

Winning Space Race with Data Science

Eduardo Orenes
06JAN23



Outline



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

Executive Summary



- For the current report the following methodologies has been employed
 - Data Collection and Data Wrangling
 - Exploratory Data Analysis via SQL and Interactive Visual Analytics
 - Interactive Graphs and Map Generation via Dash and Folium
 - Predictive Analysis via Different Models Evaluation and Optimization
- Results Summary
 - Falcon 9 rocket first stage reusability reduces launch costs
 - Recovery probability can be predicted via Data Science with available data
 - Using API and Web data, prediction accuracy score is 83.33%

Introduction



- Project background and context
 - SpaceX has gained worldwide attention for a series of historic milestones.
 - It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010.
 - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas 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.
- Problems you want to find answers
 - Determining the likelihood of Falcon 9's first stage recovery from previous attempts

Section 1

Methodology

Methodology



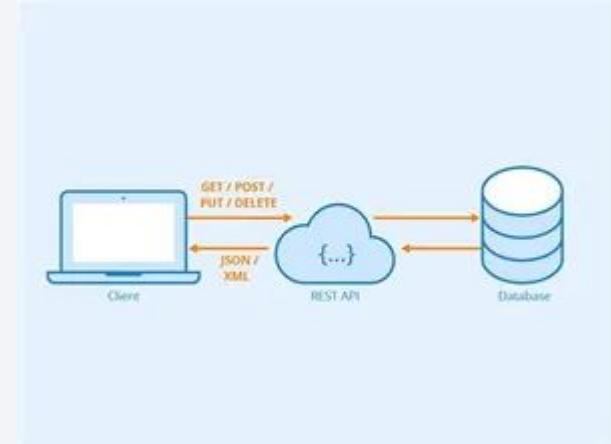
Executive Summary

- Data collection methodology:
 - SpaceX REST API
 - [Wiki Article](#) Web Scrapping
- Perform data wrangling
 - Ensure data types are appropriate to deal with them
 - NaN values identified.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - 4 models have been built and configuration parameters calculated for optimal accuracy

Data Collection



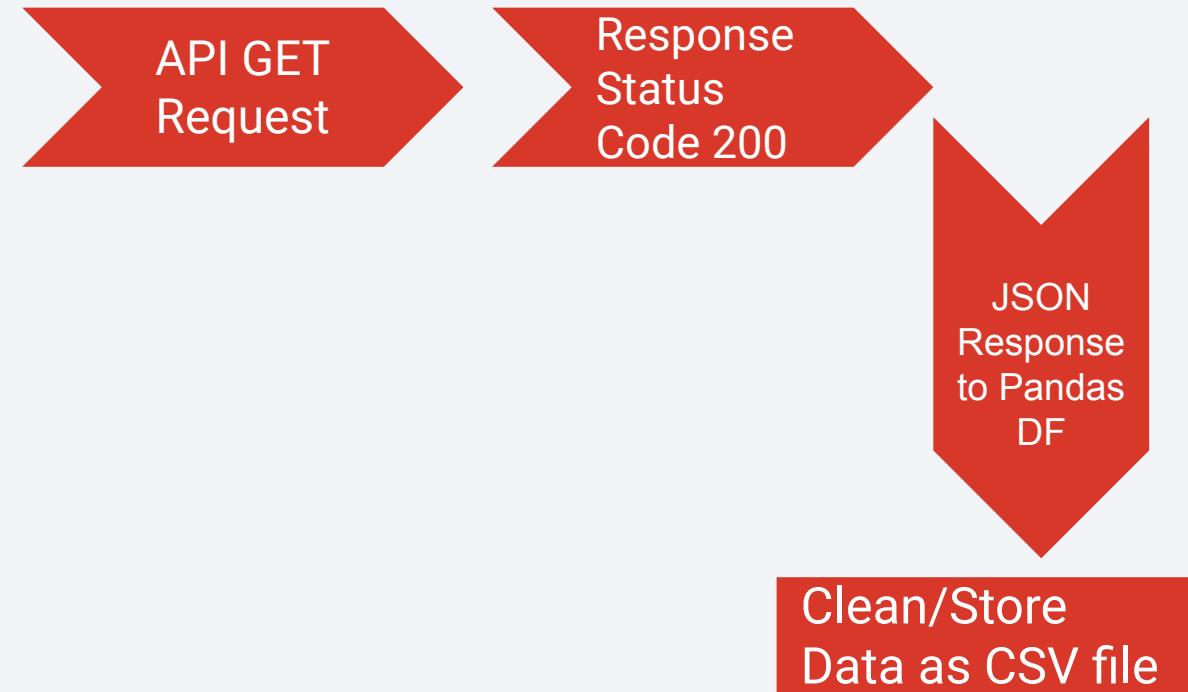
- Data sets were collected via:
 - REST API
 - Web Scrapping



Data Collection – SpaceX API



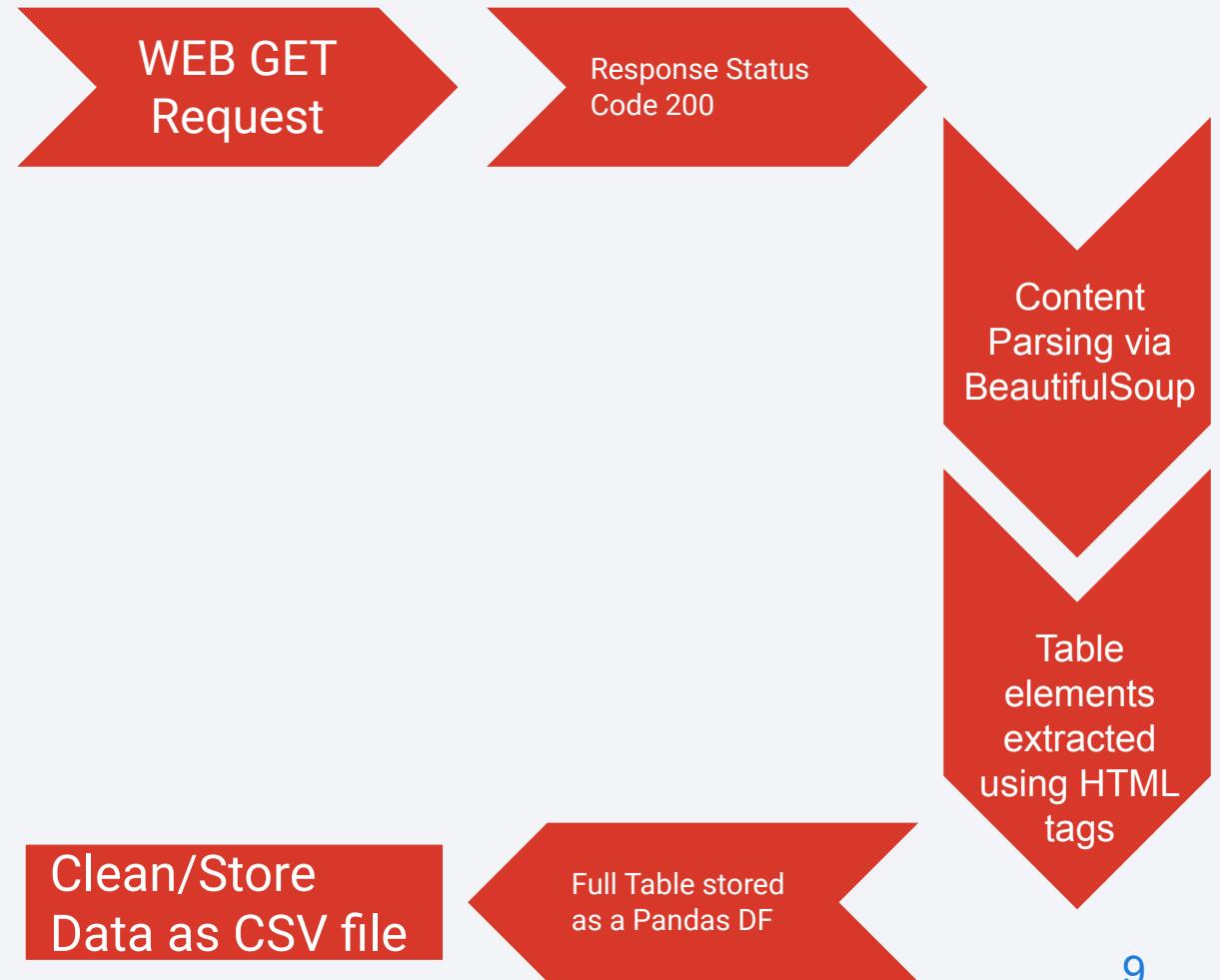
- [GitHub URL of the completed SpaceX API calls notebook](#)



Data Collection - Scraping



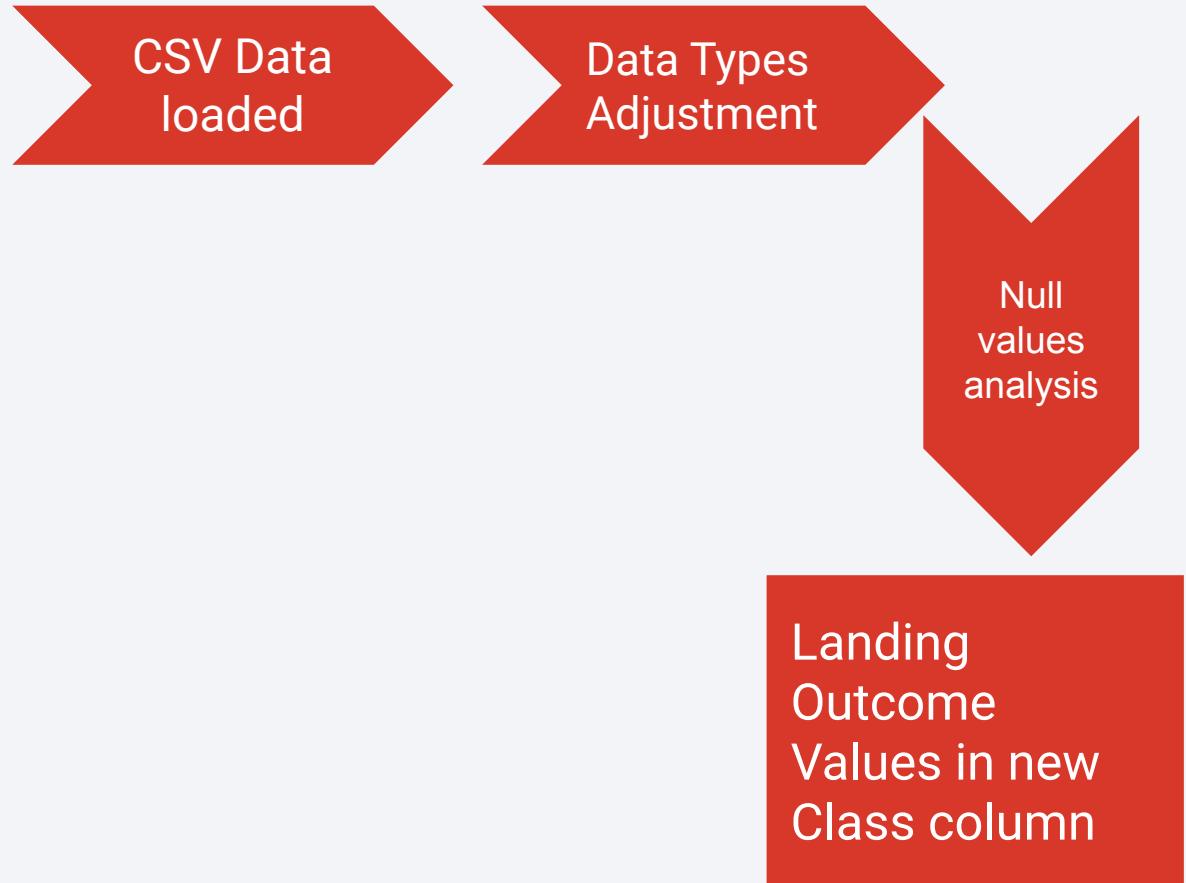
- [GitHub URL of the completed web scraping notebook](#)



Data Wrangling



- Data has been processed as follows
 - Identify data types and its conversion.
 - Calculate the percentage of the missing values in each attribute.
 - Determine the number of launches on each site.
 - Determine the number and occurrence of each orbit.
 - Determine the number of landing outcomes.
 - Create a set of outcomes where the second stage did not land successfully.
 - Creation of a classification variable that represents the outcome of each launch (1:Success, 0:Failed).



