Into the Space using Data Science

Muhammad Hassan

November 11, 2022

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Conclusion

EXECUTIVE SUMMARY

- Summary of methodologies
  - Data Collection through API
  - Data Acquisition through Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis using Data Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of results
  - Results from Exploratory Data Analysis
  - Interactive Analytics through Maps and Dashboard
  - Predictive Analysis results

INTRODUCTION

- Problem Statement

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX 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 for
  - What are the factors that determine successful landing of first stage of rocket?
  - Which feature plays a greater role in determining the successful landing of rocket?
  - What are the operating conditions to ensure a successful landing of first stage of rocket launch?

METHODOLOGY

- Data Collection through SpaceX API and Web Scraping
- Data Wrangling
- EDA with SQL & Data Visualization
- Performing interactive visual analytics with Folium and Plotly Dash
- Performing predictive analysis through supervised classification models

Data Collection

- The data is collected from SpaceX API and Web scraping from Wikipedia

  - The information retrieved from the API, https://api.spacexdata.com/v4/, includes rockets, launches and payload information

  - The information retrieved through web scraping from Wikipedia, https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches, includes launches, landing and payload information.

Data Collection – SpaceX API

- A get request was used to retrieve data from the API and some cleaning and formatting was done to make it more readable.
- The cleaned data was then exported to a CSV file for later use.

[Link to Code]

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

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

```
In [15]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
In [35]: # Calculate the mean value of PayloadMass column
avg = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, avg, inplace=True)
```

We can now export it to a **CSV** for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

```
In [37]: data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Web Scraping

- BeautifulSoup was used to scrape the Wikipedia page and collect Falcon9 rockets record.
- The records were parsed to find the correct tables and were stored in a pandas dataframe object which was then exported to a CSV file.

[Link to code]

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

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
response_object = BeautifulSoup(response.content)
```

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

In [7]:
```
# Use soup.title attribute
response_object.title
```

Out[7]: `List of Falcon 9 and Falcon Heavy launches - Wikipedia`

In [8]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = response_object.find_all('table')
```

Data Wrangling

- The dataset included several cases where the booster did not land successfully.
- The string variables in the dataframe were one-hot encoded to convert them into categorical variables.
- The number of launches at each site and the occurrence of orbit types was also noted.

[Link to code]()

```
In [5]:
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()

Out[5]:
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64

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
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64

In [7]:
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

Out[7]:
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

EDA with Data Visualization

- We explored the data by visualizing the relationships between different variables through plotting of scatter plots, bar graphs and line graphs.
- Scatter plots were used to look for correlation between two variables.
- Bar graphs were used to show the frequency of occurrences and relationship between a categorical and numeric variable.
- Line graph was used to show the trend of a variable.

Link to code

EDA with SQL

- We loaded the SpaceX dataset on IBM Db2 database.
- We performed queries on the data to understand it better:
  - Displaying the unique launch sites in space mission
  - Display 5 records where launch site begins with 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first 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 the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

[Link to code]

Build an Interactive Map with Folium

- We marked all launch sites on a folium map object and added color-coded markers and formed clusters to identify the success or failure of a landing at that particular point.
- We added markers to show the distance between the launch site and the key locations such as railway, highway, coast, and city.
- These objects were created to understand the data visually and gain other insights.

Link to code

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash
- A dashboard was created with a dropdown, pie chart, range slider and scatter plot components
  - Dropdown allowed users to select a specific launch site or all of them
  - The pie chart showed the success and failure rate of the launch site selected through the dropdown box
  - A range slider allowed users to select the range of payload mass
  - A scatter plot was also created to show the relationship between Success and Payload mass (range used from range slider)

Link to code

Predictive Analysis

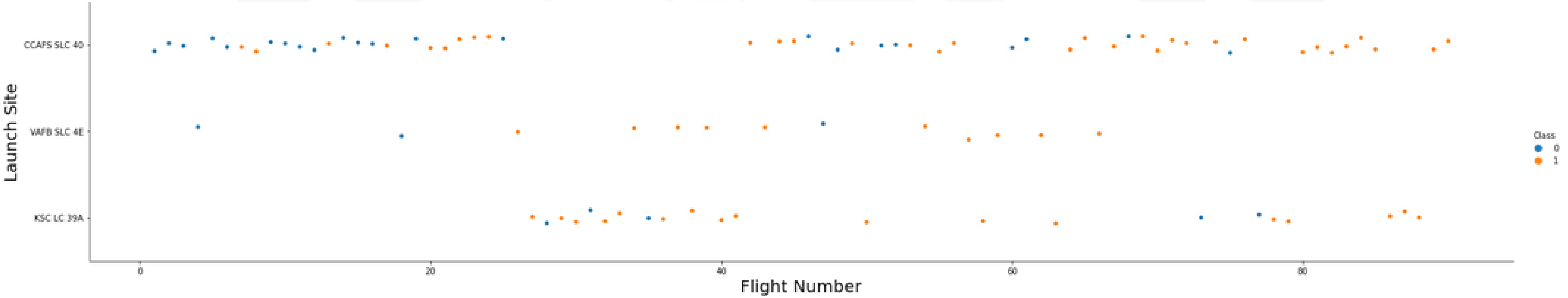
- The data was prepared by loading, then normalizing its values and then splitting data into train and test sets.
- Different machine learning algorithms were selected and GridSearchCV was used for exhaustive searching of best hyperparameters of a model.
- The accuracy of each model was evaluated on the test set and a confusion matrix was plotted to look for precision and recall of each model.
- The models were compared according to their accuracy and the best one among them was selected.

