IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Morteza Taghipour
June 2023

Outline

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

Executive Summary

- Summary of methodologies
- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Interactive map with Folium
- Interactive Dashboard with Plotly
- Predictive analysis (Classification)
- Summary of all results
- Exploratory data analysis
- Predictive analysis results

Introduction

- Project background and context

- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost 165 million dollars, and much of the difference is because SpaceX can reuse the first-stage rocket. Therefore, we try to determine if the first stage will land safely and according to the result, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

- We use previous data to predict the probability of the booster landing on the pad correlated to the launch site, mass, payload orbit, booster version and the landing pad location

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Requesting rocket launch data from SpaceX API with the following URL
  - Using Python, I have chosen just the necessary columns for my analysis
  - Filter the data frame to only include `Falcon 9` launches
  - Perform data wrangling
  - Found missing values and substitute them with the mean of data in each column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

Data collected and normalized according to below code

```
[90]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json
```

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

```
[91]: response.status_code
```

```
[91]: 200
```

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

```
[92]: # Use json_normalize meethod to convert the json result into a dataframe
data1=requests.get(static_json_url)
data2=data1.json()
data=pd.json_normalize(data2)
```

Using the dataframe `data` print the first 5 rows

```
[93]: # Get the head of the dataframe
data.head()
```

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/data-collection-api.ipynb

Data Collection - Scraping

Data collected and normalized according to below code

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[20]: # use requests.get() method with the provided static_url
      # assign the response to a object
      data=requests.get(static_url)
      data
```

```
[20]: <Response [200]>
```

Create a `BeautifulSoup` object from the HTML `response`

```
[21]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
      soup = BeautifulSoup(data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[22]: # Use soup.title attribute
      soup.title
```

```
[22]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Data%20collection-scraping.ipynb

Data Wrangling

- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcomes per orbit type
- Create a landing outcome label from the Outcome column

TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Data_wrangling.ipynb

Data Wrangling

TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [6]:

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Out[6]:

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
SO         1
HEO        1
ES-L1      1
GEO        1
Name: Orbit, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [10]:

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
```

EDA with Data Visualization

- Visualized the relationship between FlightNumber and Payload variables
- Visualized the relationship between FlightNumber vs LaunchSite
- Visualized the relationship between success rate and orbit type
- Visualized the relationship between FlightNumber and Orbit type
- Visualized the relationship between Payload and Orbit type
- Visualized launch success yearly trend

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Eda-dataviz.ipynb

EDA with SQL

- Displayed the names of the unique launch sites in the space mission
- listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listed the total number of successful and failed mission outcomes
- Listed the names of the booster_versions which have carried the maximum payload mass
- Ranked the count of successful landing_outcomes

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Eda-sql.ipynb

Build an Interactive Map with Folium

- What I have done:
- Marked all launch sites on a map
- Marked the success/failed launches for each site on the map
- Calculated the distances between a launch site to its proximities
- With this map, we can answer to the below questions:
- Are all launch sites in proximity to the Equator line?
- Are all launch sites in very close proximity to the coast?
- From the color-labeled markers in marker clusters, we are able to easily identify which launch sites have relatively high success rates.
- It is possible to zoom in a launch site and explore its proximity to see if we can easily find any railway, highway, coastline, etc.

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Interactive%20map%20with%20folium.ipynb

Build a Dashboard with Plotly Dash

- Plotted pie chart for total successful launches by all sites
- Plotted pie chart for total successful launches by each site
- Add a slider to choose the payload range
- Scatter plot which shows sites with failed and successful landing for each booster and payload

Predictive Analysis (Classification)

- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Create a logistic regression object and calculate the accuracy on the data
- Create a support vector machine object and calculate the accuracy on the data
- Create a decision tree classifier object and calculate the accuracy on the data
- Create a k nearest neighbors object and calculate the accuracy on the data

https://github.com/Morteza-Taghipour/IBM-SpaceX-capstone-Project/blob/main/Machine_Learning_Prediction.ipynb

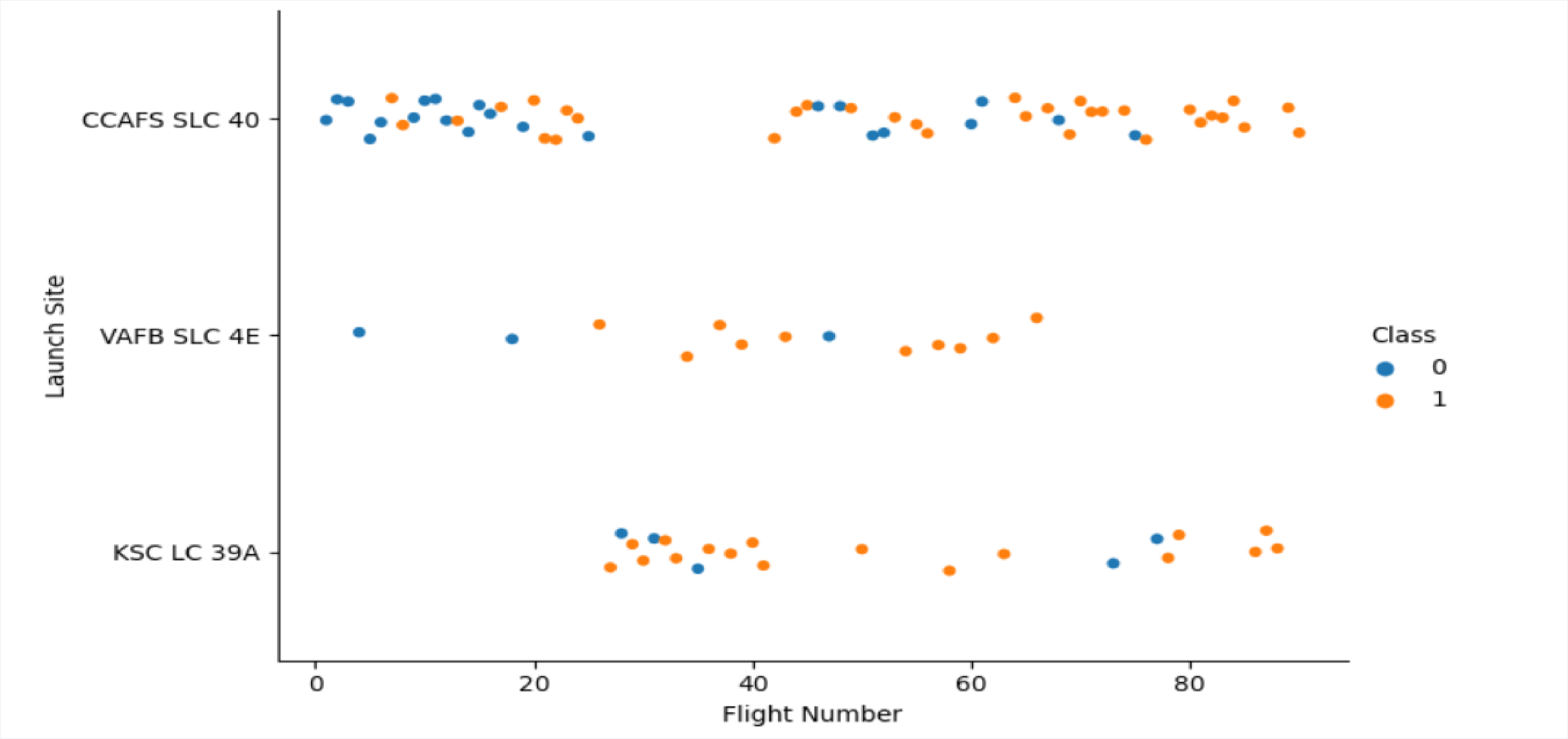
Results

