Launch, Land, Learn: Data Insights from Reusable Rockets

2025 IBM Capstone

Skills Network

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- SpaceX has revolutionized the space industry with reusable booster rockets
- Reusable boosters drastically reduces cost per kg
- Cost advantage is derived from Falcon 9's successful recovery and landing
- Reliability issues remain compared to other launch vehicles like the Soyuz
- Falcon9 recovery and landing depend on features:
  - Payload Mass
  - Orbit
  - Launch Sites
  - Booster
- Our supervised learning model was able to predict a booster recovery outcome with an accuracy of 84%

INTRODUCTION

- SpaceX Falcon9 is a revolutionary two stage launch vehicle that has reshaped the landscape to space. This rocket costs upward of $62million while other more conventional rockets cost upward of $165million.
- This dramatic cost saving is from SpaceX's revolutionary approach to reusability.
- If SpaceX can reuse its first stage rocket and successfully land its first stage, we can estimate the cost of a launch. This analysis can be beneficial to contractors or competitors.
- In this capstone project, we will predict if Falcon9 first stage will land successfully using data advertised online.

METHODOLOGY

- Data collection
- Perform data wrangling
- Perform Exploratory Data Analysis (EDA) using visualization techniques and SQL
- Interactive visualizations using Folium and Plotly
- Perform predictive analysis using machine learning classification models

Data Collection – SpaceX API

- Data collected using SpaceX REST API by making a GET request
- Parsed the SpaceX launch data
- Decoded response content into JSON file
- Converted JSON file into Pandas Dataframe

ısk 1: Request and parse the SpaceX launch data using the GET reques

make the requested JSON results more consistent, we will use the following static resp

```
atic_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
```

e should see that the request was successfull with the 200 status response code

```
sponse=requests.get(static_json_url)
```

```
sponse.status_code
```

0

ow we decode the response content as a Json using `.json()` and turn it into a Pandas

```
Jse json_normalize meethod to convert the json result into a dataframe
ta = pd.json_normalize(response.json())
```

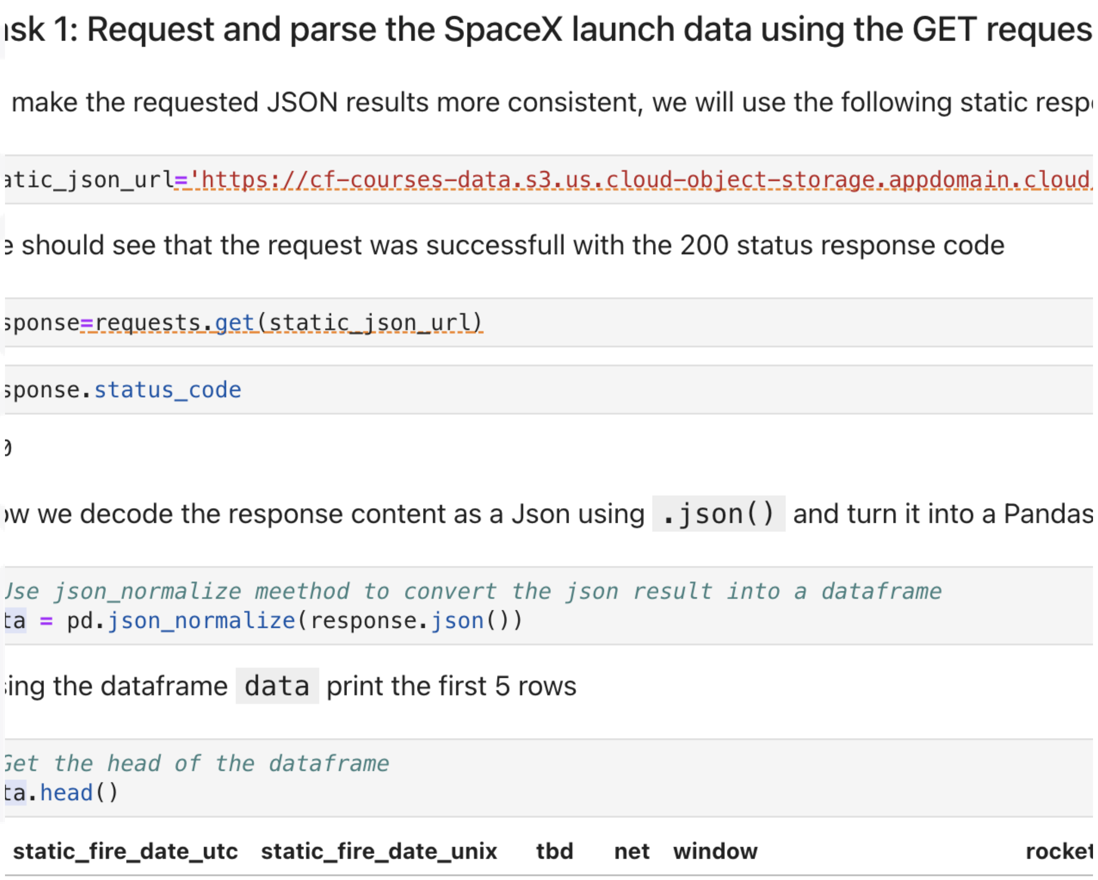
ing the dataframe `data` print the first 5 rows

```
Get the head of the dataframe
ta.head()
```

static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket

Data Collection - Scraping

- Performed Web scraping to collect Falcon 9 launch records from Wikipedia
- Utilized BeautifulSoup package to extract launch records from HTML table
- Created dataframe by parsing launch data



ısk 1: Request and parse the SpaceX launch data using the GET reques

make the requested JSON results more consistent, we will use the following static resp

```
atic_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
```

e should see that the request was successfull with the 200 status response code

```
sponse=requests.get(static_json_url)
```

```
sponse.status_code
```

```
0
```

ow we decode the response content as a Json using `.json()` and turn it into a Pandas

```
Use json_normalize meethod to convert the json result into a dataframe
ta = pd.json_normalize(response.json())
```

ing the dataframe `data` print the first 5 rows

```
Get the head of the dataframe
ta.head()
```

static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket

Data Wrangling

- After creating the Pandas DataFrame we filter for only Falcon 9 launches using the Booster-Version column
- Missing values are replaced by mean values in the Payload mass column

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Out`
Then assign it to the variable `landing_class`:

