IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Isaac Beton
31 March 2024

Outline

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

Executive Summary

SpaceX data was collected and analysed to discover trends and make predictions, which may enable SpaceY to gain a competitive advantage. The methods and results are summarised below.

Methodologies

- Data Collection – API and Webscraping
- Data Wrangling – cleaning and preparing
- Exploratory Data Analysis (EDA) – SQL, Pandas, Matplotlib
- Interactive Visual Analytics – Folium Map and Plotly Dashboard
- Predictive Analysis –
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K-Nearest Neighbour (KNN) classifiers

Results

- Determined best Hyperparameters for models
- All models performed equally – approx. 83% accuracy
- False-positives were the main errors within the models

Introduction

Commercial Spaceflight is expensive and competitive – cost savings from re-using first-stage rockets gives a commercial advantage.

SpaceY has conducted an analysis of our main competitor, SpaceX, to determine their likelihood of a successful landing.

We have used SpaceX launch data to build Machine Learning models, which enables us to discover trends and make predictions.

Predicting whether a landing is successful enables us to estimate their costs, which will inform our future commercial decisions.

Section 1

Methodology

Methodology

Executive Summary

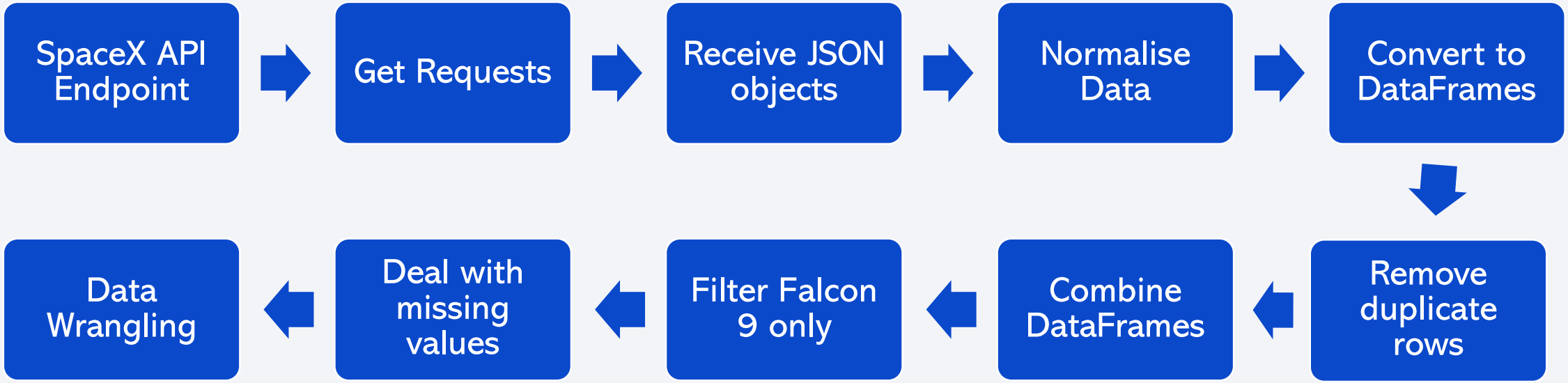
- Data collection methodology:
  - Data was collected from SpaceX API and Webscraping using a get-request.
- Perform data wrangling
  - API data was pre-formatted in a JSON object.
  - Webscraped data was parsed using BeautifulSoup to extract HTML tables.
  - These were converted to Pandas DataFrames
- Perform exploratory data analysis (EDA) using visualisation and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Split data in train and test sets
  - Used GridSearchCV to tune Machine Learning Models hyperparameters
  - Plot Confusion Matrices
  - Compare models

Data Collection

Data Collection can be broadly described by the below, but varies with method:

Data Collection – SpaceX API

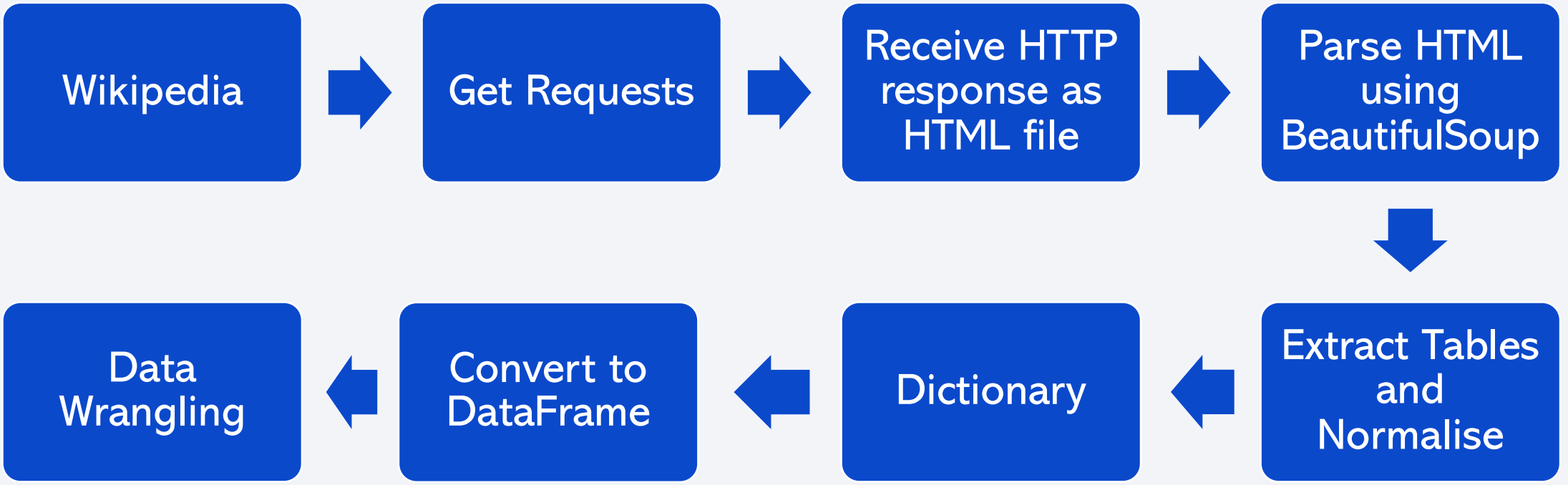
Collect data from SpaceX API and correctly format for analysis



[Data Collection – API notebook]

Data Collection – Webscraping

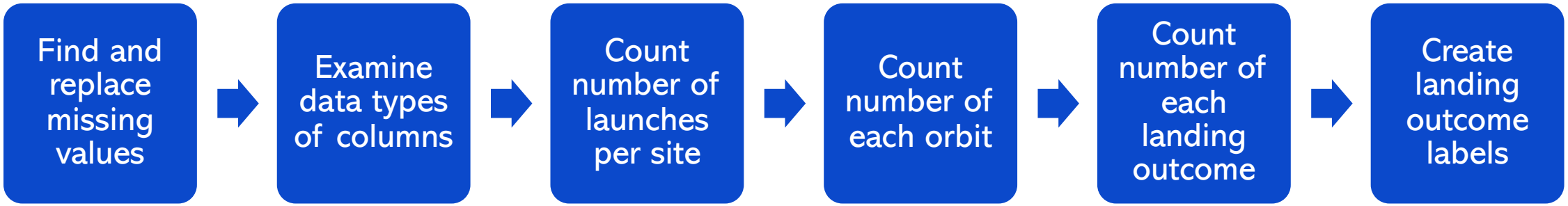
Collect data from HTML tables on Wikipedia and correctly format for analysis



Data Collection – Webscraping notebook

Data Wrangling

Clean data and assign each of the 8 landing outcomes a pass (1) or fail (0)



Data Wrangling notebook

EDA with Data Visualisation

Summary of charts:

- Scatter plot (Catplot and Relplot), with marker colour corresponding with PASS/FAIL:
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload Mass vs Launch Site
  - Flight Number vs Orbit Type
  - Payload Mass vs Orbit Type
- Bar Chart
  - Orbit Type vs Success Rate
- Line Chart
  - Mean Yearly Success Rate

EDA – Data Visualisation notebook

EDA with SQL

Summary of SQL queries:

- Number of unique Launch Sites:

```
%sql select distinct(launch_site) from SPACEXTBL
```

- 5 records of Launch Sites beginning with “CCA”:

```
%%sql
select * from SPACEXTBL
where launch_site like "CCA%"
limit 5
```

- Total Payload Mass carried by boosters launched by NASA (CRS):

```
%sql select sum(PAYLOAD_MASS__KG_) as 'total payload mass carried by boosters launched by NASA (CRS)'
from SPACEXTBL where customer = 'NASA (CRS)'
```

- Average Payload Mass carried by booster version F9 v1.1:

```
%sql select avg(PAYLOAD_MASS__KG_) as 'average payload mass carried by booster version F9 v1.1' from
SPACEXTBL where booster_version='F9 v1.1’
```

- Date of first successful ground pad landing:

```
%%sql
select min(date) from SPACEXTBL
where Landing_Outcome='Success (ground pad)'
```

EDA – SQL notebook

EDA with SQL

Summary of SQL queries:

- Names of Boosters successfully landed on drone ships with 4000kg < mass <6000kg:

```
%%sql
select Booster_Version from SPACEXTBL
where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4001 and 5999
```

- Total number of each Mission Outcome:

```
%%sql
Select Mission_Outcome, Count(Mission_Outcome)
From SPACEXTBL
Group by Mission_Outcome
```

- Names of Boosters that have carried the maximum Payload Mass:

```
%%sql
Select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL
where PAYLOAD_MASS__KG_ = (Select Max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

EDA – SQL notebook

EDA with SQL

Summary of SQL queries:

- List the month, Landing Outcome, Booster Version, and Launch Site for failed drone ship landings in 2015:

```
%%sql
Select substr(date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL
where Landing_Outcome = 'Failure (drone ship)'
and substr(Date,0,5)='2015'
```

- In descending order, list the Landing Outcomes between 2010-06-04 and 2017-03-20:

```
%%sql
Select Landing_Outcome,count(*) as count_outcomes
from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
Order by count_outcomes DESC
```

EDA – SQL notebook

Build an Interactive Map with Folium

Summary of Folium Map objects:

- *folium.Circle* : Adds shaded circle area – visually locates the launch sites
- *folium.map.Marker* : Adds text label – used to identify launch sites and distances on the map
- *folium.Icon* : Adds colour-code to Markers – shows successful and unsuccessful launches
- *MarkerCluster()* : Groups nearby markers together – visually simplifies and shows launch site distribution
- *MousePosition()* : Adds coordinate readout of the mouse position
- *folium.Polyline* : Adds a line from a launch site to specified coordinates – visually identifies the distance from launch site to coast, closest city, railways, highways

Interactive Visualisation notebook

Build a Dashboard with Plotly Dash

Summary of Interactive Objects and Visualisations:

- *dcc.Dropdown* : Adds a dropdown list input – enables user to select a specific launch site to view data for
- *dcc.Rangeslider* : Adds a rangeslider input – enables user to select a range of payloads to view data for
- *px.pie* : Adds a pie chart – shows the proportion of successful and unsuccessful launches for the selected launch site
- *px.scatter* : Adds a scatter plot – shows the relationship (if any) between two variables, in this case Success vs Payload Mass

Note:

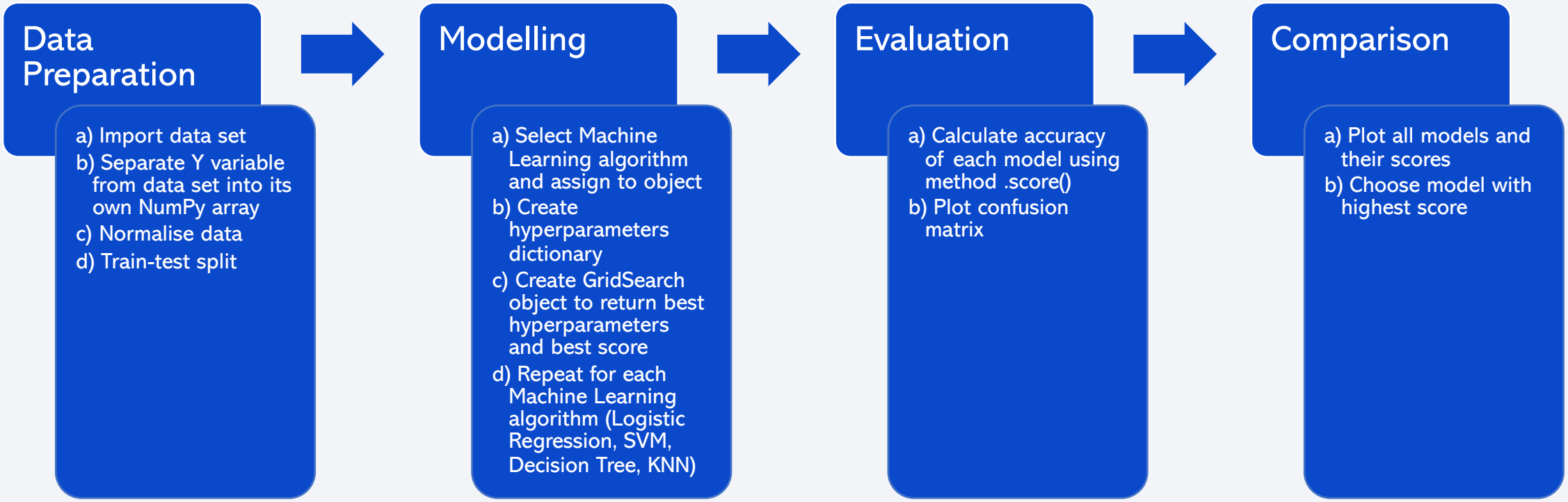
