IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name> Sahir Dev
<Date> 11-05-2025

OUTLINE

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

EXECUTIVE SUMMARY

- **Summary of methodologies:**
- 1. Data collection through api.
- 2. Data collection with web scrapping.
- 3. Data wrangling.
- 4. Exploratory Data Analysis with SQl.
- 5. Exploratory Data Analysis with Data Visualization.
- 6. Interactive Visual Analytics with Folium.
- 7. Interactive Dashboard with Plotly Dash.
- **Summary of all results:**

  1.Exploratory Data Analysis result

  2.Interactive analytics in screenshots

  3.Predictive Analytics result

INTRODUCTION

- **Project background and context:**

- Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- **Problems you want to find answers**
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

METHODOLOGY

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

DATA COLLECTION – SPACEX API

Get Request for Rocket Launch Data using API

```
[3]: spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

Now we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

```
[6]: response_norm= response.json()
#response_norm= response_norm.json_normalize()
data= pd.json_normalize(response_norm)
```

Filter the data dataframe using the BoosterVersion column to only keep the Falcon 9 launches)

```
data_falcon9= data_falcon9[data_falcon9['BoosterVersion']!='Falcon 1']
```

Data Cleaning and fill in missing values

```
[ ]: mean_pl=data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].replace(np.nan,mean_pl, inplace=True)
```

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- The link to the notebook is
- https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/Data%20Collection%20API.ipynb

DATA COLLECTION - SCRAPING

HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[22]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon
      r= requests.get(static_url)
```

Create a BeautifulSoup object from the HTML response

```
[ ]: soop= BeautifulSoup(r.text,'html.parser')
     soop.title
```

apply the provided extract_column_from_header() to extract column name one by one

```
[ ]: column_names = []
     for cname in first_launch_table.find_all('th'):
         if (cname is not None) & (len(cname)>0):
             column_names.append(extract_column_from_header(cname))
```

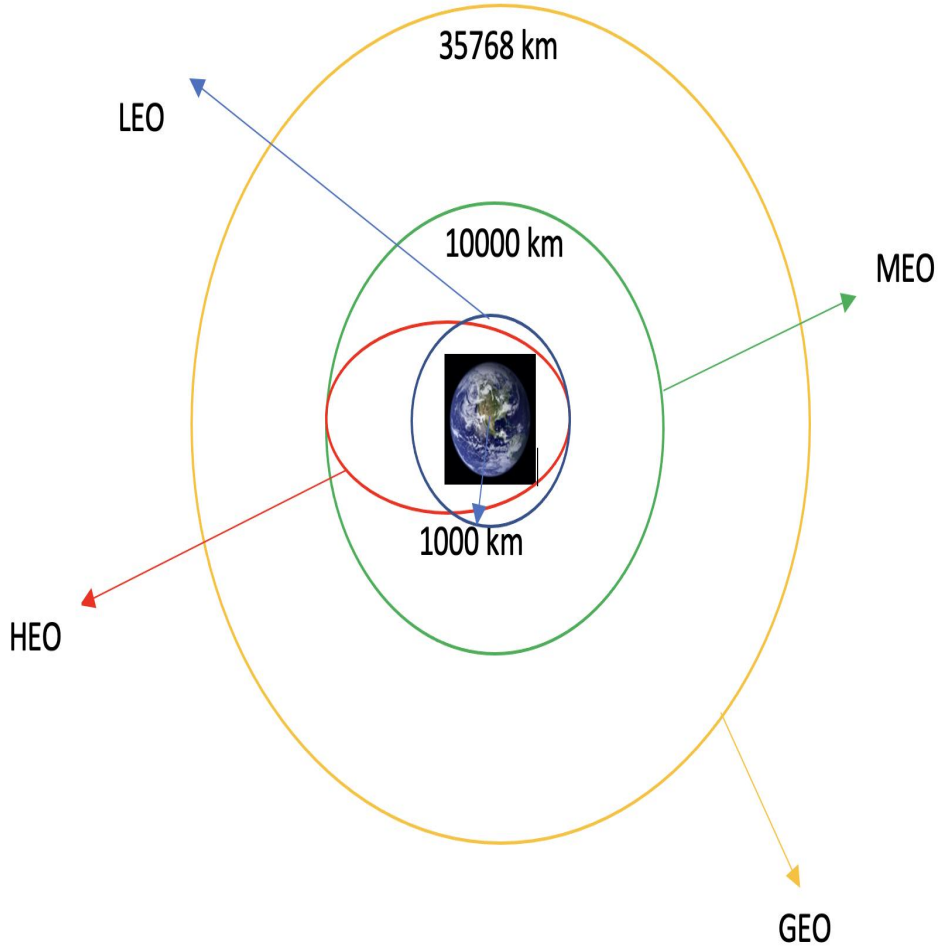
Create a data frame by parsing the launch HTML tables

Export Data to csv

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/Data%20Collection%20Web%20Scrapping.ipynb

DATA WRANGLING

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is
- https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/Data%20Wrangling.ipynb