[12]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df.Outco
```

This variable will represent the classification variable that represents the outcome of each lau
successfully; one means the first stage landed Successfully

[13]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

[13]:

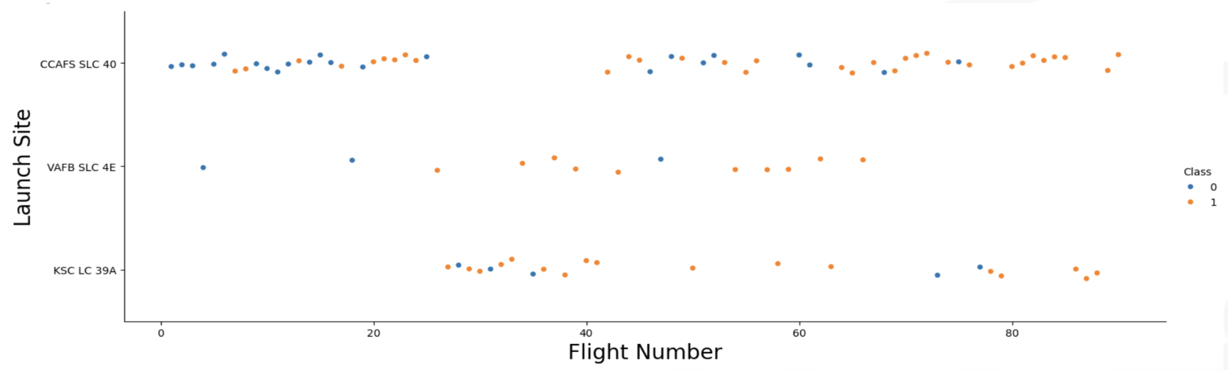
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

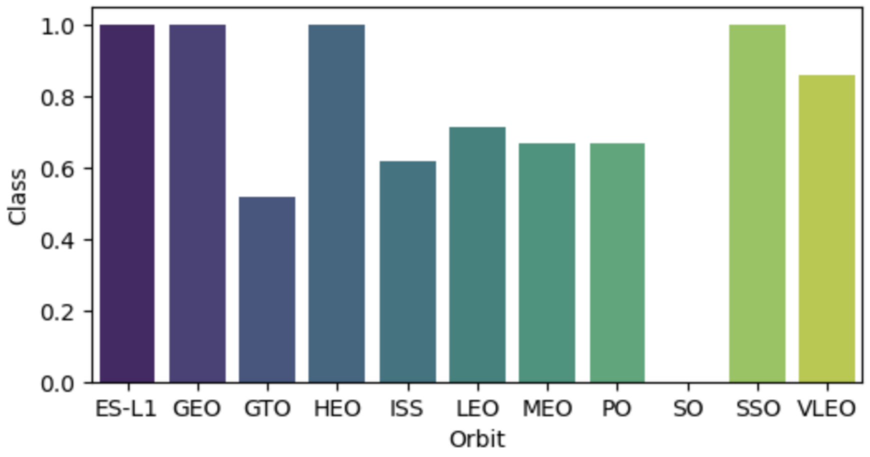
[14]:
```
df.head(5)
```

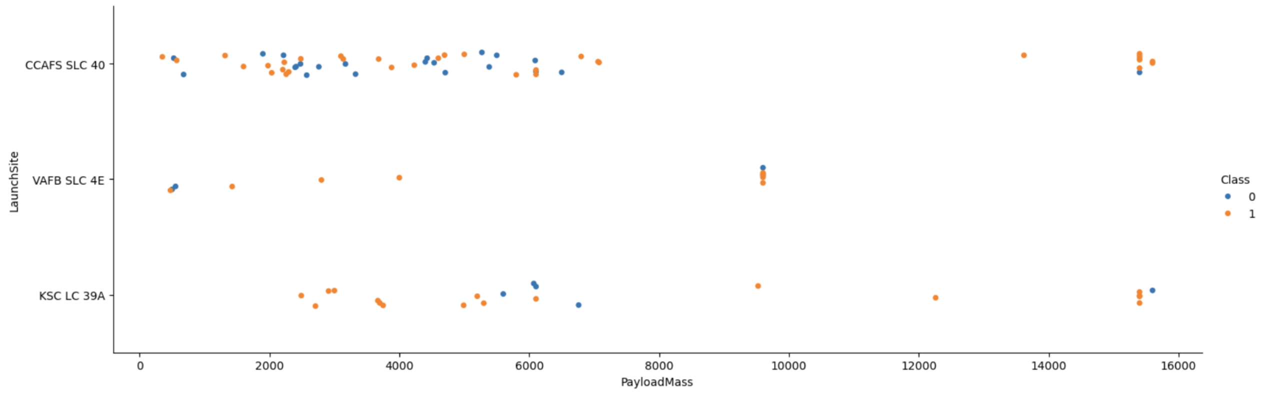
Exploratory Data Analysis

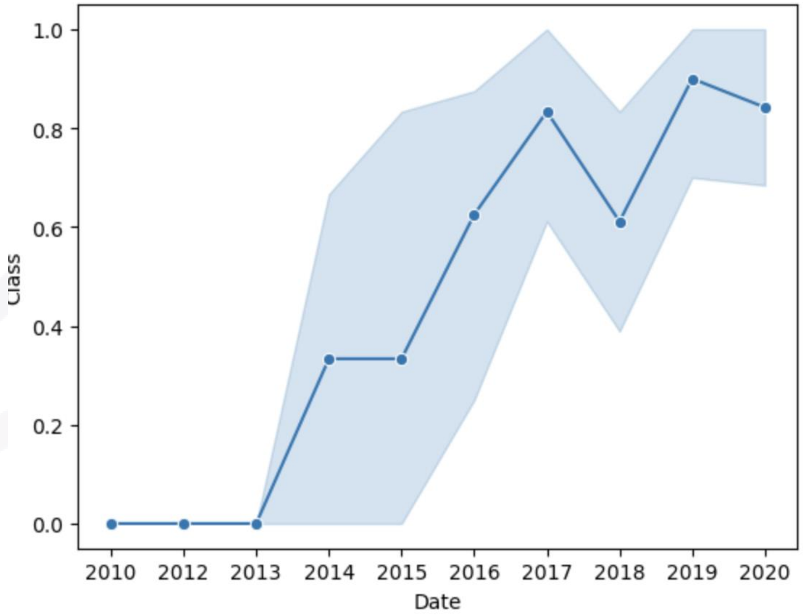
- Performed EDA and feature engineering using Pandas and Matplotlib
- Visualizations:
  - Scatterplot to visualize multiple columns of data
  - Bar chart to visualize success rate and orbit type
  - Line chart to visualize launch success over time

EDA – Visualizations









EDA - SQL

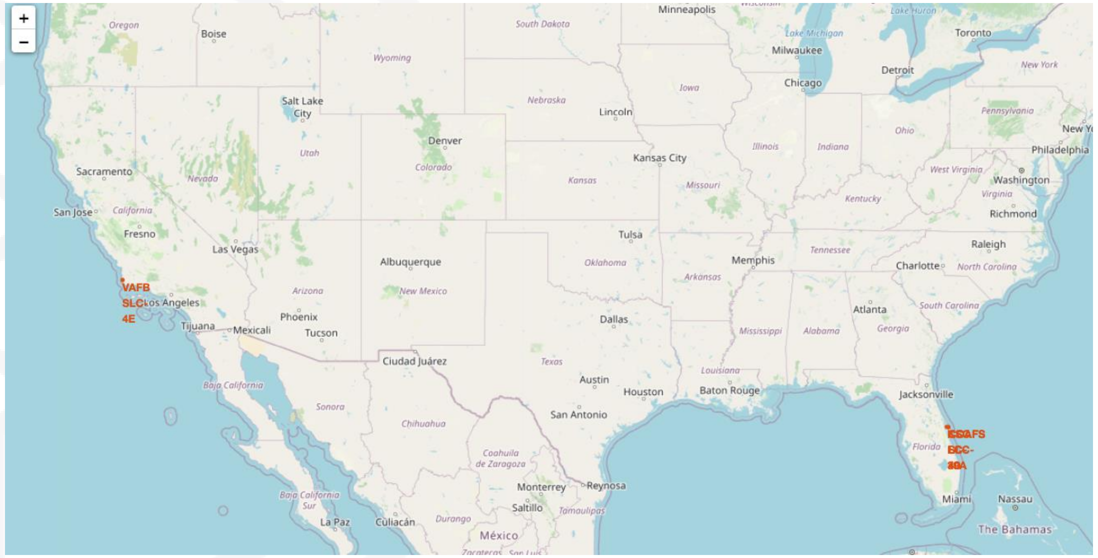
- Using SQL the following Queries were made:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1

EDA - SQL

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

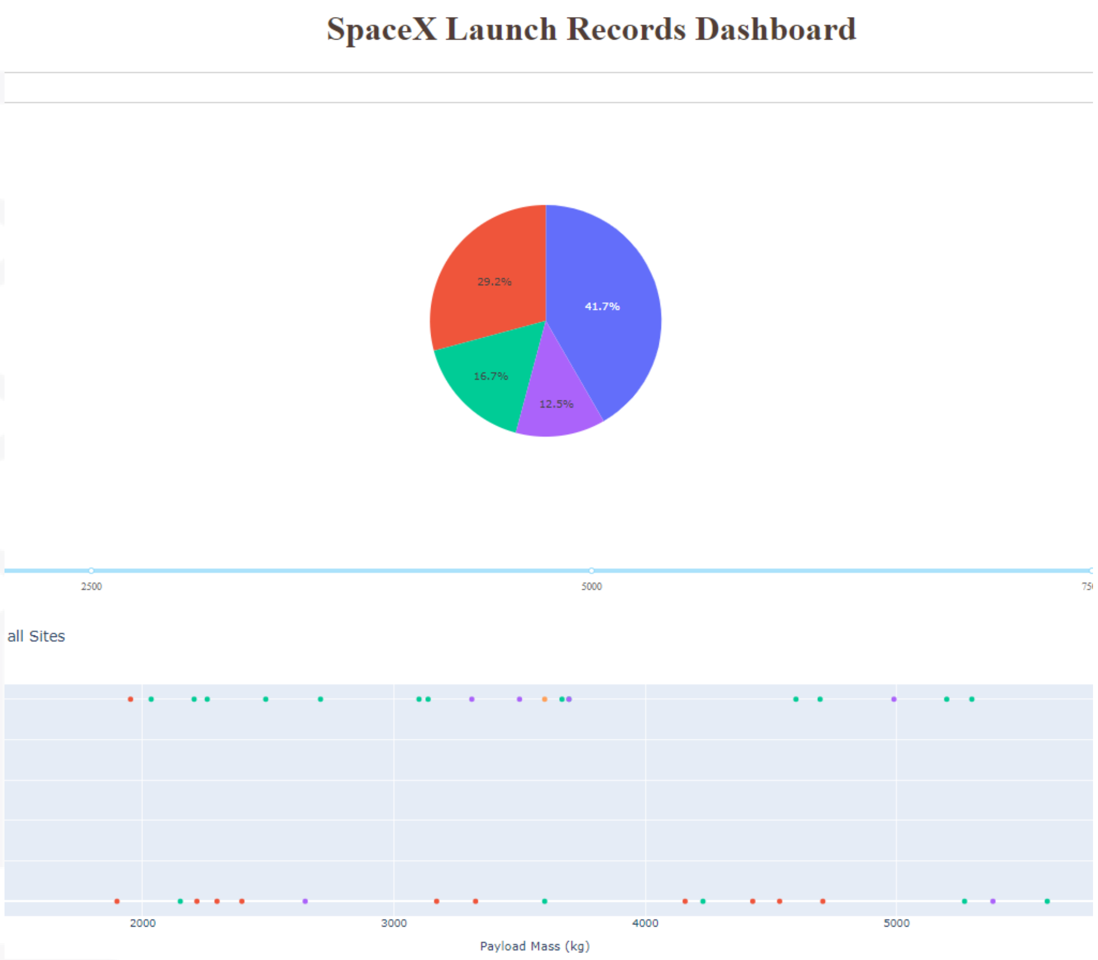
Interactive Visualization - Folium

- Using the Folium library, a map of all Falcon 9 launch sites was created
- Map objects such as markers, circles, lines and success/failure of launch was added



Interactive Visualization - Plotly

- An interactive dashboard was created using Plotly
- Features:
    - A drop-down input for launch site
    - Callback function to render a pie chart based on selection site dropdown
    - Range slider to select payload
    - Callback function to render the scatter plot of success vs payload



Predictive Analysis – Overview

- After loading in data, performing EDA and creating training labels
  - Created NumPy array from the Class column in our data assigning it to Variable Y
  - Standardized feature dataset (x) by transforming it using preprocessing StandardScalar() function in Sklearn.
  - Split into training and testing set using Sklearn with test size parameter set to .2 (20%) and random state to 2

Predictive Analysis – Classification

- Models Created:
  - SVM
  - Classification Tree
  - K nearest Neighbors
  - Logistic Regression
- Utilized GridSearchCV to find best hyperparameter for each model

- GridSearchCV object set for 10 then fit with training data to find best hyperparameter
- After fitting training set and outputting the GridSearchCV object, we display the best parameters using best_params_
- Calculated accuracy on test data and plotted confusion matrix for each model

Predictive Analysis – Classification Results

Results of Model Accuracy

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333

RESULTS

- Exploratory Data Analysis Results
- Interactive Visualizations
- Predictive Analysis Results

Insights: Exploratory Data Analysis

Insights - SQL

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