- [GitHub URL of your completed data wrangling related notebooks](#)

EDA with Data Visualization



- Summary of plotted charts
 - Relationship between Flight Number and Launch Site
 - Relationship between Payload and Launch Site
 - Relationship between success rate of each orbit type
 - Relationship between FlightNumber and Orbit type
 - Relationship between Payload and Orbit type
 - Launch success yearly trend
- [GitHub URL of your completed EDA with data visualization notebook](#)

EDA with SQL



- Display the names of the unique launch sites in the space mission
 - `SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL`
- Display 5 records where launch sites begin with the string 'CCA'
 - `SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5`
- Display the total payload mass carried by boosters launched by NASA (CRS)
 - `SELECT SUM(PAYLOAD_MASS_KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER == 'NASA (CRS)'`
- Display average payload mass carried by booster version F9 v1.1
 - `SELECT AVG(PAYLOAD_MASS_KG_) AS "Average Payload Mass" FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'`
- List the date when the first successful landing outcome in ground pad was achieved.
 - `SELECT substr(MIN(Timestamp), 1, 10) AS Date FROM SPACEXTBL WHERE "Landing _Outcome" == 'Success (ground pad)'`
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
 - `SELECT Booster_Version FROM SPACEXTBL WHERE "Landing _Outcome" == 'Success (drone ship)' AND PAYLOAD_MASS_KG_ BETWEEN 4000 AND 6000`
- List the total number of successful and failure mission outcomes
 - `SELECT (SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%') AS SUCCESS, (SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%') AS FAILURE`
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
 - `SELECT DISTINCT(Booster_Version) FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ == (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)`
- 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.
 - `SELECT SUBSTR(Date, 6, 2) AS Month, "Landing _Outcome", Booster_version, launch_site FROM SPACEXTBL WHERE SUBSTR(Date, 3, 2) == '15'`
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
 - `SELECT DISTINCT("Landing _Outcome"), COUNT(*) as "Count" FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND "Landing _Outcome" LIKE "%Success%" GROUP BY "Landing _Outcome" ORDER BY "Count" DESC`
- [GitHub URL of your completed EDA with SQL notebook](#)

Build an Interactive Map with Folium



- Different markers and circles added to highlight NASA location, launch locations. Lines have also been added to indicate distances to different relevant entities like railways, highways, cities, coastline.
- All those indicators help to visually identify our location of interest and some associated metrics like distance or success/fail launch.
- [GitHub URL of your completed interactive map with Folium map](#)
 - Dear Peer Reviewer, please notice that GitHub does not show the maps. You will need to download and execute the Notebook locally, installing and troubleshooting any python library requirement if you want to check all generated maps.

Build a Dashboard with Plotly Dash

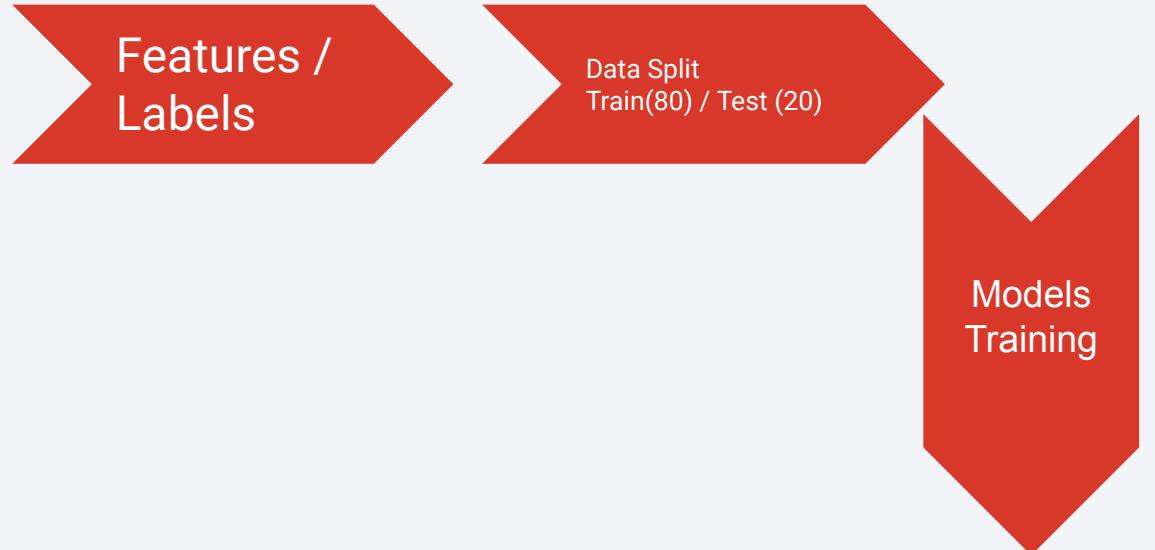


- User input dropdown menu for launch site
 - It allows to easily select the launch site and see the data
- Pie chart showing success rate at the user chosen site
 - It allows to visualize success rate very rapidly
- User input payload mass range slider
 - It allows to play with a payload mass range to see results per location
- [GitHub URL of your completed Plotly Dash lab](#)

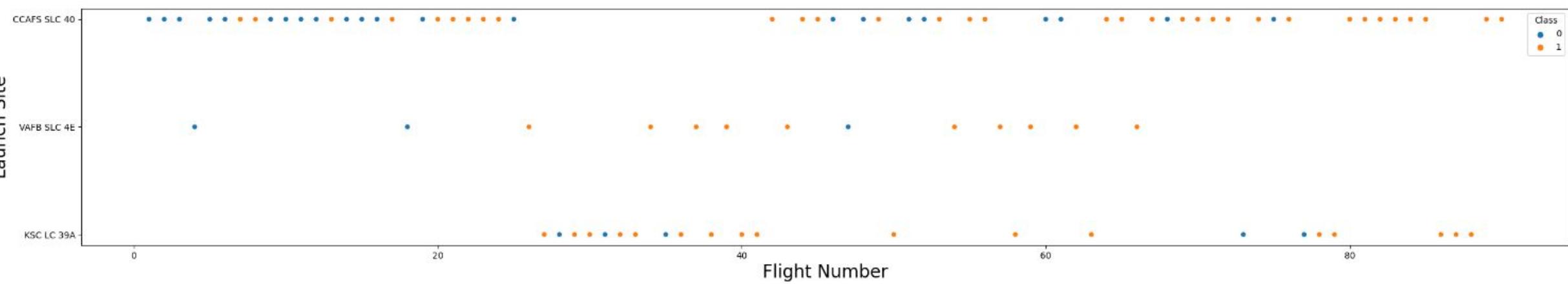
Predictive Analysis (Classification)



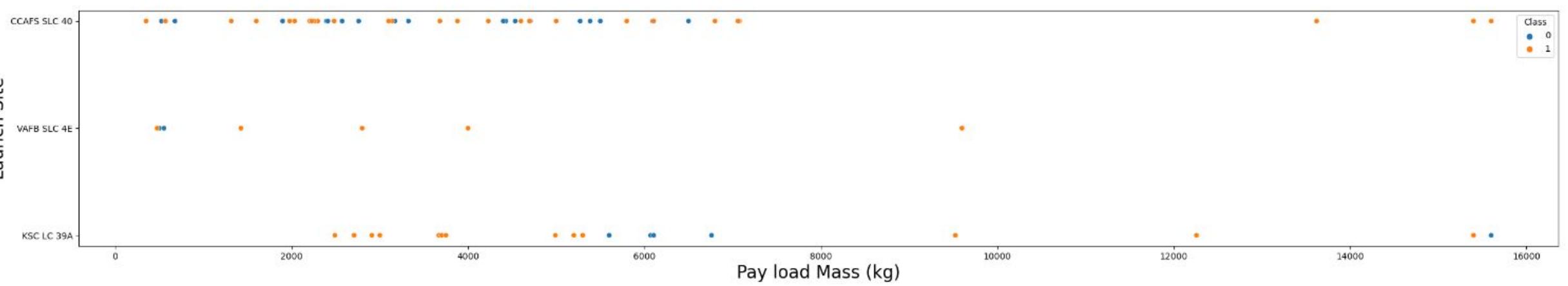
- Data separated into features and labels, setting the test sample to 20%
 - Following scikit-learn models have been instantiated and trained
 - Logistic Regression
 - Support Vector Machines
 - Decision Tree
 - KNN
 - GridSearchCV has been used to perform an exhaustive search over specified parameters, in order to find out the best configuration (higher accuracy score)
 - Score calculation
- Dear Peer Reviewer, please notice that the parameter `random_state=0` has been set to avoid variations in results after repeated tests. That makes all the models to have the same score. Without it, the Tree model would have an inferior score, most of the tested times.
- [GitHub URL of your completed predictive analysis lab](#)