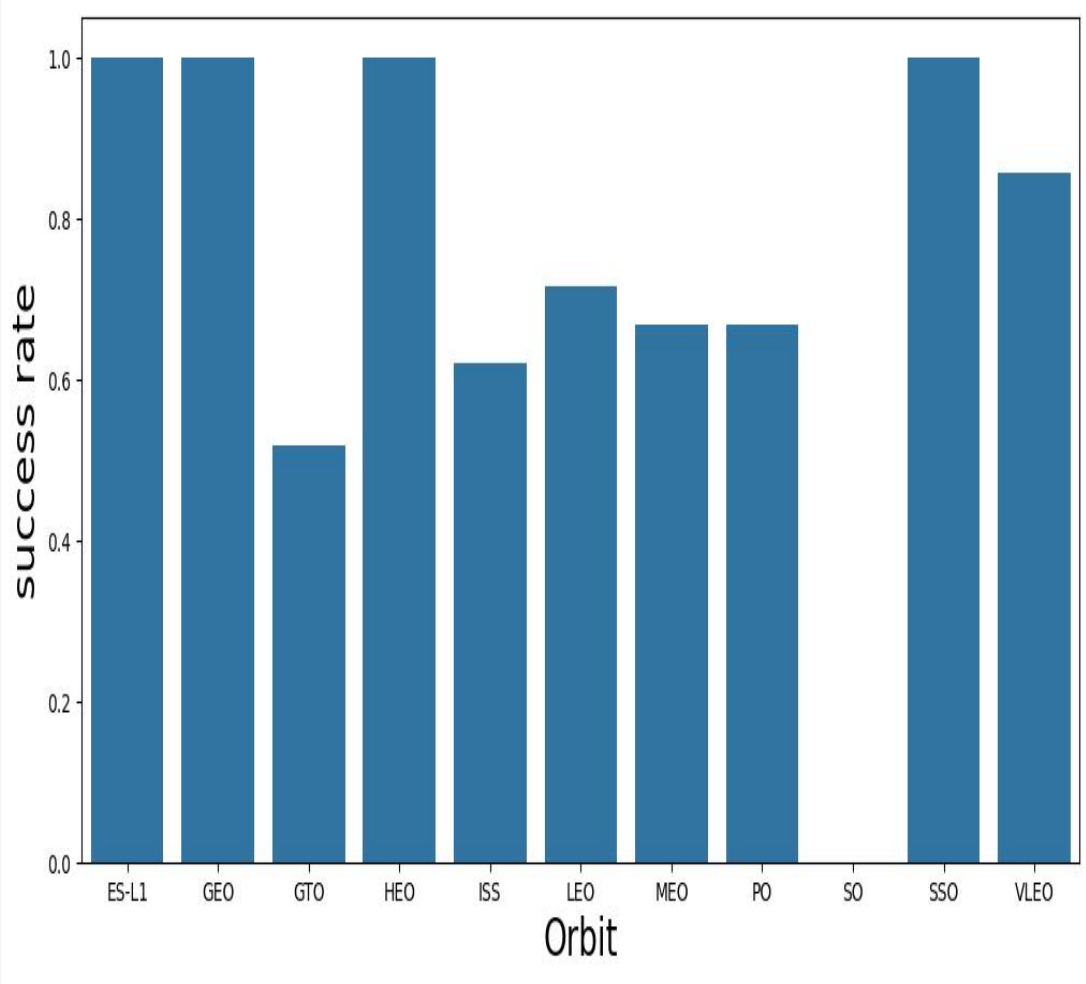


EDA with Data Visualization

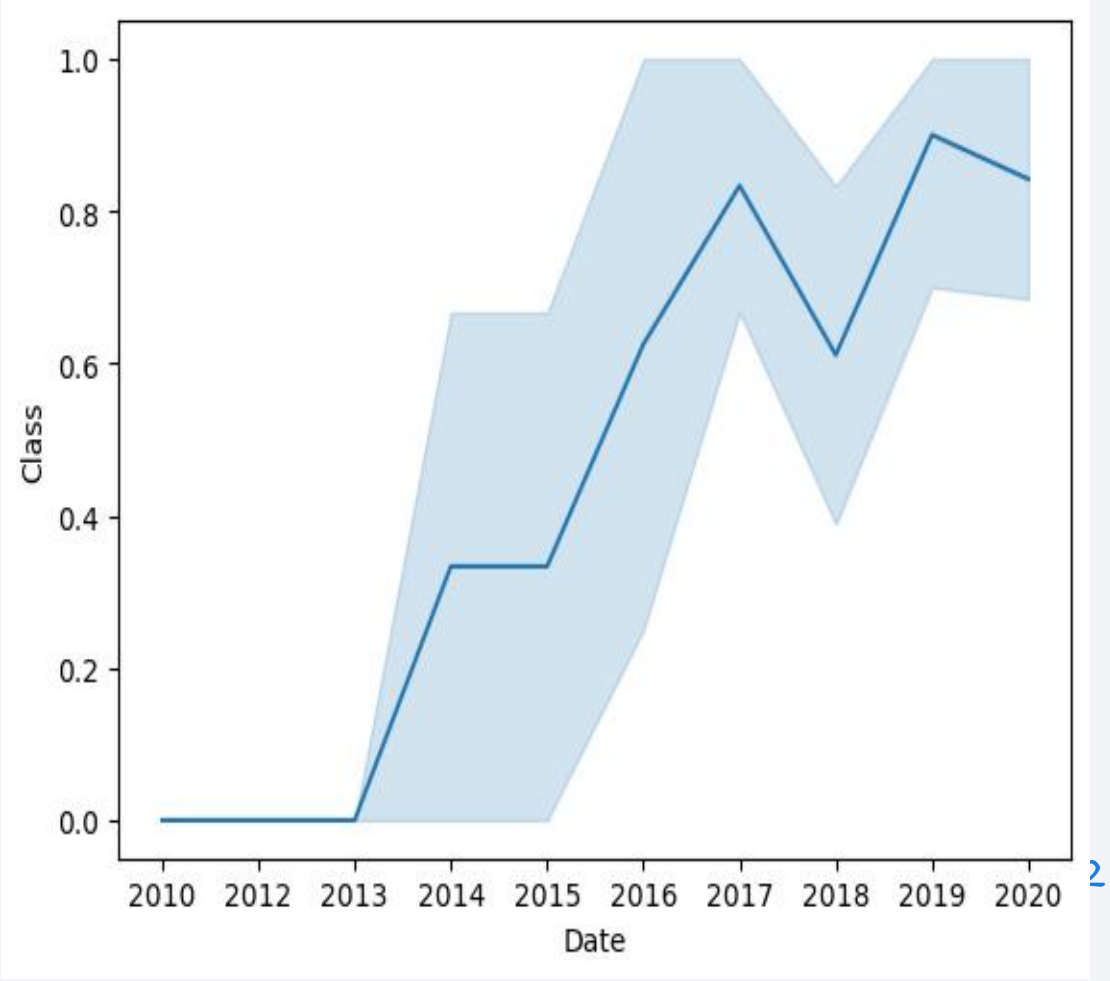
- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit. A bar diagram makes it easy to compare sets of data between different groups at a glance.
- The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.
- Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.
- LINK: https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/EDA%20with%20Data%20Visualization.ipynb

EDA WITH DATA VISUALIZATION

Visualize the relationship between success rate of each orbit type



Visualize the launch success yearly trend



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 succesful landing outcome in ground pad was acheive.
- 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 all the booster_versions that have carried the maximum payload mass. Use a subquery.
- 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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- **LINK**: https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/EDA%20With%20SQL.ipynb

BUILD AN INTERACTIVE MAP WITH FOLIUM

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- **LINK**: https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/Interactive_Visual_Analytics_Folium.ipynb
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

BUILD A DASHBOARD WITH PLOTLY DASH

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- **link:** https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/spacex-dash-app.py

PREDICTIVE ANALYSIS (CLASSIFICATION)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models using different methods and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- LINK: https://github.com/sahir-dev/IBM-DSC-SpaceX/blob/main/Machine%20Learning%20Prediction.ipynb

RESULTS

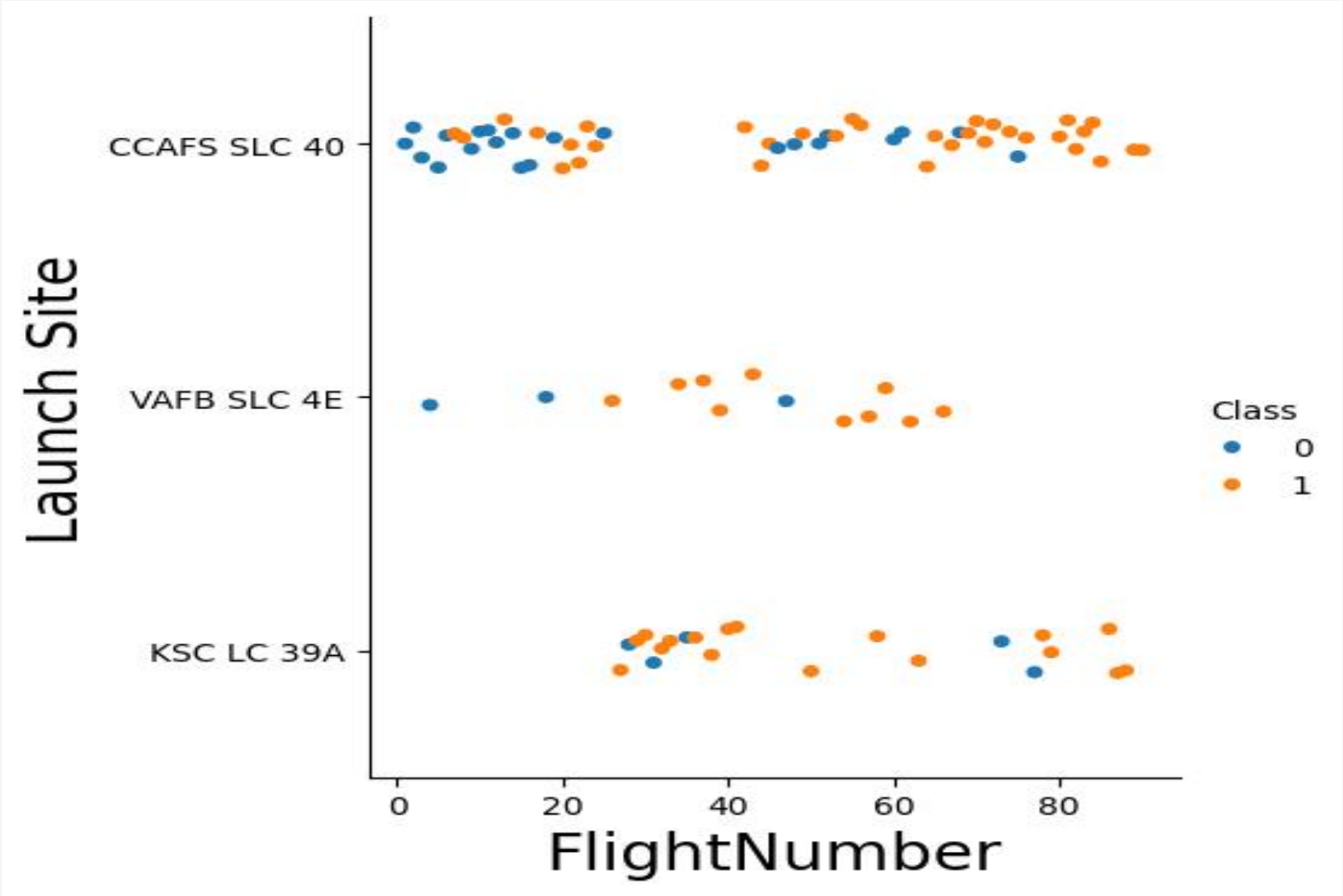
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