[Link to code]

RESULTS

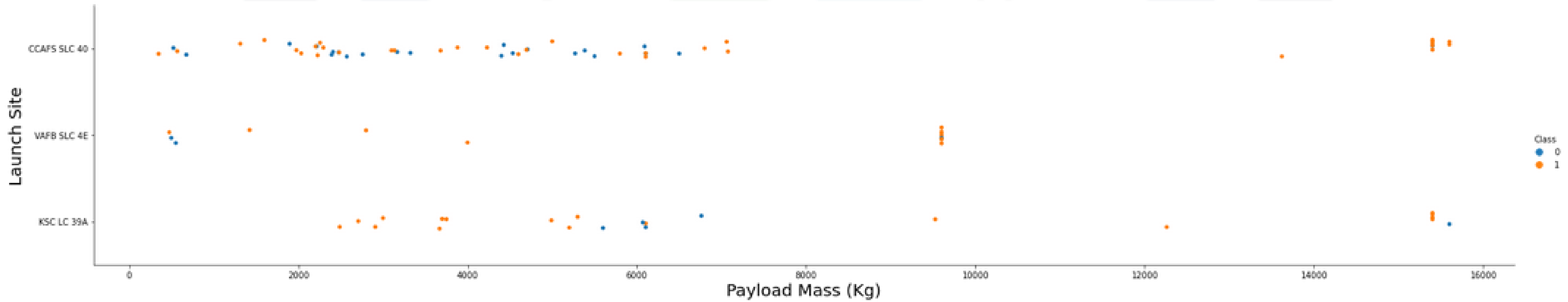
- Exploratory data analysis results
- Interactive analysis through screenshots
- Predictive analysis results

Flight Number vs. Launch Site



Here we observe as the flight number increases, the success for each launch site also increases.

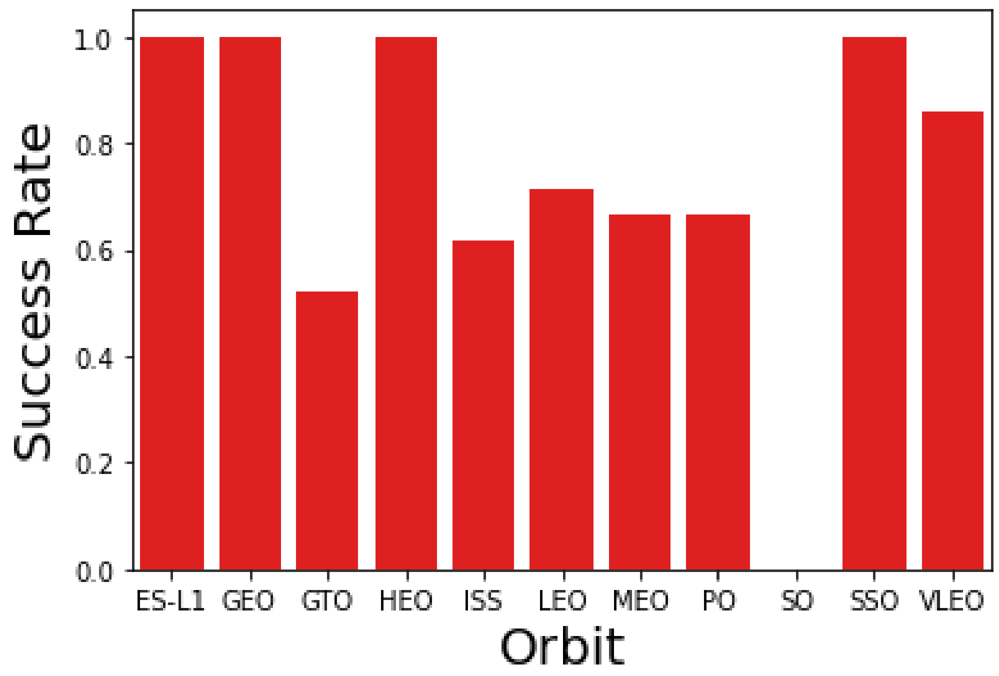
Payload vs. Launch Site



We see that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

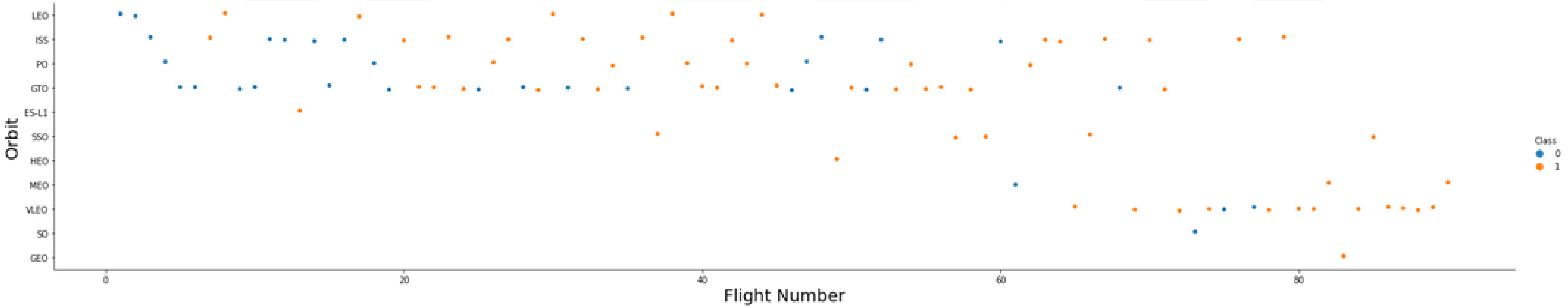
We can also see that greater the payload mass is, higher are the chances for successful landing.