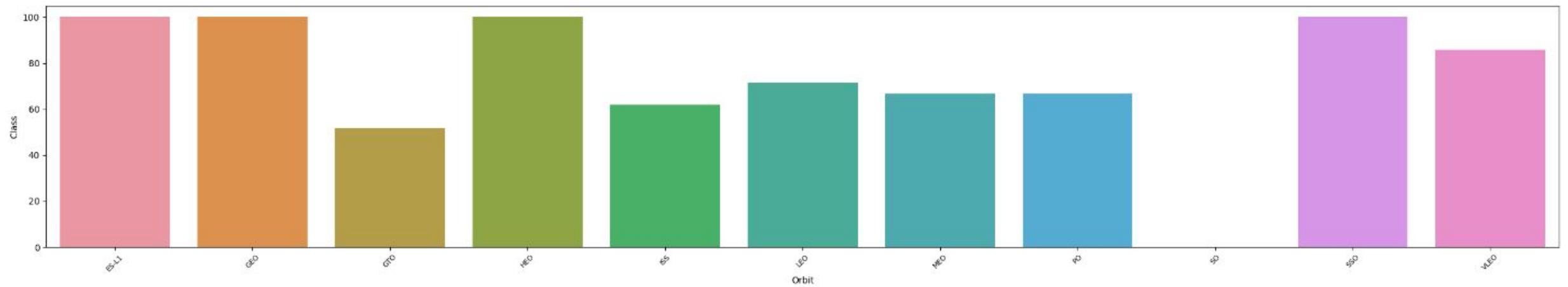
Results - EDA



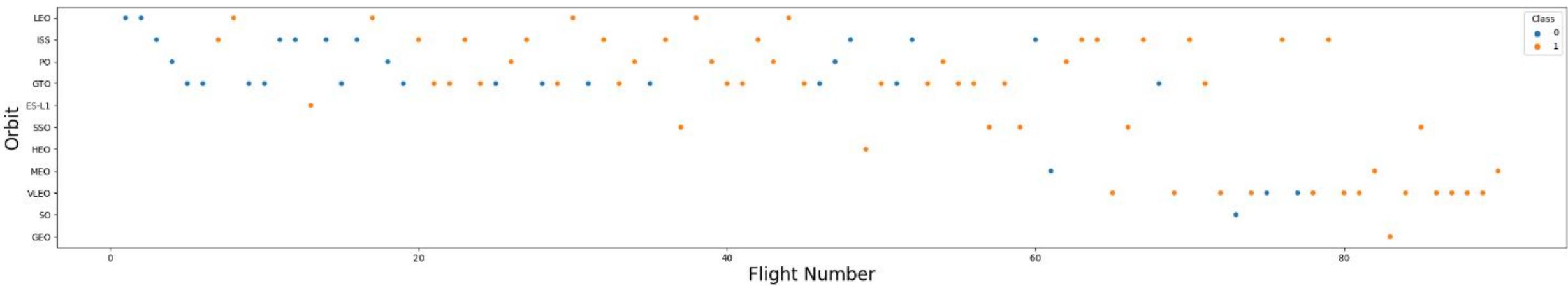
Results - EDA



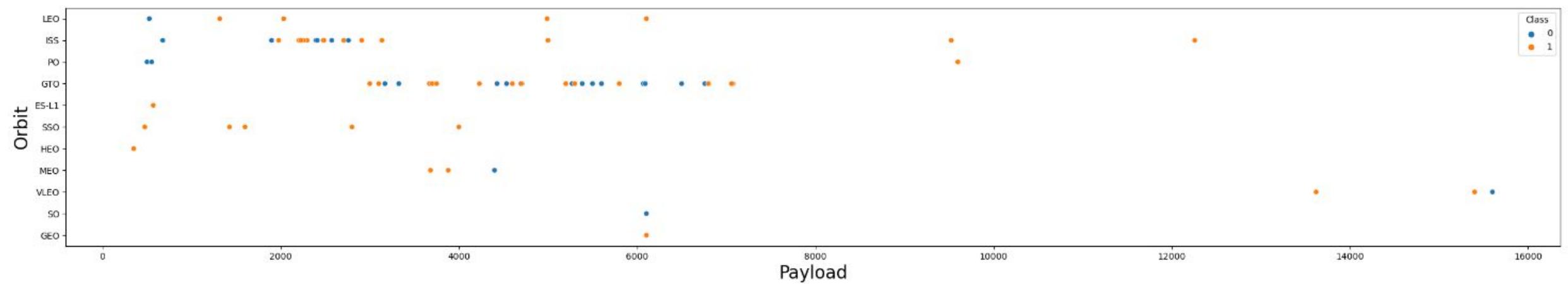
Results - EDA



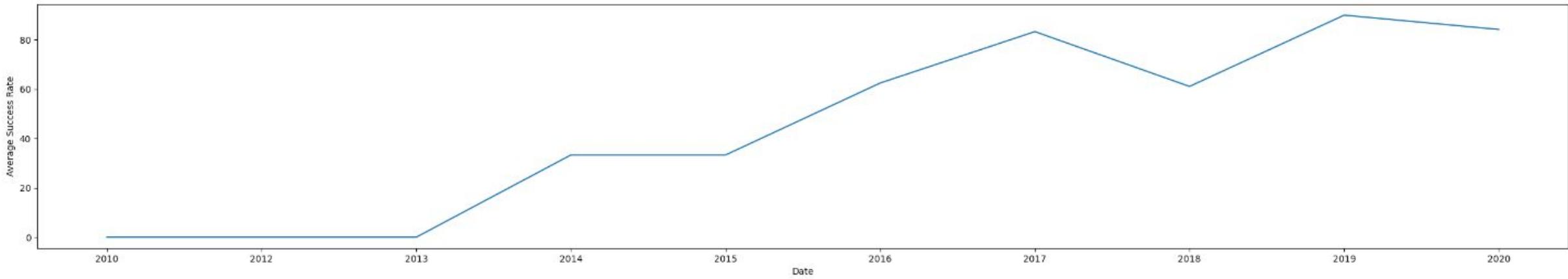
Results - EDA



Results - EDA



Results - EDA



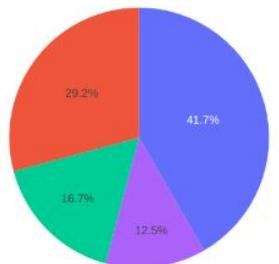
Results - Dashboards



SpaceX Launch Records Dashboard

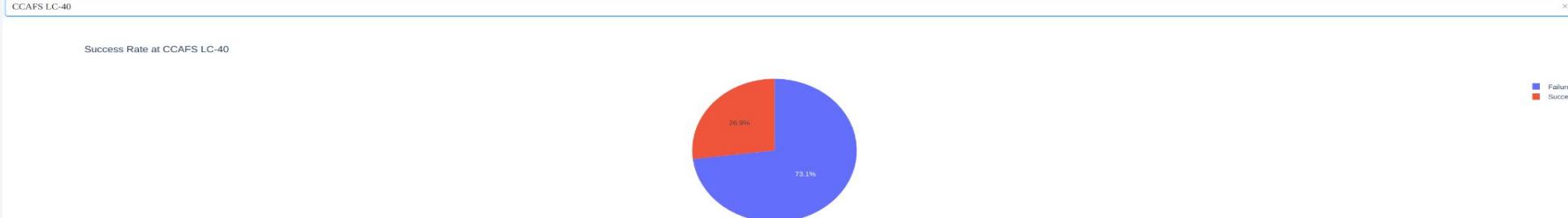
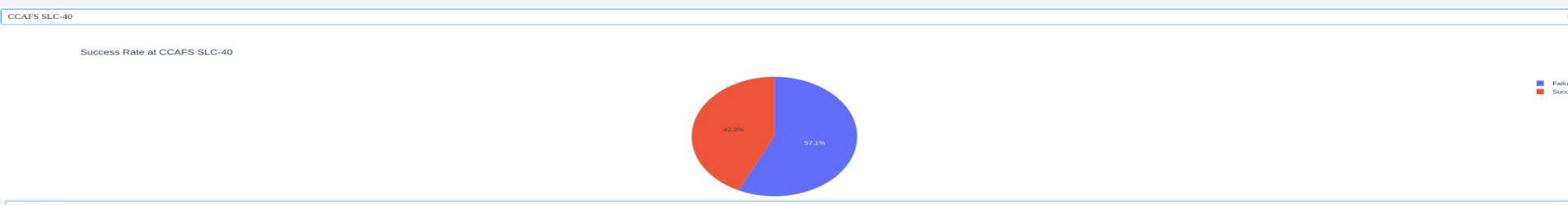
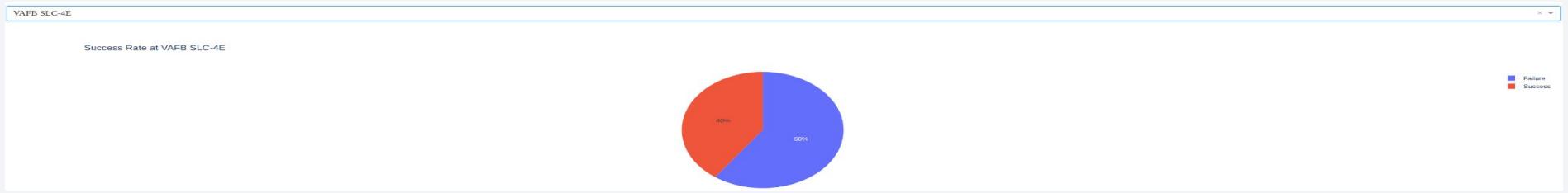
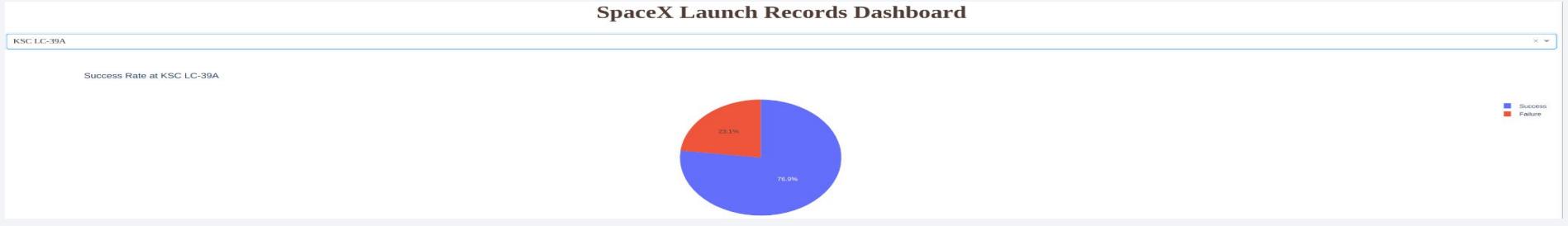
All Sites

