IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Lucky Chav
December 25, 2021

Outline

Executive Summary

- Data was collected using API and web scraping
- Data was then cleaned for preparations for exploratory analysis and modeling
- Exploratory data analysis was performed using visualizations and SQL
- Interactive analytics was performed using Folium and Plotly Dash
- Predictive analysis was performed using classification models

- Success rate appears to be related each site as the number of flights increase

- Success rate since 2013 has been increasing

- Orbits with high success rate: ES-L1, GEO, HEO, SSO, VLEO

- All launch sites are in very close proximity to the coast

- Site CCAFS LC-40 has the highest success rate out of all sites, with a success rate of of 73.1%

- FT booster has a relatively high success rate compared to the other boosters within the payload range of 2000 - 5500 kg

- Support Vector Machine model has the highest classification accuracy of about 88.9%

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone.
- Perhaps the most successful is SpaceX with accomplishments that include:
    - Sending spacecraft to the International Space Station.
    - Starlink, a satellite internet constellation providing satellite Internet access.
    - Sending manned missions to Space.
- One reason SpaceX can do this is the rocket launches are relatively inexpensive.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.
- We are taking a role of a data scientist working for a fictional rocket company, Space Y (founded by billionaire industrialist, Allon Musk), and would like to compete with SpaceX.
- Problems to derive insights from:
    - Determine the price of each launch.
    - Determine if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Launch data was collected with the SpaceX Rest API and also by web scraping related Wiki Pages
- Perform data wrangling
  - A plethora of processes were conducted to clean the data in preparations for analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different Classifications models were built, tuned, and evaluated

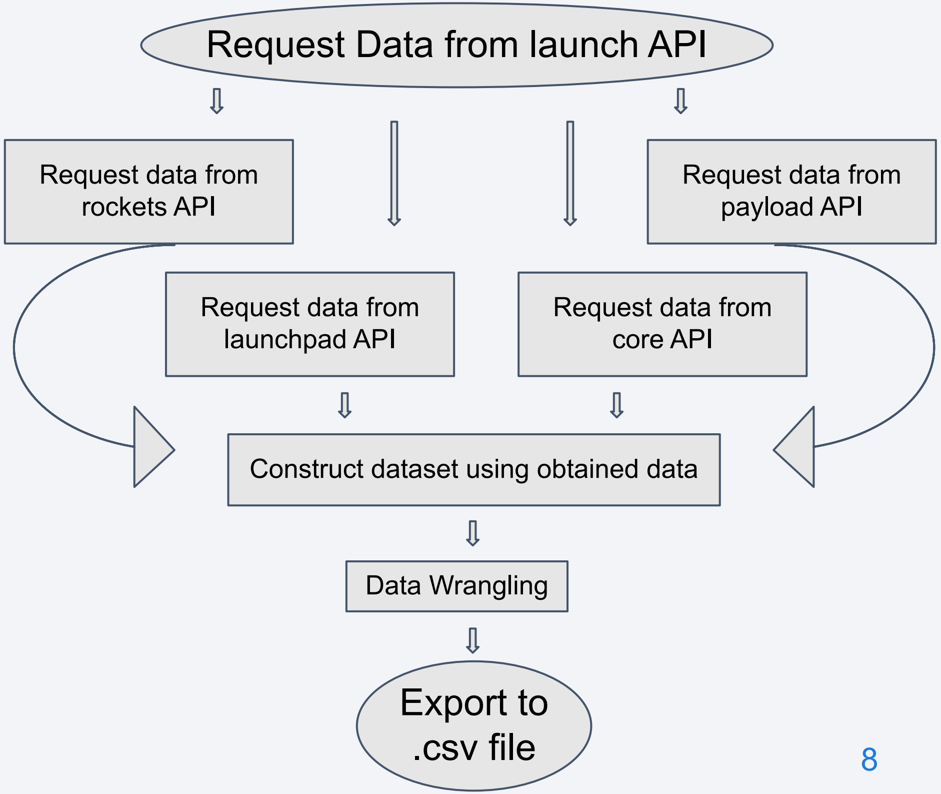
Data Collection

- SpaceX launch data was collected with the SpaceX Rest API.
- This API gave us data about launches, including information on the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Falcon 9 Launch data was also collected by web scraping related Wiki pages.
- Python BeautifulSoup package was used to web scrape some HTML tables that contain valuable Falcon 9 launch records.

Data Collection – SpaceX API

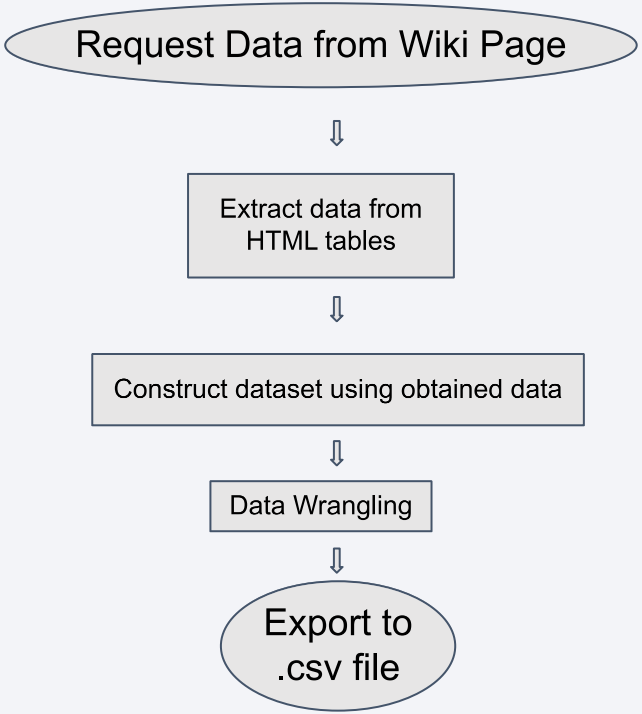
- The SpaceX REST API endpoint, or URL, worked with was: api.spacexdata.com/v4/launches/past