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

Section 2

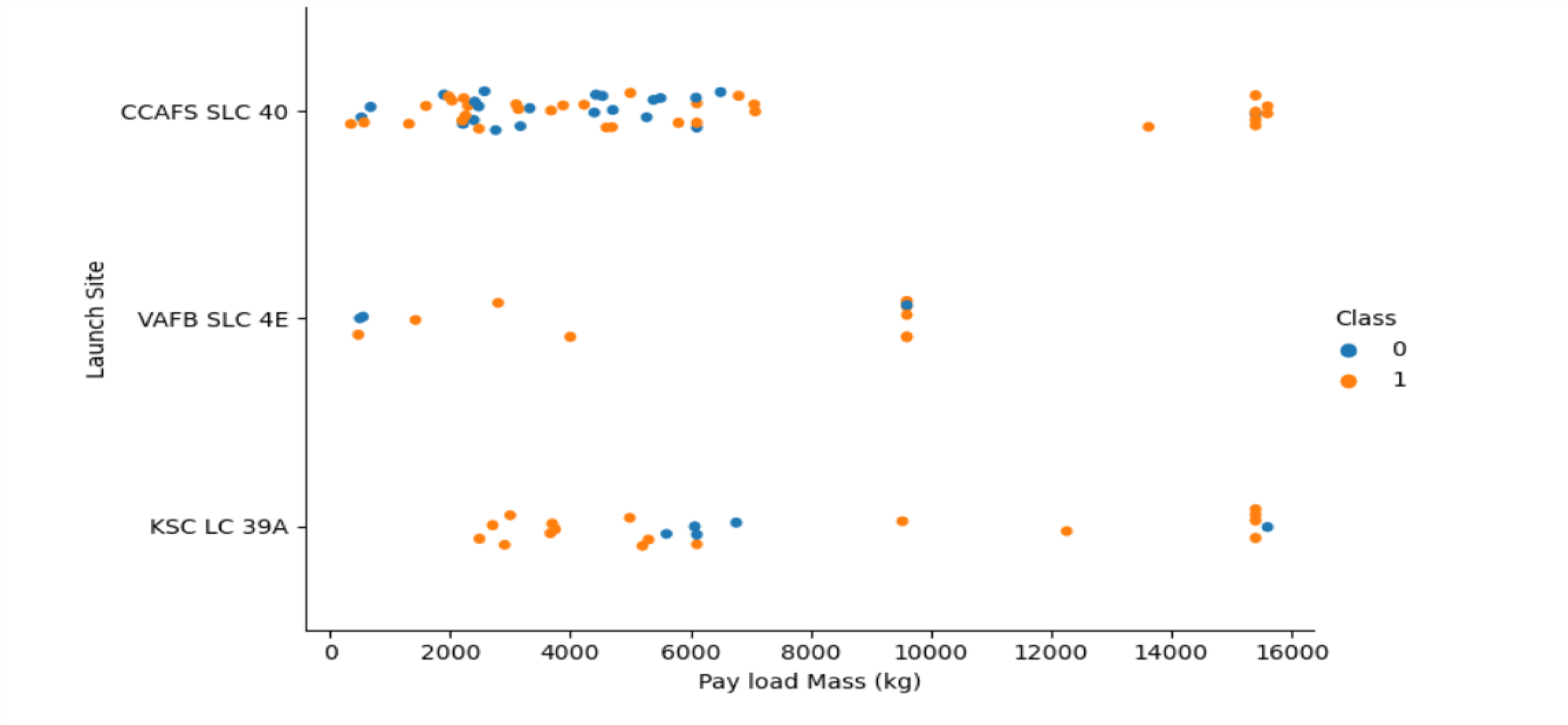
Insights drawn from EDA

Flight Number vs. Launch Site



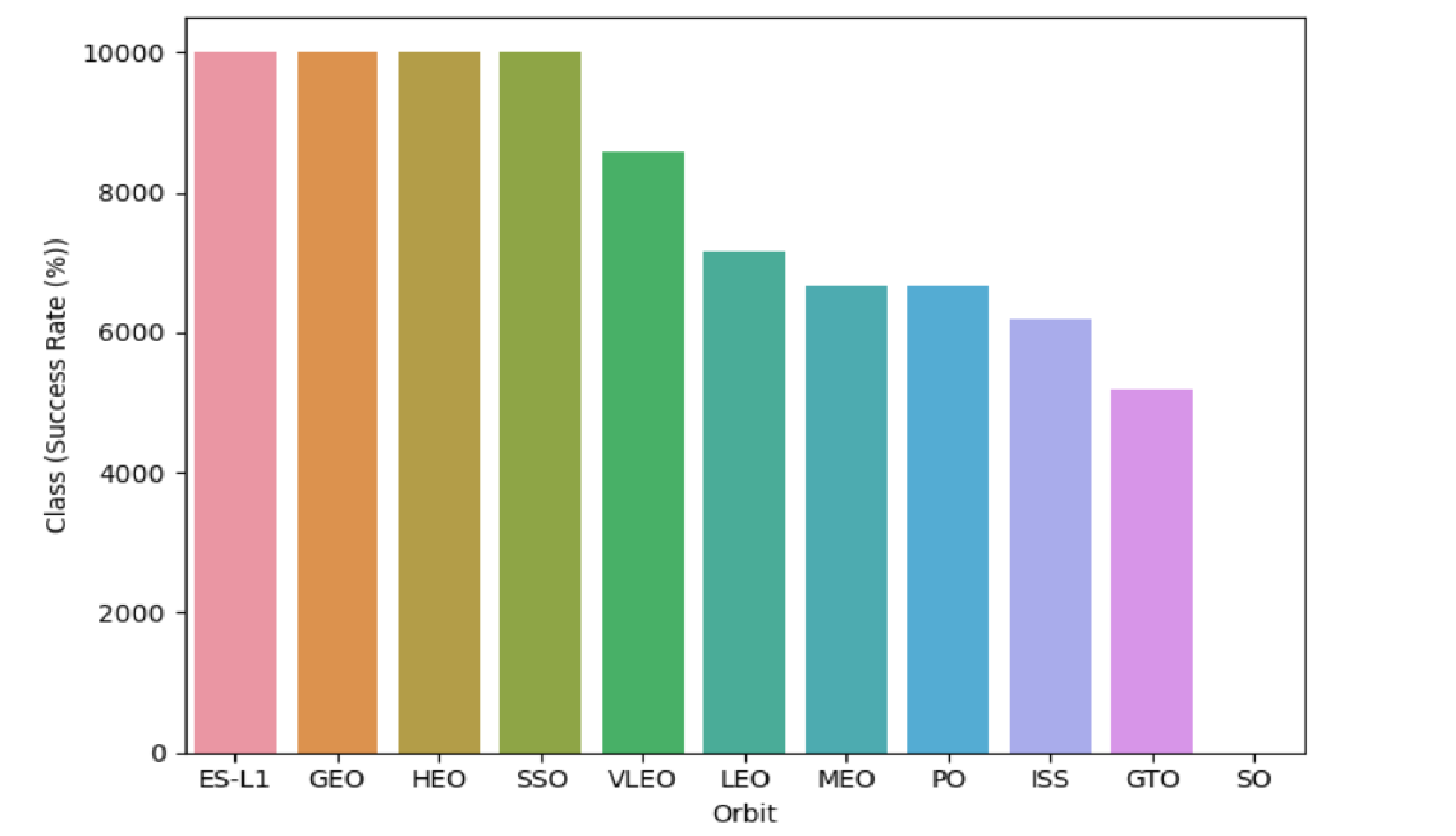
This plot shows that more flight numbers, more probability of successful landing

Payload vs. Launch Site



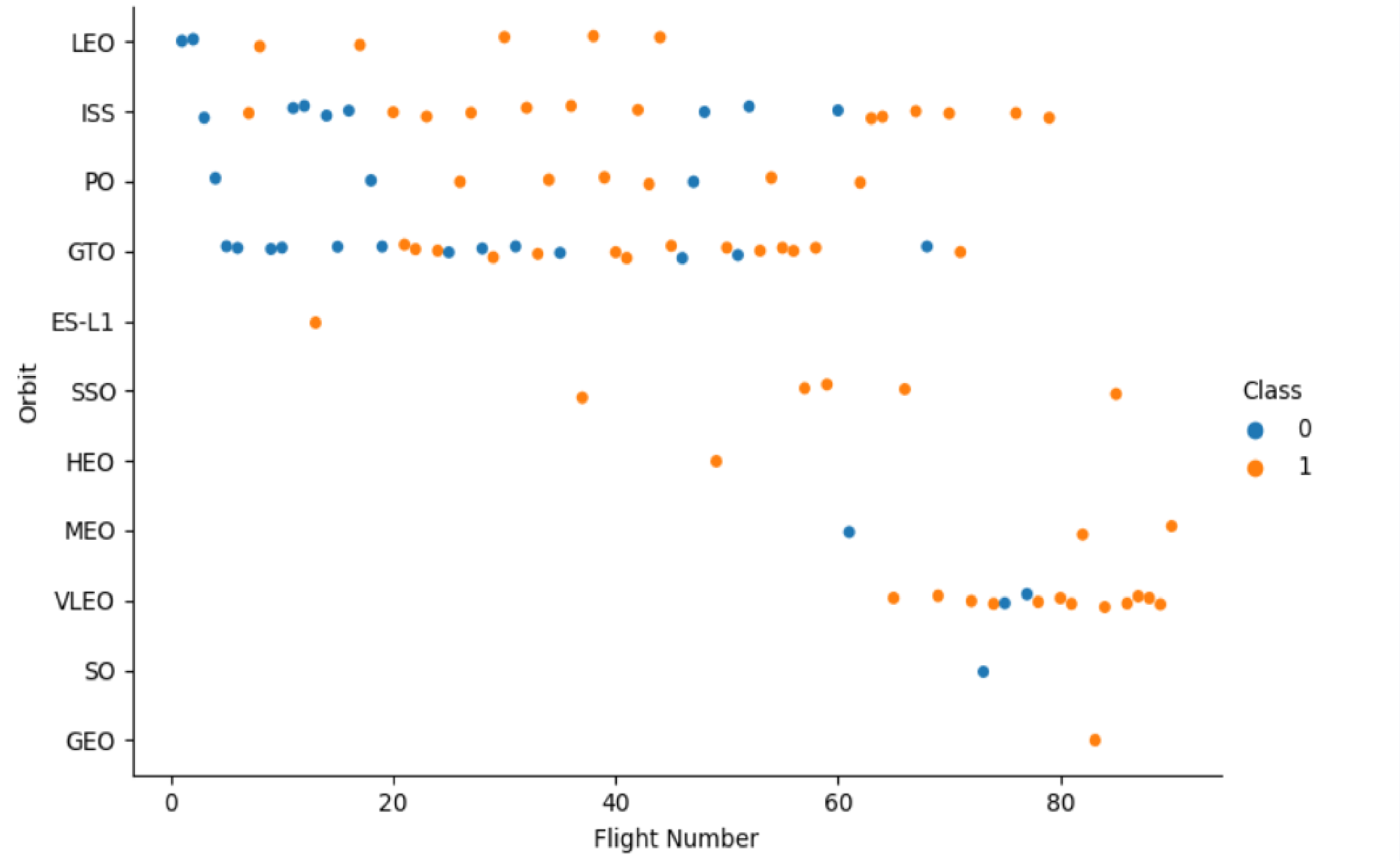
In the “CCAFS SLC 40” site, more payload results in more successful landing

Success Rate vs. Orbit Type



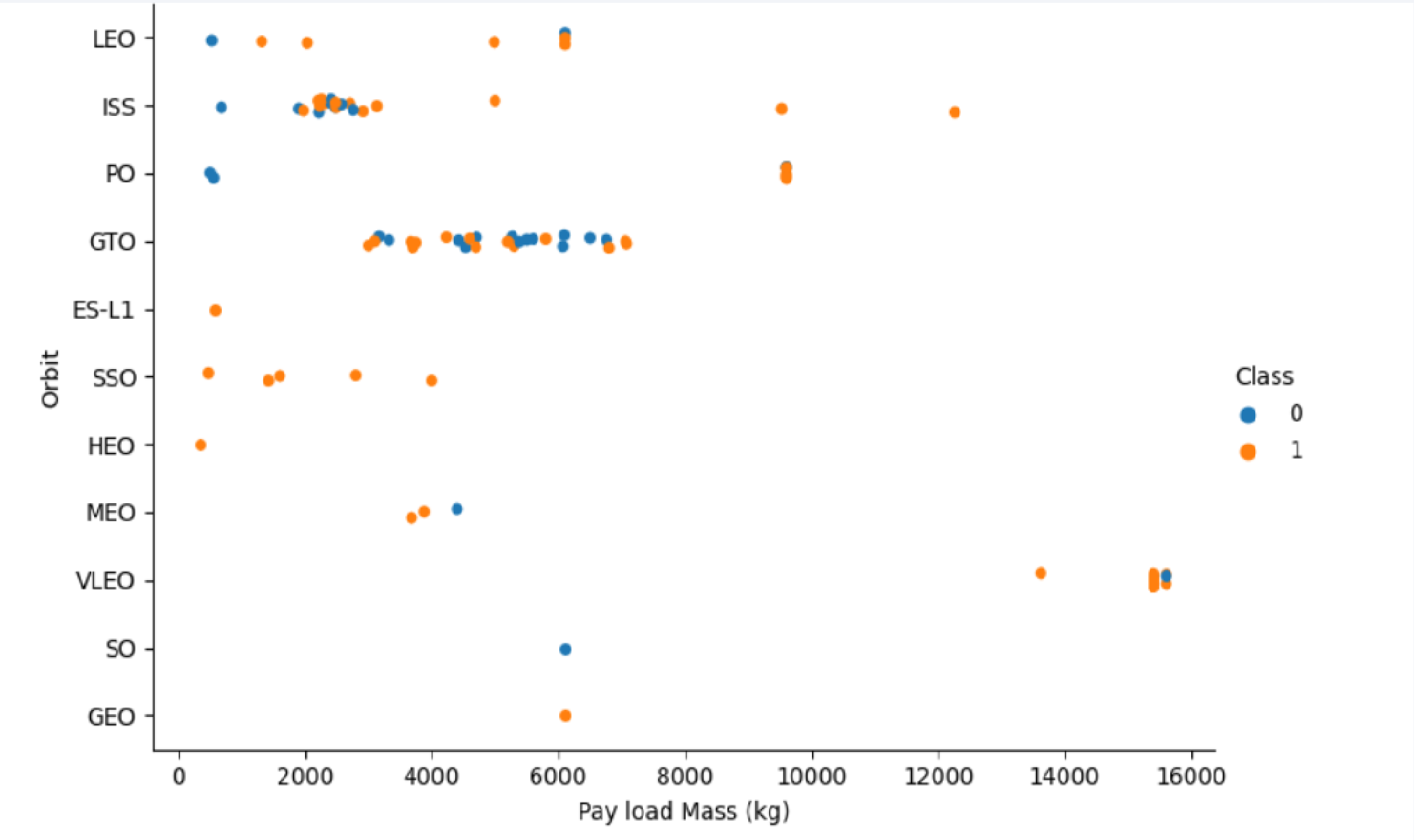
The first 4 orbits have 100% successful landing

Flight Number vs. Orbit Type



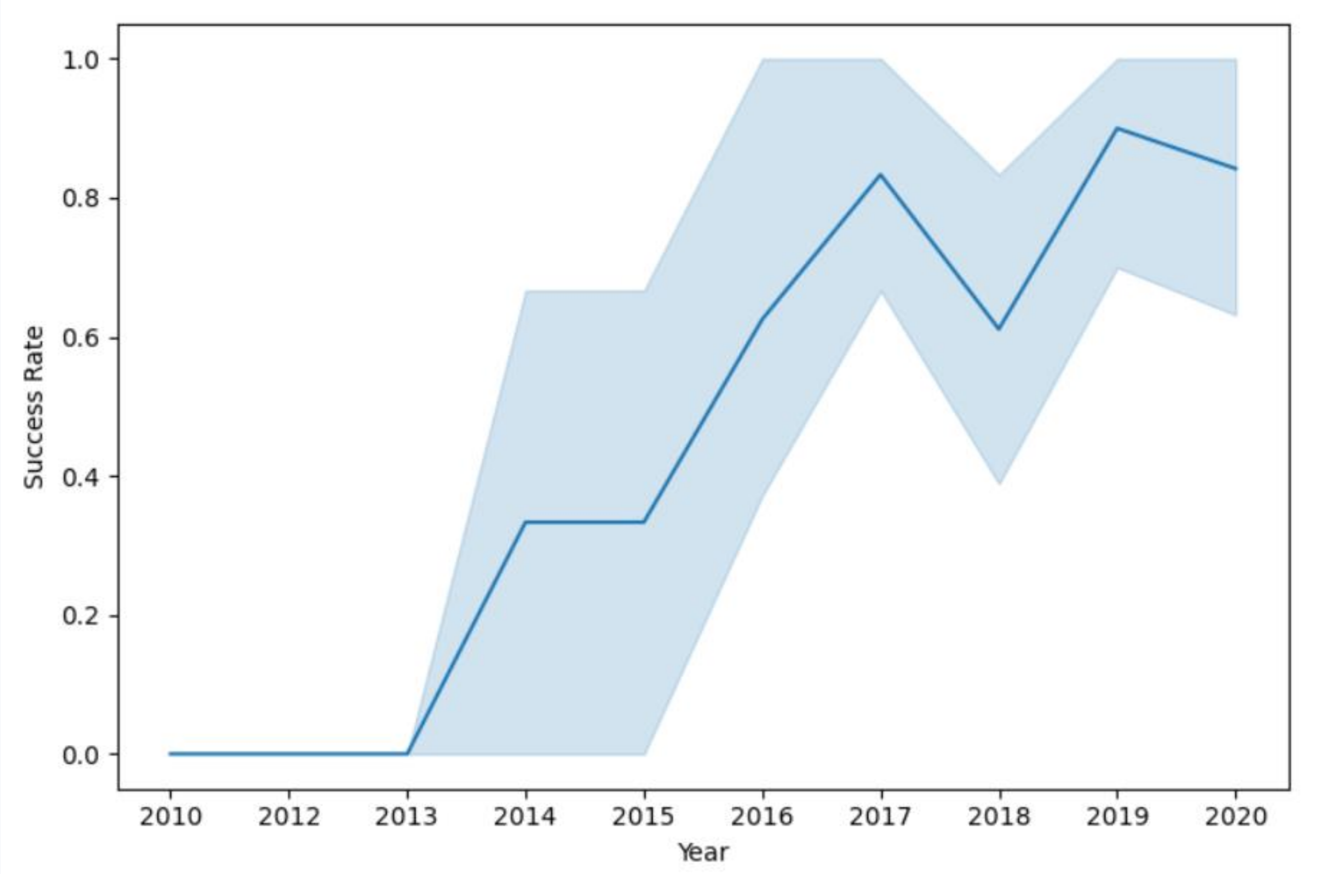
The flight number increases, the first stage is more likely to land successfully

Payload vs. Orbit Type



With Heavy payloads the successful landing rate is more for Polar,LEO, and ISS

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020

All Launch Site Names