Success Rate at all sites



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

Results - Dashboards



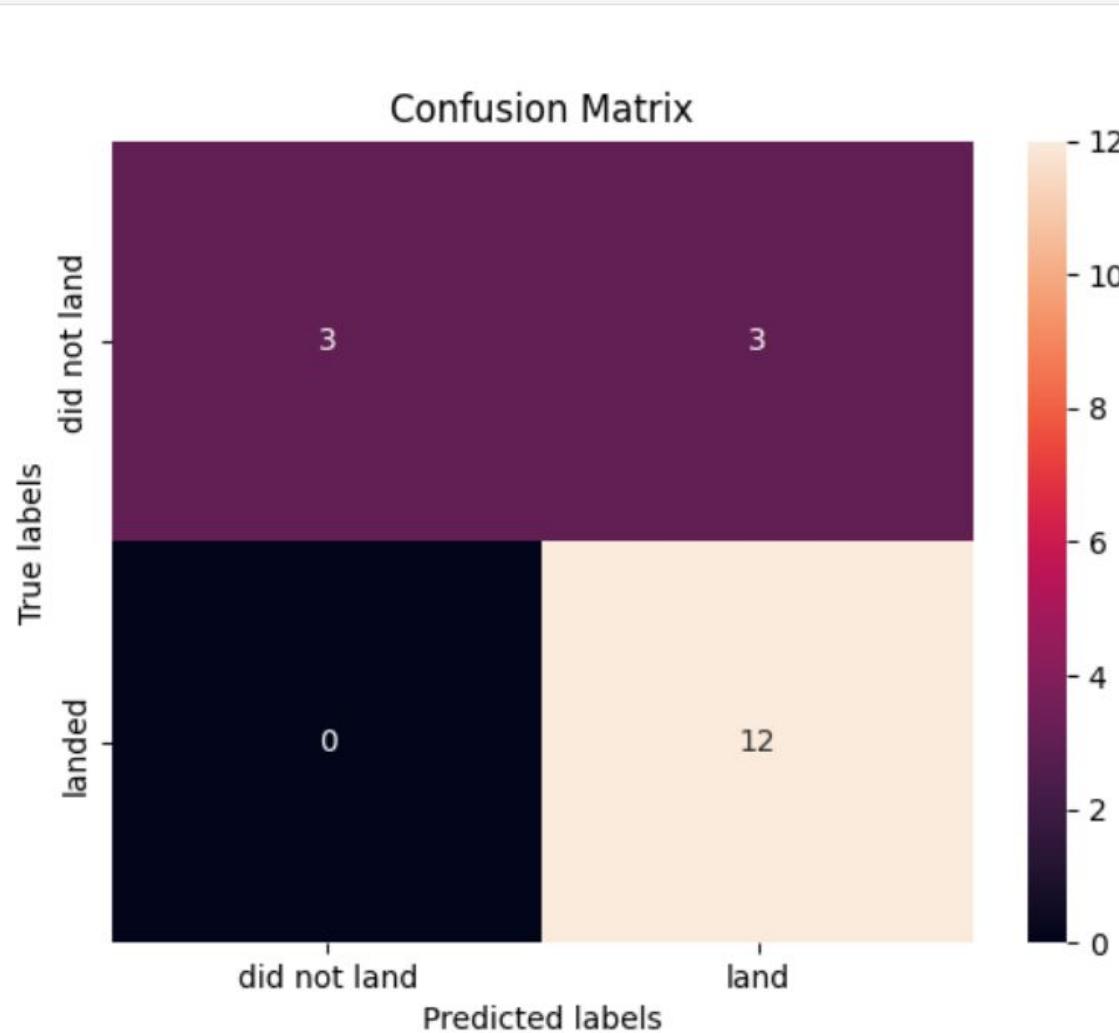
Results - Dashboards



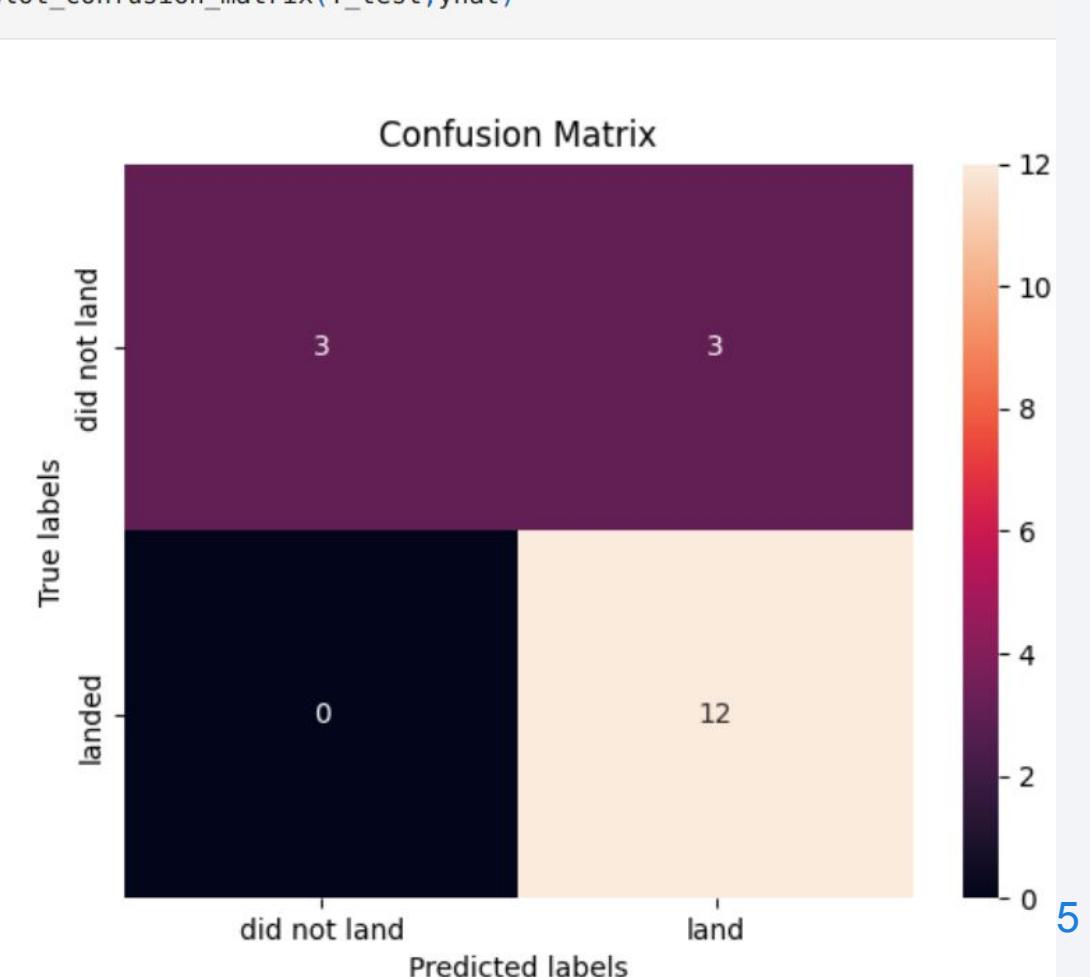
Results - Predictive Analysis



```
yhat=logreg_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



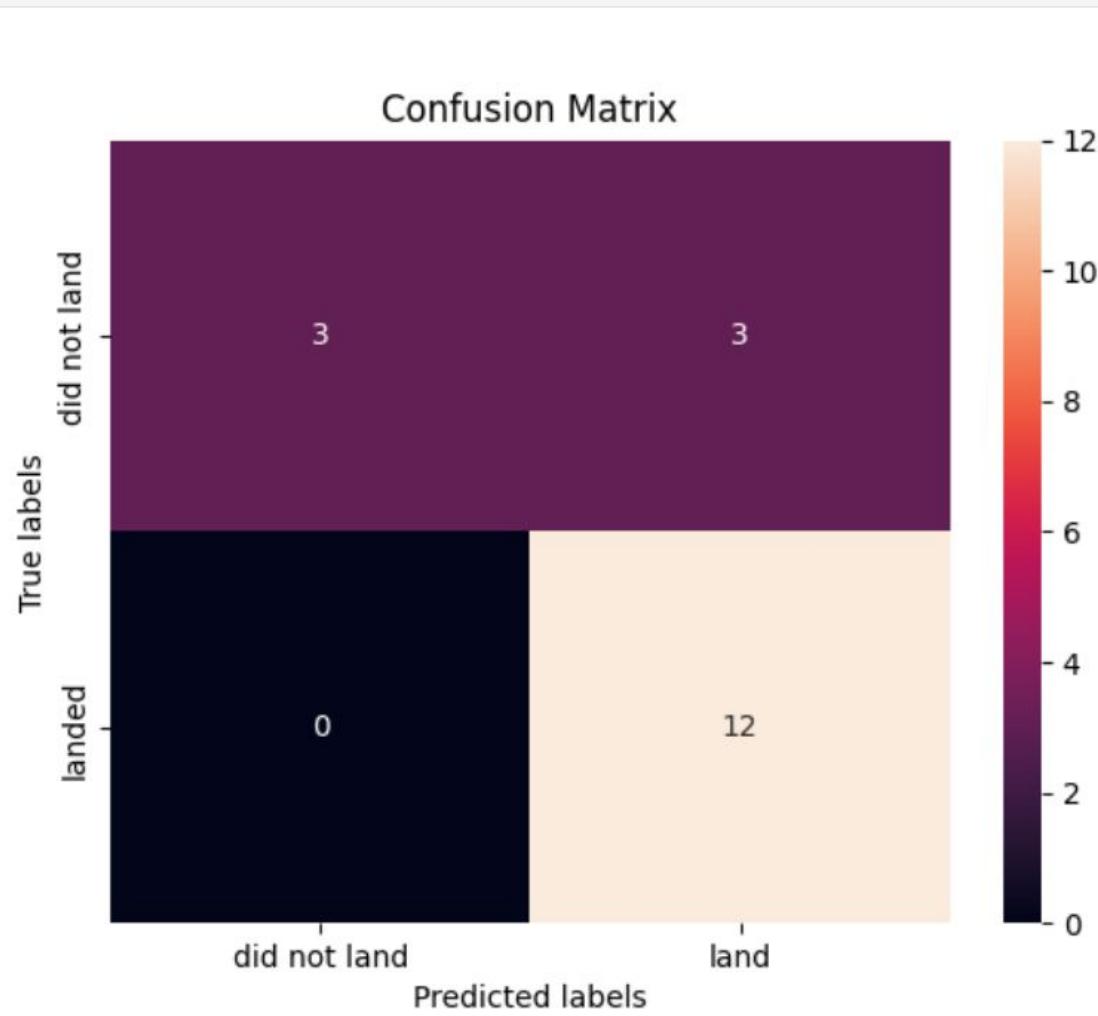
```
yhat=svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



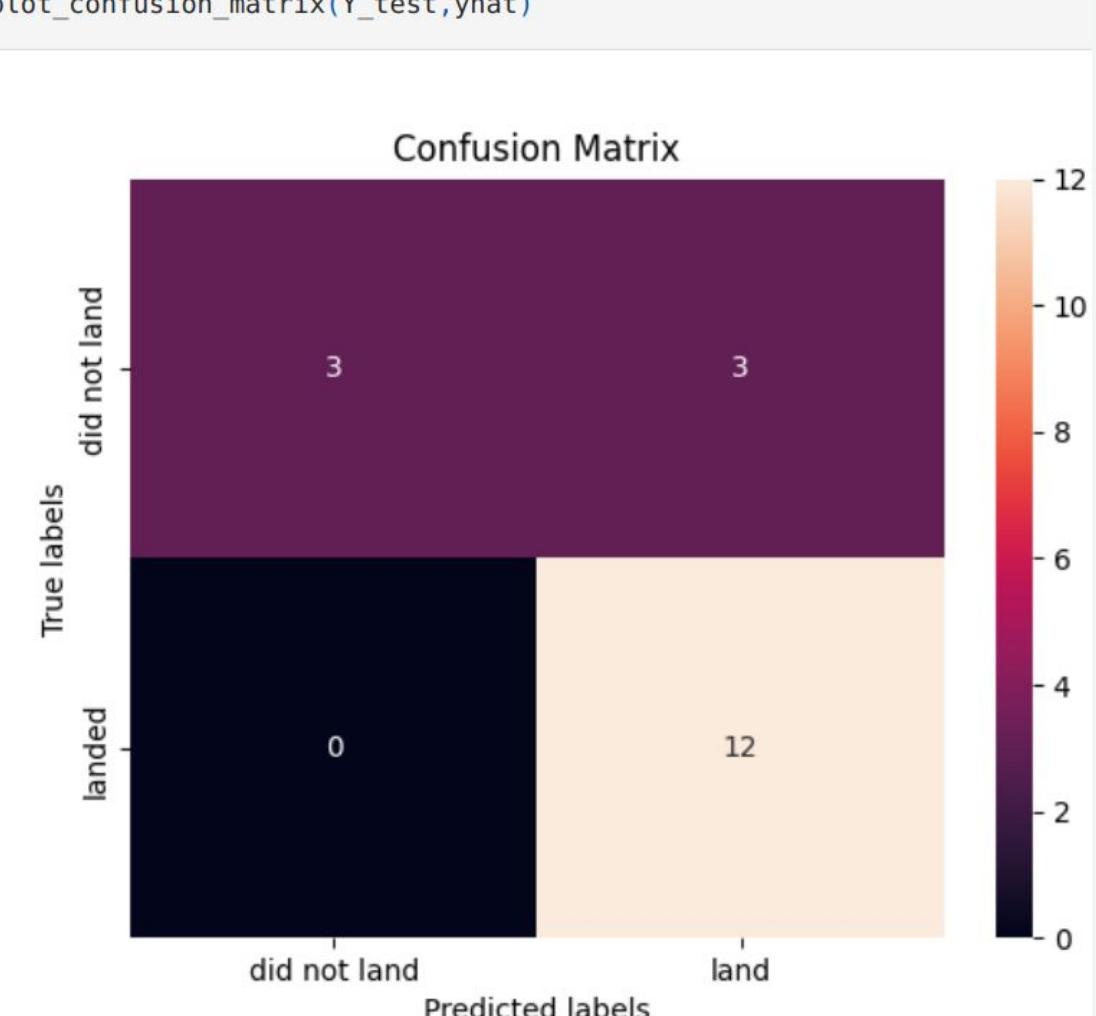
Results - Predictive Analysis



```
yhat = svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



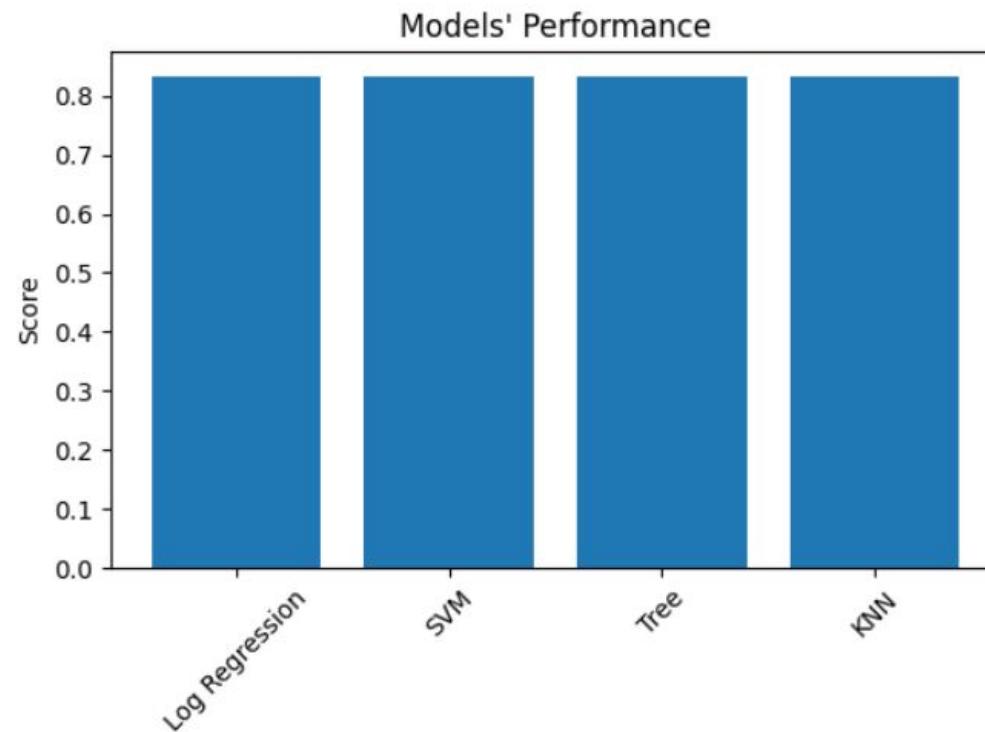
```
yhat = knn_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