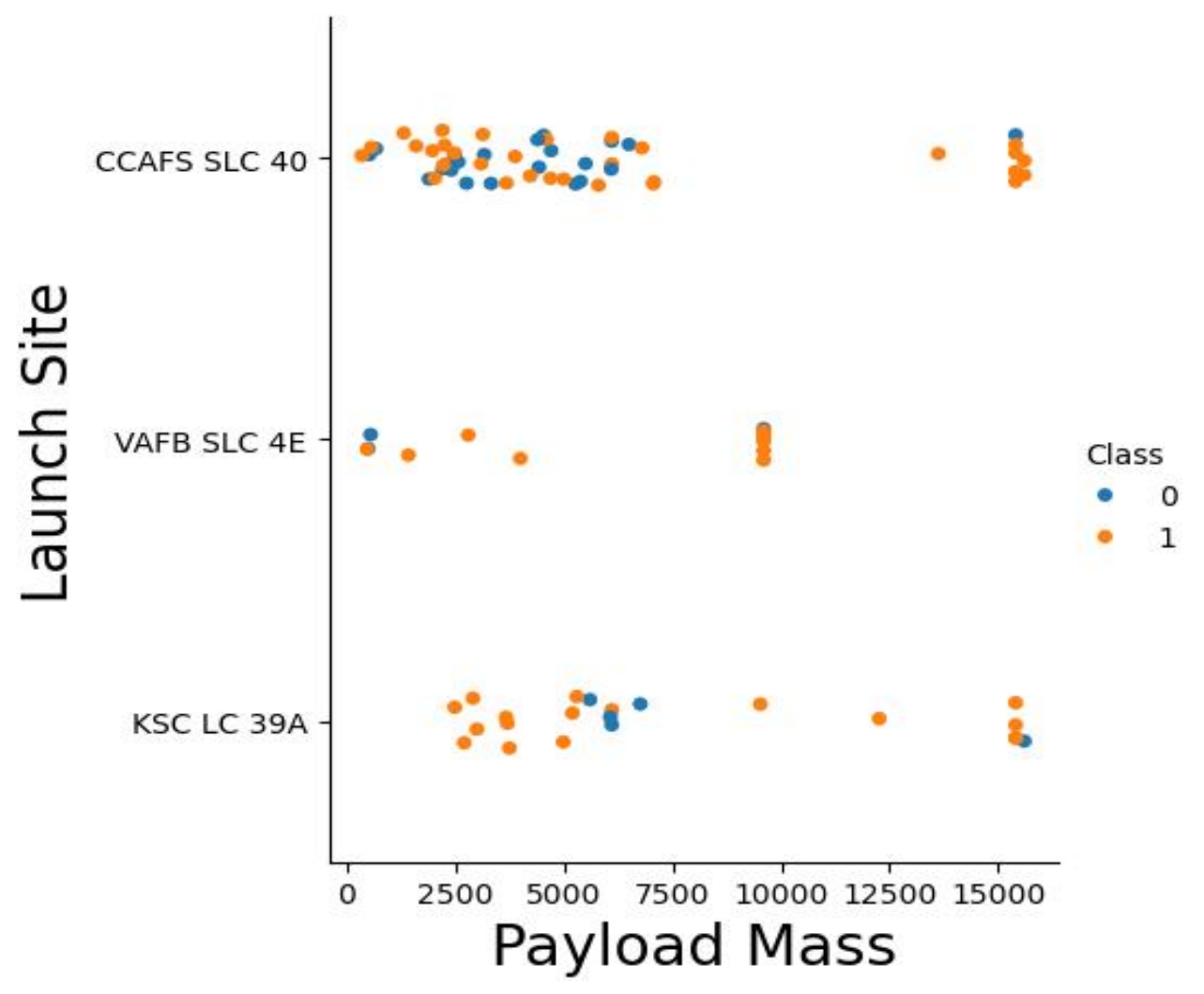
FLIGHT NUMBER VS. LAUNCH SITE

- CCAFS SLC-40 (top row) and KSC LC-39A (bottom row) have many high- flight-number missions, and most of those later flights are successful (more orange dots as x increases)
- VAFB SLC-4E (middle row) has fewer total missions, and successes (orange) start appearing at mid-range flight numbers.
- **Overall, success rate improves with higher flight numbers (more orange than blue on the right side), reflecting growing operational maturity.**



PAYLOAD VS. LAUNCH SITE

- CCAFS SLC-40 (top row) launches payloads between ~2 000 kg and ~8 000 kg, with a mix early on but nearly all successes at higher masses.
- VAFB SLC-4E (middle) has fewer launches, clustered under ~3 000 kg and around ~10 000 kg, with success improving over time.
- KSC LC-39A (bottom) handles the widest range (up to ~15 000 kg), and almost every high-mass mission from this site succeeded.
- **Overall, heavier payloads and later flights show higher success rates across all sites.**



SUCCESS RATE VS. ORBIT TYPE

100% success for deep-space and high-inclination orbits:

ES-L1, GEO, HEO, and SSO missions never failed in your dataset.

Very high success for very low Earth orbit (VLEO).

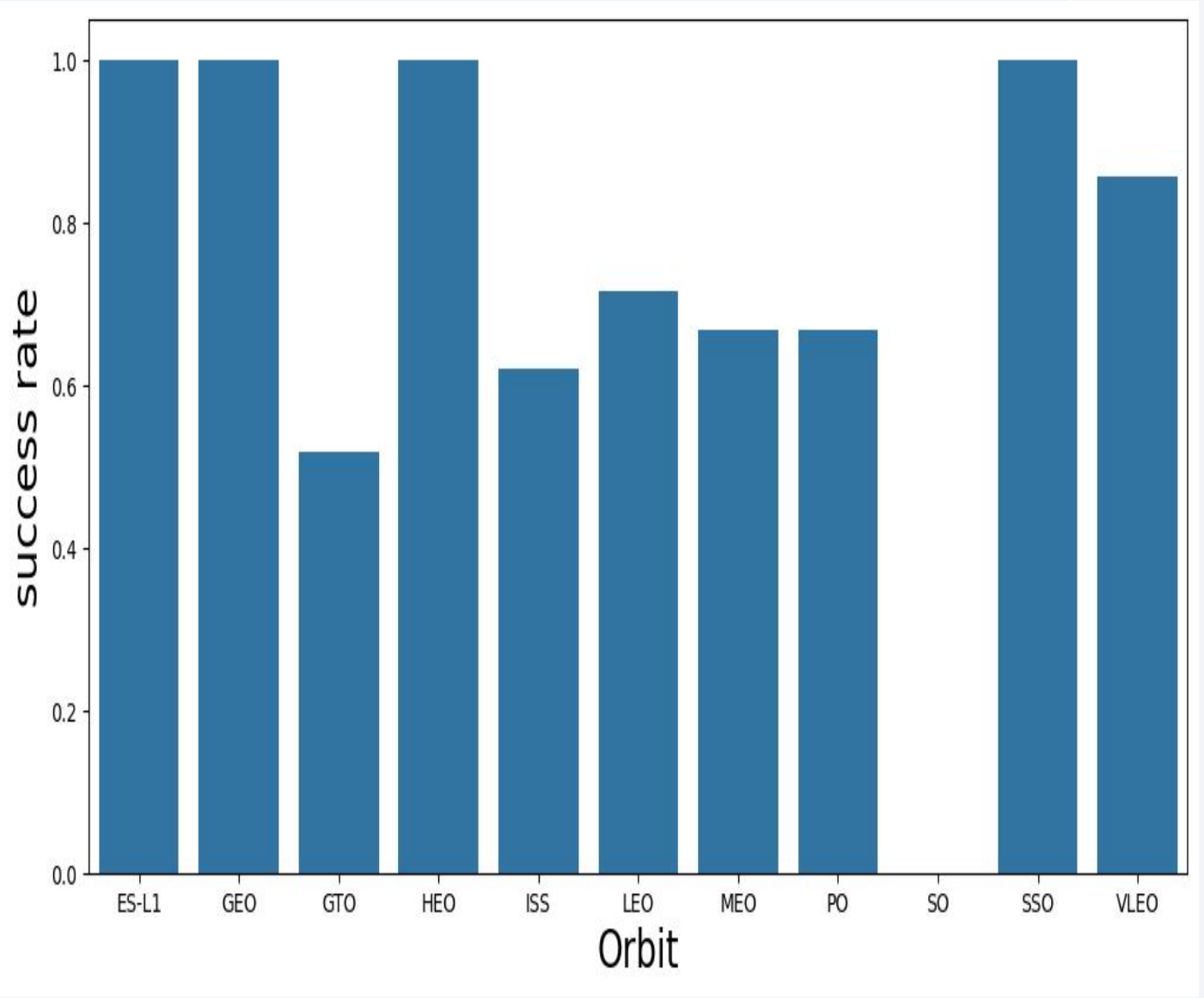
Moderate success for LEO , MEO , and PO.