```
%sql select distinct Launch_Site from SPACEXTABLE;
```

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

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

- Created a list of all launch site names. SQL's distinct method allows us to get all unique launch site names.

Insights - SQL

```
%sql SELECT * from SPACEXTABLE 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
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

- Also gathered all launch site names beginning with 'CCA'

Insights - SQL

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like 'NASA%';
```

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

SUM(PAYLOAD_MASS__KG_)
99980

- SUM total payload mass from payload_mass_kg_ column resulting in a total payload mass of 99,980kg

Insights - SQL

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1%';
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- Average Payload mass for booster version F9 v1.1 was found by using the AVG function on the payload_mass_kg_ column.

Insights - EDA

```
%sql SELECT MIN(Date) from SPACEXTABLE where Mission_Outcome = 'Success';
```

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

MIN(Date)
2010-06-04

- The first date of successful landing selected by utilizing the min() function

Insights - SQL

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and
```

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

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

- Next, we find what boosters had a successful drone ship landing with payload between 4,000 and 6,000

Insights - SQL

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

- Using Count and Group By functions to find the total number of mission outcomes

Insights - SQL

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEX
```

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

boosterversion
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

- Use SUBQUERY to return the mass payload and use it to list all the boosters that have carried the maximum payload of 15600kgs.

Insights - SQL

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_O
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

- Using substr() to get the month and year from date column and returned the matching records.

Insights - SQL

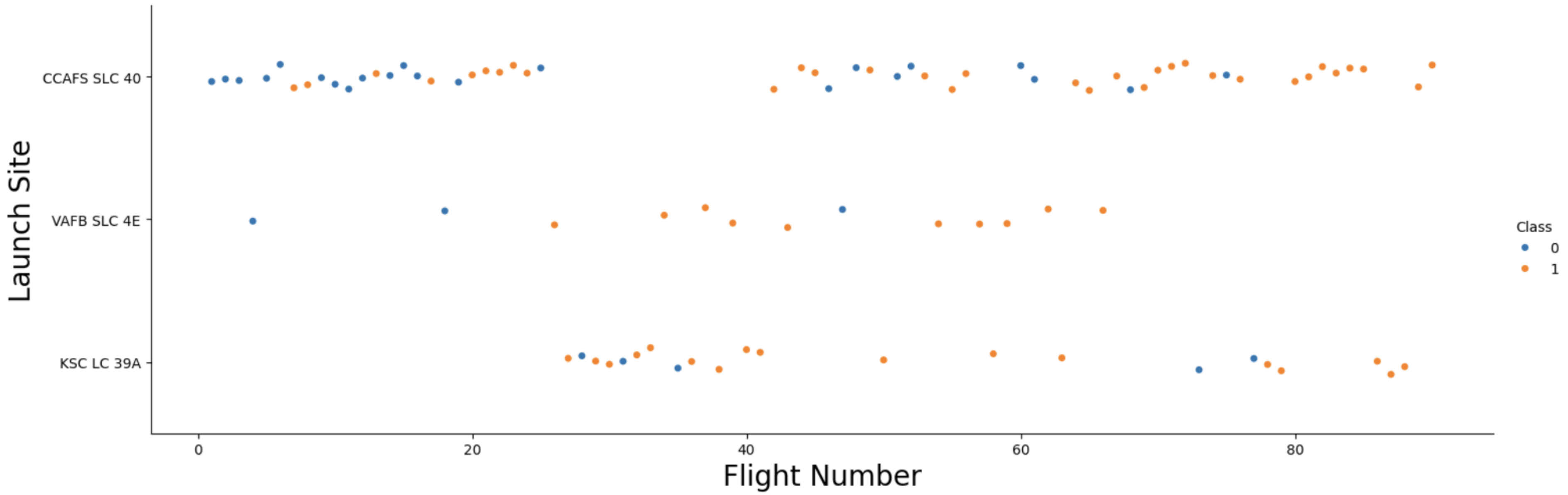
```
%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE "DATE" BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY "DATE" DESC;
```

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

Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)

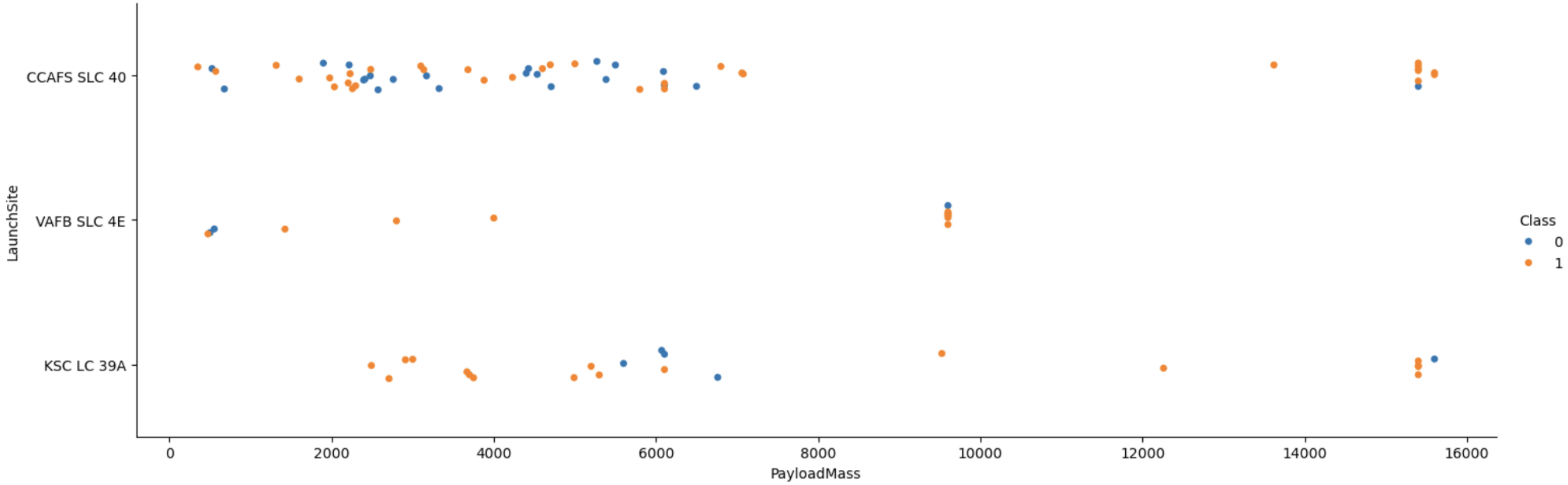
- Finally, we rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