dcc : dash_core_components

px : plotly express

Dashboard code

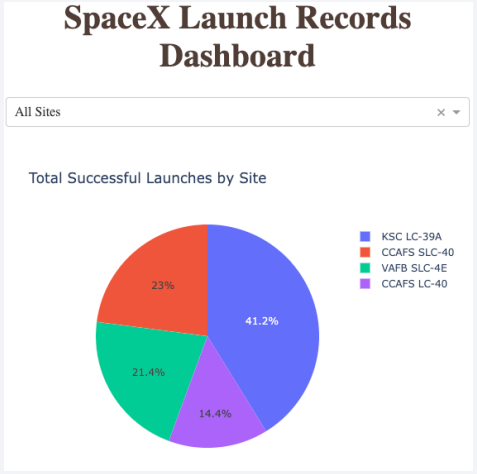
Predictive Analysis (Classification)

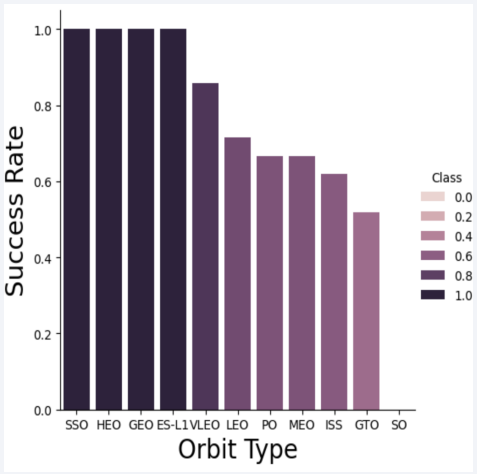
Summary of building and assessing Classification Models:

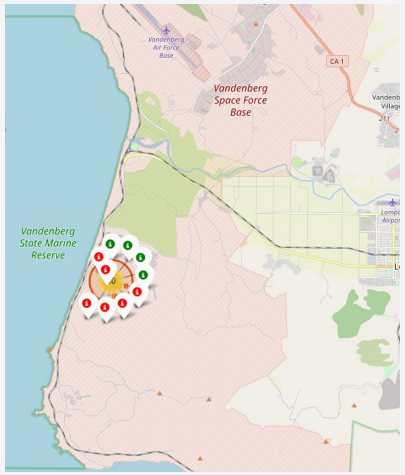


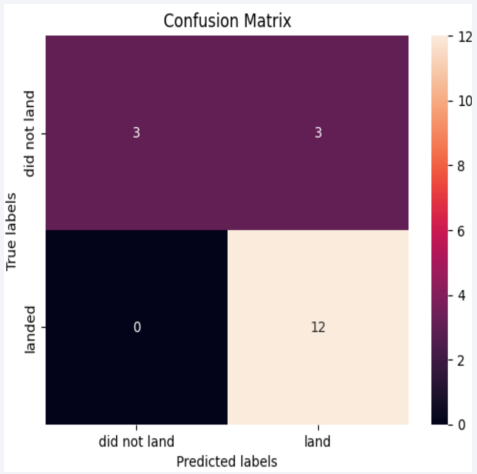
Predictive Analysis notebook

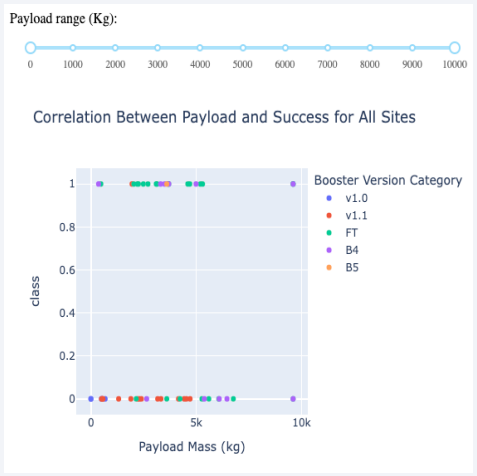
Results

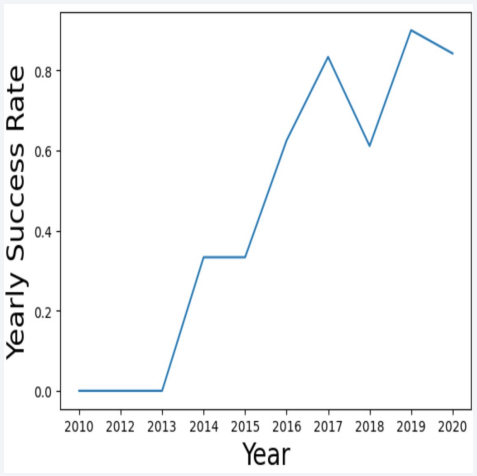

SpaceX Launch Records Dashboard
All Sites
Total Successful Launches by Site
41.2%
23%
21.4%
14.4%
KSC LC-39A
CCAFS SLC-40
VAFB SLC-4E
CCAFS LC-40

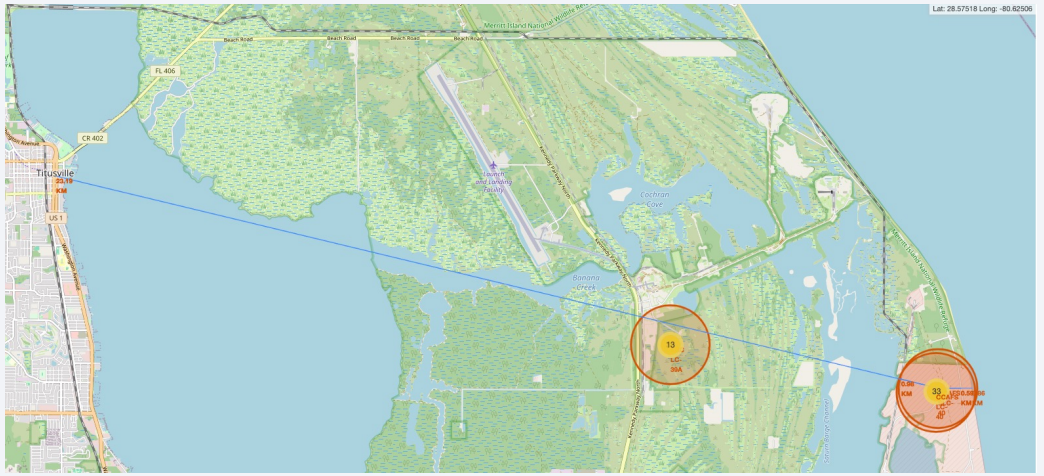

Success Rate
Orbit Type
SSO HEO GEO ES-L1 VLEO LEO PO MEO ISS GTO SO
Class

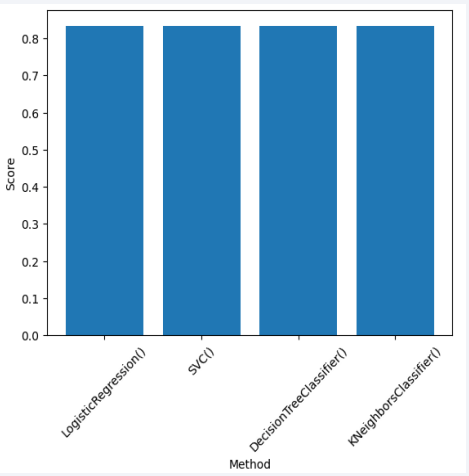

Vandenberg Space Force Base
Vandenberg State Marine Reserve

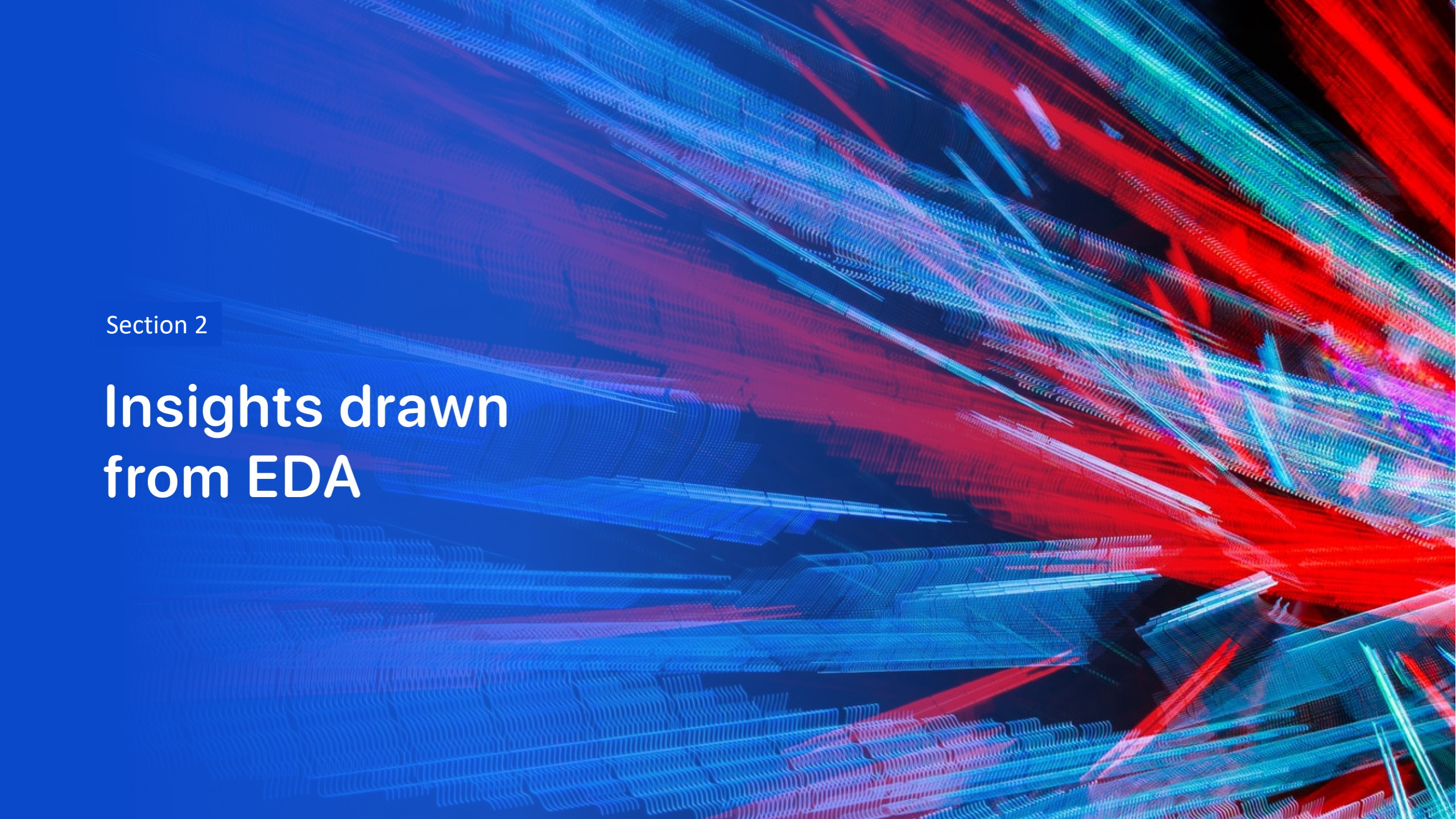

Confusion Matrix
True labels
did not land
landed
Predicted labels
did not land
land
3
3
0
12


Payload range (Kg):
Correlation Between Payload and Success for All Sites
Booster Version Category
v1.0
v1.1
FT
B4
B5
class
Payload Mass (kg)

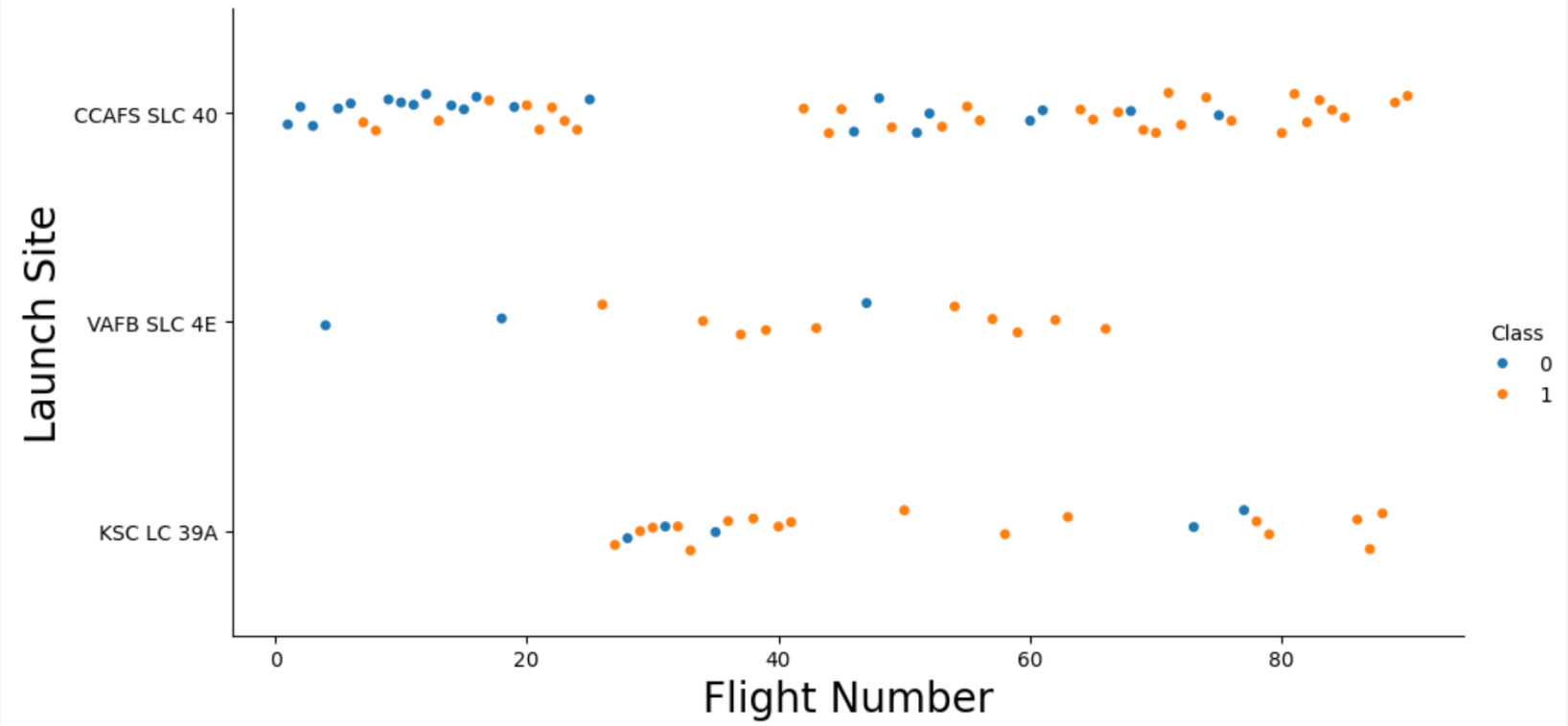

Yearly Success Rate
Year




Score
Method
LogisticRegression()
SVC()
DecisionTreeClassifier()
KNeighborsClassifier()

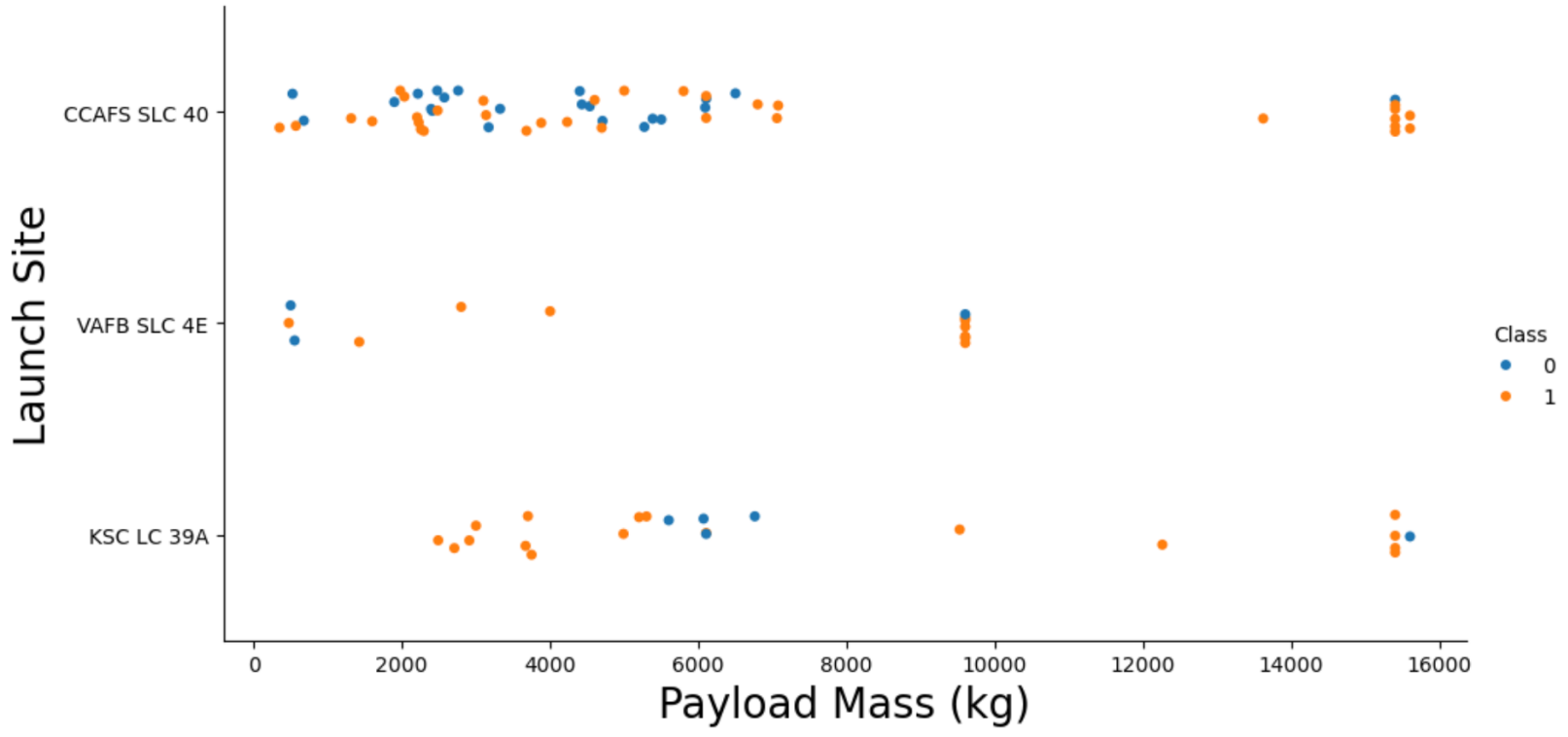

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



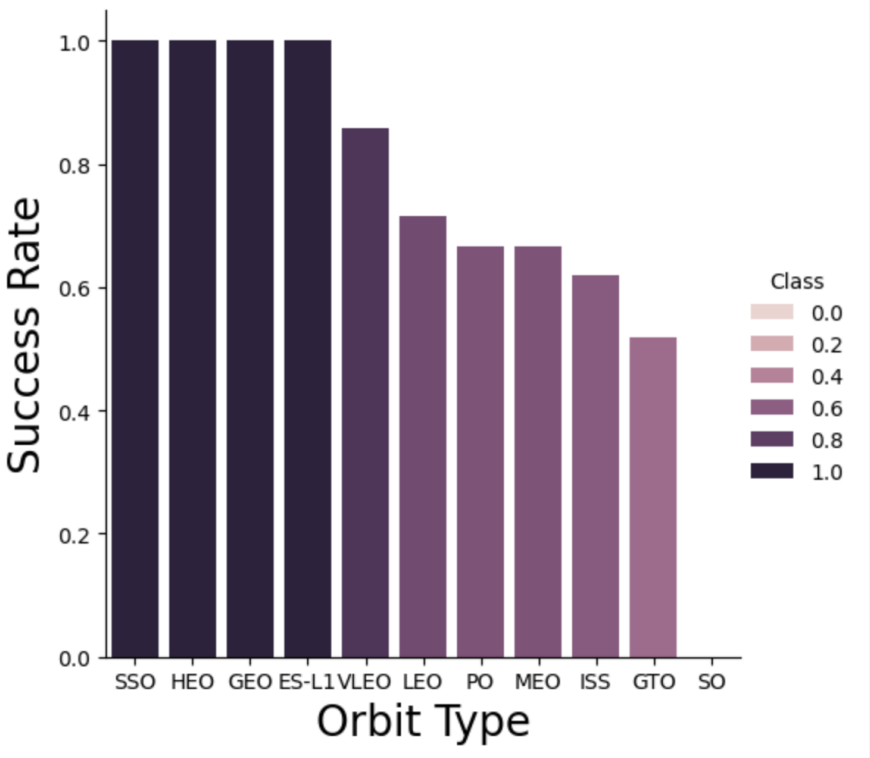
- Class 1, the successful launches, occur more frequently as Flight Number increases, which implies that each launch enables learning and improvement