Success Rate vs. Orbit Type



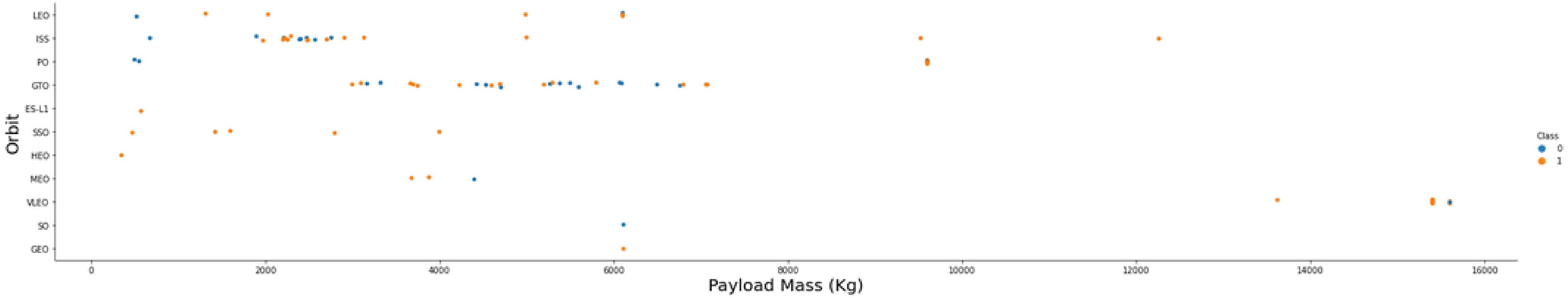
We can see that ES-L1, GEO, HEO, SSO and VLEO have a greater success rate.

Flight Number vs. Orbit Type



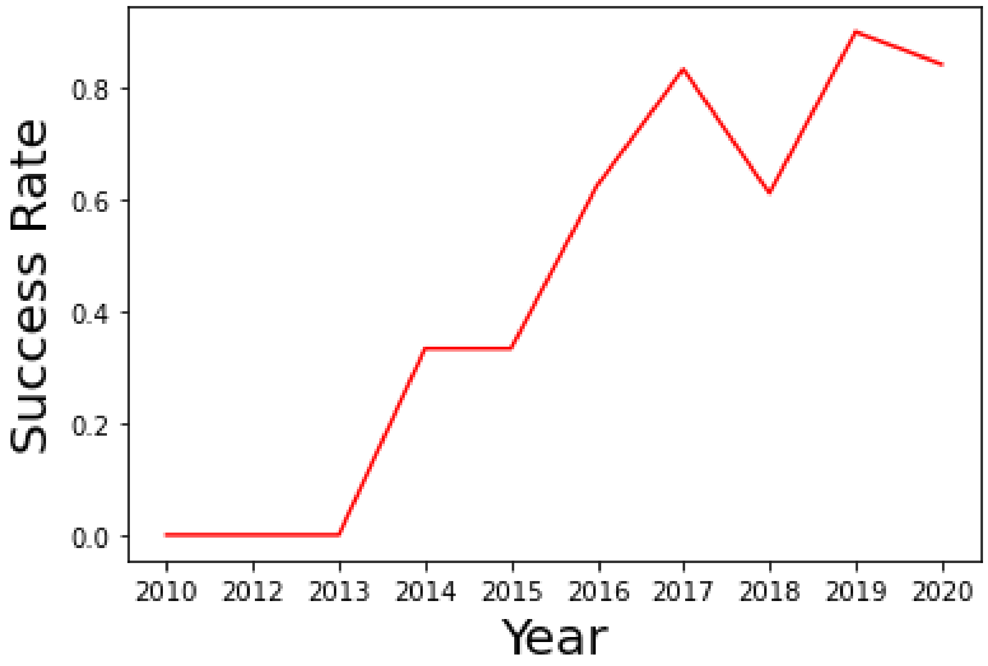
We can see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



We can see that for heavy payloads, the successful landing rate is greater for Polar, LEO and ISS.

Launch Success Yearly Trend



We can see from the line graph above that the success rate for SpaceX has been increasing since 2013.

All Launch Site Names

Distinct keyword is used to return only unique launch site names

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

In [5]:

```
%%sql

SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5
Done.
```

Out[5]:

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

Launch Site Names begin with CCA

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

In [14]:

```
%%sql

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

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[14]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Filtering is done through use of WHERE and LIKE clause along with keyword LIMIT to bring only the top 5 rows.

Total Payload Mass

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

In [26]:

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_) AS "TOTAL PAYLOAD MASS BY NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376,
Done.
```

Out[26]:

TOTAL PAYLOAD MASS BY NASA (CRS)
45596

Aggregation function is used to return the sum of payload mass carried by NASA (CRS).