Lower success for ISS-resupply.

Lowest success for GTO missions.

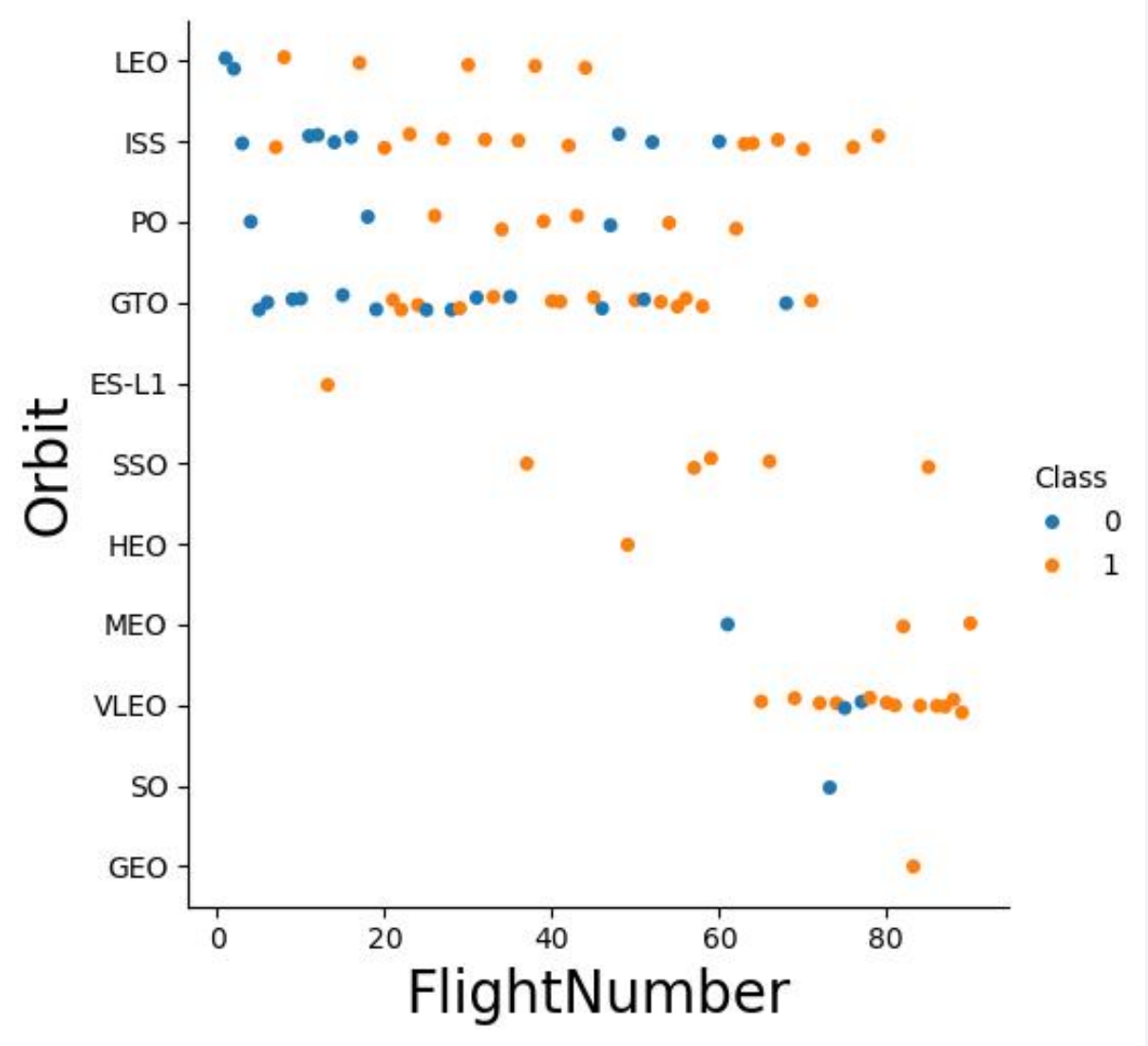
SO shows 0% because there were no successful missions (likely only failures or no data).

In short, the hardest orbits (GTO, ISS) see the most failures, while unique deep-space (ES-L1, GEO, HEO) and Sun-synchronous (SSO) missions have perfect records.