```
In [74]: %sql select distinct(LAUNCH_SITE) from SPACEXTBL;

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

Out[74]:

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

The key word DISTINCT shows unique launch sites

Launch Site Names Begin with 'CCA'

Out[75]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

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

Total Payload Mass

```
In [76]: %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

Out[76]:

payloadmass
619967.0

Total payload mass carried by boosters launched by NASA

Average Payload Mass by F9 v1.1

```
In [77]: %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

Out[77]:

payloadmass
6138.287128712871

Average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
In [78]: %sql select min(DATE) from SPACEXTBL;

 * sqlite:///my_data1.db
Done.

Out[78]:
```

min(DATE)
01/06/2014

Dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

In [79]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 a
```

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

Out[79]:

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

Names of boosters that have successfully landed on drone ships and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

[18]:
```
%sql select count(Mission_Outcome) as successful from SPACEXTBL where Mission_Outcome like 'success%';
```

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

[18]:

successful
100

[19]:
```
%sql select count(Mission_Outcome) as Failure from SPACEXTBL where Mission_Outcome like 'failure%';
```

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

[19]:

Failure
1

Total number of successful and failed mission outcome

Boosters Carried Maximum Payload

```
[81]: %sql select Booster_Version as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);

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

[81]:

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

List the names of the booster which have carried the maximum payload mass

2015 Launch Records

```
[82]: %sql SELECT substr(Date, 4, 2) as Month,Landing_Outcome,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL where substr(Date,7,4)='2015' and La
```

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

[82]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