Results - Predictive Analysis (random_state=0)



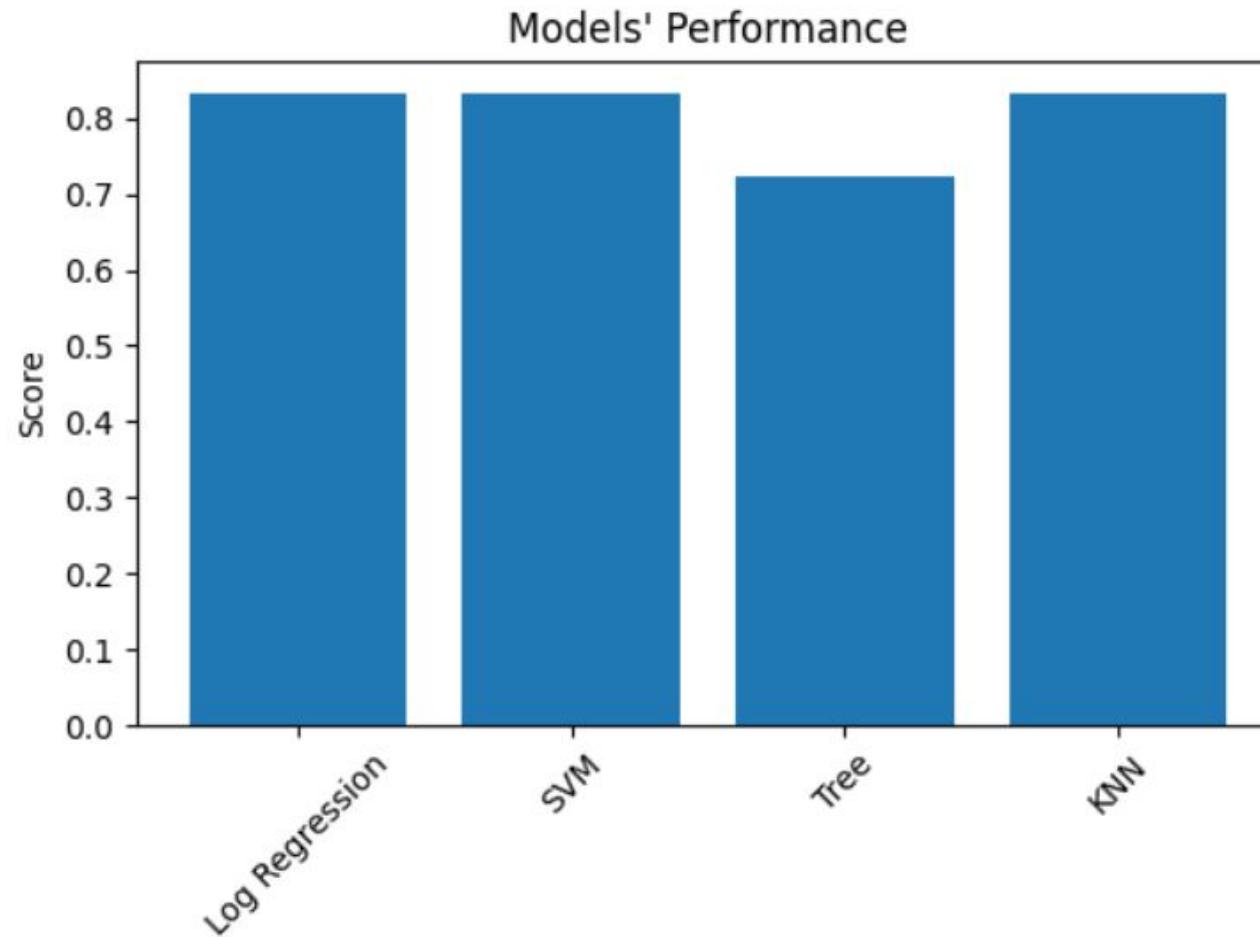
All methods performs equally

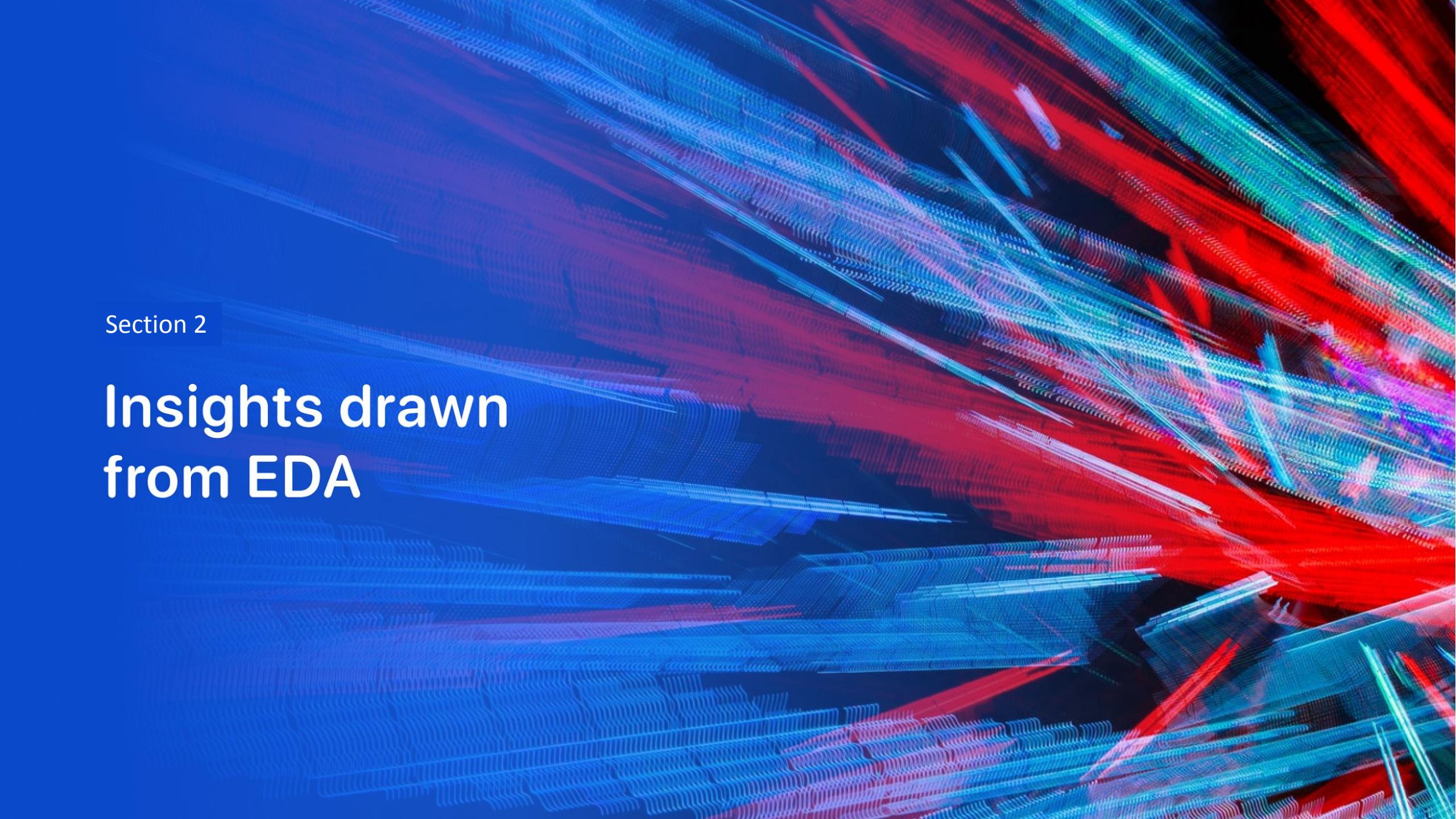


Results - Predictive Analysis (random_state=1)



Method Performs Best: Log Regression

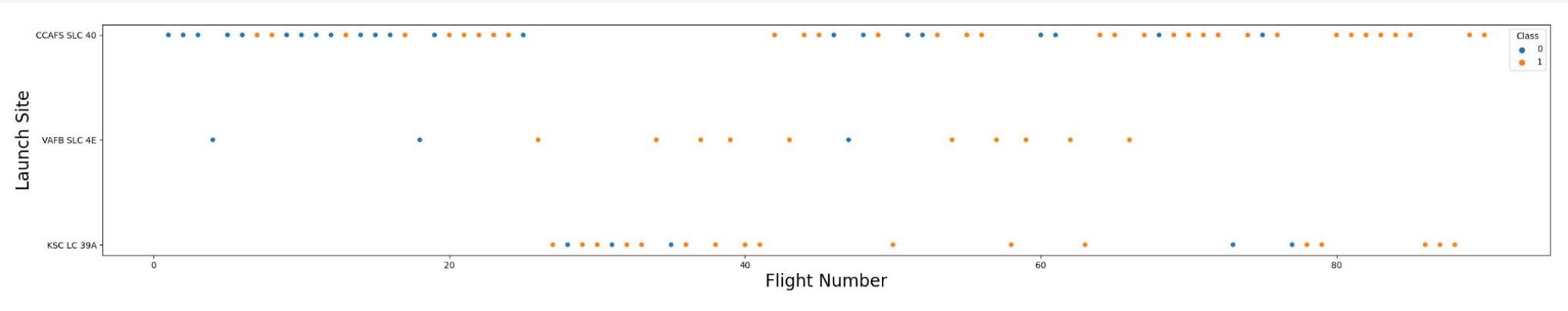


The background of the slide features a complex, abstract pattern of glowing lines. These lines are primarily blue and red, creating a sense of depth and motion. They appear to be composed of numerous small, glowing particles or dots, giving them a textured, almost liquid-like appearance. The lines converge and diverge, forming various shapes and directions across the dark, solid-colored background.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



- CCAFS SLC 40 presents the highest number, being evident a predominance of 1 class (success) in the last third.
 - CCAFS SLC 40 is the most used
- VAFB SLC 4E presents a very low frequency, being interrupted definitively after flight number 66

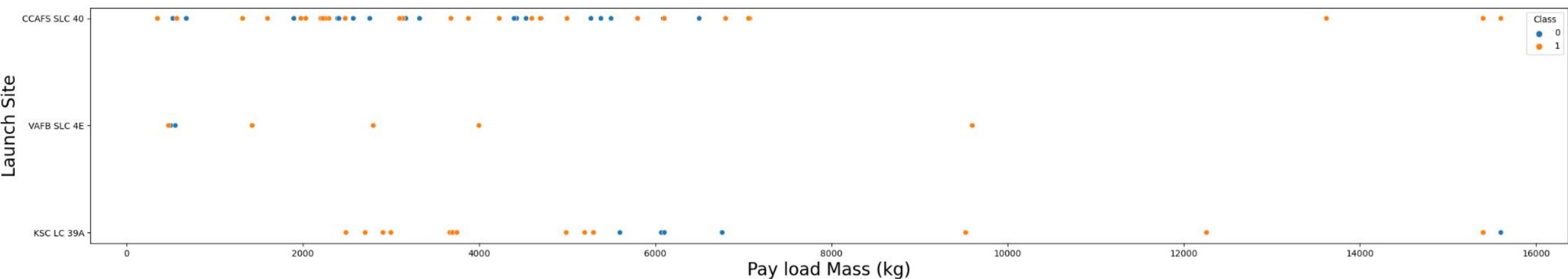
```
In [15]: df[df['LaunchSite'] == 'VAFB SLC 4E'].sort_values(by='FlightNumber', ascending=False)
```

Out[15]:

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	Ret
65	66 2019-06-12	Falcon 9	1425.0	SSO	VAFB SLC 4E	True RTLS	2	True	True	True	5e9e3032383ecb554034e7c9	5.0	

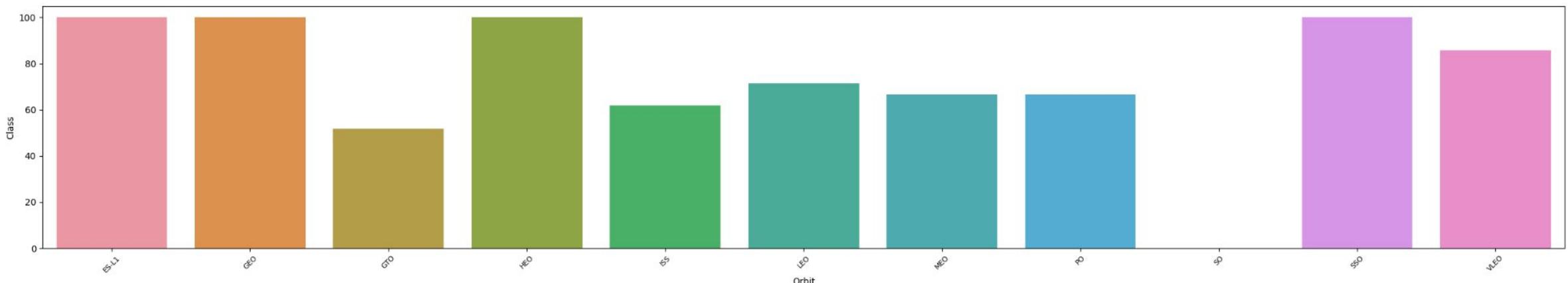
- VAFB SLC 4E is the least employed

Payload vs. Launch Site



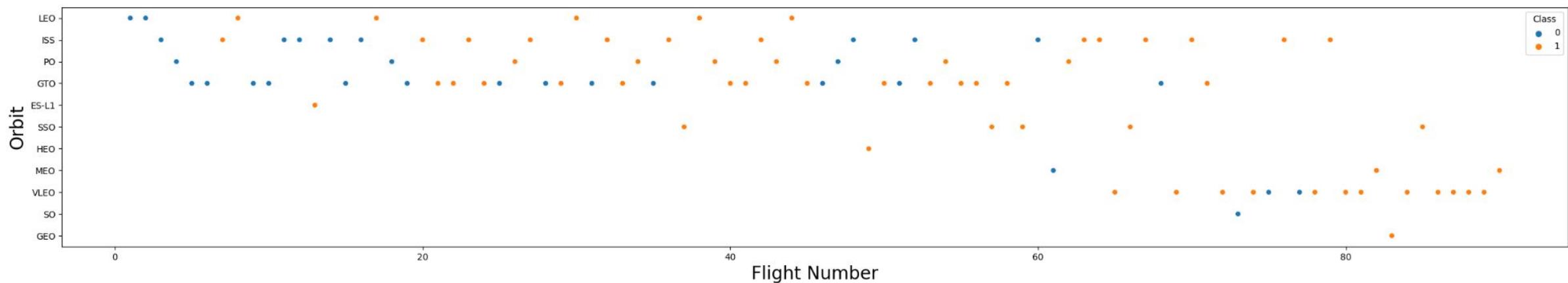
- Payload Mass between 8000 and 16000 Kg are much less frequent.
- VAFB SLC 4E presents a smaller number of payloads, being the maximum less than 10.000 Kg

Success Rate vs. Orbit Type



- Class = Success
- SO orbit presents the slower success rate
- GEO, ES L1, HEO and SSO present the higher success rate

Flight Number vs. Orbit Type



- LEO was interrupted after flight number 44

```
df[df['Orbit'] == 'LEO'].sort_values(by='FlightNumber', ascending=False).head(1)
```

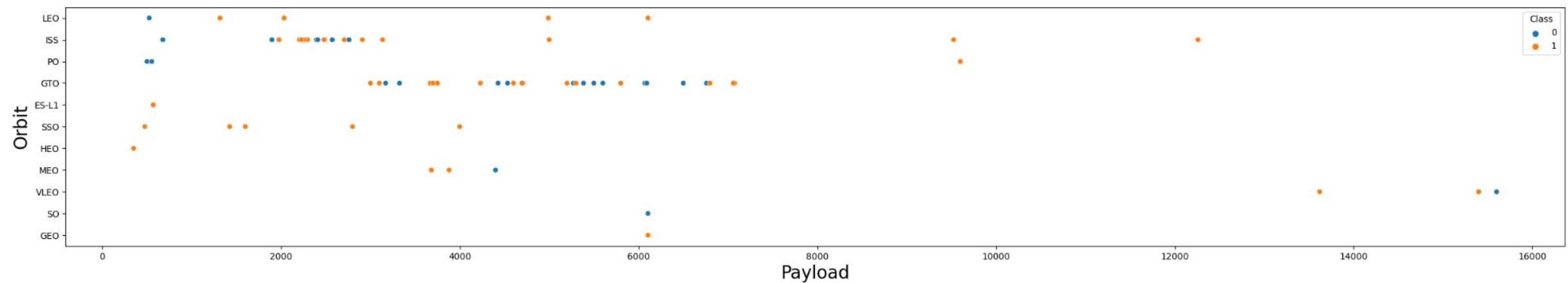
FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused
43	44 2018-01-08	Falcon 9	6104.959412	LEO	CCAFS SLC 40	True RTLS	1	True	False

- ISS and GTO are more used
- VLEO starts being used from 65 on

```
df[df['Orbit'] == 'VLEO'].sort_values(by='FlightNumber', ascending=True).head(1)
```

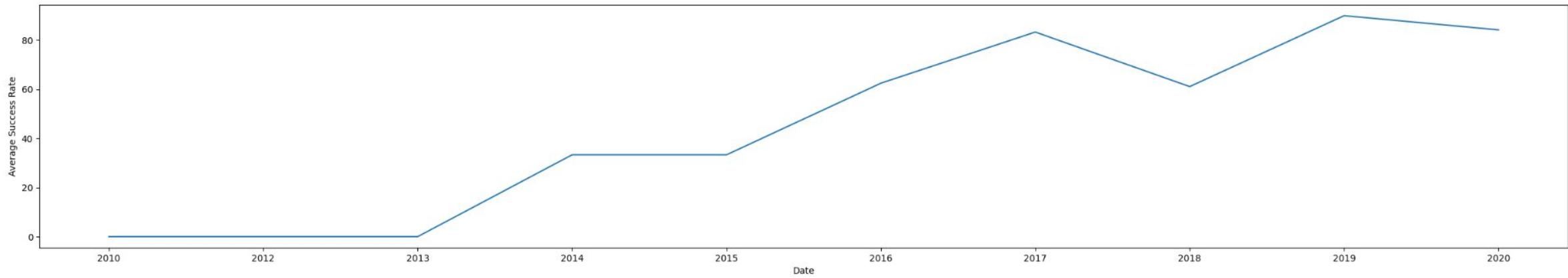
FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reu
64	65 2019-05-24	Falcon 9	13620.0	VLEO	CCAFS SLC 40	True ASDS	3	True	1

Payload vs. Orbit Type



- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

Launch Success Yearly Trend



All Launch Site Names



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

```
%%sql  
SELECT DISTINCT(LAUNCH_SITE)  
FROM SPACEXTBL
```

```
* 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'



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

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome	Timestamp	Timestamp_Format
2010-06-04 00:00:00	675000000000000	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	2010-06-04 18:45:00	04-06-2010 18:45
2010-12-08 00:00:00	565800000000000	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)	2010-12-08 15:43:00	08-12-2010 15:43
2012-05-22 00:00:00	278400000000000	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt	2012-05-22 07:44:00	22-05-2012 07:44
2012-10-08 00:00:00	210000000000000	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	2012-10-08 00:35:00	08-10-2012 00:35
2013-03-01 00:00:00	546000000000000	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	2013-03-01 15:10:00	01-03-2013 15:10

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 by NASA (CRS)"
FROM SPACEXTBL
WHERE CUSTOMER == 'NASA (CRS)'
```

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

Total Payload Mass by NASA (CRS)

45596

Average Payload Mass by F9 v1.1



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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass"
FROM SPACEXTBL
WHERE Booster Version LIKE 'F9 v1.1%'
-- # Booster Version Included for the average calculation:
-- # F9 v1.1 B1003, F9 v1.1, F9 v1.1 B1011, F9 v1.1 B1010,
-- # F9 v1.1 B1012, F9 v1.1 B1013, F9 v1.1 B1014,
-- # F9 v1.1 B1015, F9 v1.1 B1016, F9 v1.1 B1018, F9 v1.1 B1017
```

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

Average Payload Mass

2534.6666666666665

First Successful Ground Landing Date



List the date when the first successful landing outcome in ground pad was achieved.

Hint: Use min function

```
%%sql
SELECT substr(MIN(Timestamp), 1, 10) AS Date
FROM SPACEXTBL
WHERE "Landing _Outcome" == 'Success (ground pad)'
```

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

Date
2015-12-22

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

```
%%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.
```

```
Booster_Version
```

```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
F9 FT B1031.2
```

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
%%sql
SELECT
  (
    SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%'
  ) AS SUCCESS,
  (
    SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%'
  ) AS FAILURE

* sqlite:///my_data1.db
Done.

SUCCESS FAILURE
-----  
100      1
```

Boosters Carried Maximum Payload



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

```
%%sql
SELECT DISTINCT(Booster_Version)
FROM SPACEXTBL
WHERE PAYLOAD_MASS_KG_ == (
    SELECT MAX(PAYLOAD_MASS_KG_)
    FROM SPACEXTBL
)
```

```
* 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



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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql
SELECT SUBSTR(Date, 6, 2) AS Month, "Landing _Outcome", Booster_version, launch_site
FROM SPACEXTBL
WHERE SUBSTR(Date, 3, 2) == '15'
```