- CCAFS SLC 40 was launched at most frequently, which implies it was likely viewed as the most practical/economical/accessible by the SpaceX team.

Payload vs. Launch Site



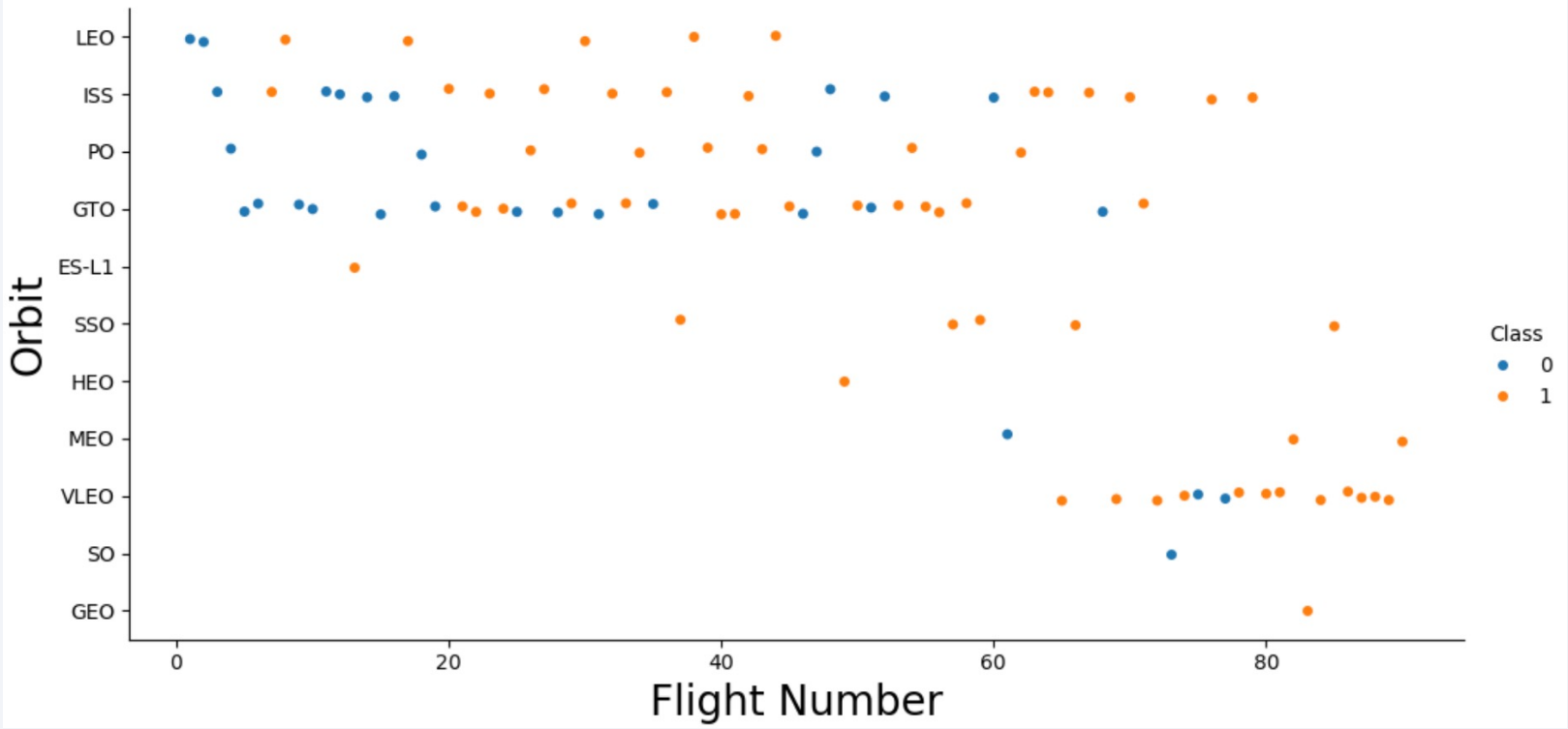
- Payload masses were most frequently less than 7500kg at all sites, for which there does not appear to be a correlation between Payload Mass and Success Rate
- For heavier Payload Masses, the Success Rate appears to be far higher

Success Rate vs. Orbit Type



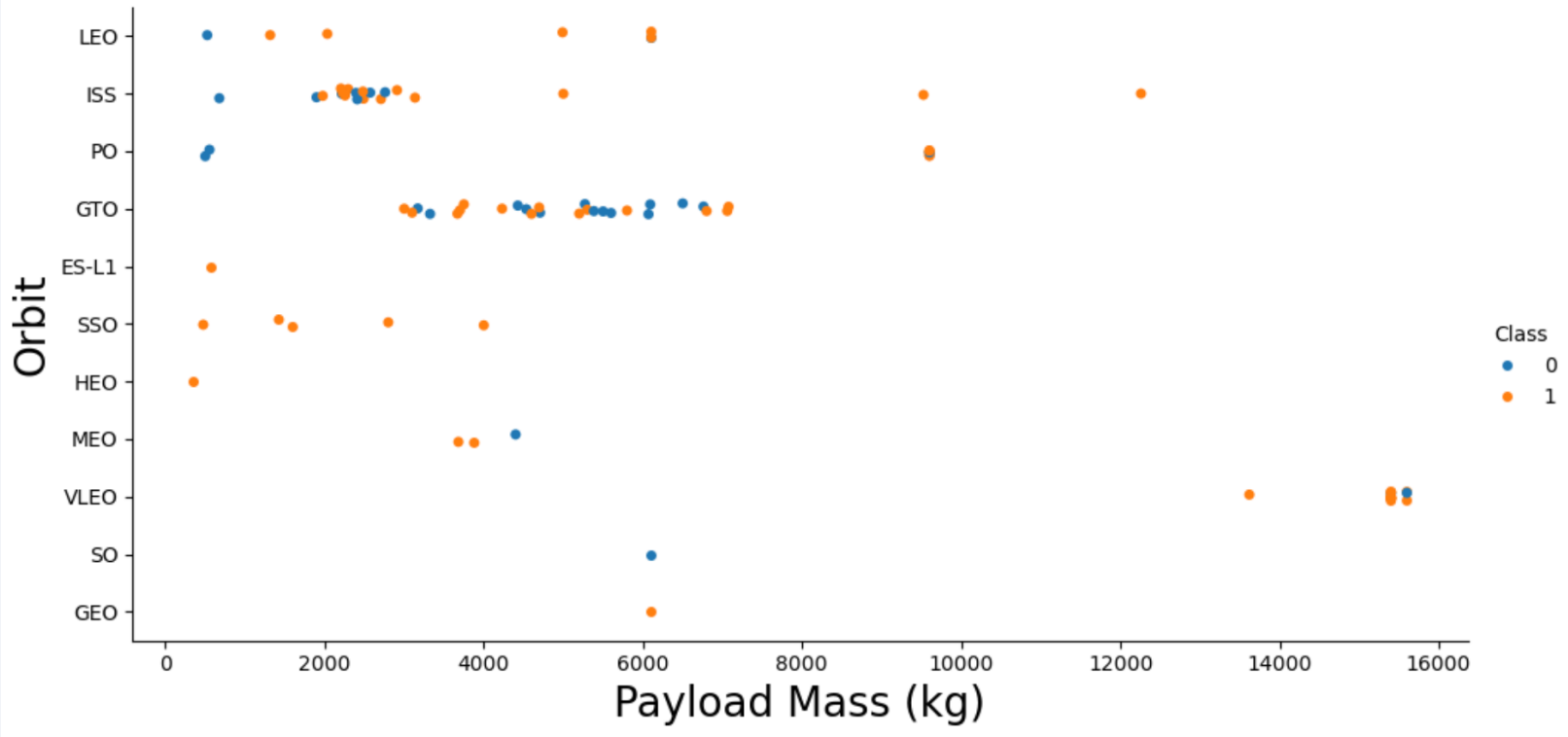
- Orbits SSO, HEO, GEO, and ES-L1 had no failures, whilst Orbit SO had no successes
- High or low success rates may be skewed by sample size – it is not clear how many launches have occurred from this plot alone

Flight Number vs. Orbit Type



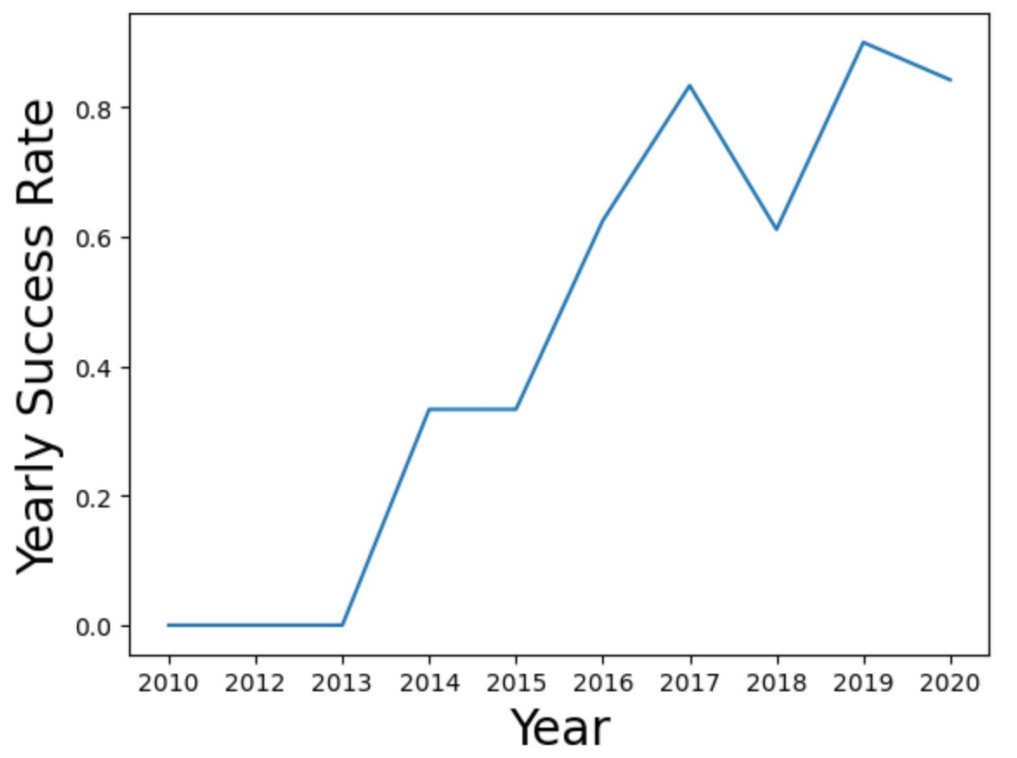
- Orbit GTO is the most frequently launched
- The least frequently launched orbits were SSO, HEO, GEO, ES-L1, and SO, which were the most/least successful orbits in the previous slide

Payload vs. Orbit Type



- Orbit GTO, the most frequently launched Orbit, has the smallest range of Payload Masses
- Orbit ISS, the 2nd most frequently launched Orbit, has the largest range of Payload Masses but appears to most frequently launch Payload Masses between 2000kg to 4000kg

Launch Success Yearly Trend



- Success rate is generally increasing
- 2018 and 2020 were less successful than their previous years

All Launch Site Names

- SQL query