- GitHub URL of the completed SpaceX API calls notebook as an external reference and for peer-review purposes: (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Data%20Collection%20API.ipynb)



Data Collection - Scraping

- Python BeautifulSoup package was used to web scrape some HTML tables that contain valuable Falcon 9 launch records.
- GitHub URL of the completed web scraping notebook as an external reference and for peer-review purposes: (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb)



Data Wrangling

- Replaced Missing values of Payload Mass with mean.
- Created new column called ‘class’ to determine success rate of launches
- Reset index for ‘Flight Number’ column
- One Hot Encoding - Created dummy variables for categorical columns
- Cast all numeric columns as 'float64’
- Standardized data for classification analysis

GitHub URLs of data wrangling related notebooks as an external reference and peer-review purposes:

- (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Data%20Collection%20API.ipynb)
- (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/EDA.ipynb)
- (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/EDA%20with%20Data%20Visualization.ipynb)
- (https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Machine%20Learning%20Prediction.ipynb)

EDA with Data Visualization

The following charts were plotted to get some preliminary insights about how each variable would affect the success rate:

- Flight Number vs. Payload
- Flight Number vs. Launch Site
- Flight Number vs. Orbit Types
- Success Rate of Orbit Types
- Payload vs. Orbit Types
- Payload vs Launch Site
- Launch Success Yearly Trend

GitHub URL of EDA with data visualization notebook as an external reference and peer-review purpose:
(https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/EDA%20with%20Data%20Visualization.ipynb)

EDA with SQL

The following SQL queries were performed:

- Finding names of unique launch sites
- Finding records where launch sites begin with “CCA”
- Finding total payload carried by boosters launched by NASA (CRS)
- Finding average payload carried by booster version F9 v1.1
- Finding date of the first successful landing on a ground pad
- Finding successful drone ship boosters with payloads between 4000 - 6000(kg)
- Finding total number of successful and failure mission outcomes
- Finding name of boosters which have carried the maximum payload mass
- Finding boosters of failed drone ship outcomes in 2015
- Finding landing outcomes by count between 2010-06-04 to 2017-03-2020

GitHub URL of completed EDA with SQL notebook as an external reference and peer-review purposes:

(https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/EDA%20with%20SQL.ipynb)

Build an Interactive Map with Folium

- All launch sites were added to a folium map, with circles and markers to indicate launch sites as a highlighted area with a text label.
- Successful and failed launch outcomes were then marked for each site to see which sites have high success rates. Red markers to indicate successful launches and green to indicated failed launches.
- Marker clusters were used to simplify the map containing many markers having the same coordinate.
- Lines were added to show distance between a launch site and its proximities to points of interests.
- The launch success rate may also depend on the location and the proximities of a launch site. (i.e., the initial position of rocket trajectories)
- Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some factors by analyzing the existing site locations.

GitHub URL of completed interactive map with Folium map as an external reference and peer-review purposes:
(https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb)

Build a Dashboard with Plotly Dash

The following plots/graphs and interactions were added to a dashboard:

- Pie chart showing count of successful launches for all launch sites
- If a specific launch site was selected, the Pie chart will show the count of successful vs failed launches for the site
- Scatter graph to show the correlation between payload and launch success for all sites
- A slider was also added to select payload range

GitHub URL for external reference and peer-review purposes:
(https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/spacex_dash_app.py)

Predictive Analysis (Classification)

- The data was split into sets, 80% for training and 20% for testing
- The following models were built using the training set: Logistic Regression, Support Vector Machine, Decision Tree, K Nearest Neighbors
- Hyperparameters were found and fitted for each model and accuracy was calculated on test data.
- Confusion matrices were also plotted for each model to distinguish different classes
- Best performing classification model was found by comparing best score and accuracy.

Split data into training/test ⇨ Build models with training data ⇨ Find best parameters ⇨ Evaluate Accuracy with test data ⇨ Compare models

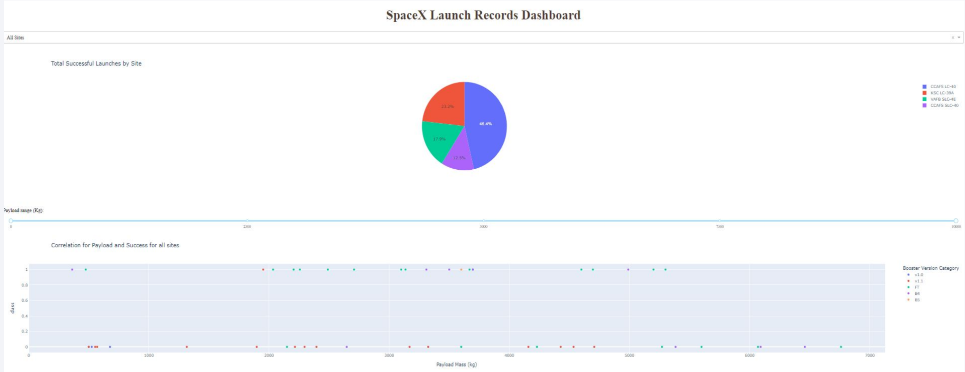
- GitHub URL for external reference and peer-review purposes:
(https://github.com/Laidbackluck/IBM-Data-Science-Project-Repository/blob/master/Machine%20Learning%20Prediction.ipynb)