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

```
Done.
```

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

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



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql
SELECT DISTINCT("Landing _Outcome"), COUNT(*) as "Count"
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND "Landing _Outcome" LIKE '%Success%'
GROUP BY "Landing _Outcome" ORDER BY "Count" DESC
```

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

Landing _Outcome	Count
Success (drone ship)	5
Success (ground pad)	3

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper right, there are greenish-yellow bands of light, likely the Aurora Borealis or Australis. The overall atmosphere is dark and mysterious.

Section 3

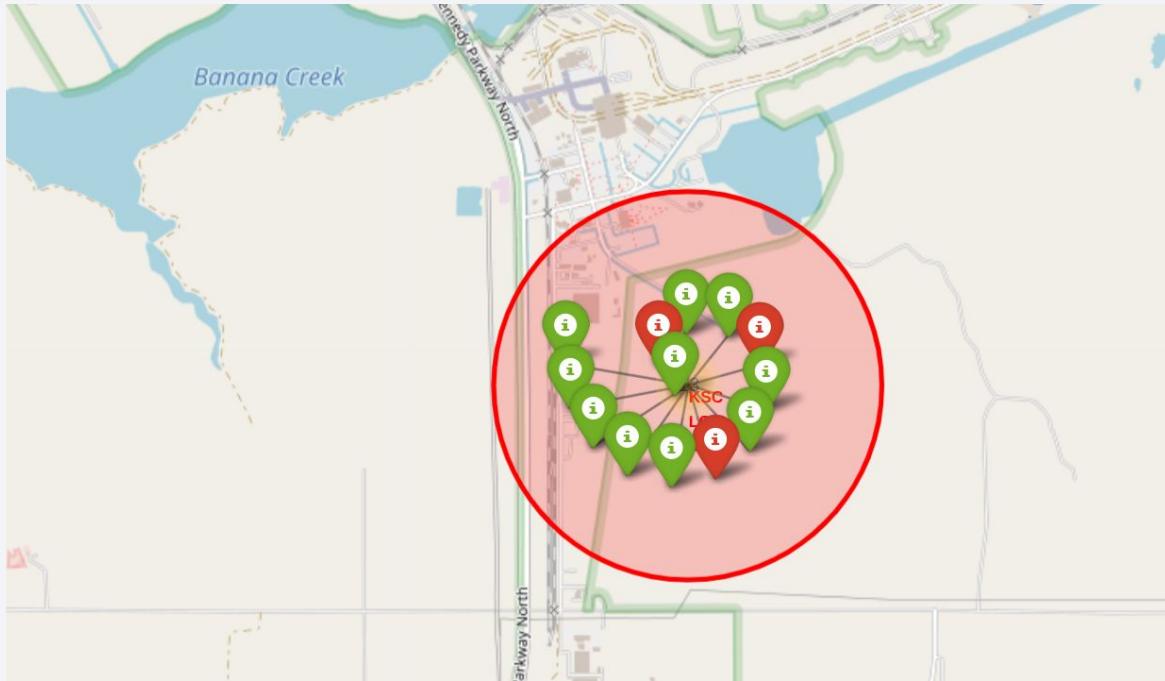
Launch Sites Proximities Analysis

All Launch Sites (Falcon 9)



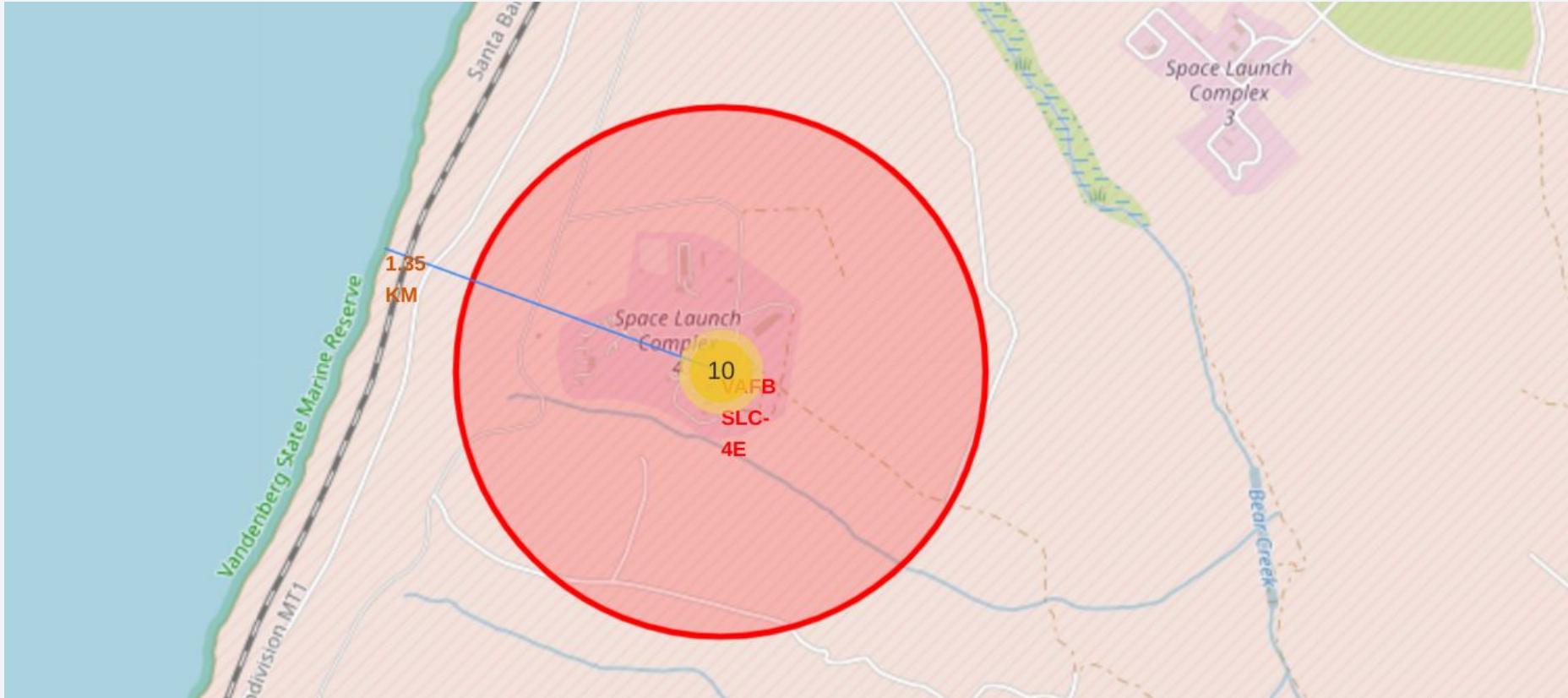
- Red circles to mark the place
- Labels also in red

Launch Outcomes for KSC LC 39A



- KSC LC 39A
 - Green informational icons indicate successful outcomes
 - Red information icons indicate failed outcomes

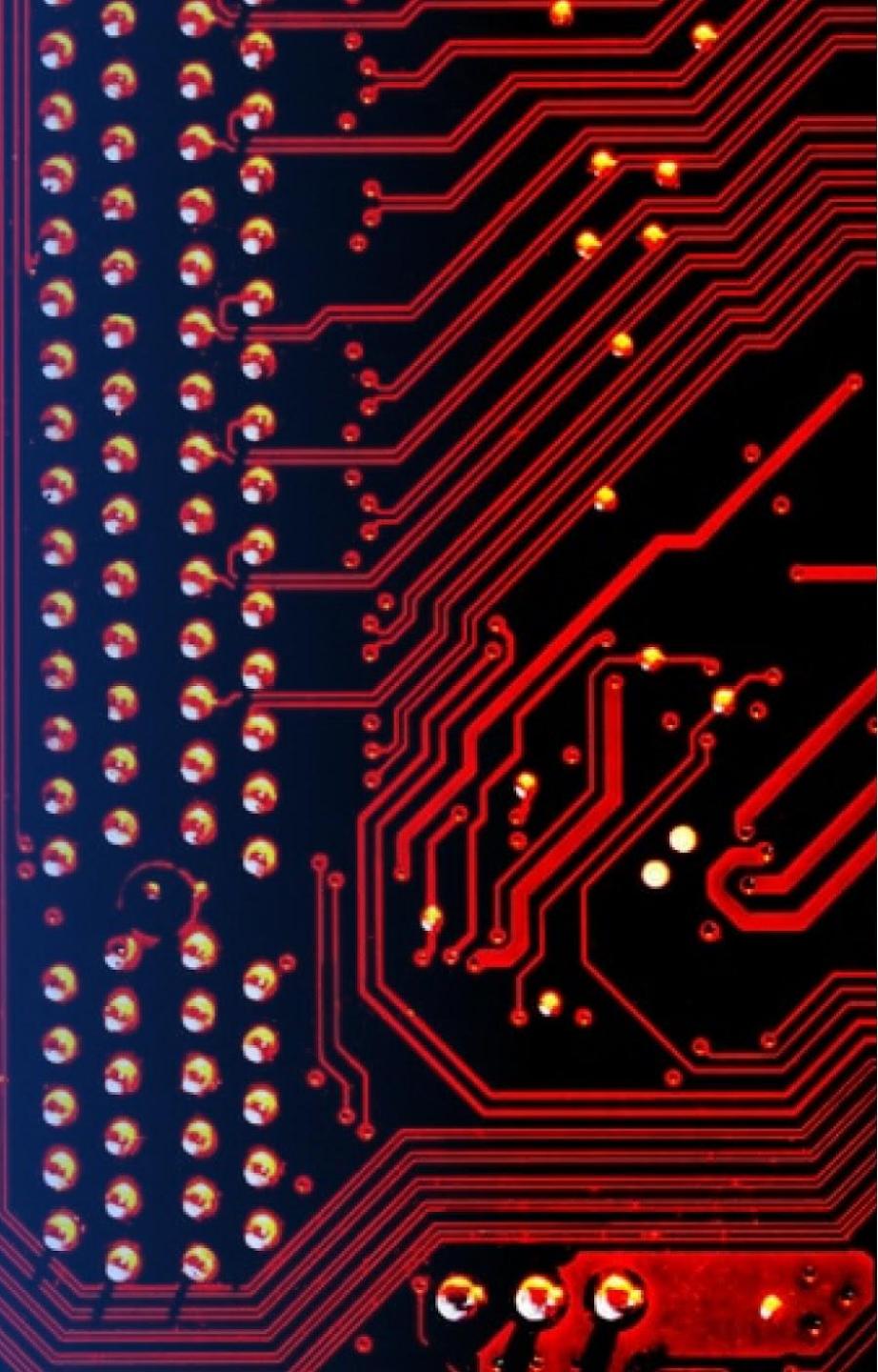
Distance from VAFB SLC 4E to Coastline



- Distance to coastline from VAFB SLC 4E is 1.35 Km

Section 4

Build a Dashboard with Plotly Dash



Success Rates at All Sites

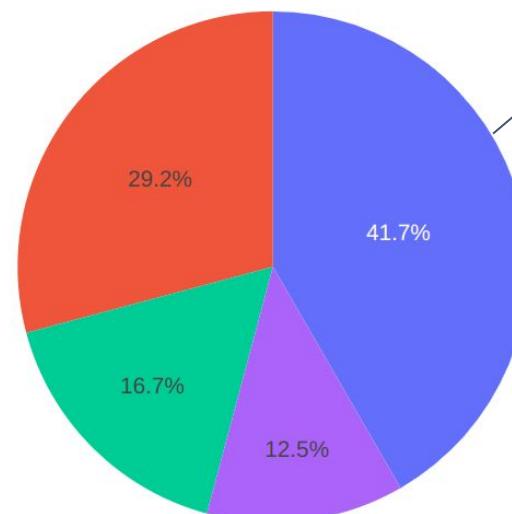


SpaceX Launch Records Dashboard

All Sites

X

Success Rate at all sites



Observe that 41.7% corresponds to

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

Higher Failure Rate is for CCAFS SLC-40

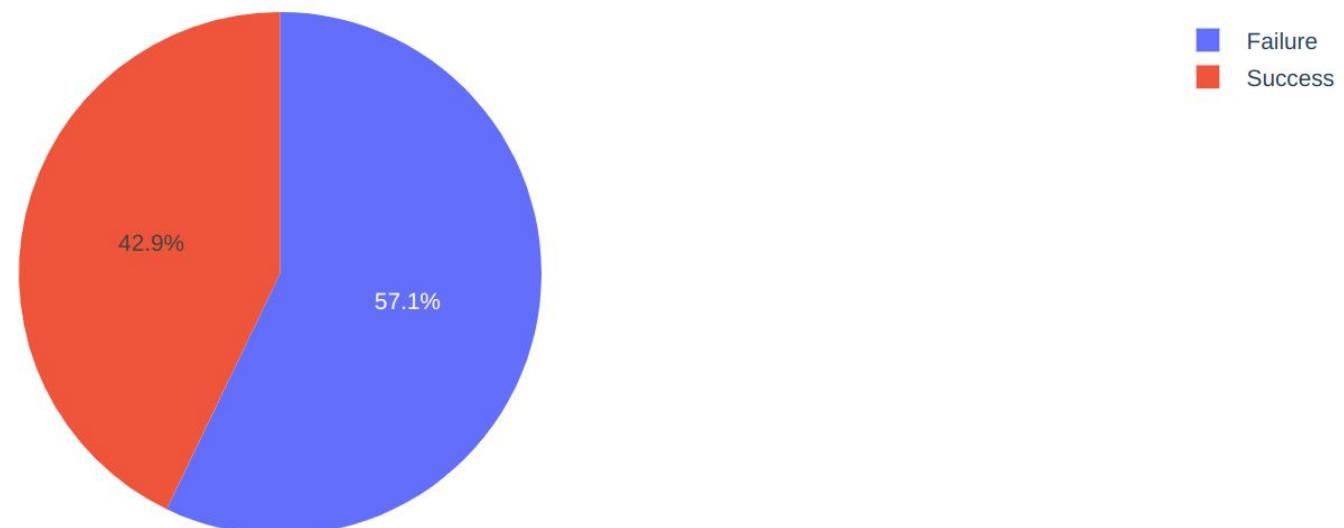


SpaceX Launch Records Dashboard

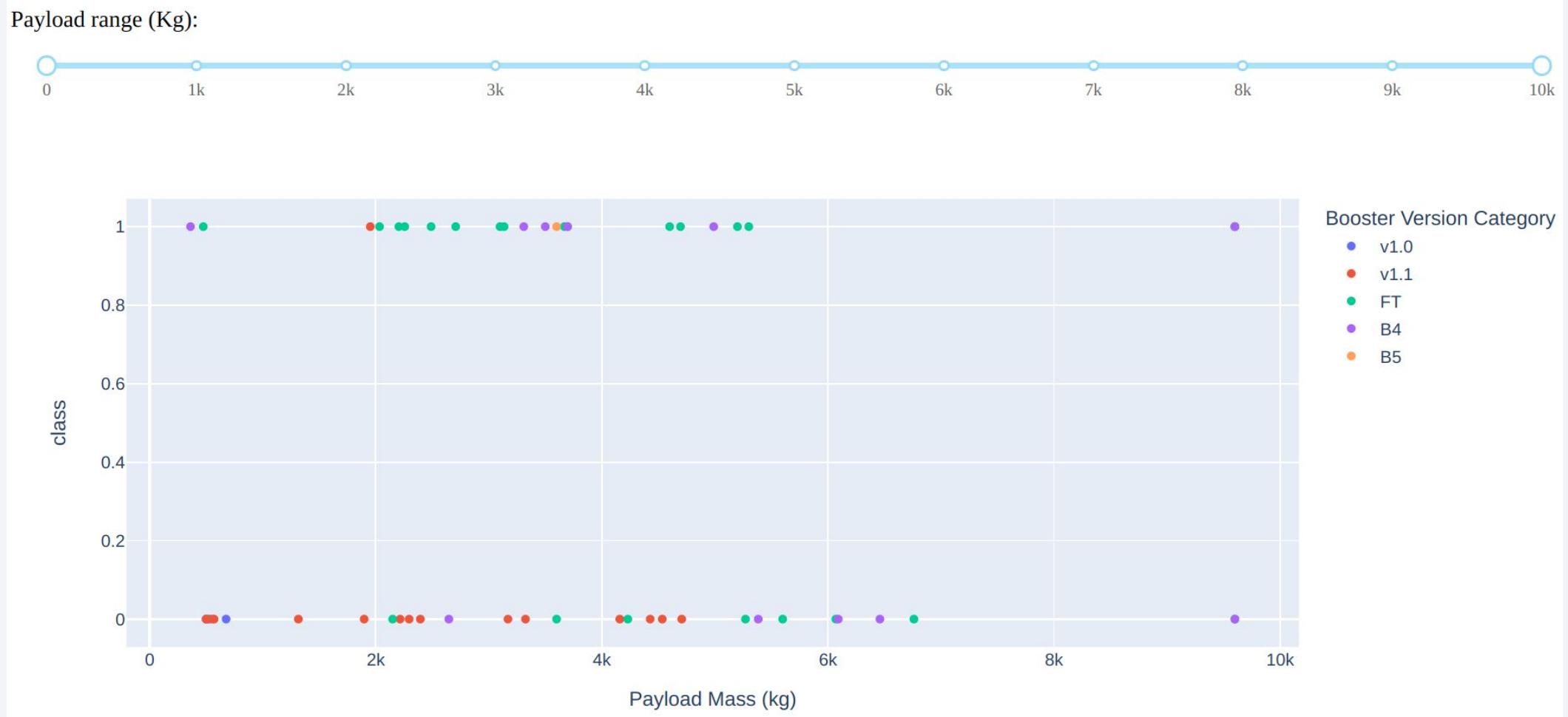
CCAFS SLC-40

x ▾

Success Rate at CCAFS SLC-40