```
%sql select distinct(launch_site) from SPACEXTBL
```

- Explanation

`%sql` – line magic – makes Python read anything within that line as if it were written in SQL

`distinct(launch_site)` – selects unique names from the column 'Launch_Site'

- Query Result:

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

Launch Site Names Begin with 'CCA'

- SQL query

```
%%sql
select * from SPACEXTBL
where launch_site like "CCA%"
limit 5
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

`select *` - selects all

`launch_site like "CCA%"` – search for entries in `launch_site` starting with `CCA` and ending with anything, `%` (a wildcard)

`limit 5` - only return the first 5 instances

- Query Result:

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

Total Payload Mass

- SQL query

```
%sql select sum(PAYLOAD_MASS__KG_) as 'total payload
mass carried by boosters launched by NASA (CRS)' from
SPACEXTBL where customer = 'NASA (CRS)'
```

- Explanation

`%sql` – line magic – makes Python read anything within that line as if it were written in SQL

`sum(PAYLOAD_MASS__KG_)` – return the sum of the column `PAYLOAD_MASS__KG_`

`where customer = 'NASA (CRS)'` – search for entries in `customer` that equal `NASA (CRS)`

- Query Result:

total payload mass carried by boosters launched by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- SQL query

```
%sql select avg(PAYLOAD_MASS__KG_) as 'average payload
mass carried by booster version F9 v1.1' from SPACEXTBL
where booster_version='F9 v1.1'
```

- Explanation

`%sql` – line magic – makes Python read anything within that line as if it were written in SQL

`avg(PAYLOAD_MASS__KG_)` – return the average of the column `PAYLOAD_MASS__KG_`

`booster_version='F9 v1.1'` – search for entries in `booster_version` that equal `F9 v1.1`

- Query Result:

average payload mass carried by booster version F9 v1.1
2928.4

First Successful Ground Landing Date

- SQL query

```
%%sql
select min(date) from SPACEXTBL
where Landing_Outcome='Success (ground pad)'
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

`min(date)`– return the first (minimum) of the column `date`

`Landing_Outcome='Success (ground pad)'`– search for entries in `Landing_Outcome` that equal `Success (ground pad)`

- Query Result:

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL query

```
%%sql
select Booster_Version from SPACEXTBL
where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4001 and 5999
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

`Landing_Outcome='Success (drone ship)'` – search for entries in `Landing_Outcome` that equal `Success (drone ship)`

`PAYLOAD_MASS__KG_ between 4001 and 5999` – search for entries with value 4001 to 5999 (inclusive), which are the values between 4000 and 6000

- Query Result:

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

Total Number of Successful and Failure Mission Outcomes

- SQL query

```
%%sql
Select Mission_Outcome, Count(Mission_Outcome)
From SPACEXTBL
Group by Mission_Outcome
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

select entries in `Mission_Outcome` and the count of each mission outcome, grouped by `Mission_Outcome`

- Query Result:

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

Boosters Carried Maximum Payload

- SQL query

```
%%sql
Select Booster_Version, PAYLOAD_MASS__KG_ from
SPACEXTBL
where PAYLOAD_MASS__KG_ = (Select
Max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

`where PAYLOAD_MASS__KG_ = (Select Max(PAYLOAD_MASS__KG_)` – Filter the entries within `PAYLOAD_MASS__KG_` that are equal to the maximum value of `PAYLOAD_MASS__KG_`

- Query Result:

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

2015 Launch Records

- SQL query

```
%%sql
Select substr(date,6,2) as Month, Landing_Outcome,
Booster_Version, Launch_Site from SPACEXTBL
where Landing_Outcome = 'Failure (drone ship)'
and substr(Date,0,5)='2015'
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

`substr(date,6,2)` – Select 2 characters of the strings within the `date` column, starting from on the 6th character. Since dates are stored as YYYY-MM-DD, the 6th and 7th characters are the month number.

`where Landing_Outcome = 'Failure (drone ship)'` – Filter the entries within `Landing_Outcome` that are equal to `2015`

- Query Result:

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

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

- SQL query