Insights - EDA



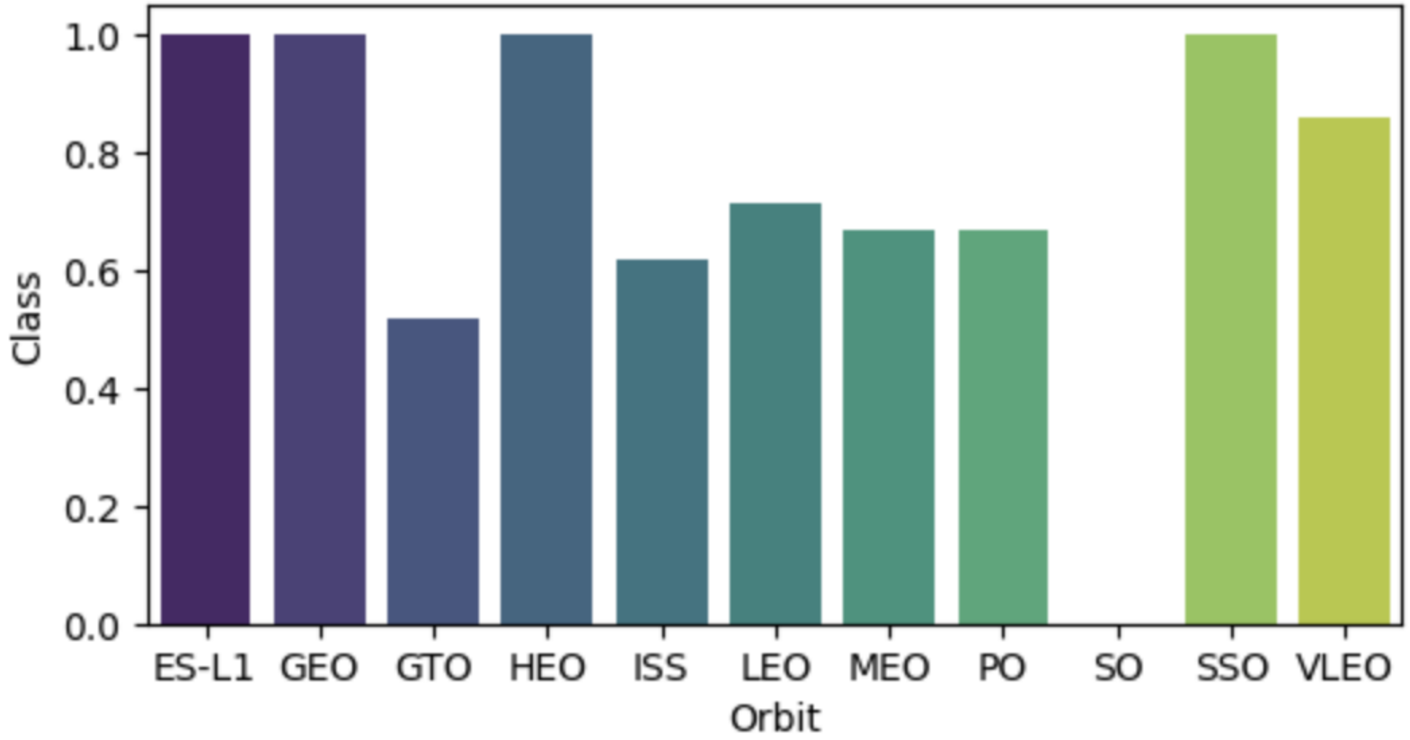
- As flight number increases, so does the rate of success. There is a noticable increase of success after flight number 20 for CCAFS SLC 40/VAFB SLC 4E and after flight number 40 for KSC LC 39A

Insights - EDA



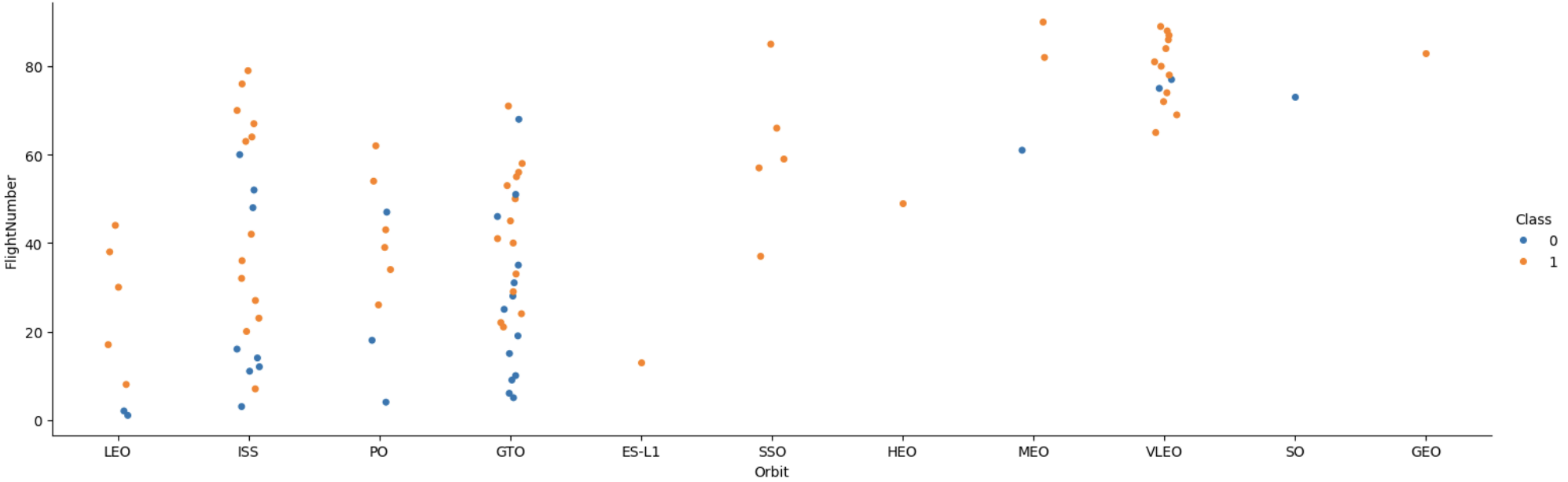
- Notice in this payload vs launch site plot that there are no launches with a payload greater than 10000 at VAFB SLC 4E.

Insights - EDA



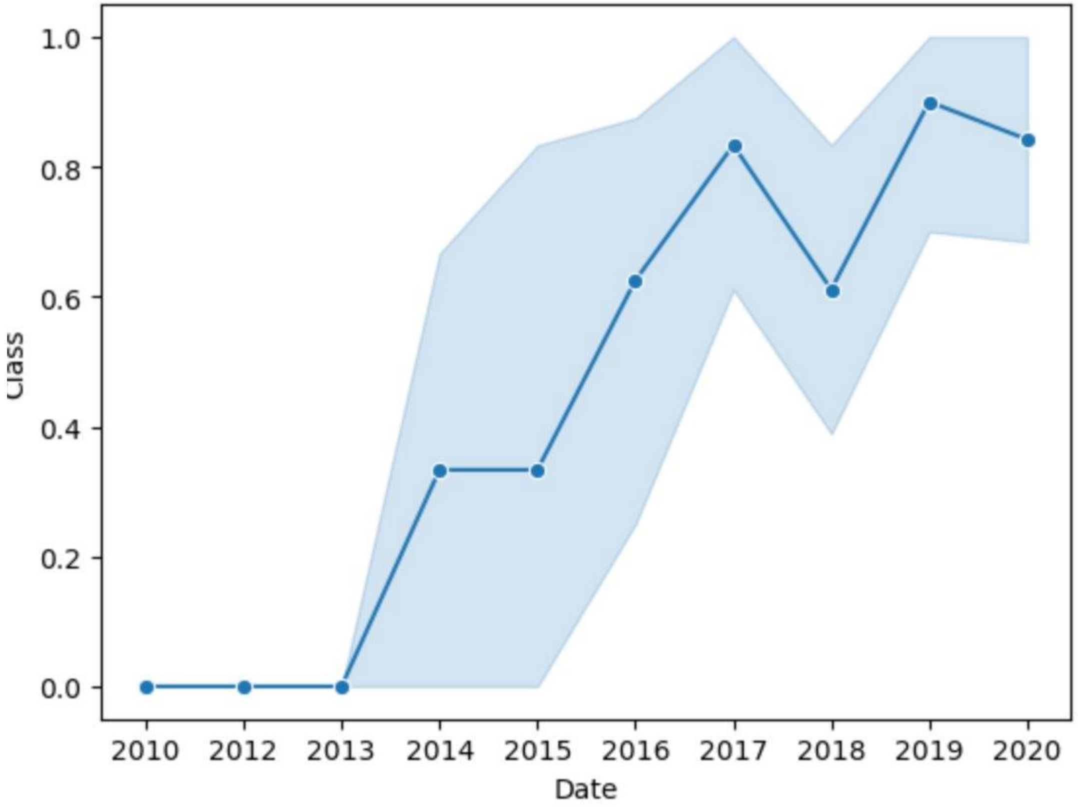
- Orbits ES-L1, GEO, HEO and SSO have the highest success rate at 100% meanwhile SO has the lowest success rate at 0%.

Insights - EDA



- LEO success shows good relation to number of flights. Inversely, there is no relation between flight number and Orbit type for GTO.

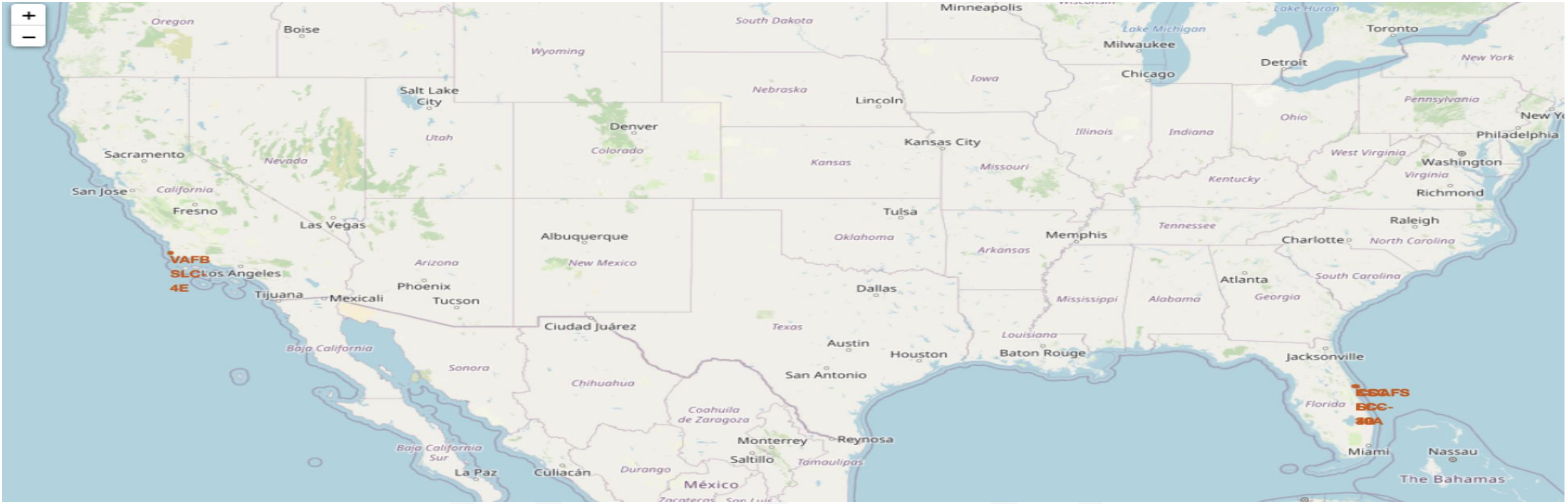
Insights - EDA



- Success rate has increased since 2013 showing great progress

Launch Site Analysis

Insights - Launch Site Analysis



- Four total SpaceX launch sites:
  - VAFB SLC-4E: Vandenberg Space Launch Complex
  - KSC-LC29A: Kennedy Space Center
  - CCAFS-LC40: Cape Canaveral Launch Complex 40
  - CCAFS-SLC40: Cape Canaveral Space Launch Complex 40

Insights - Launch Site Analysis

- Analyzed successful and unsuccessful launch sites. Green shows a successful launch while red is unsuccessful