```
[84]: %sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DES

 * sqlite:///my_data1.db
Done.
[84]:
```

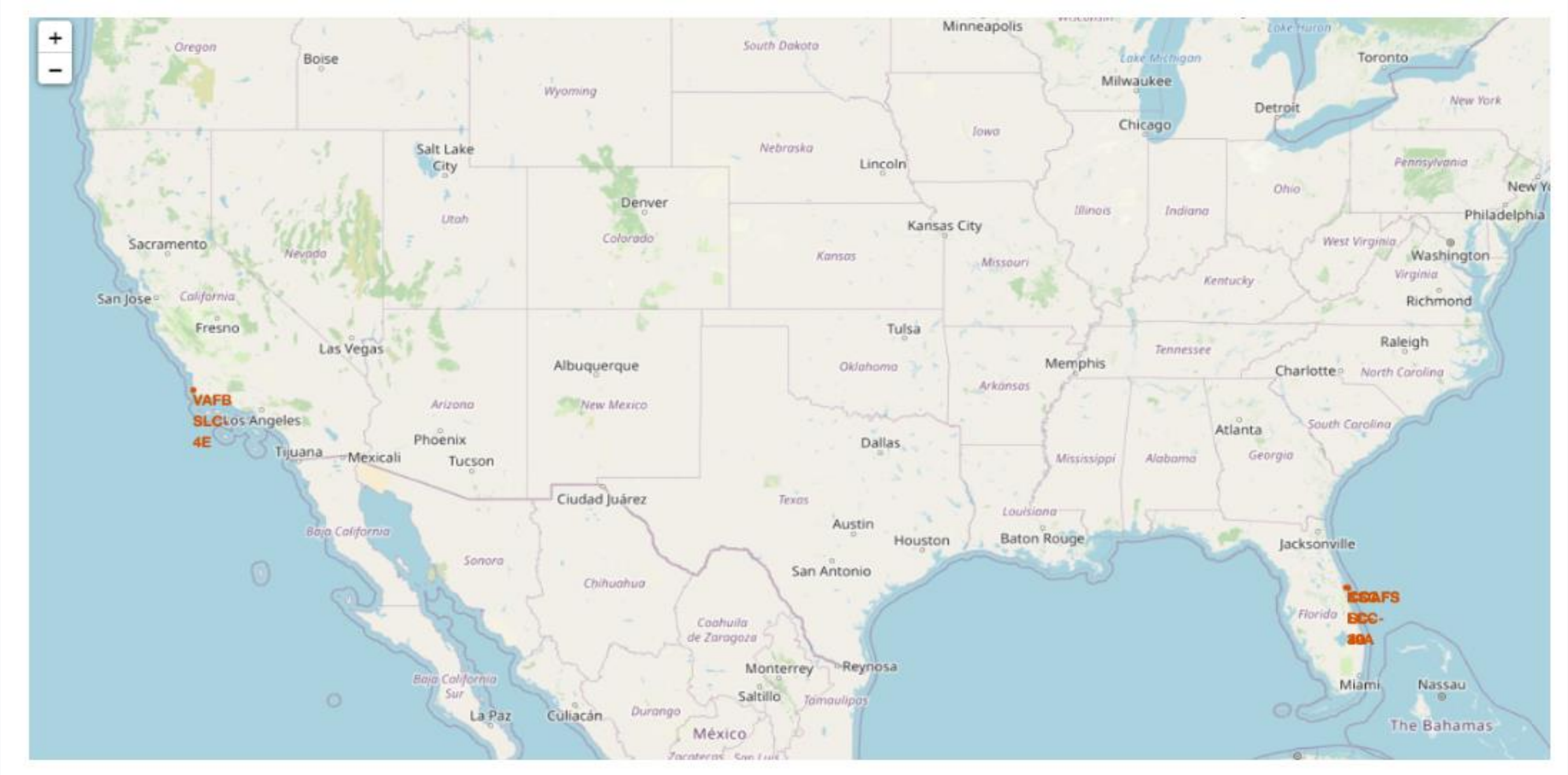
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
19/02/2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18/10/2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600.0	LEO	SpaceX	Success	Success
18/08/2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440.0	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18/07/2016	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)

- 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

Section 3

Launch Sites Proximities Analysis

Launch sites global map



The position of the launch site and its proximity to the Equator line or coast

Launch sites with color labels

From the color-labeled markers in marker clusters, It is possible to easily identify which launch sites have relatively high success rates

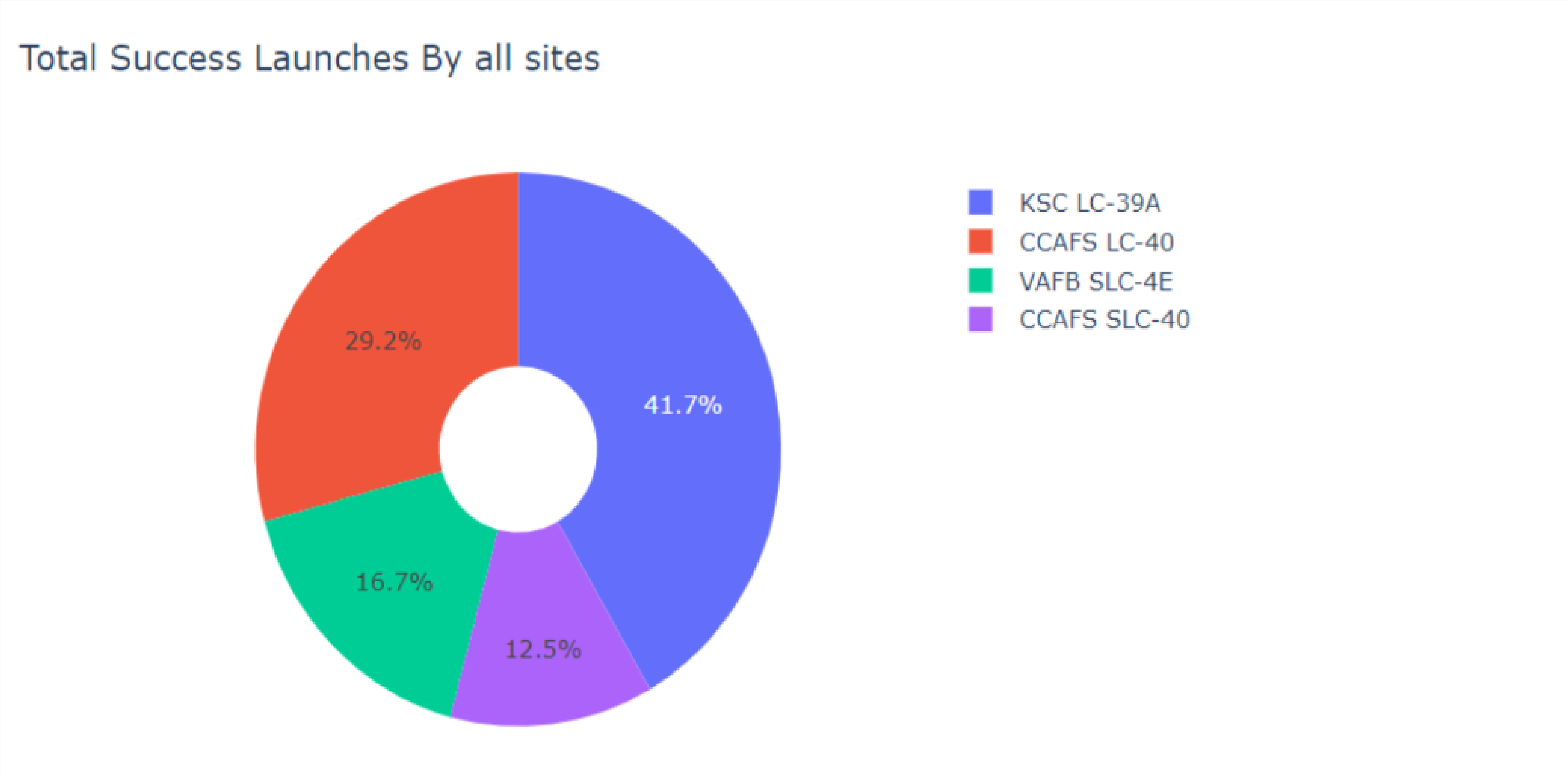
Launch Site distance to landmarks

The screenshot of a selected launch site to its proximities such as railway, highway, coastline

Section 4

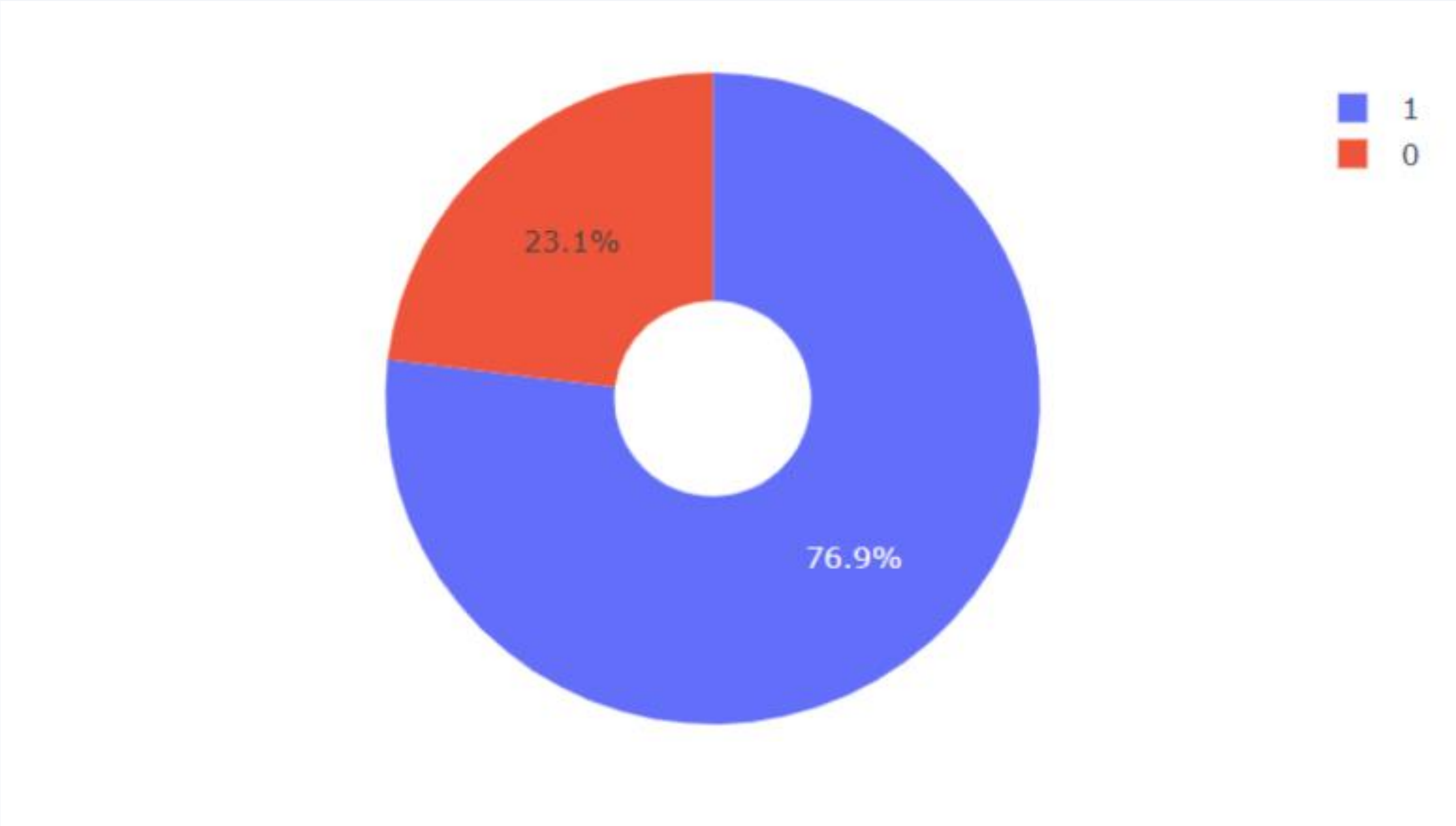
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