```
%%sql
Select Landing_Outcome,count(*) as count_outcomes
from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
Order by count_outcomes DESC
```

- Explanation

`%%sql` – cell magic – makes Python read anything within that cell as if it were written in SQL

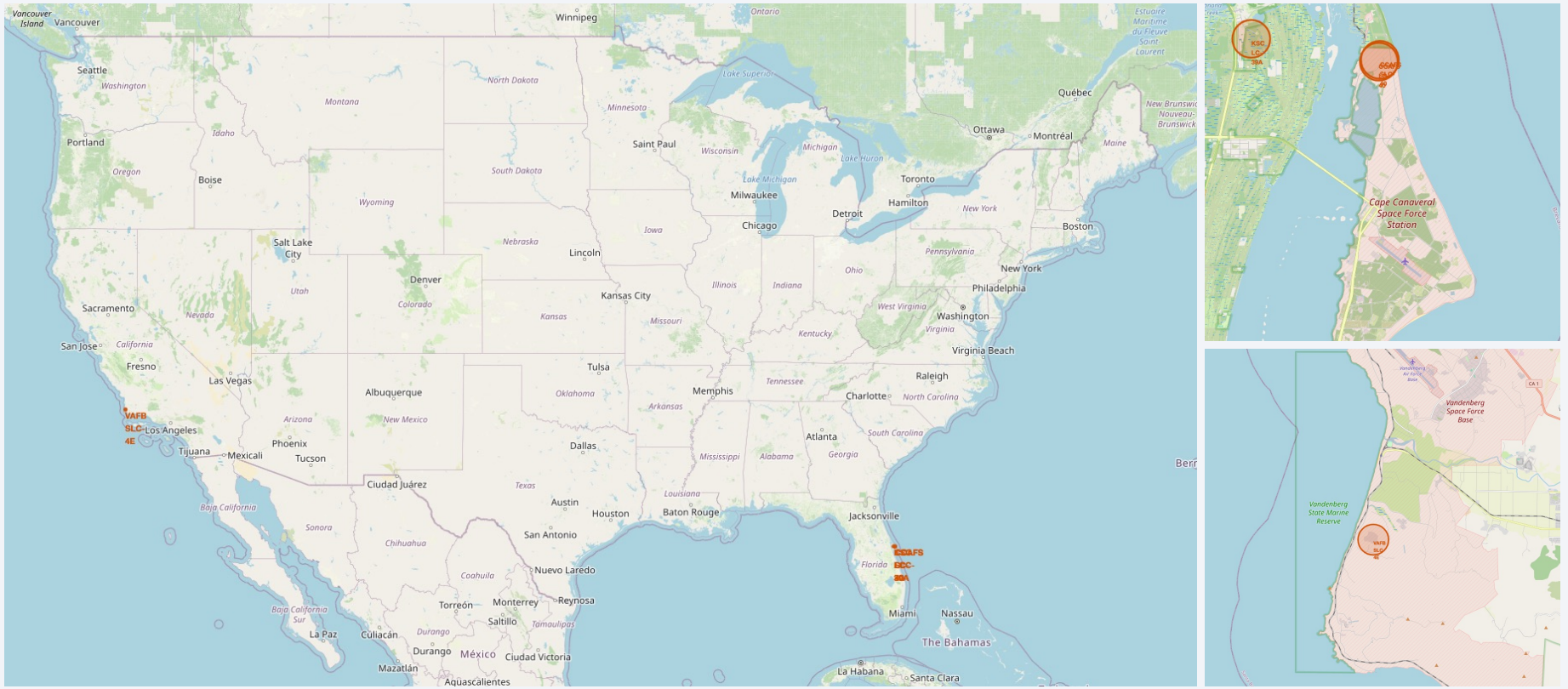
`Select Landing_Outcome,count(*)` – Select from the column `Landing_Outcome` and their counts where the date is between `2010-06-04` and `2017-03-20`. The landing outcomes are grouped and then the result is arranged in descending order.

- Query Result:

Landing_Outcome	count_outcomes
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

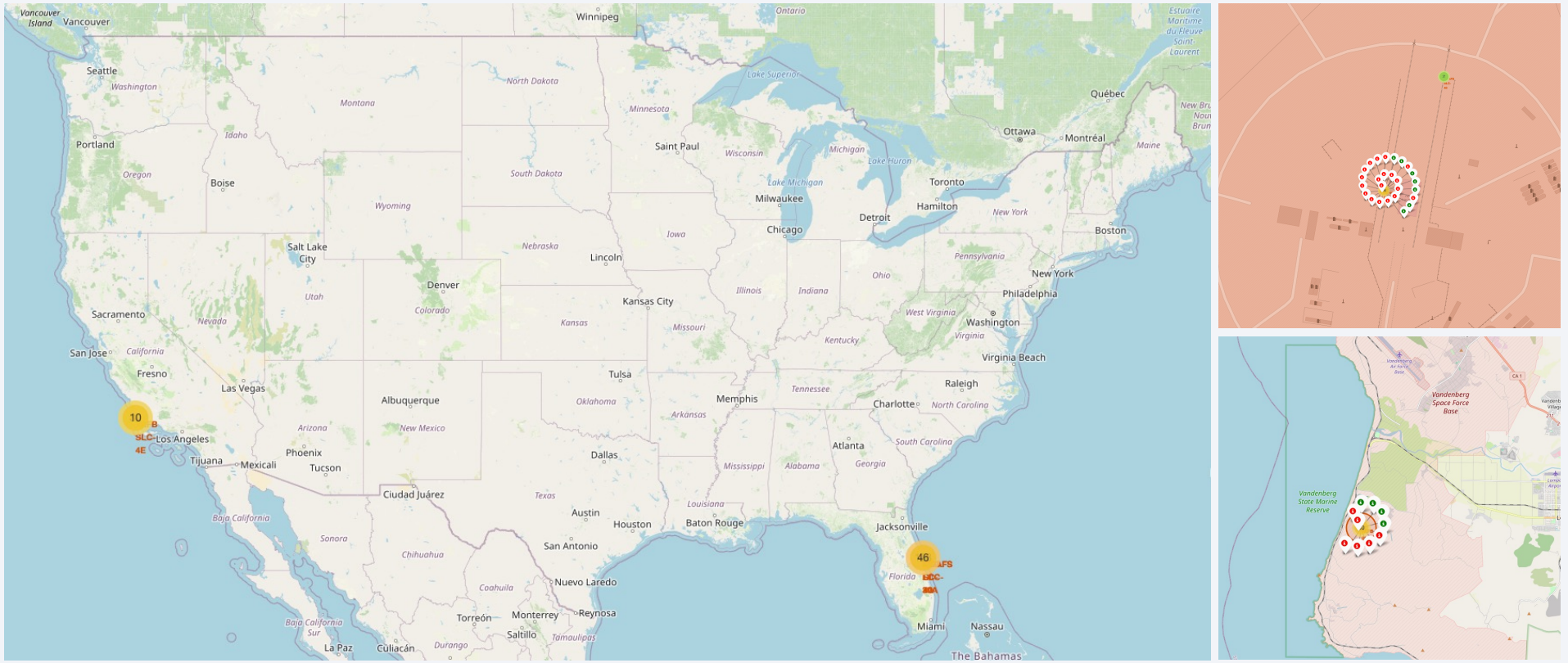

Section 3
Launch Sites
Proximities Analysis

All Launch Sites



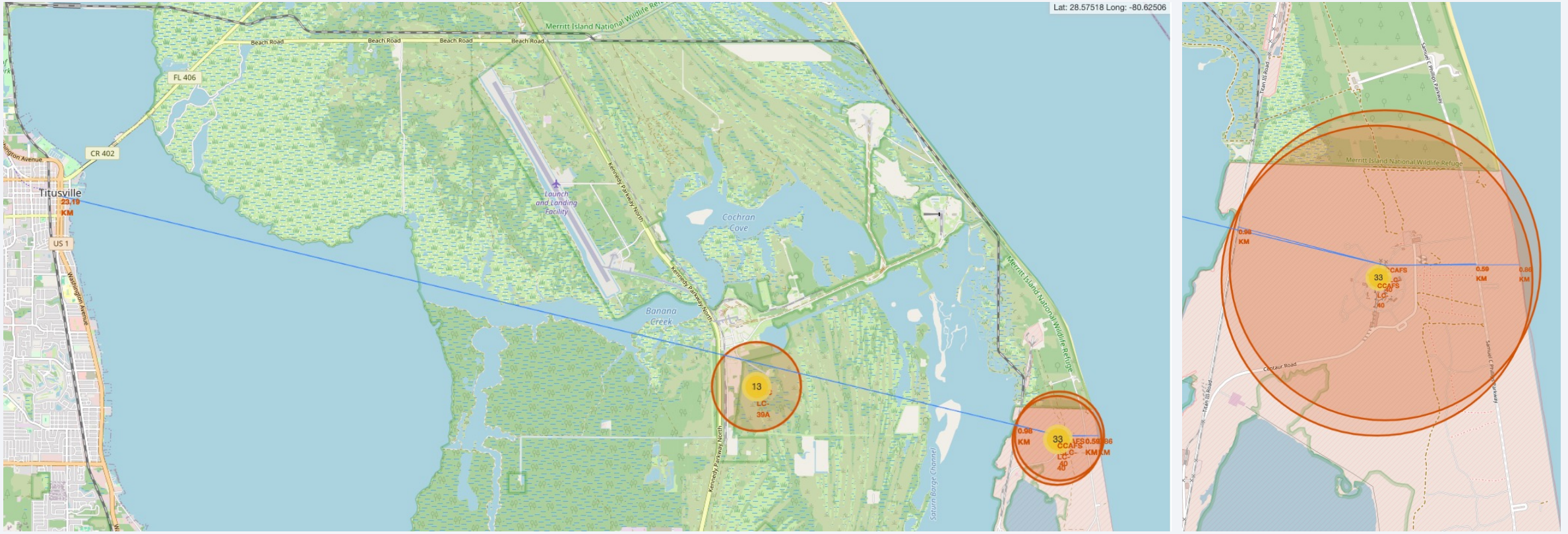
- Launch sites are on the coast
- Launch sites are close to or within the restricted areas of U.S. Space Force bases

Launch Outcomes

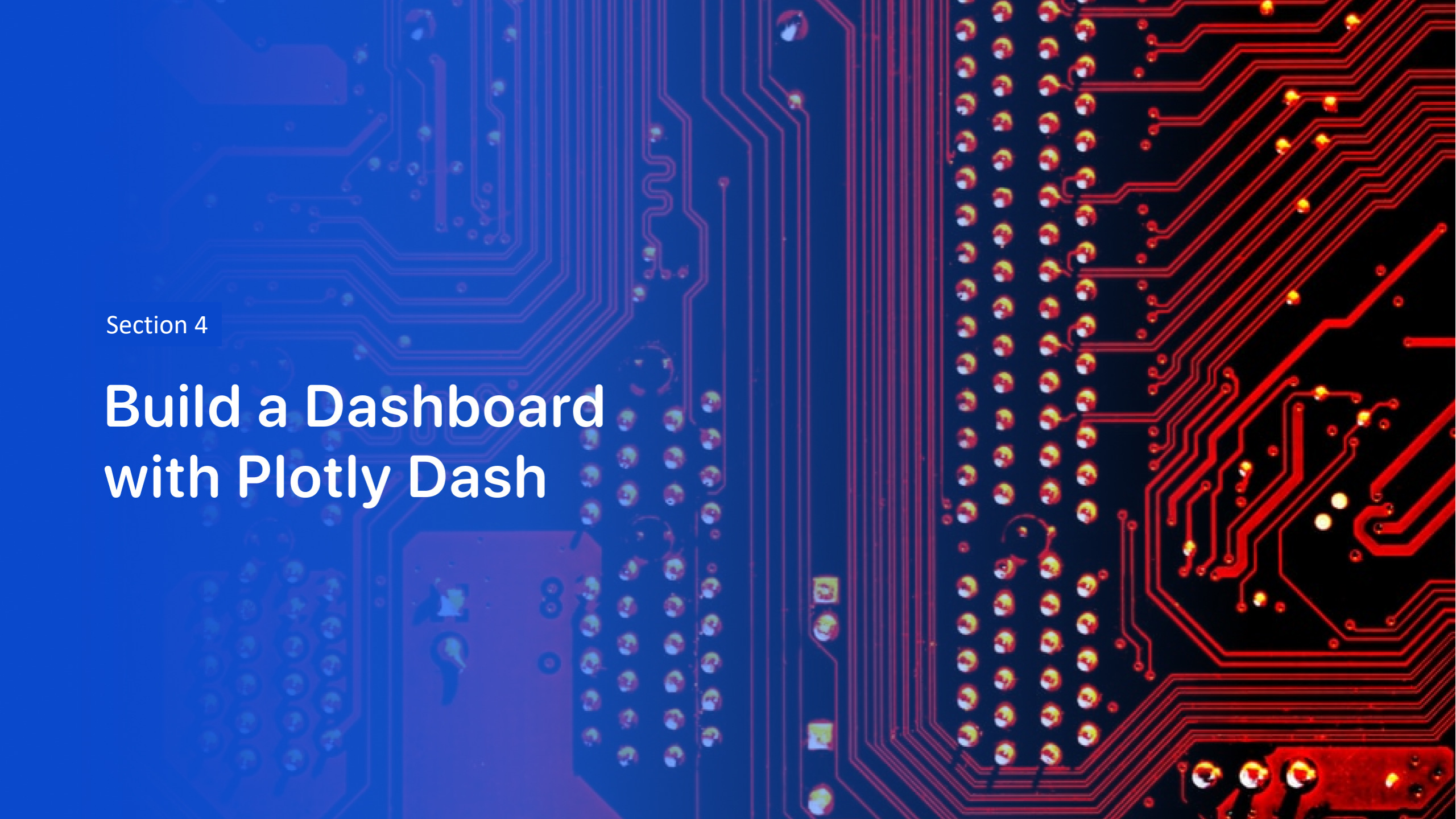


- Launches sites are clustered together and the total number of launches are displayed
- When a launch site is clicked on, it shows successful launches with a green icon and unsuccessful with a red icon

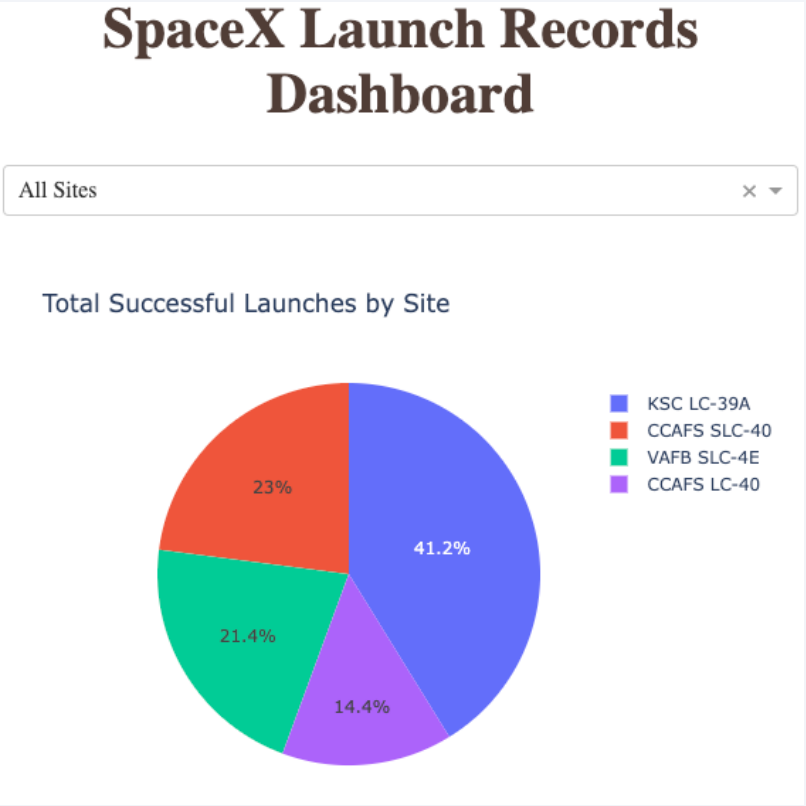
Launch Site Proximity



- Launch sites are close to railways, highways, and coastlines
- Launch sites are far enough away from cities to maintain safe distance, but close enough for supplies

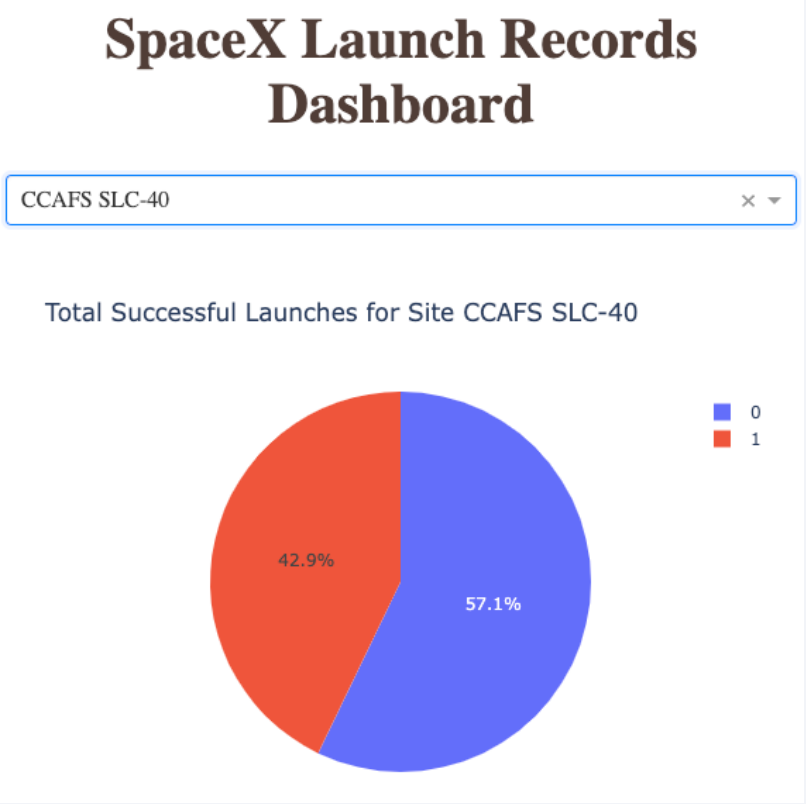

Section 4
Build a Dashboard with Plotly Dash