- KSC LC 39A (Kennedy Space Center) had the most successful launches with 10

Insights - Launch Site Analysis

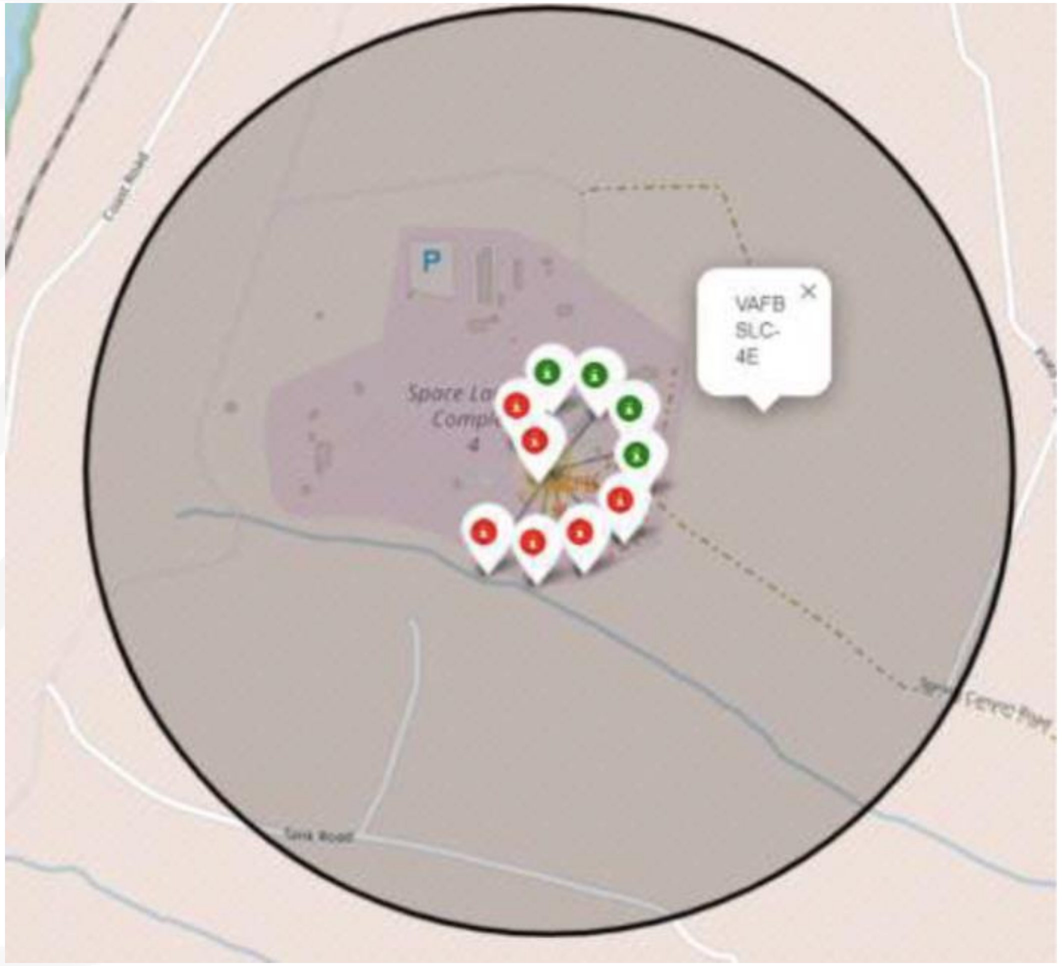
- Cape Canaveral has two launch sites close to each other.
- CCAFS LC-40 has 26 launches total launches out of which 7 launches were successful

Insights - Launch Site Analysis

- Second launch site at Cape Canaveral CCAFS LC-40 has 7 launches
- Only 3 were successful

Insights - Launch Site Analysis

- Vandenburg Space Launch complex is located in California, on the opposite coast.
- Vandenburg had 10 total launches with only 4 being successful.

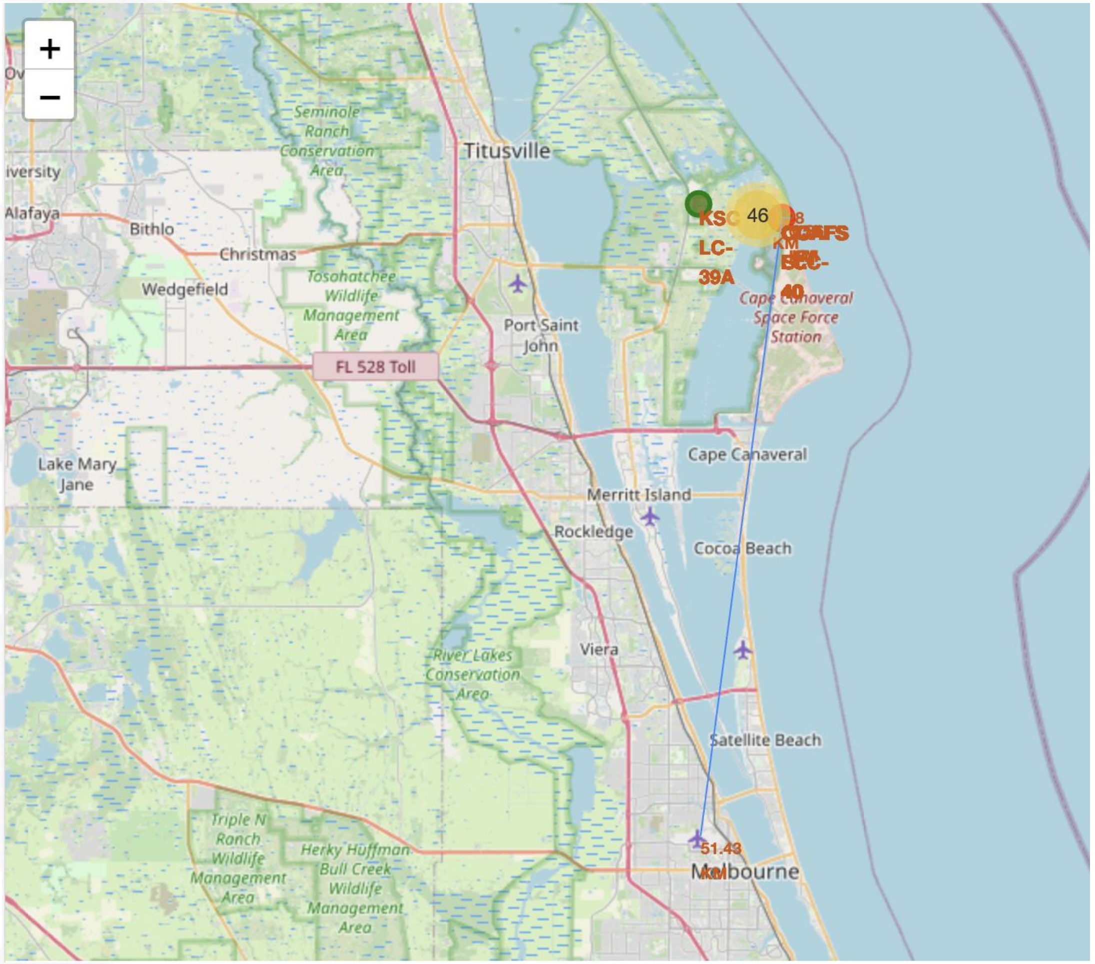


Insights – Launch Site Analysis

- Launch Sites are chosen for their proximity to the ocean and the equator.
- Launch Sites on the east coast head Southeast over the Atlantic while launch sites on the west coast head Southwest over the pacific.

Insights – Launch Site Analysis

- Launch sites are also far enough from cities to minimize risk and disruption



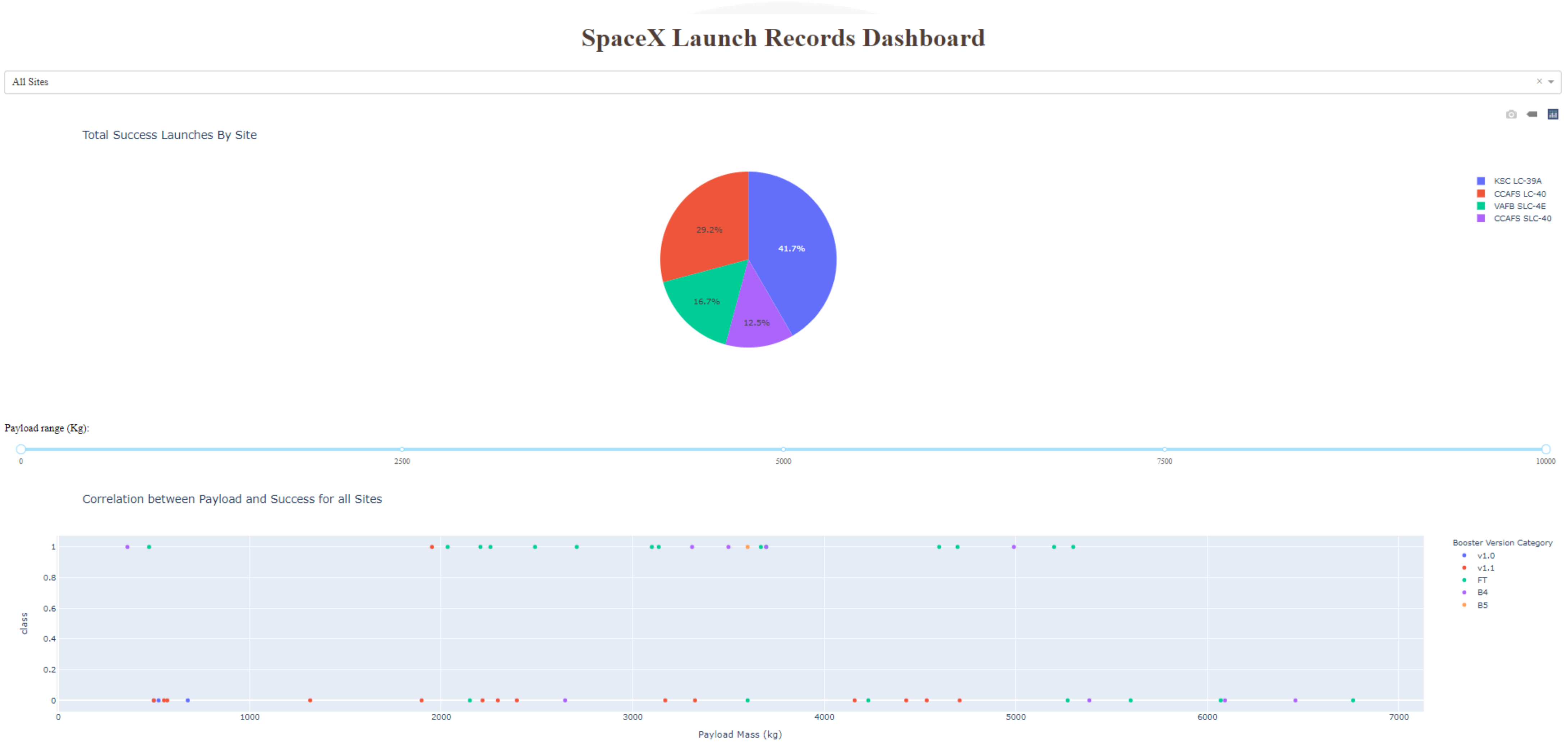
Plotly Dashboard

Insights – Plotly Dashboard

- ilt a Dashboard with:
  - Dropdown menu for selecting launch sites
  - Pie charts displaying success rates
  - Scatter plot displaying multiple columns of data
  - Range slider for selecting ranges of payload mass

- Analyze launch sites by success rate