Average Payload Mass by F9 v1.1

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

In [28]:
```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) AS "AVG PAYLOAD MASS(KG) OF F9 V1.1" FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[28]:

AVG PAYLOAD MASS(KG) OF F9 V1.1
2534

The query is used to return the average mass carried by any F9 v1.1 booster.

First Successful Ground Landing Date

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

Hint:Use min function

In [30]:

```
%%sql

SELECT MIN(DATE) AS "DATE OF FIRST SUCCESSFUL LANDING ON GROUND PAD" FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[30]:

DATE OF FIRST SUCCESSFUL LANDING ON GROUND PAD
2015-12-22

From this query we find out the date of first successful landing.

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

In [31]:

```
%%sql

SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE (LANDING__OUTCOME = 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000);
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[31]:

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

The query returns that boosters that landed successfully carrying a mass between 4000kg and 6000kg. The WHERE and AND clause were used for conditional filtering.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [32]:

```
%%sql

SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS "TOTAL" FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdo
Done.
```

Out[32]:

mission_outcome	total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

The query grouped the data according to the mission outcome and then the different outcomes were counted to give the total number of successful and failed outcomes.

Boosters Carried Maximum Payload

A subquery was used here to find the booster versions that have carried the maximum payload mass.

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

In [33]:

```
%%sql

SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3
Done.
```

Out[33]:

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

In [36]:

```
%%sql

SELECT DATE, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[36]:

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

The query returned the Date, the outcome, the booster version and the launch site for failed landing outcome in the year 2015.

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

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

In [54]:

```
%%sql

SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) COUNTS FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER B
```

