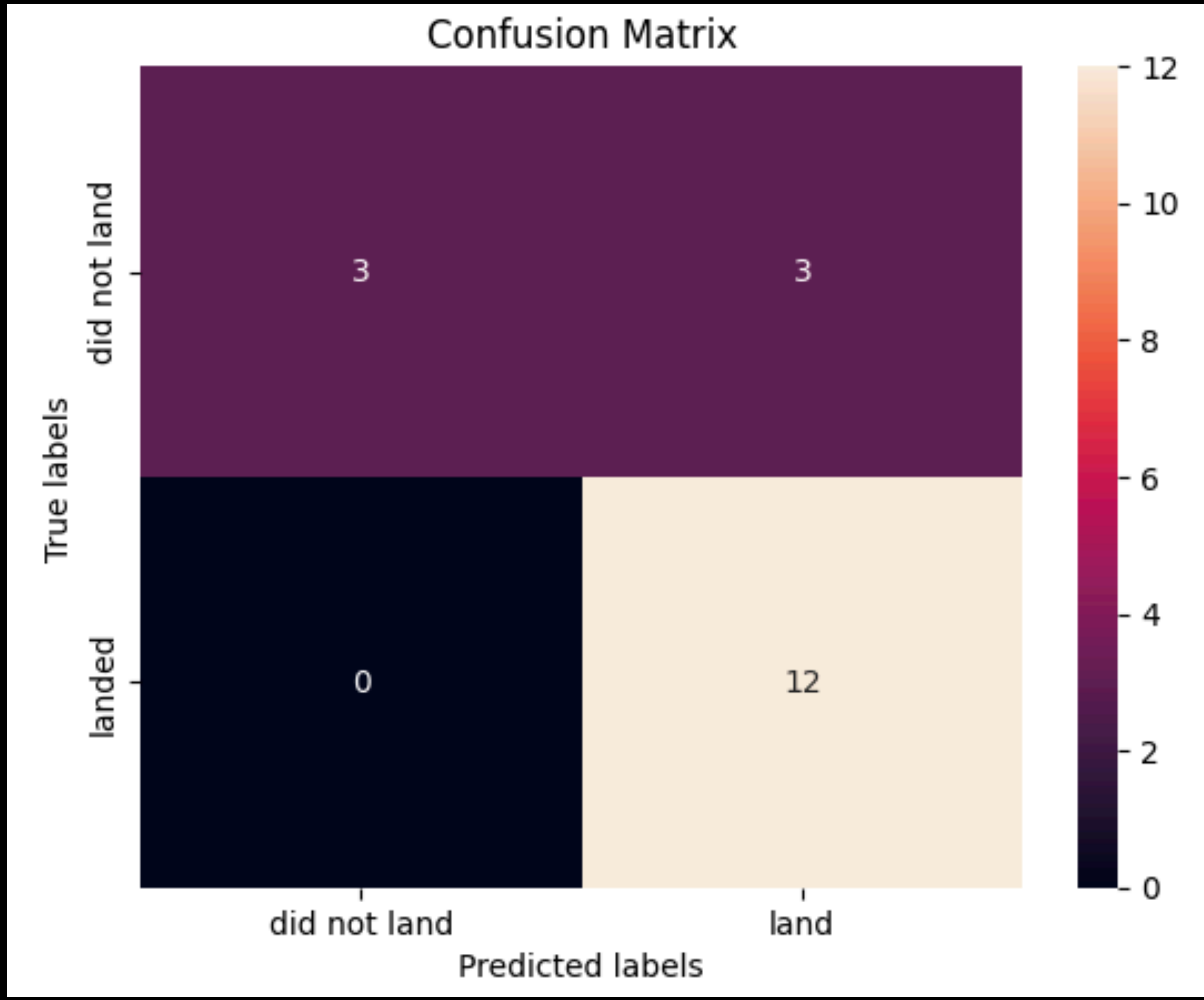
PART 2: RESULTS — PREDICTIVE ANALYSIS

- Comparing the test accuracy scores of 4 models. 3 models (log_reg, svm, knn) tie in terms of predictive performance.



PART 2: RESULTS — PREDICTIVE ANALYSIS

- Confusion matrix of best performing models (log_reg, svm, knn).
  - 12 true positives.
  - 3 true negatives.
  - 3 false positives (predicted to land, but failed to land).
  - 0 false negative (predicted to fail, but landed).