- Analyze payload ranges with highest and lowest success rate
- Analyze which booster version had highest success rate

Insights – Plotly Dashboard



Prediction Analysis

Insights – Prediction Analysis

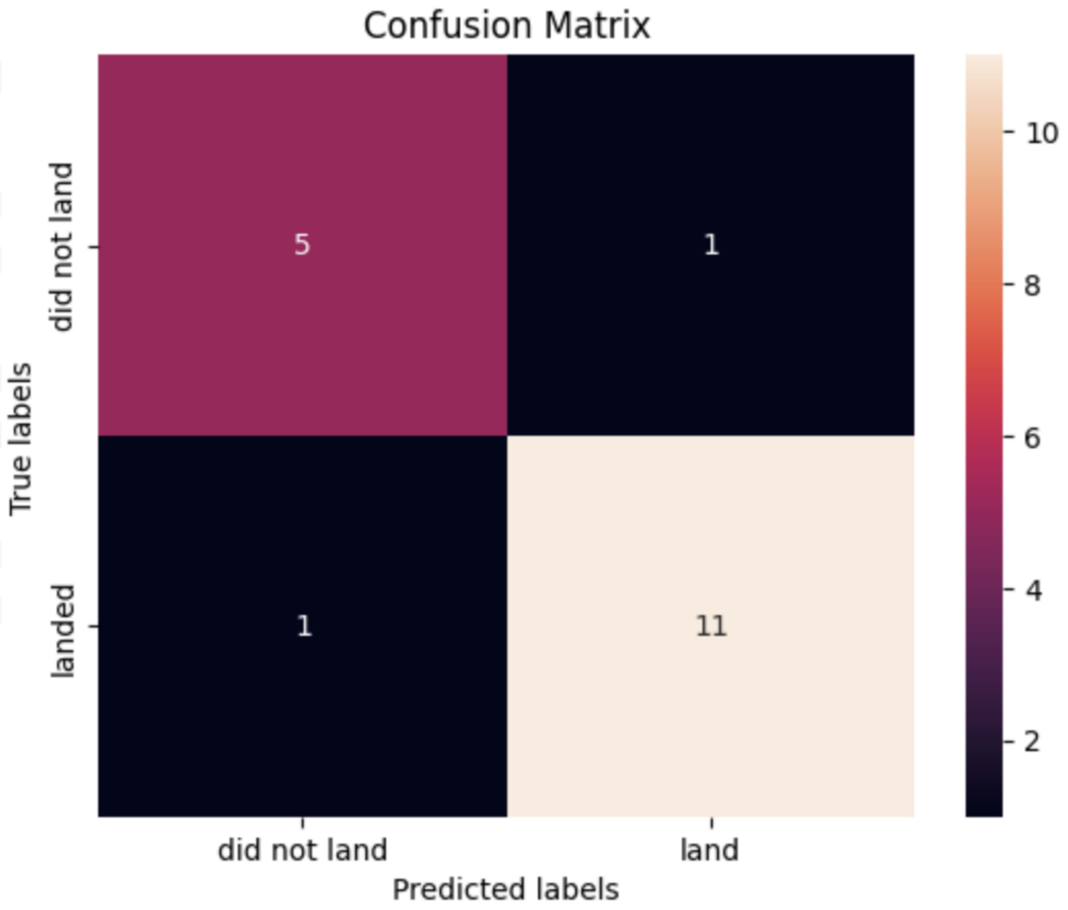
- Used 4 different predictive methods. Decision Tree has the highest accuracy of any model with 88%

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333

Insights – Confusion Matrix

- Decision Tree Confusion Matrix did a better job of predicting successful landings

- Decision Tree did not do as well predicting unsuccessful landings compared to all other models



Conclusions

- As flight number increases, the success rate increases.
  - VAFB SLC 4E has a success rate of 100% after flight number 50.
  - KSC LC 39A and CCAFS SLC 40 have a success rate of 100% after flight number 80.
- VAFB SLC 4E had no launches with a payload greater than 10000
- Different launch sites have different success rates
  - CCAFS LC-40 has a success rate of 60%
  - KSC LC 39A and VAFB SLC 4E have a success rate of 77%
- Orbits ES-L1, GEO, HEO and SSO have the highest success rate at 100% meanwhile SO has the lowest success rate at 0%.
- Heavy payloads have a higher landing success rate in Polar, LEO and ISS missions
- Success rate continuously increased from 2013-2020.


Thank you!
Skills Network