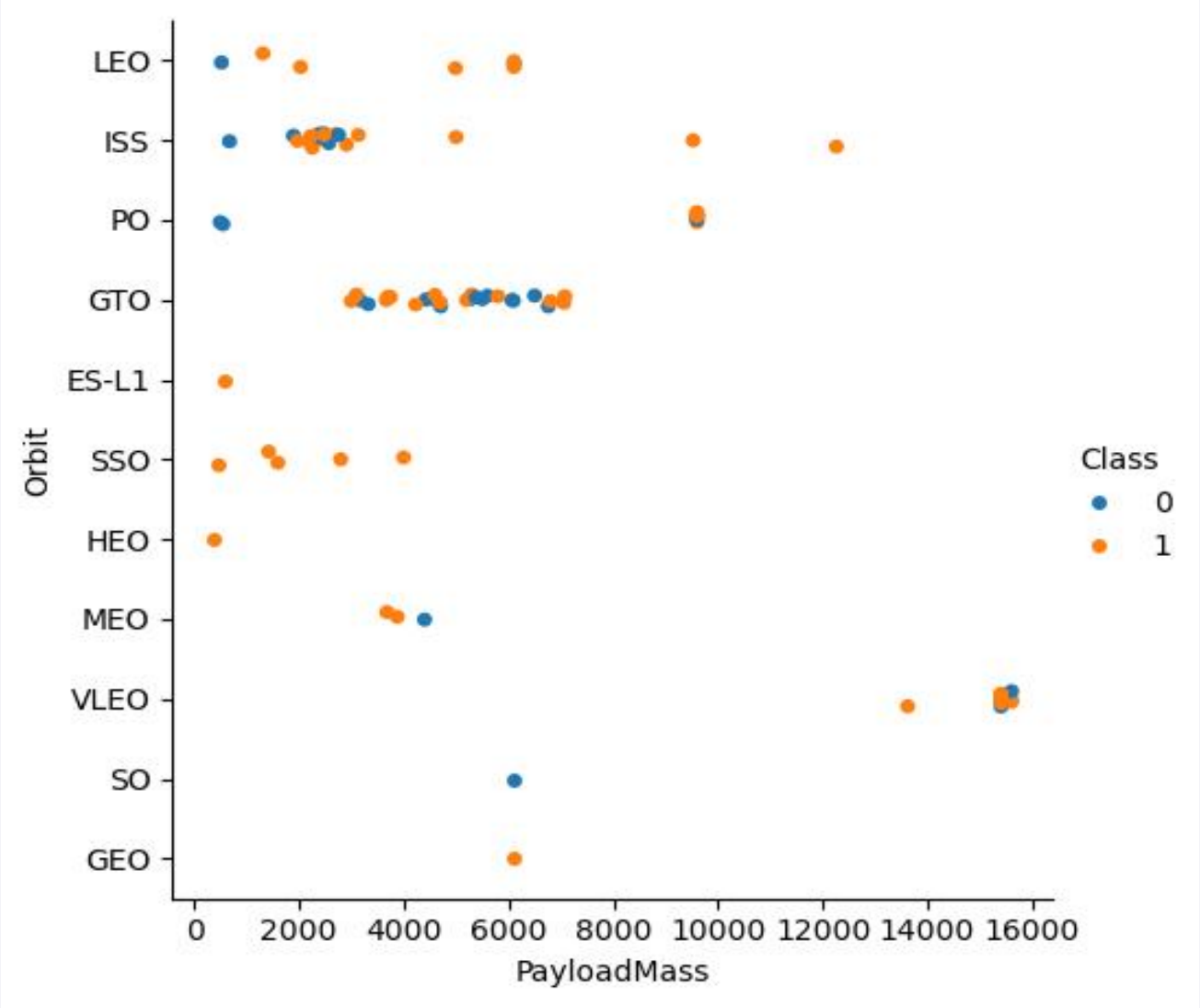
FLIGHT NUMBER VS. ORBIT TYPE

- Early Flights across all orbits have more failures (blue), but success rates climb as flight number increases.
- LEO and ISS missions (top rows) see a transition from mixed outcomes at low flight numbers to mostly successes beyond ~30.
- GTO (mid row) also improves—early GTO flights had failures, but later flights become reliably successful.
- High-mass orbits like VLEO and MEO (lower rows) start later (higher flight numbers) and are predominantly successful.
- Rare orbits (SO, GEO, ES-L1) cluster at specific flight numbers with a few failures early on, then full success.
- In short, operational maturity shows up clearly: as SpaceX gains experience (higher flight numbers), success becomes the norm across every orbit.



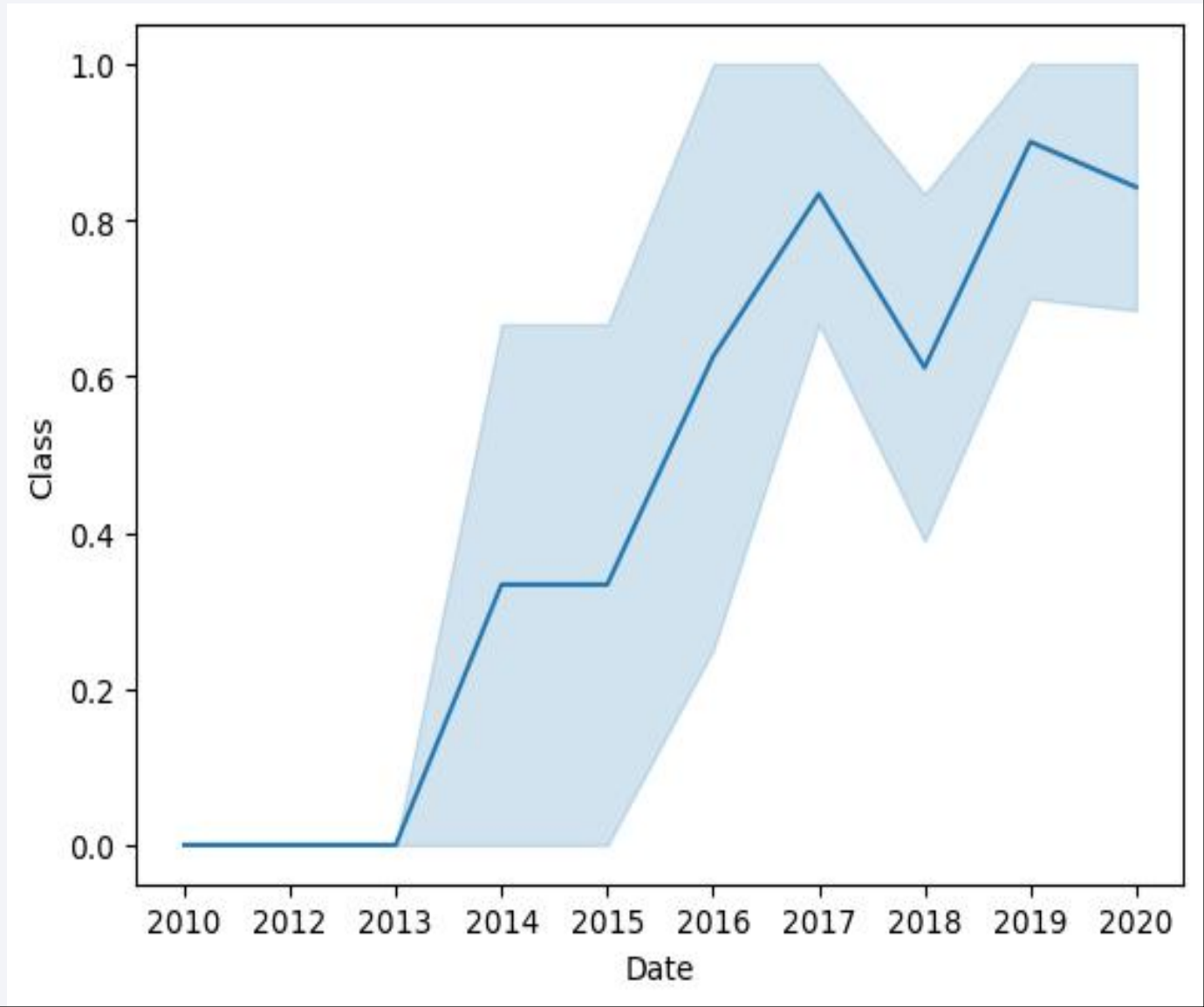
Payload vs. Orbit Type

- LEO/ISS (top rows) cluster at lower masses (<4 000 kg). Early missions show mixed outcomes, but beyond ~2 000 kg almost all are successful.
- GTO missions (mid-row) dominate the 3 000–8 000 kg range. Failures occur early in that band, then orange successes take over.
- High-mass orbits like VLEO (lower row) appear at >12 000 kg and are almost entirely successful.
- Specialty orbits (ES-L1, SSO, HEO, GEO, SO) have only a handful of points—mostly orange—indicating strong success rates once those services ramped up.
- **Bottom line: as payload mass and operational experience grow, SpaceX's success rate climbs across every orbit type.**



LAUNCH SUCCESS YEARLY TREND

- 2010–2013: 0% success (no wins until 2013).
- 2014–2015: Climb into the 30–35% range.
- 2016: Jumps above 60%.
- 2017: Peaks near 85%.
- 2018: Dips to ~60%.
- 2019–2020: Rises again to around 90% in 2019 and stays in the mid-80% range by 2020.
- **Bottom line: SpaceX's year-on-year launch success rate has steadily improved, stabilizing above 80% in recent years.**



ALL LAUNCH SITE NAMES

We used distinct Keyword so that we can only show unique launch sites. %sql is a line magic so that we execute that line as a sql.

```
[21]: %sql select distinct launch_site from spacextbl
 * sqlite:///my_data1.db
Done.
```

[21]:

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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

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

[13]:
```
%sql select * from spacextbl where launch_site like "CCA%" limit 5
```

[13]:

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

We used the query above to display 5 records where launch sites begin with `CCA

TOTAL PAYLOAD MASS

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

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_mass from spacextbl where customer = 'NASA (CRS)'
```

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

Total_Payload_mass
45596

We calculated the total payload carried by boosters from NASA as 45596 using the query below

AVERAGE PAYLOAD MASS BY F9 V1.1

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

```
25]: %sql select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version ='F9 v1.1'
```

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

25]:

avg(PAYLOAD_MASS__KG_)
2928.4

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

FIRST SUCCESSFUL GROUND LANDING DATE

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

Hint:Use min function