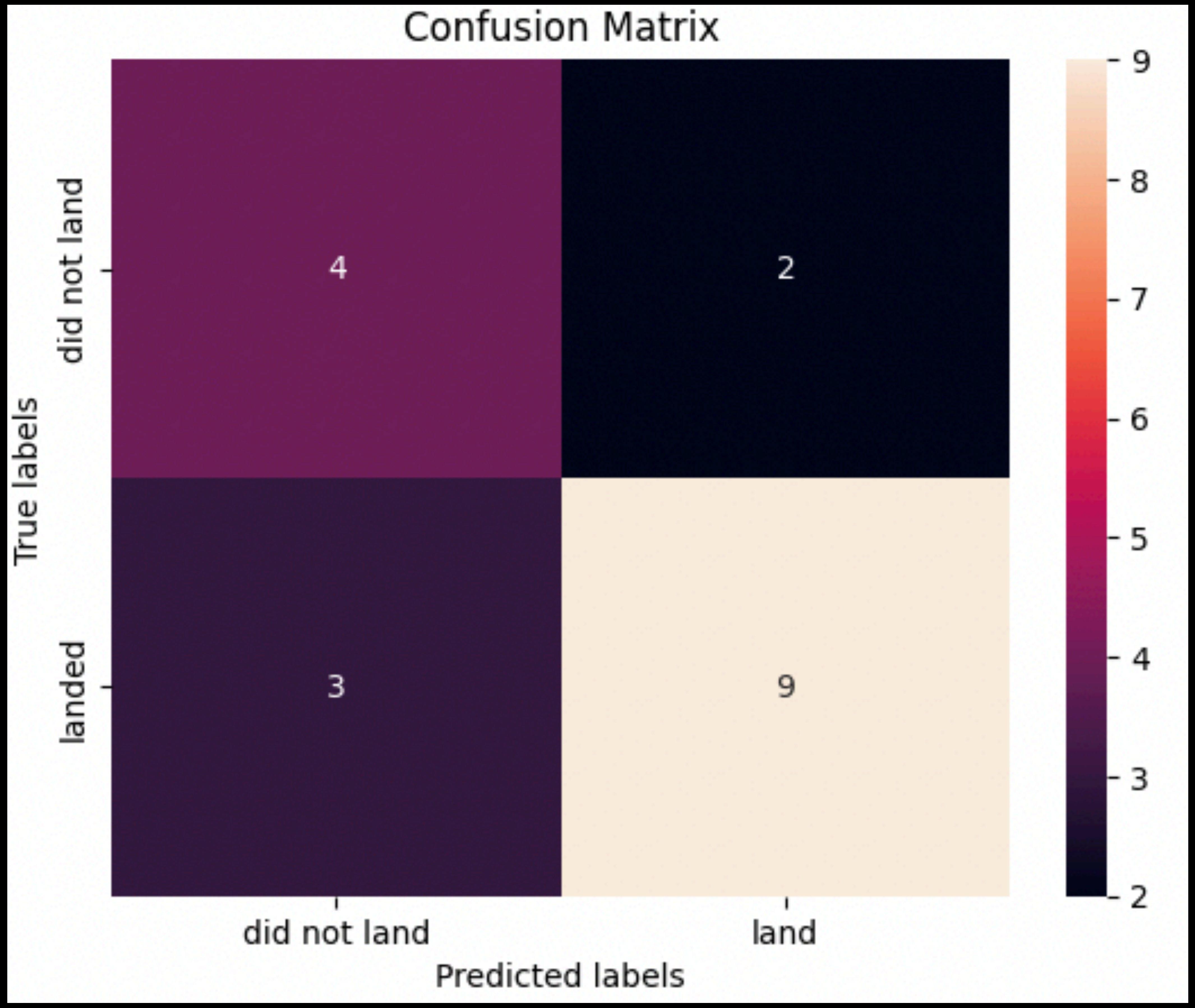


PART 3: CONCLUSIONS

- Data preprocessing is essential before analyzing and modeling it.
- It is possible to construct predictive models to classify the success/fail of future launches.

PART 4: APPENDIX

- Confusion matrix of the worse performing model (decision tree).
  - 9 true positives.
  - 4 true negatives.
  - 2 false positives (predicted to land, but failed to land).
  - 3 false negative (predicted to fail, but landed).



THANK YOU!