Success Rate – All Sites



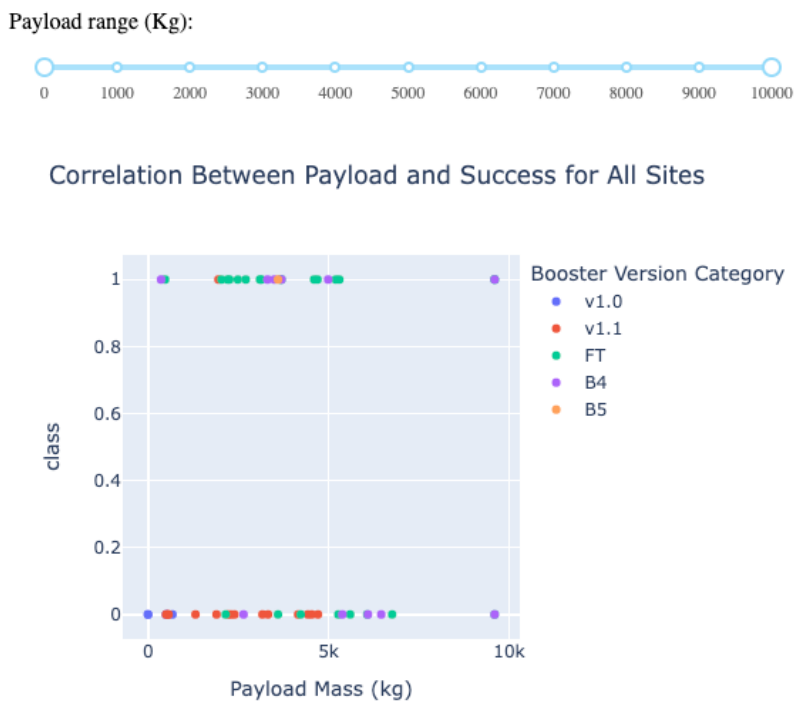
- Site KSC LC-39A has contributed the largest number of successful launches
- This does not imply it has the highest success ratio – if it launches significantly more flights than the rest, it could have a low success ratio and still contribute the same number of successful launches

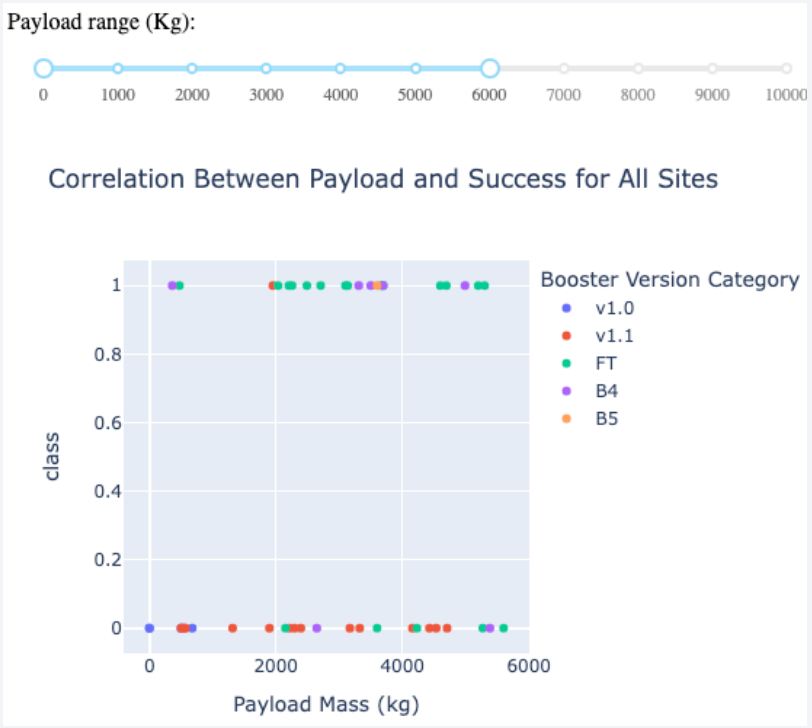
Success Rate – CCAFS SLC-40

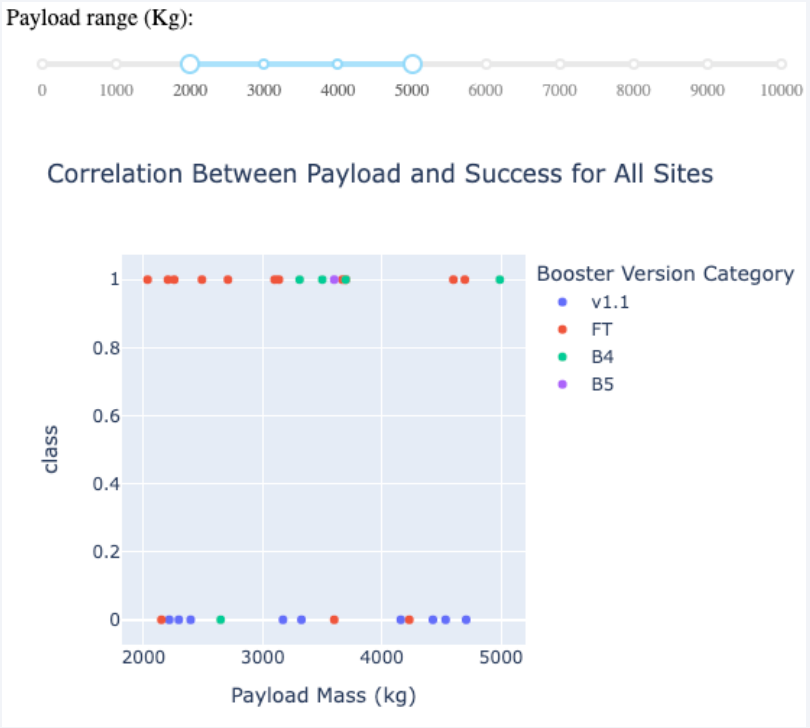


- CCAFS SLC-40 had the 2nd highest number of successful launches, but the highest success ratio.
- This implies that the site with the most successful launches likely launched many more flights than CCAFS SLC-40

Payloads vs Success – All Sites





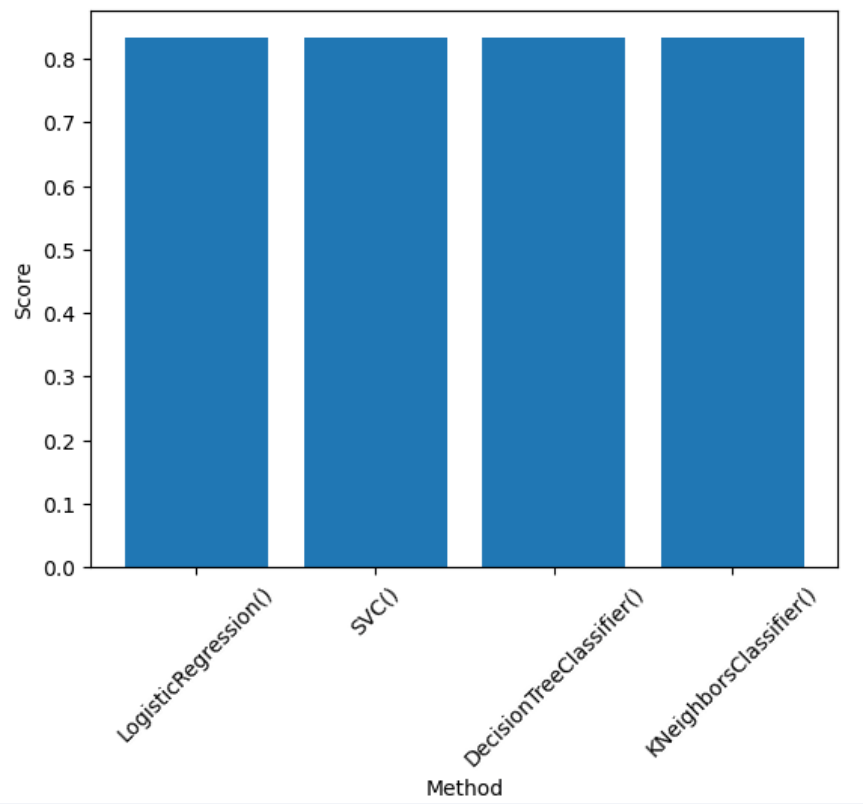


- Most launches were less than 7000kg
- The most successful Booster Version was “FT”
- The most successful Payload range was approximately between 2000kg to 4000kg

Section 5

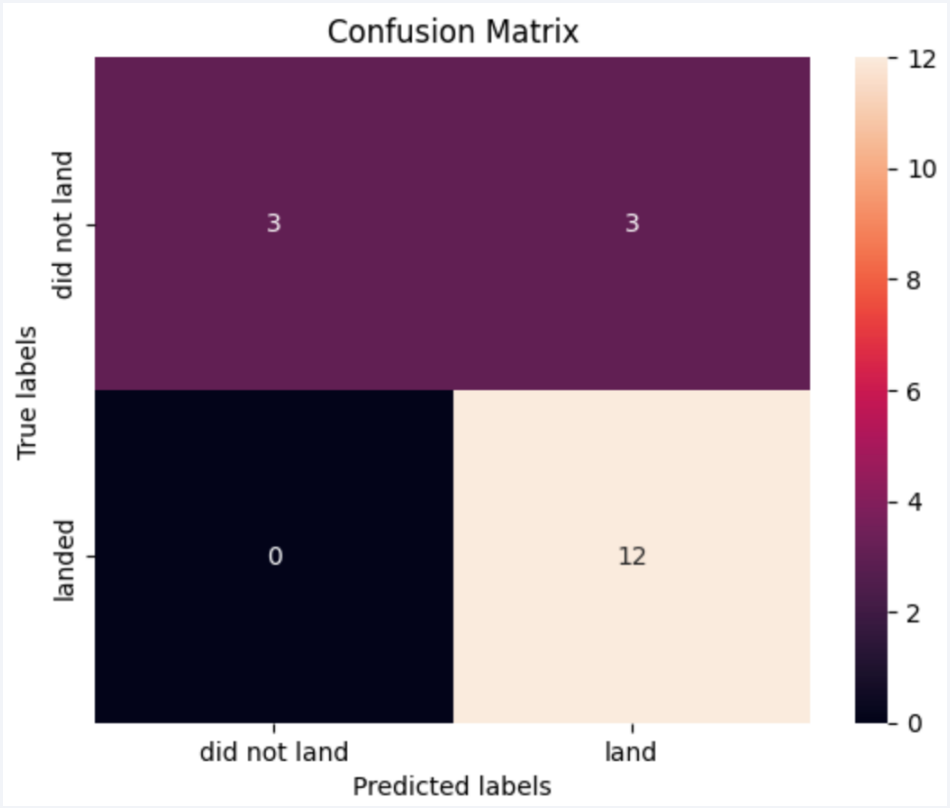
Predictive Analysis (Classification)

Classification Accuracy



- All models had the same score
- All models perform equally well, noting that *best_score* from *GridSearch* should not be used to differentiate between models at it is an optimistically biased estimator of future model performance.

Confusion Matrix



- The confusion matrix for all models is the same – all models performed equally well
- The most common error was a false-positive error, with no false-negative errors returned for the test set