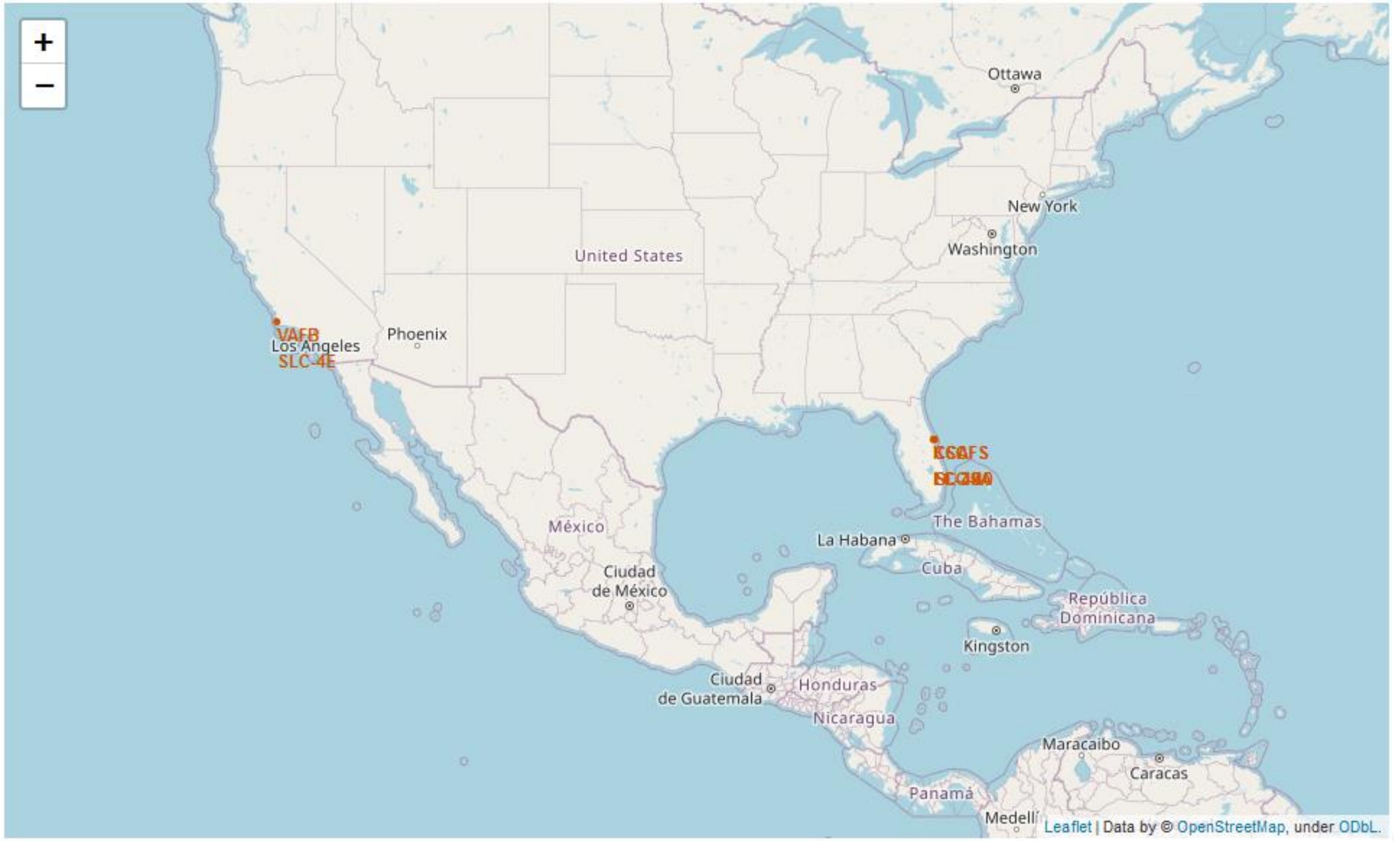
```
 * ibm_db_sa://dzy74444:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[54]:

landing__outcome	counts
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

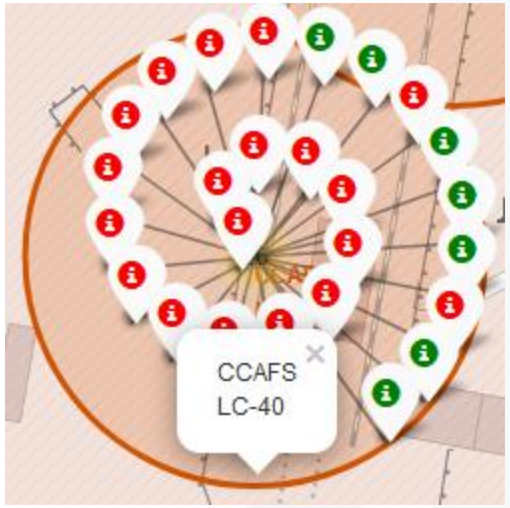
This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

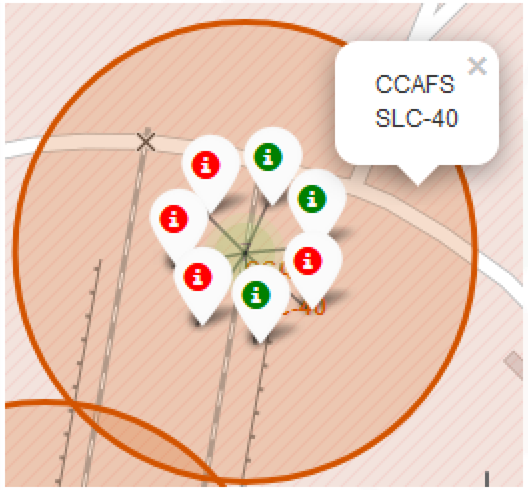
All Launch Sites

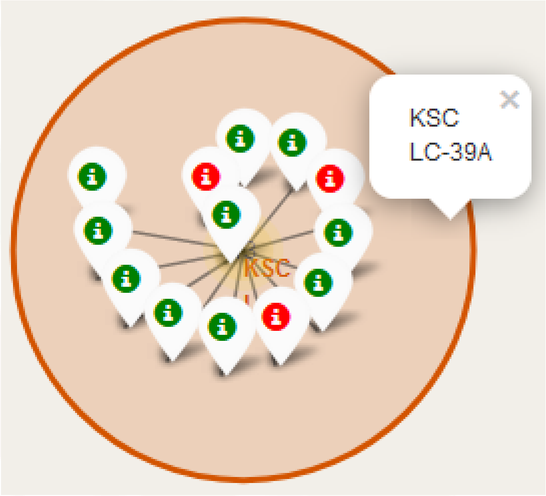


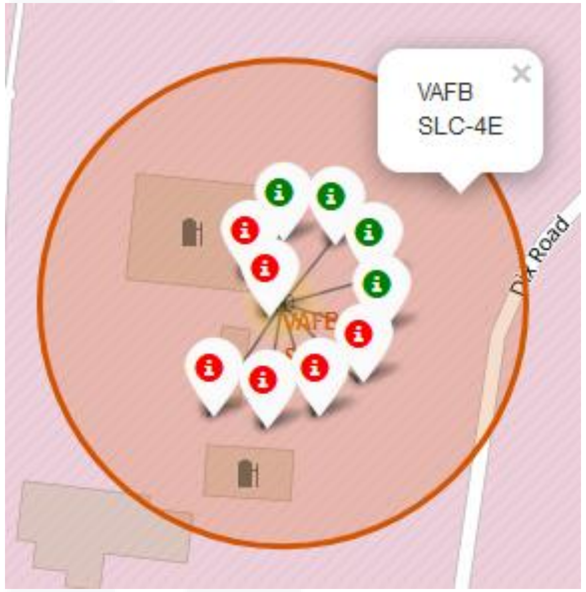
We can see that all the launch sites are on the coast of US.

Color-coded Markers



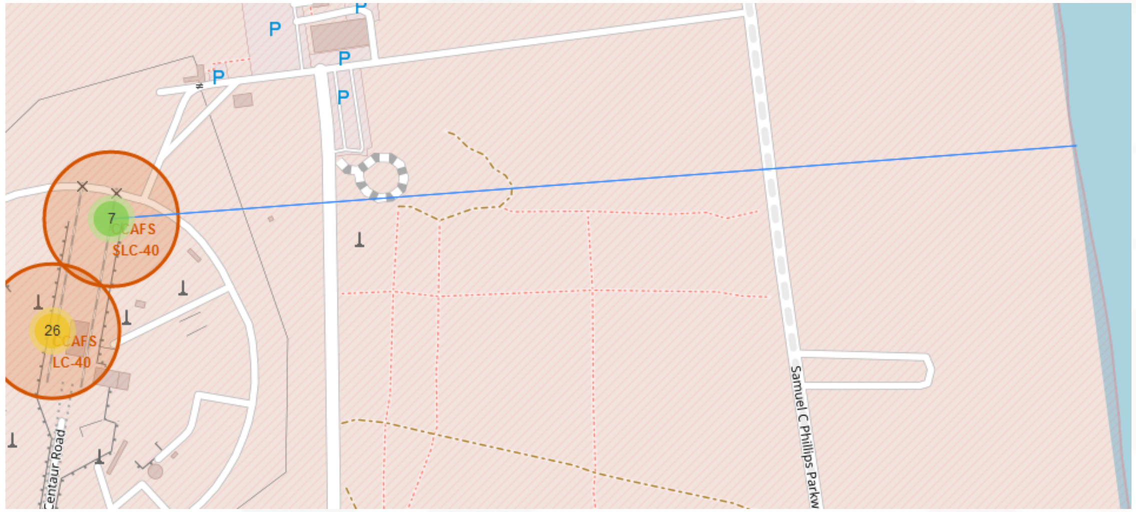