Payload Mass for All Locations



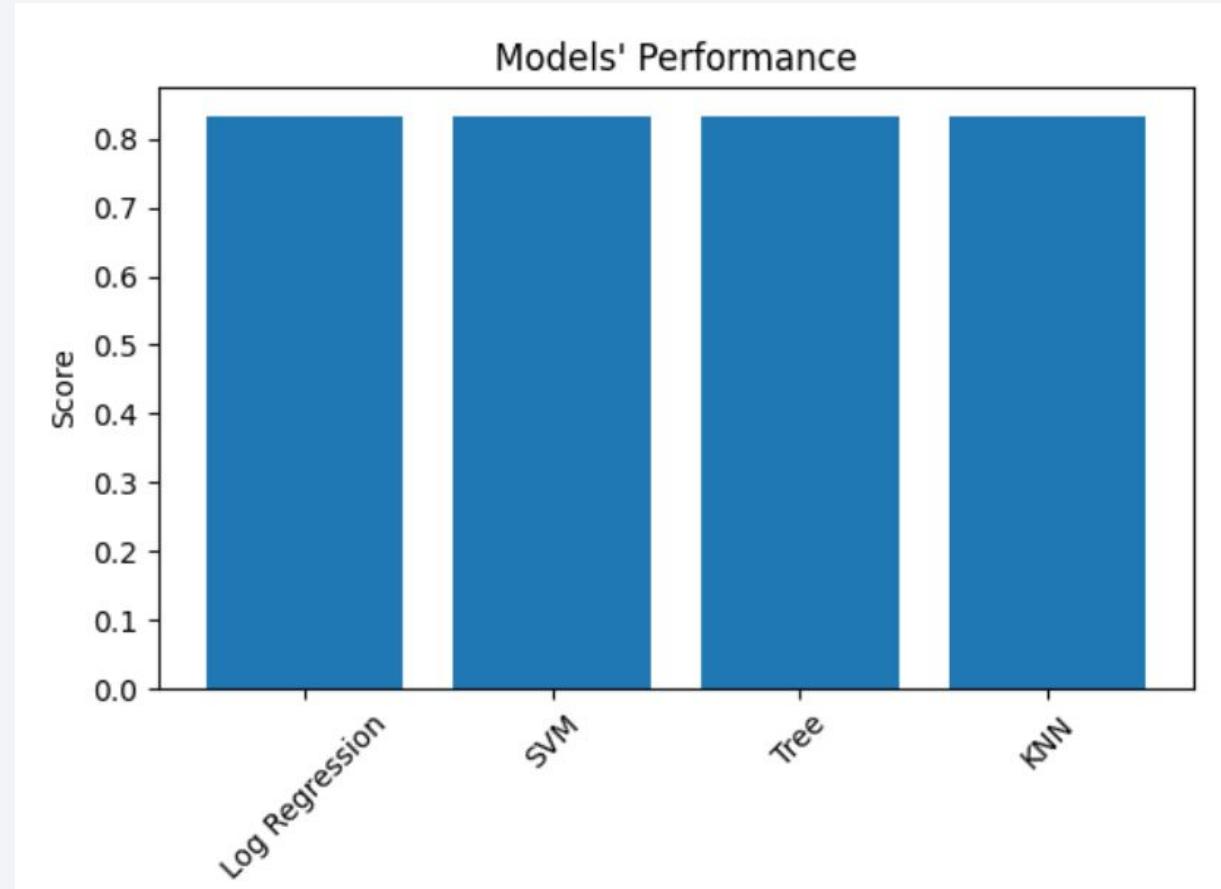
Section 5

Predictive Analysis (Classification)

Classification Accuracy



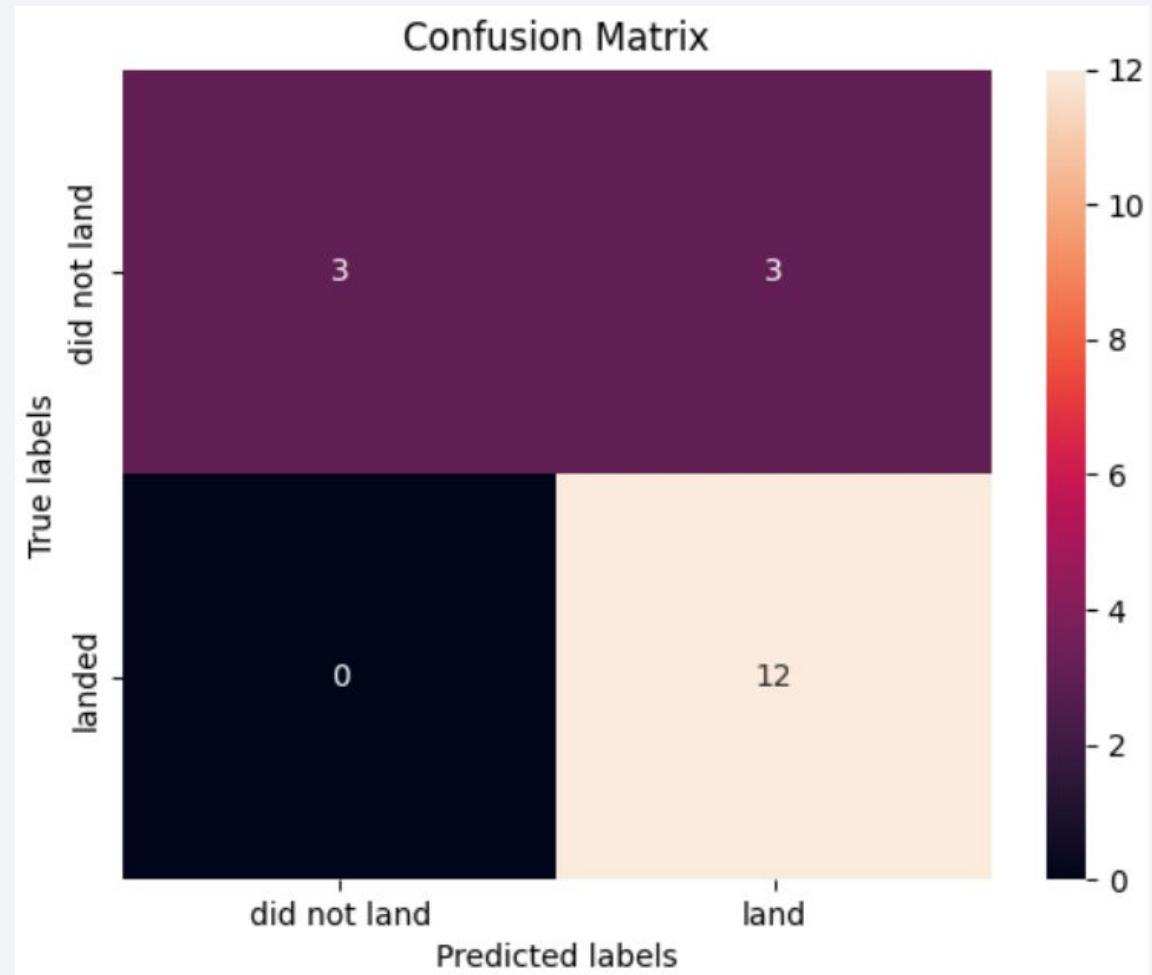
- According to the graph and the programmed function, all models present the same accuracy score for the `cv` models.
 - 0.8333333333333334
- For this to be true, we need to set the **`random_state = 0`** in the tree model in order to avoid different results with different executions of the Notebook's Code.
- Optimal CV configurations have been manually tested in the last cell of the Notebook for checking purposes with same results.
- When `random_state` is not set to 0, the tree model presents an inferior score in the CV models.



Confusion Matrix



- Logistic Regression
 - Observe the great difference depending on the case
 - 3/3 when not-landed
 - 0/12 when landed



Conclusions



- Launch Sites Location follows security and logistics principles
 - As far as possible from cities
 - As close as possible to highways, railways and coastline.
- There is enough data to predict with an acceptable accuracy score
 - At this point, highest probability of first stage recovery: KSC LC 39A
- Falcon 9 First Stage Landing Average Success Increases Over Time
- Future launches will increase accuracy score with retraining of the prediction model

Appendix



- Creativity to improve the presentation
 - Observe the blue check icon  to verify that all intended data has been included.
- Innovative Insights
 - All the models have been manually configured to check different accuracy scores which led to discover the importance to check the random parameter for the Tree Classification mechanism, mentioned during the lecture.

Thank you!