Results

In [11]:
```python
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs',
features.head()
```

Out[11]:

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0003
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0005
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0007
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	NaN	1.0	0	B1003
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B1004

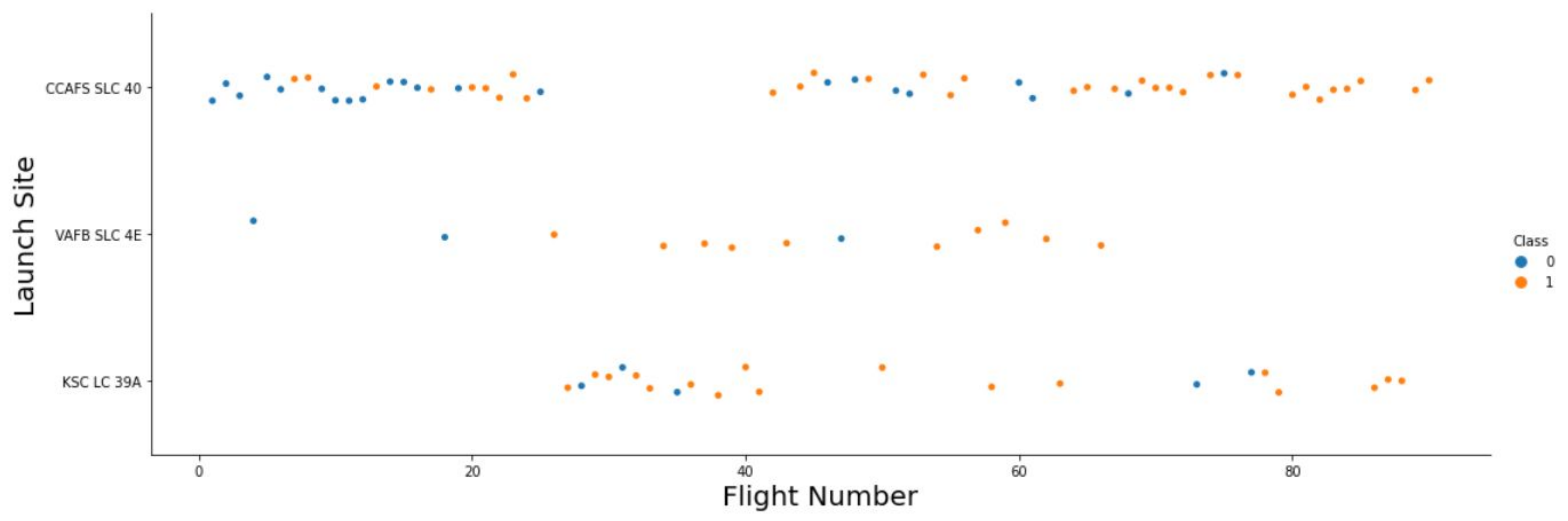



```
Log Reg Best Score: 0.8464285714285713
Log Reg Accuracy: 0.875

SVM Best Score: 0.8482142857142856
SVM Accuracy: 0.8888888888888888

Decision TreeBest Score: 0.8892857142857145
Decision TreeAccuracy: 0.8472222222222222

KNN Best Score: 0.8482142857142858
KNN Accuracy: 0.8611111111111112
```