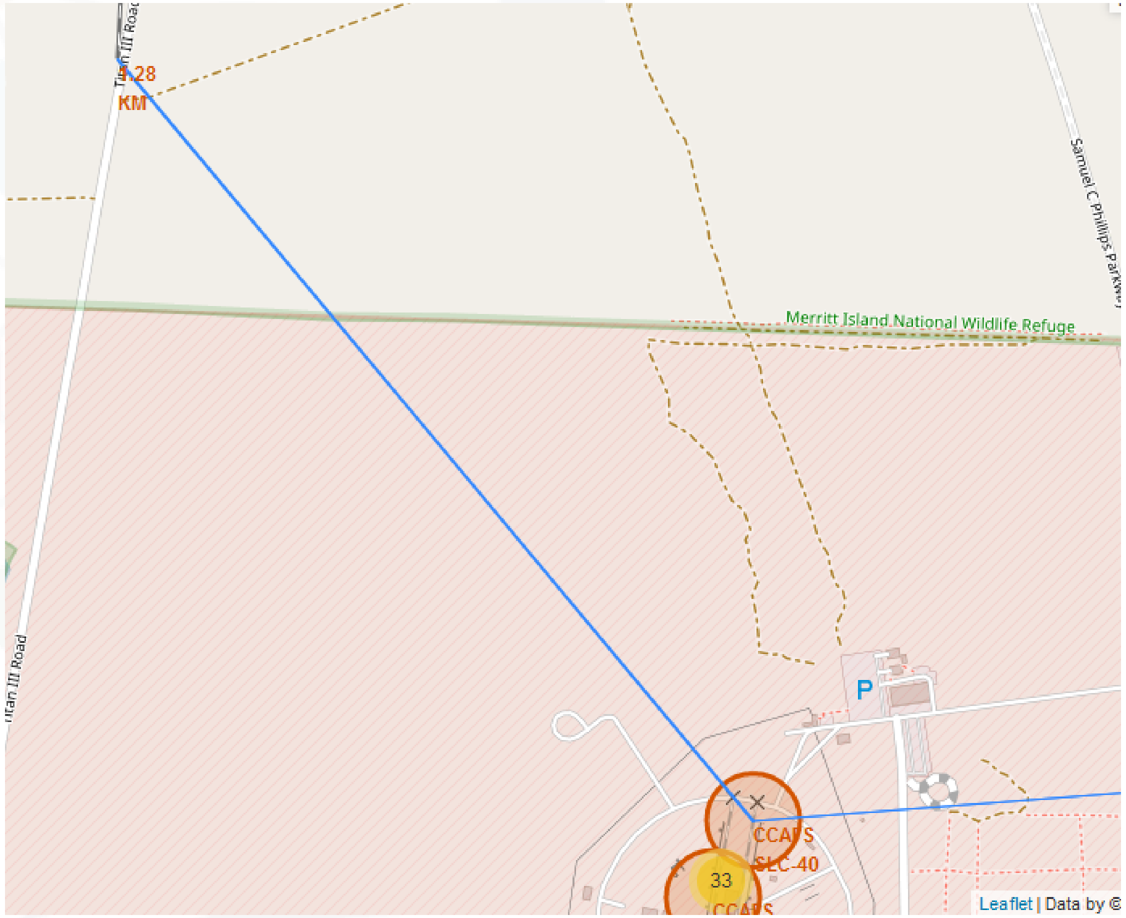




The green markers represent successful landings while the red ones indicate unsuccessful landings. We can see that KSC LC-39A has most successful landings.

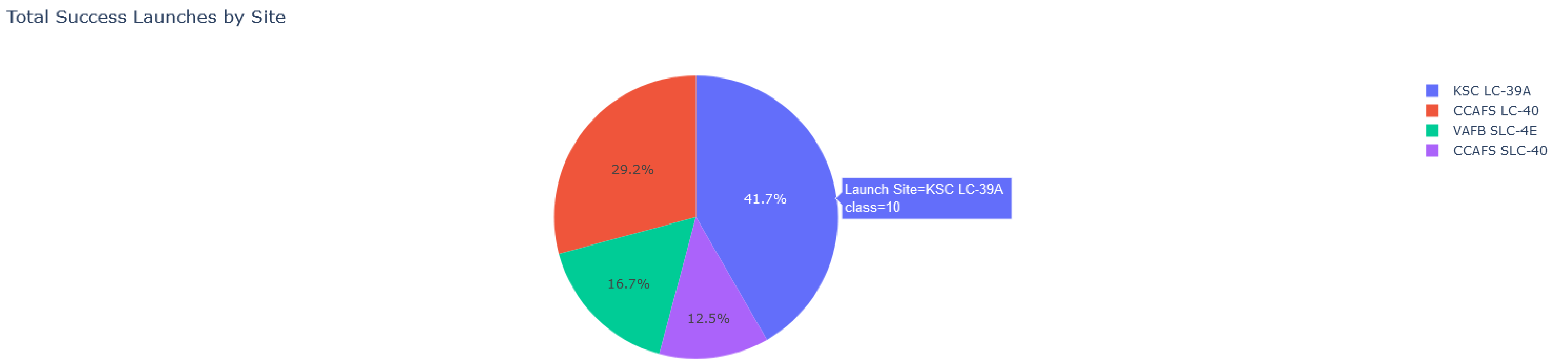
Launch Site distance to Landmarks





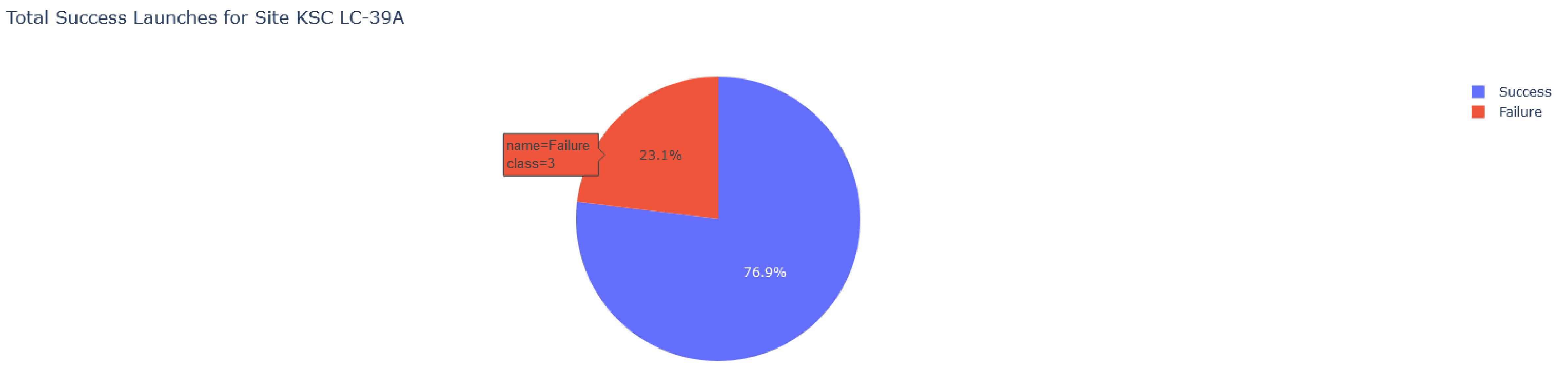
Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

DASHBOARD- Total Success by Sites



We can see that KSC LC-39A has the highest success rate.

DASHBOARD- Total Success of KSC LC-39A

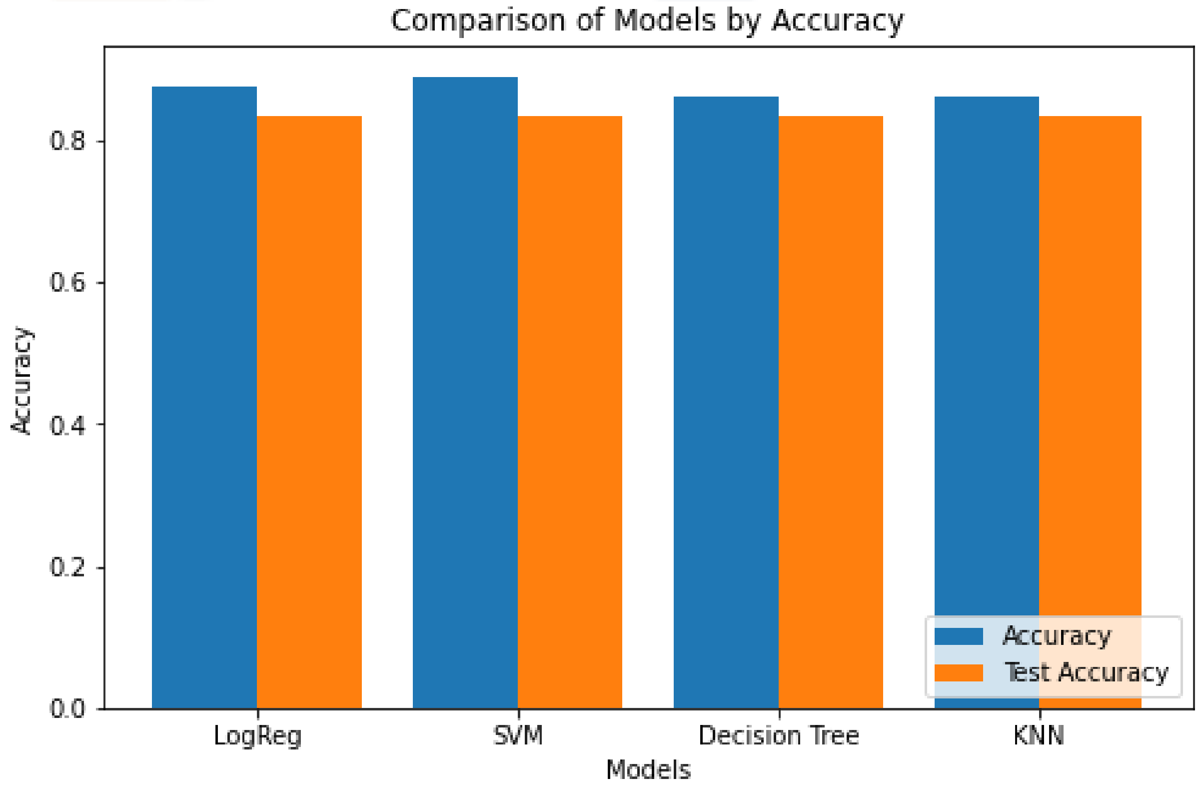


We see that this site has achieved a success rate of 76.9% with 23.1% as failure rate.