```
[16]: %sql select min(Date) from spacextbl where landing_outcome='Success (ground pad)'
```

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

[16]:

min(Date)
2015-12-22

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

[27]:
```
%%sql
select booster_version from spacextbl where (landing_outcome='Success (drone ship)') & ((PAYLOAD_MASS__KG_> 4000) & (PAYLOAD_MASS__KG_<6000))
```

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

[27]:

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

We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

List the total number of successful and failure mission outcomes

```
[35]: %sql select mission_outcome,count(*) from spacextbl group by mission_outcome
```

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

[35]:

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

Total number of successful mission outcome = 100
Total number of failed mission outcome = 1

BOOSTERS CARRIED MAXIMUM PAYLOAD

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
[36]: %sql select booster_version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
 * sqlite:///my_data1.db
Done.
```

[36]:

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

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 LAUNCH RECORDS

[39]:
```
%%sql select substr(Date,6,2) as month, landing_outcome, booster_version, launch_site
from spacextbl where (landing_outcome='Failure (drone ship)') & (substr(Date,0,5)='2015')
```

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

[39]:

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

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. so we used substr.
%%sql is a cell magic which treat the entire cell as a sql.

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

40]:
```
%%sql select landing_outcome,count(landing_outcome) from spacextbl
where Date between '2010-06-04' and '2017-03-20'
group by landing_outcome
order by count(landing_outcome) desc;
```

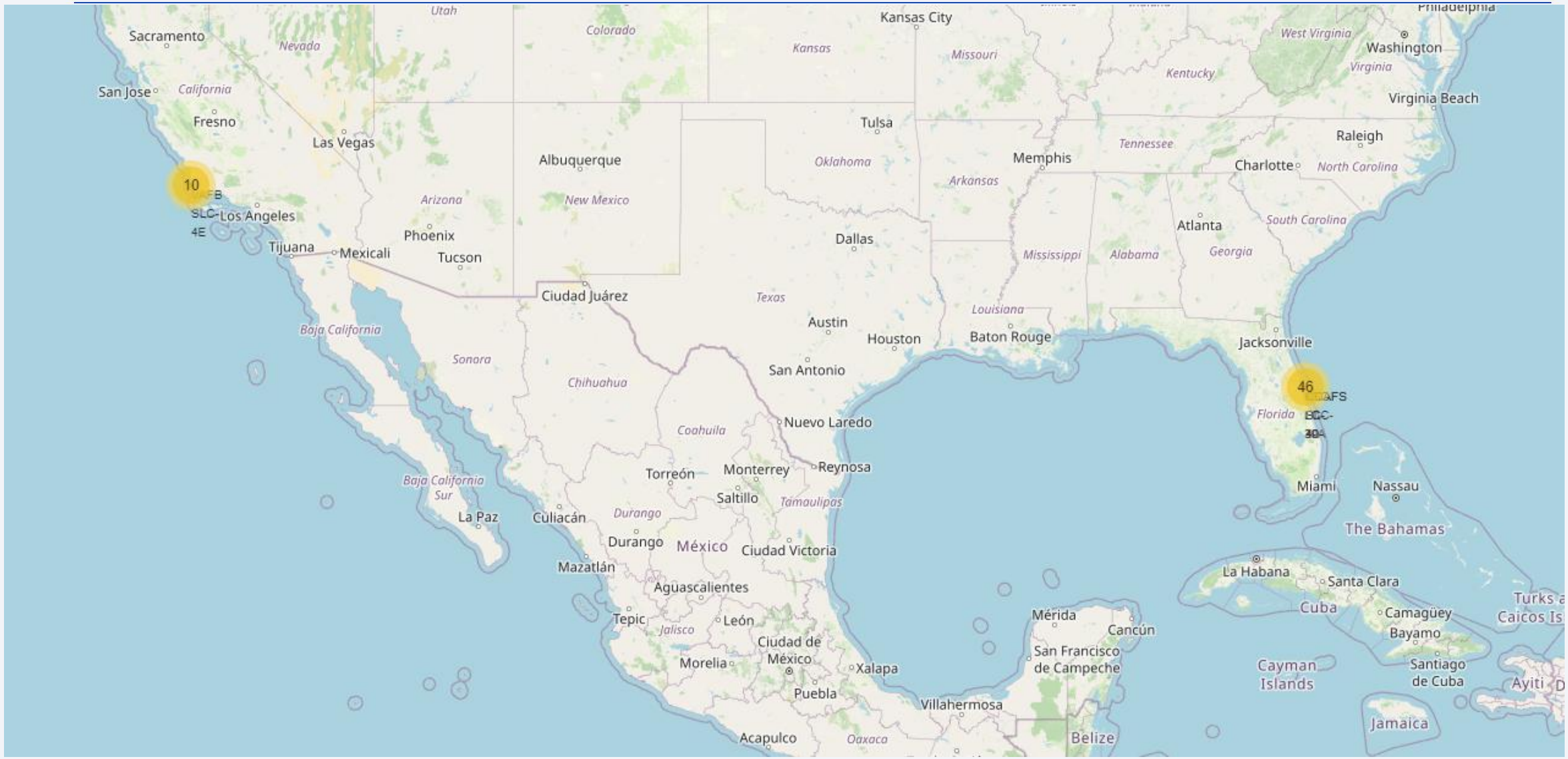
```
 * sqlite:///my_data1.db
Done.
```

40]:

Landing_Outcome	count(landing_outcome)
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