Section 2

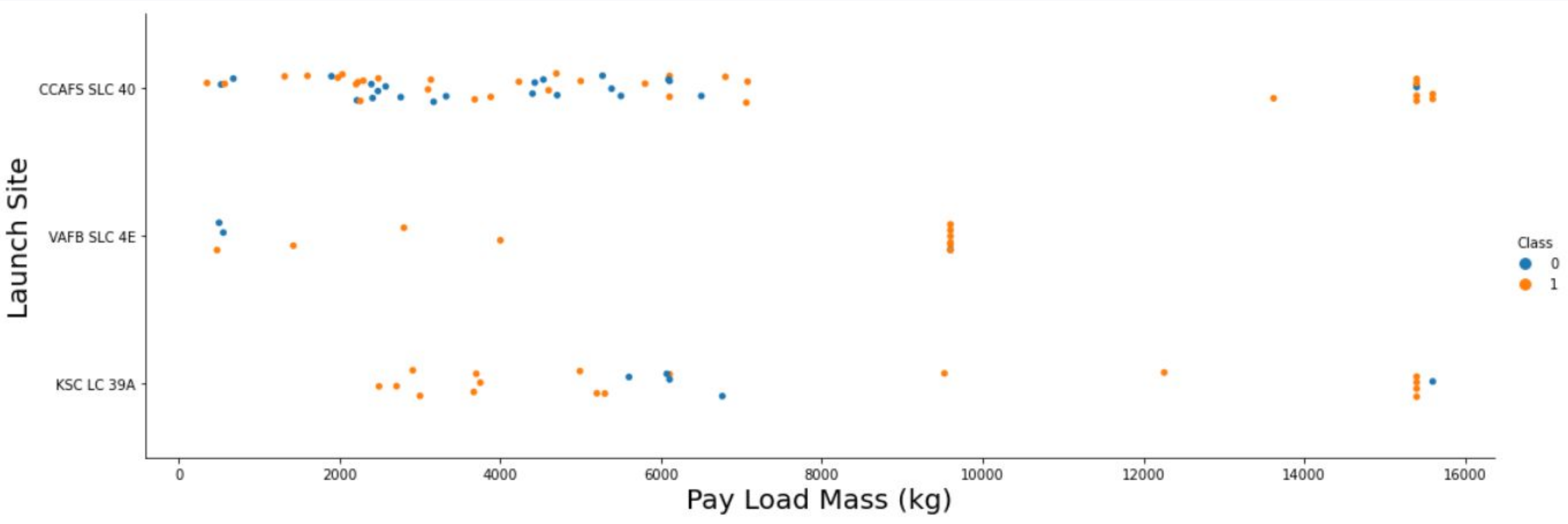
Insights drawn from EDA

Flight Number vs. Launch Site



- Success rate appears to be related each site as the number of flights increase

Payload vs. Launch Site

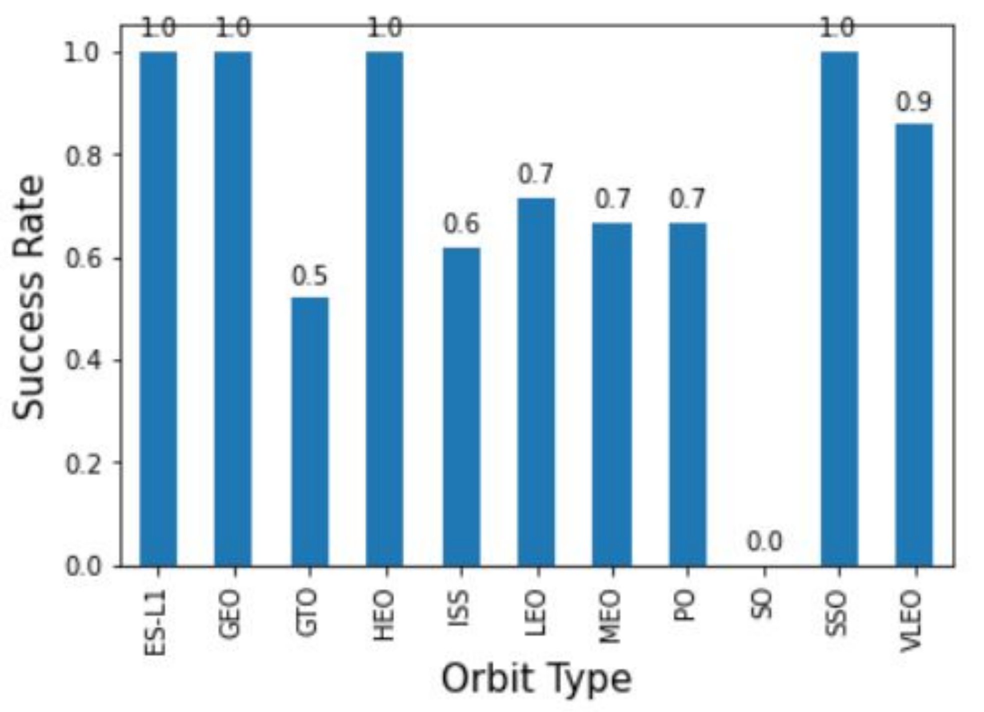


- For the VaFB SLC 4E site, we can see that there are no rocket launched for heavy payload mass (greater than 10,000kg)

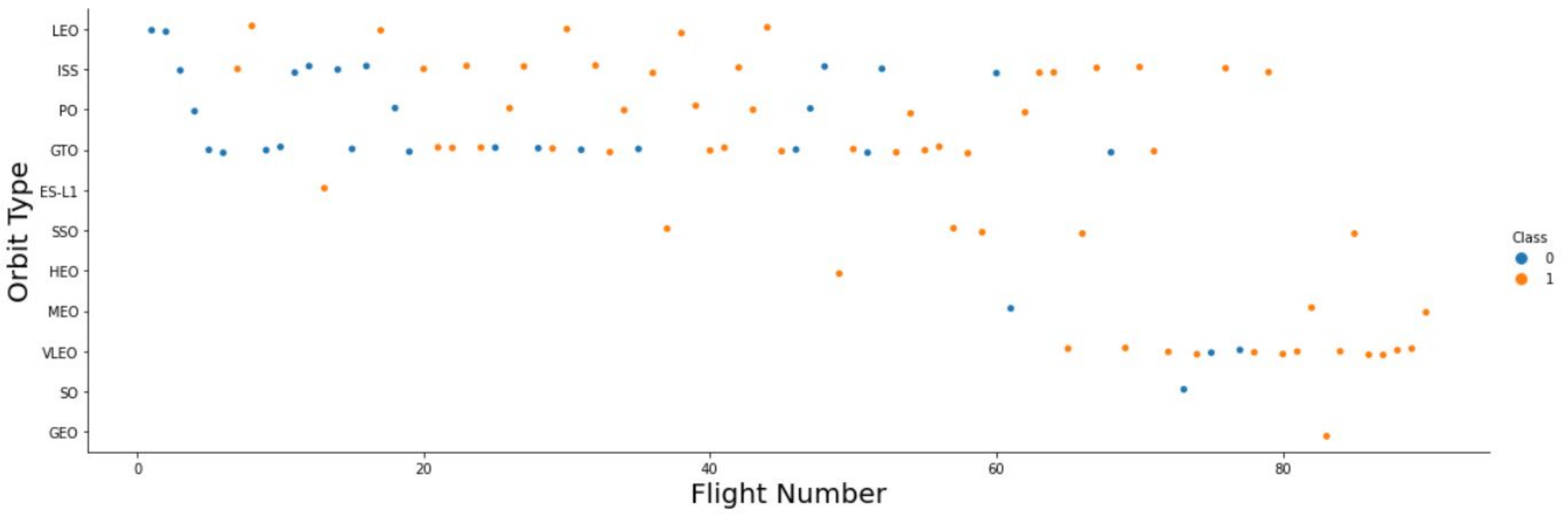
Success Rate vs. Orbit Type

Orbits with high success rate:

- ES-L1
- GEO
- HEO
- SSO
- VLEO

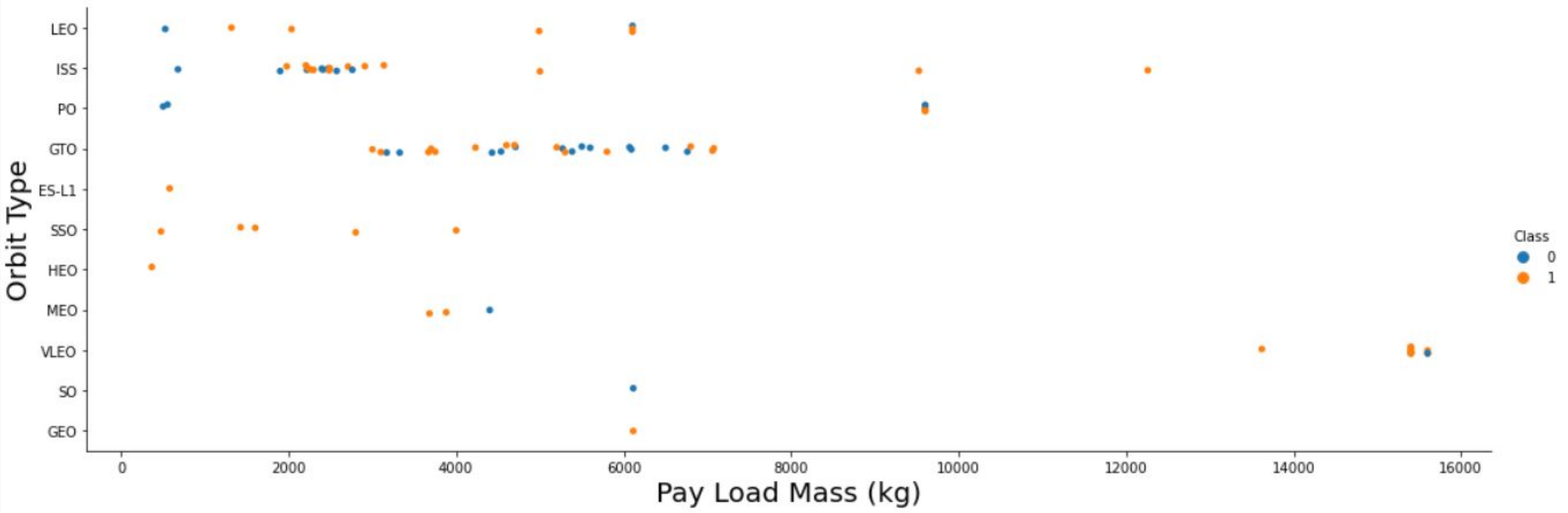


Flight Number vs. Orbit Type



- The LEO orbit 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

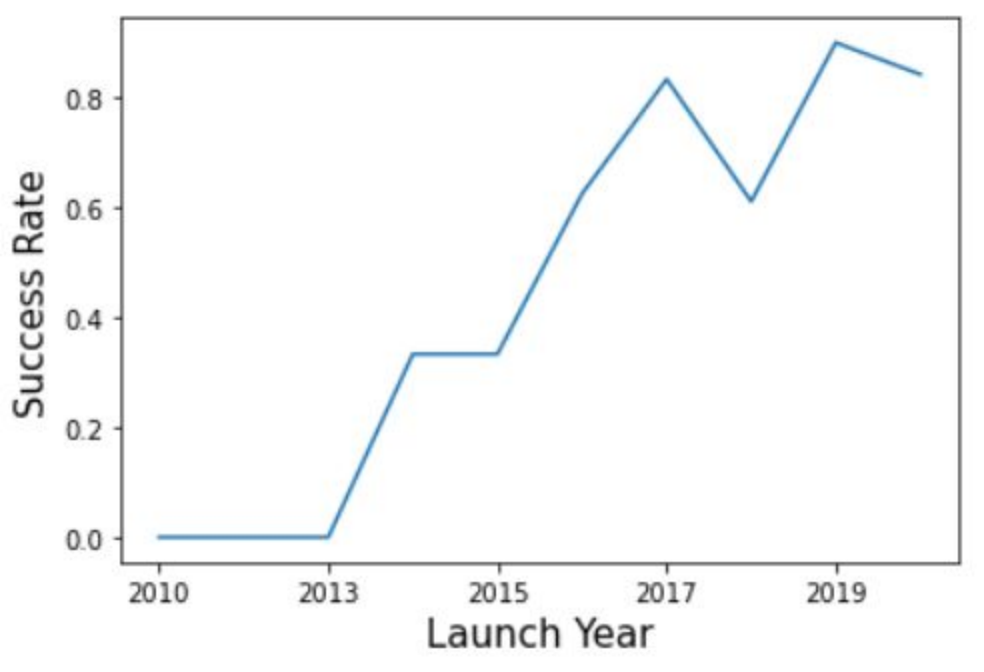
Payload vs. Orbit Type



- With heavy payloads, the successful landing, or positive landing rate are more for Polar, LEO, and ISS
- However, for GTO, we cannot distinguish this well as both positive landing rate and negative landing are both there