DASHBOARD- Payload vs. Outcome

Classification Accuracy

The Support Vector Machine produced the best results with maximum accuracy on both, train as well as test data.



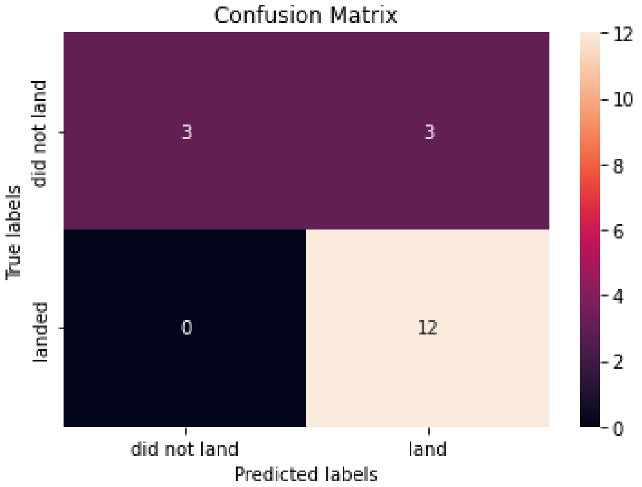
Best parameters of SVM

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

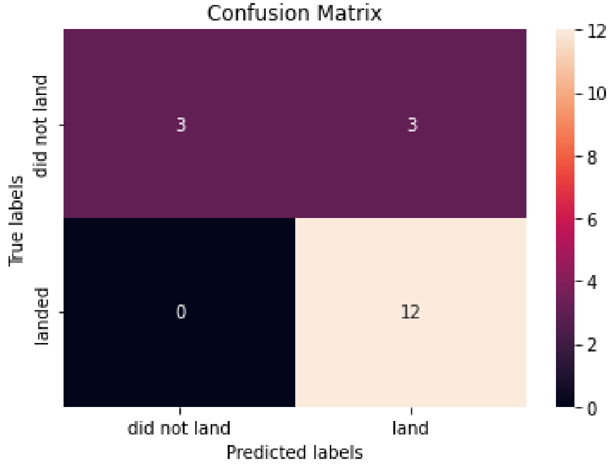
Confusion Matrices

We can see that the test matrices of all models are same. The major problem in all these models is the False Positives

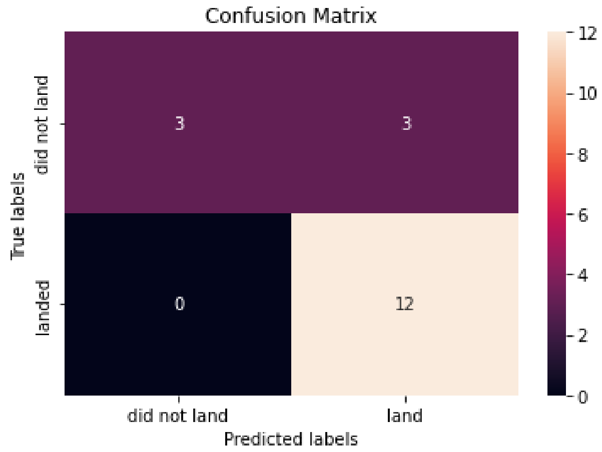
Logistic Regression



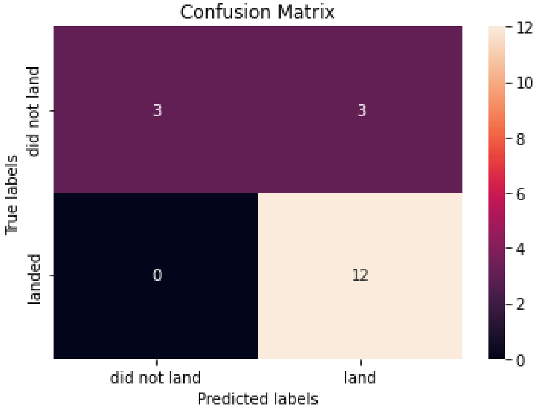
SVM



Decision Tree



KNN



CONCLUSION

- The orbits with best success rate are GEO, HEO, SSO, ES-L1.
- KSC LC-39A had the highest success rate.
- The mission success can be explained by several factors such as orbit type, payload mass, launch site and previous launches as well.
- Generally lower payloads performed better as compared to heavy ones but for certain orbits there is a safe range of load which can successfully land stage 1 with heavier payload as well.
- SVM was chosen as the best model due to its better training accuracy, although all the models had equal test accuracy.

THANK YOU!