Conclusions

- The most successful Payload Masses were typically between 2000kg and 4000kg with the “FT” Booster
- All Machine Learning models tested performed equally well
- The models typically are optimistically-biased (false-positive errors), so caution should be used to not oversell the chances of success to stakeholders
- Higher volume Launch Sites or Orbits often skewed the perceived success as they produced a higher number of successful launches, but may not have a high success ratio
- Recommend investigating the highest volume Launch Sites/Orbits to determine the factors that contributed towards a high Launch volume, but the highest Success Ratio Sites/Orbits to determine factors that contributed towards a high Success Ratio, and combine these factors to increase volume and Success.

Appendix

- GitHub repository

- Finding the Machine Learning method(s) that scored best:

```
methods = [logreg_cv, svm_cv, tree_cv, knn_cv]
results=[]
for method in methods:
results_dict={'Method':str(method.estimator), 'Score':str(method.score(X_test, Y_test))}
results.append(results_dict)
results_df=pd.DataFrame(results)
sns.catplot(x='Method',y='Score',hue='Score',data=results_df,ascending=False,kind = 'bar')
plt.xlabel("Method",fontsize=20)
plt.ylabel("Score",fontsize=20)
plt.show()

print('The method(s) that perform best are:\n',results_df.loc[results_df['Score'] ==
results_df['Score'].max()]['Method'].to_string(index=False))
```

Thank you!