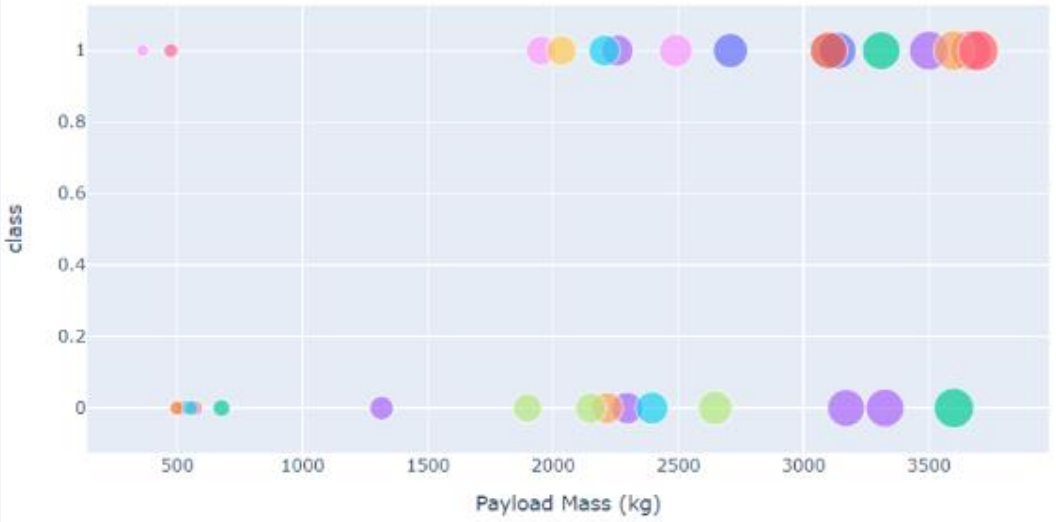
KSC LC-39A is the most successful launch site

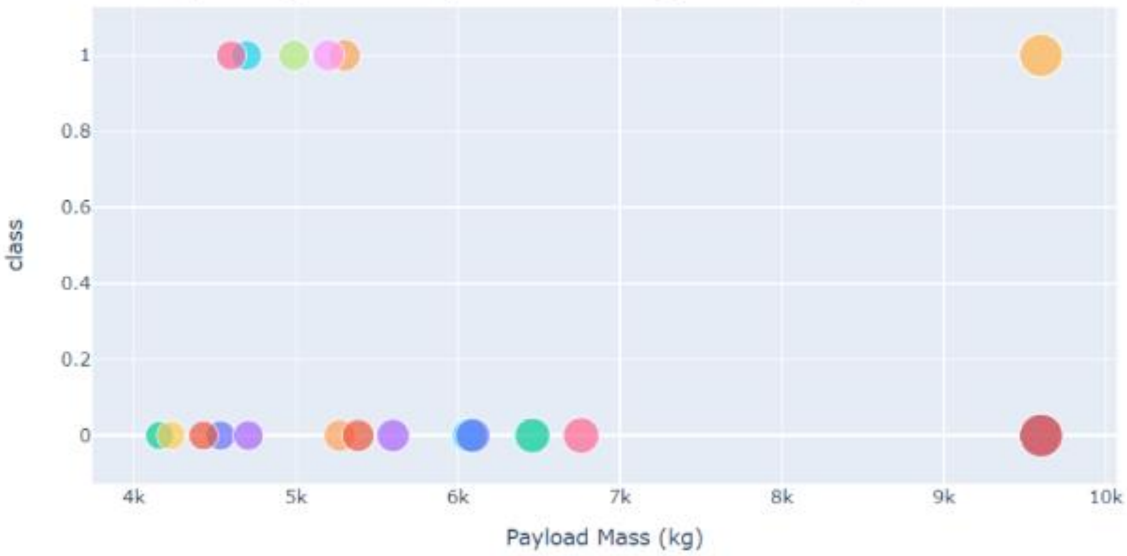
Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A with a 76.9% success rate has the highest launch success

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





Payload vs. Launch Outcome scatter plot for all sites, with different payload

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

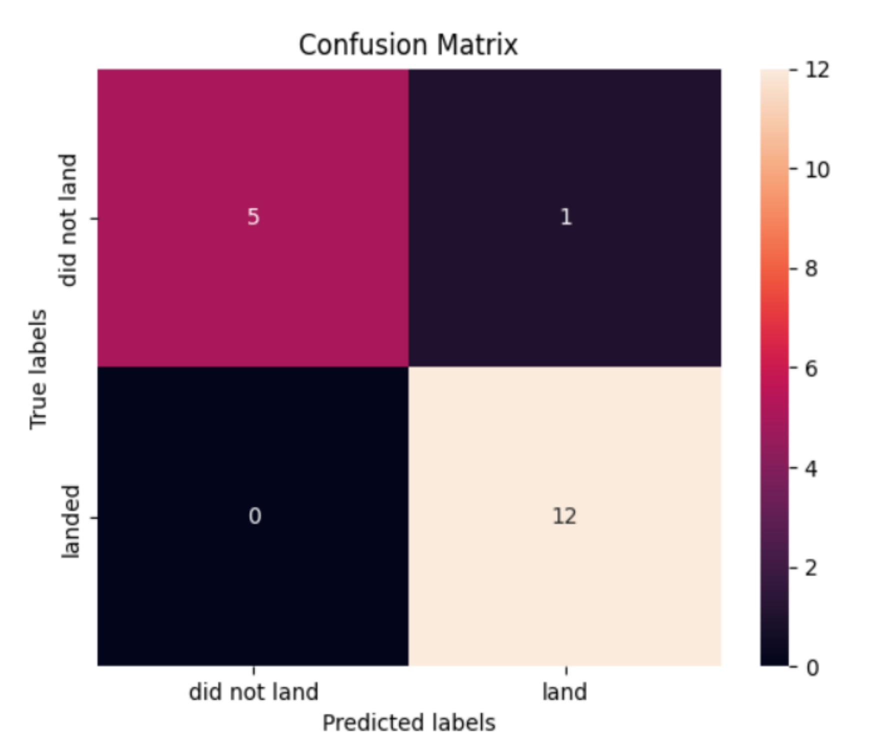
In [39]:
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

best = max(models, key=models.get)
print('Best model is', best,'with a score of', models[best])
```

```
Best model is DecisionTree with a score of 0.875
```

Decision tree classifier is the model with the highest classification accuracy

Confusion Matrix



Confusion matrix for the decision tree classifier

Conclusions

We can conclude that:

- More flight result in more successful landings on each site.
- Launch success rate increased from 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most successful rate.
- KSC LC-39A had the most successful launches.
- The Decision tree classifier is the best machine learning algorithm.

Thank you!