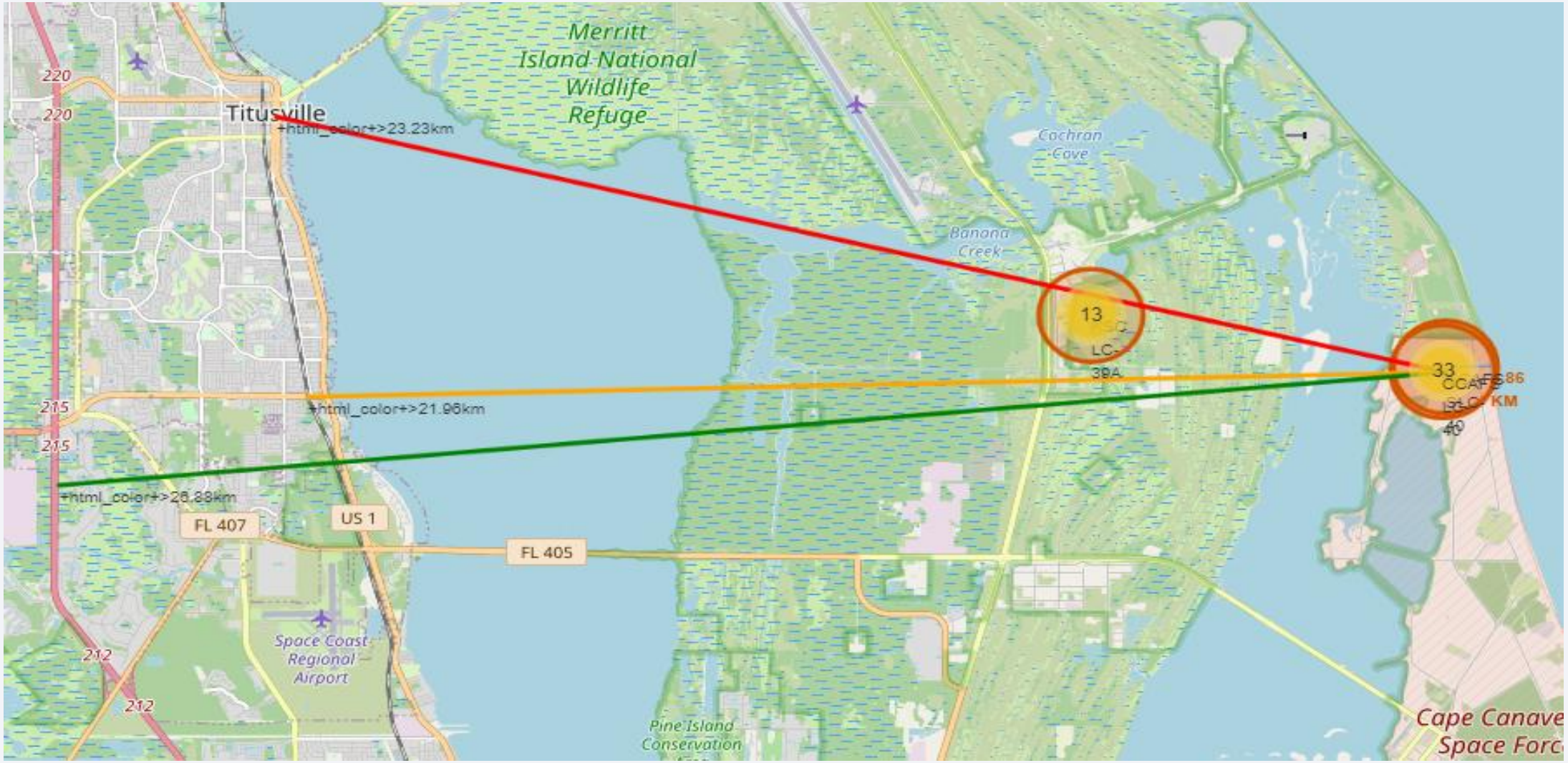
LAUNCH OUTCOMES

CONTINUED


Saturn Causeway
Kennedy Parkway North
Kennedy Parkway North
Instrumentation Road
Contractors Road
P
P
KSC
LC-
39A

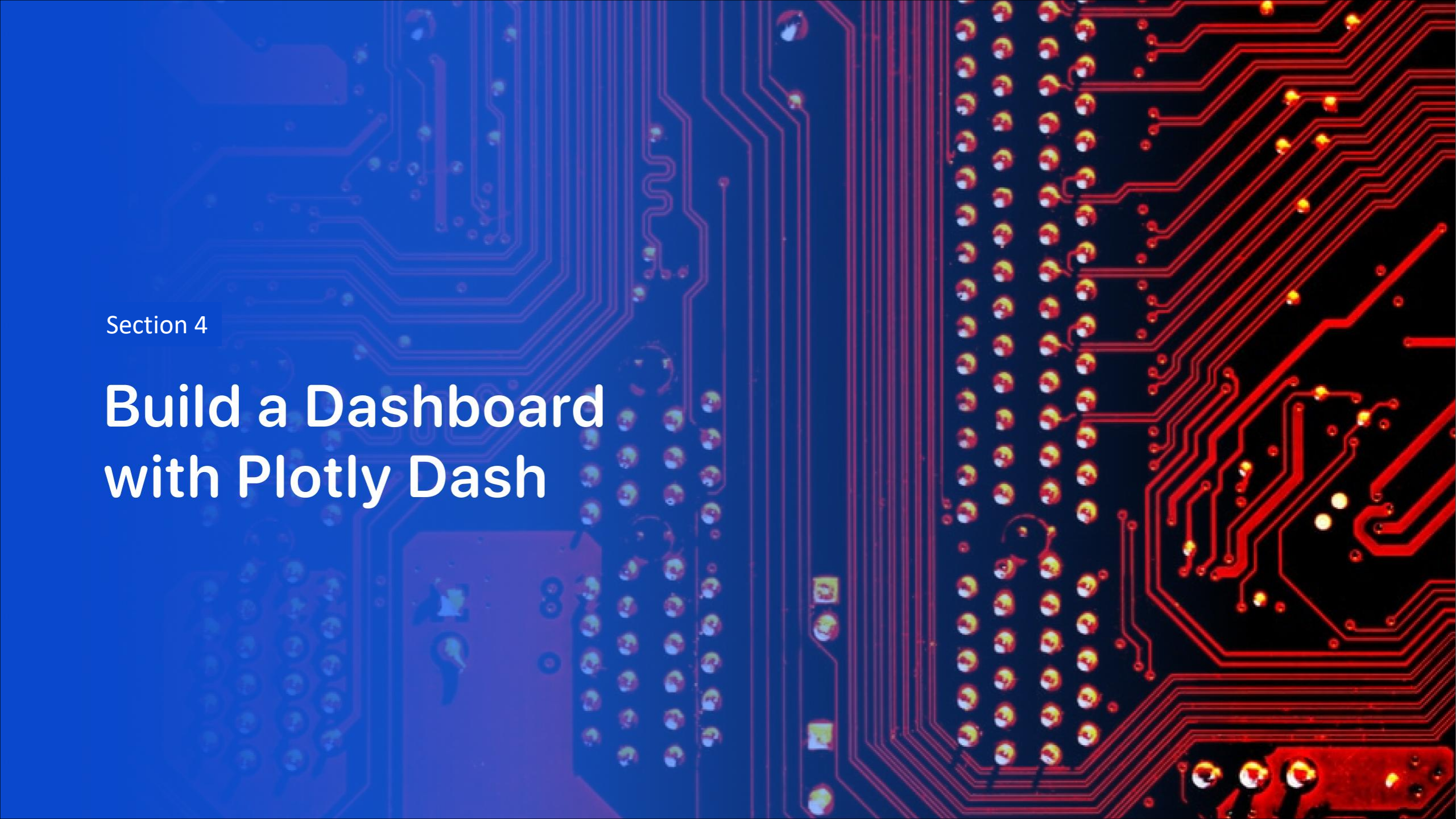

Vandenberg
Space Launch
Complex
4
Dix Road
VAFB
SLC-
4E

DISTANCES TO PROXIMITIES

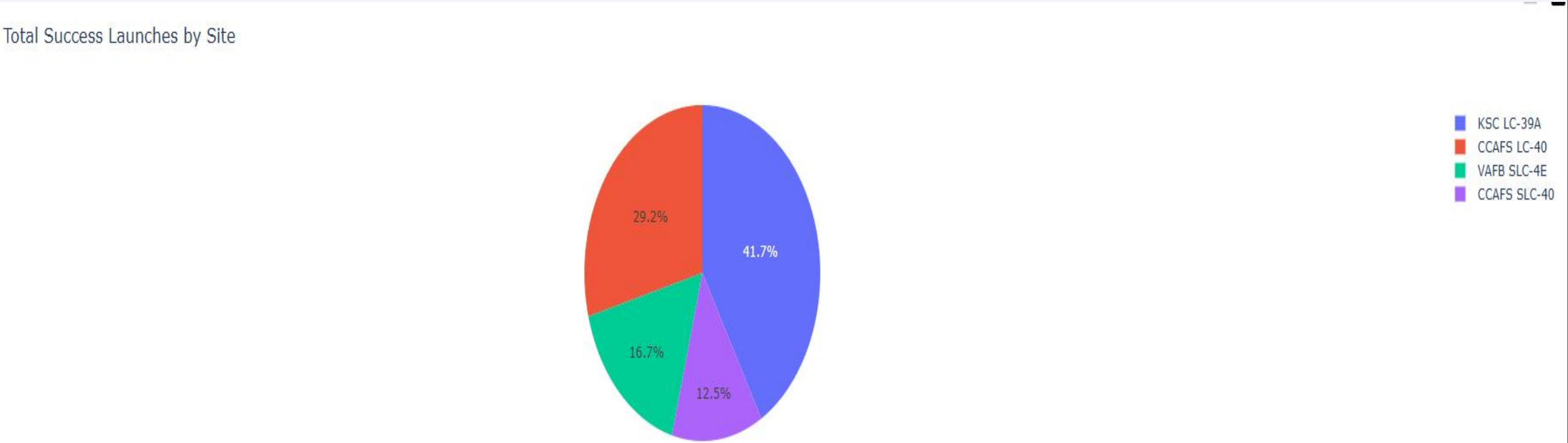


Section 4

Build a Dashboard with Plotly Dash

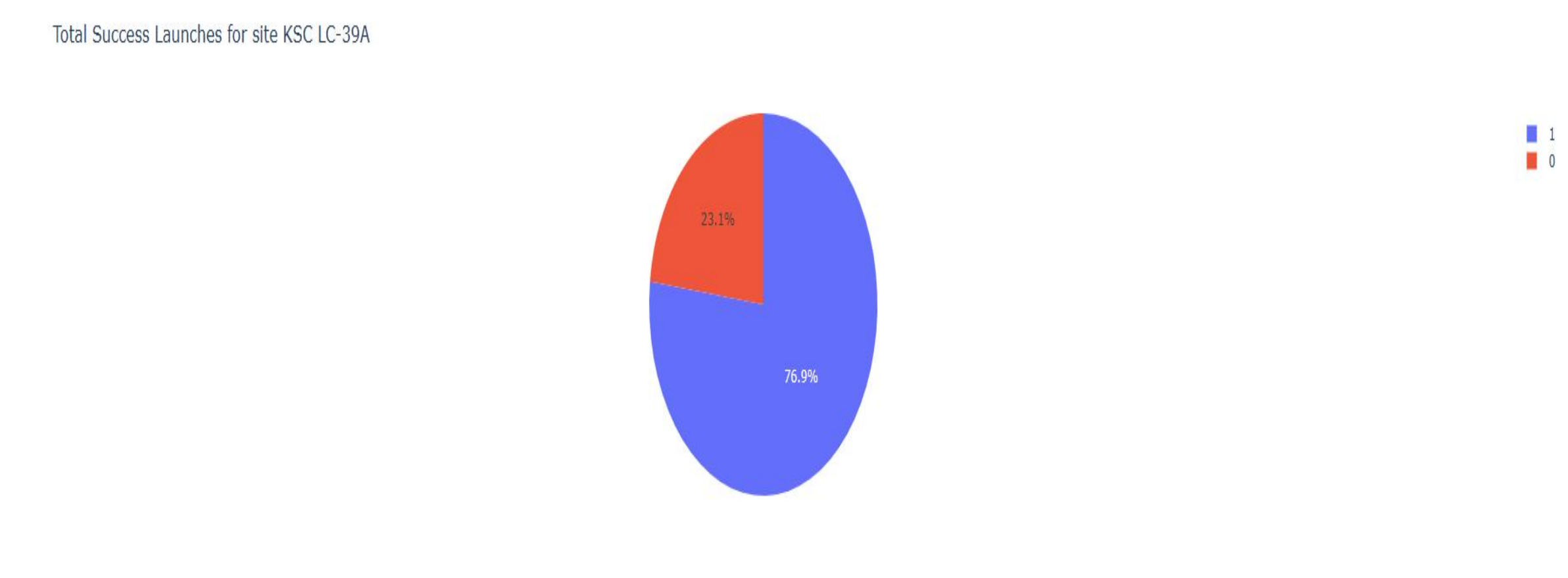


SHOW THE SCREENSHOT OF LAUNCH SUCCESS COUNT FOR ALL SITES, IN A PIECHART



KSC LC-39A has the most successful launches amongst launch sites .(**41.2%**)

PIECHART FOR THE LAUNCH SITE WITH HIGHEST LAUNCH SUCCESS RATIO



KSC LC-39A has the highest success rate amongst launch sites (**76.9%**)

10 successful launches and 3 failed launches.

Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Section 5

Predictive Analysis (Classification)

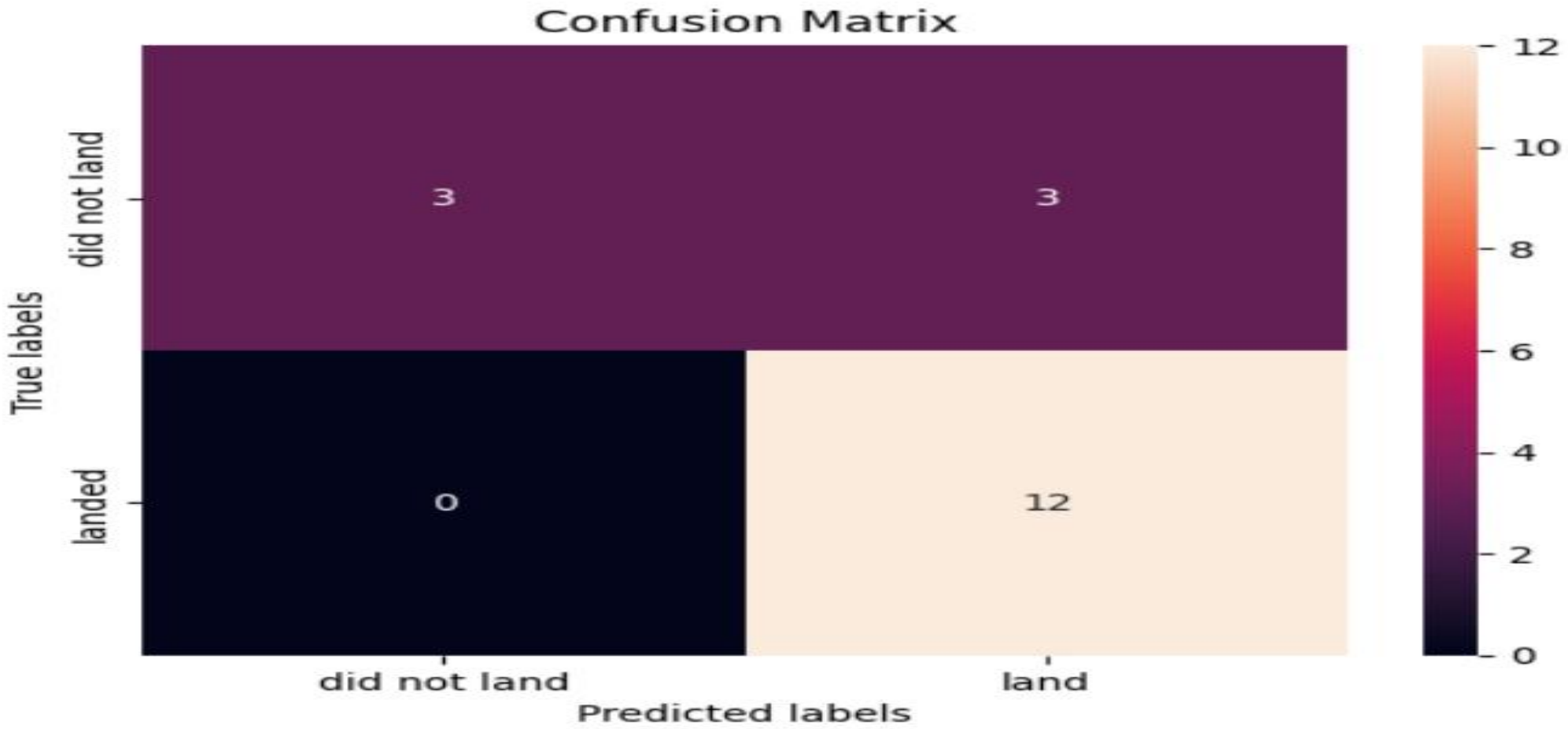
CLASSIFICATION ACCURACY

```
grid_searches = {
    'Decision Tree': tree_cv,
    'SVM':           svm_cv,
    'KNN':           knn_cv,
    'Logistic Reg.':  logreg_cv
}

best_name, best_cv = max(
    grid_searches.items(),
    key=lambda item: item[1].best_score_
)

print(f"🍗 Best model (by CV score): {best_name}")
print(f"Best cross-validation accuracy: {best_cv.best_score_ * 100:.2f}%")
```

```
🍗 Best model (by CV score): Decision Tree
Best cross-validation accuracy: 87.50%
```

CONFUSION MATRIX

```
]: yhat = tree_cv.predict(X_test)
   plot_confusion_matrix(Y_test,yhat)
```



CONCLUSIONS

- **The larger the flight amount at a launch site, the greater the success rate at a launch site.**
- **All the launch sites are close to the coast**
- **Launch Success: Increases over time**
- **Launch success rate started to increase in 2013 till 2020.**
- **Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.**
- **KSC LC-39A had the most successful launches of any sites.**
- **The Decision tree classifier is the best machine learning algorithm for this task.**

APPENDIX

- **https://github.com/sahir-dev/IBM-DSC-SpaceX/tree/main**

Thank you!