Launch Success Yearly Trend

- Success rate since 2013 has been increasing



All Launch Site Names

The unique launch site names are:

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

In [4]:

```
%%sql
SELECT
    DISTINCT launch_site
FROM
    SPACEXDATASET
;
```

Out[4]:

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

Launch Site Names Begin with 'CCA'

In [5]:

```
%%sql
SELECT
    *
FROM
    SPACEXDATASET
WHERE
    launch_site LIKE 'CCA%'
LIMIT
    5
;
```

Showing the first 5 results of Names beginning with “CCA”

```
 * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/BLUDB
Done.
```

Out[5]:

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

Total Payload Mass

```
In [6]:  %%sql
         SELECT
             SUM(payload_mass__kg_) AS "Total 'NASA (CRS)' Payload Mass "
         FROM
             SPACEXDATASET
         WHERE
             customer = 'NASA (CRS)'
         ;

          * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/BLUDB
         Done.
```

Out[6]:

Total 'NASA (CRS)' Payload Mass
45596

- The total payload carried by boosters from NASA is 45,596 k.g.

Average Payload Mass by F9 v1.1

In [7]:
```
%%sql
SELECT
    AVG(payload_mass__kg_) AS "Average payload mass carried by booster version F9 v1.1"
FROM
    SPACEXDATASET
WHERE
    booster_version = 'F9 v1.1'
;
```

```
 * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/BLUDB
Done.
```

Out[7]:

Average payload mass carried by booster version F9 v1.1
2928

- The average payload mass carried by booster version F9 v1.1 is 2,928 k.g.

First Successful Ground Landing Date

```
In [8]:
%%sql
SELECT
    min(DATE) AS "Date of first successful landing outcome in ground pad"
FROM
    SPACEXDATASET
WHERE
    landing__outcome = 'Success (ground pad)'
;

 * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/BLUDB
Done.
```

Out[8]:

Date of first successful landing outcome in ground pad
2015-12-22

- The date of the first successful landing outcome on ground pad is December 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:
- F9 FT B1021.2
- F9 FT B1031.2
- F9 FT B1022
- F9 FT B1026

In [9]:
```
%%sql
SELECT
    DISTINCT booster_version, payload_mass__kg_, landing__outcome
FROM
    SPACEXDATASET
WHERE
    payload_mass__kg_ > 4000 AND
    payload_mass__kg_ < 6000 AND
    landing__outcome LIKE 'Succ%' AND
    landing__outcome LIKE '%drone%'
;
```

```
 * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1c
Done.
```

Out[9]:

booster_version	payload_mass__kg_	landing__outcome
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

In [10]:

```
%%sql
SELECT
    DISTINCT Mission_outcome,
    Count(Mission_outcome) AS "Count"
FROM
    SPACEXDATASET
GROUP BY
    mission_outcome
;
```

```
 * ibm_db_sa://mmw60838:***@fbd88901-ebdb
Done.
```

Out[10]:

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass

```
In [11]:
%%sql
SELECT
    DISTINCT booster_version,
    payload_mass__kg_
FROM
    SPACEXDATASET
WHERE payload_mass__kg_ = (
    SELECT
        MAX(payload_mass__kg_)
    FROM
        SPACEXDATASET
)
;
```

Out[11]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [12]:

```
%%sql
SELECT
    landing__outcome,
    booster_version,
    launch_site,
    DATE
FROM
    SPACEXDATASET
WHERE
    landing__outcome LIKE 'Fail%' AND
    landing__outcome LIKE '%drone%' AND
    YEAR(DATE) = 2015
;
```

```
 * ibm_db_sa://mmw60838:***@fbd88901-ebdb-4a4f-a32e-9
Done.
```

Out[12]:

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

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

- Rank of 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 [13]:
```
%%sql
SELECT
    landing__outcome AS "Landing Outcomes (2010-06-04 - 2017-03-20)",
    COUNT(*) AS "Count"
FROM
    SPACEXDATASET
WHERE
    DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    landing__outcome
ORDER BY
    COUNT DESC
;
```

Out[13]:

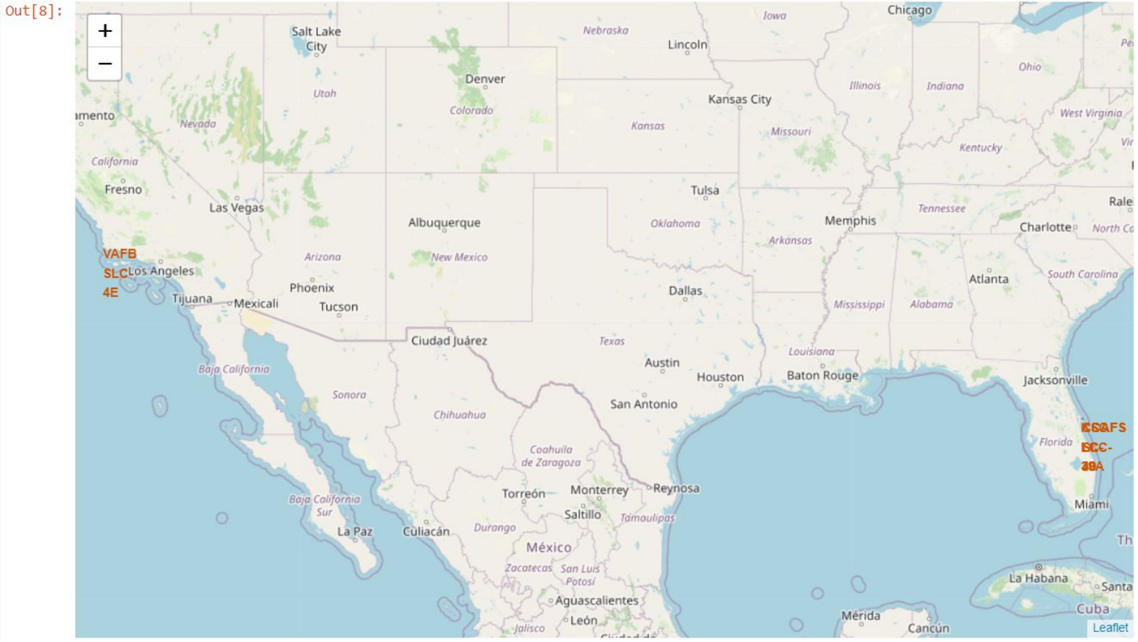
Landing Outcomes (2010-06-04 - 2017-03-20)	Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

Folium Map of all launch sites

- All launch sites are not in proximity to the Equator line
- All launch sites are in very close proximity to the coast

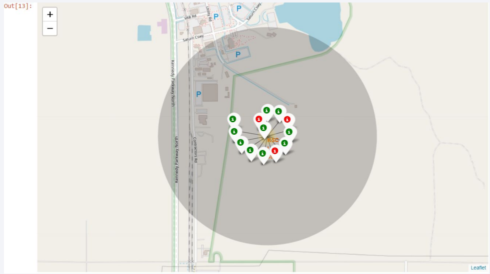


Folium Map of Launch Outcomes

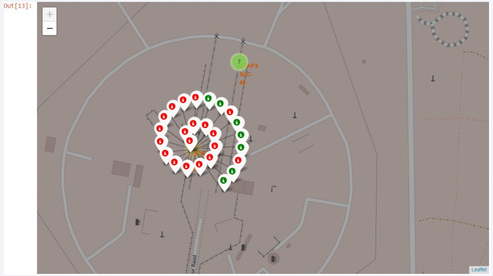
- From the color-labeled markers, we identify launch site KSC - LC39A has a relatively high success rates

VAFB SLC-4E

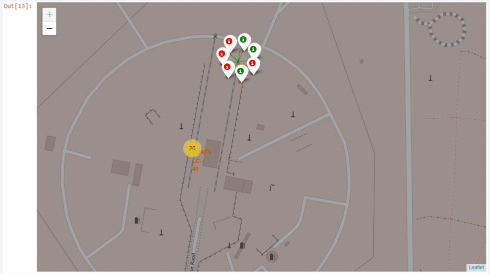
KSC LC-39A



CCAFS LC40



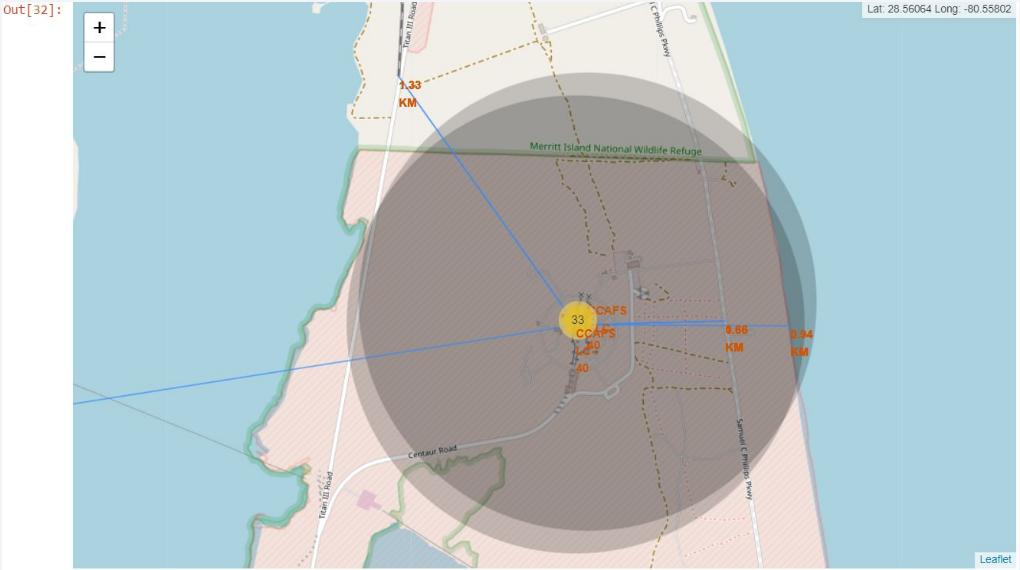
CCAFS SLC40



Folium Map of Proximities to CCAFS LC-40

Proximities from launch site CCAFS LC-40:

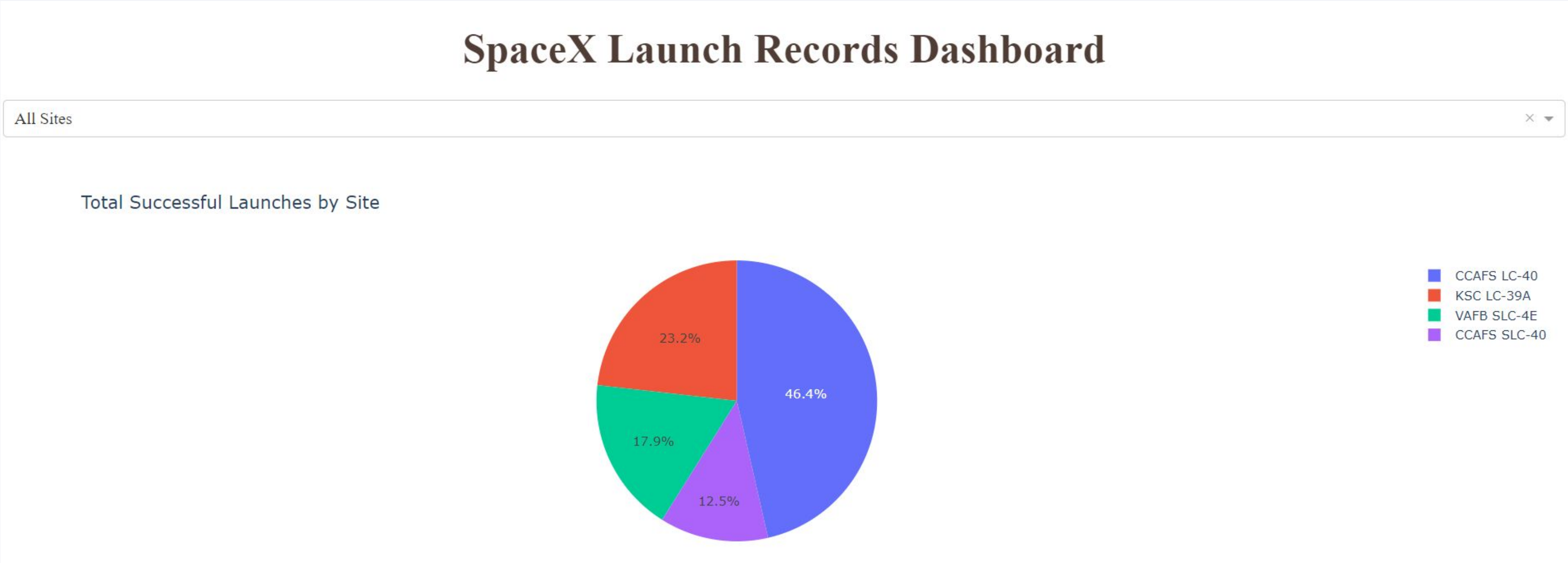
- Railway - 1.33 km
- Highway - 0.66 km
- Coastline 0.94 km
- City - 19.89 km



Section 5

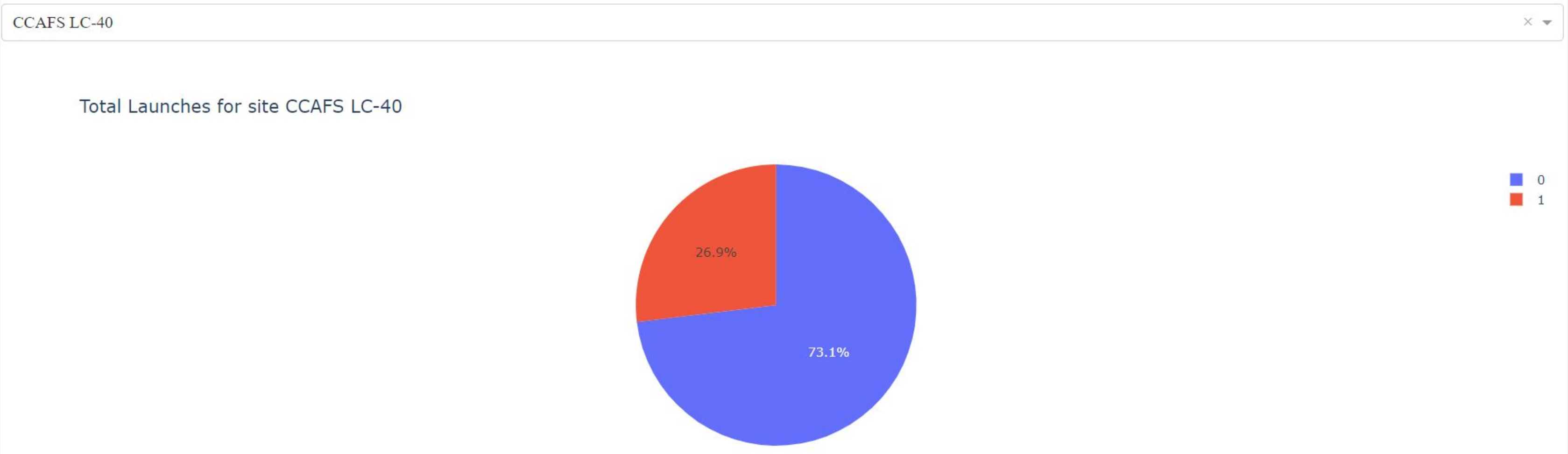
Build a Dashboard with Plotly Dash

Dashboard - Launch Success for All Sites



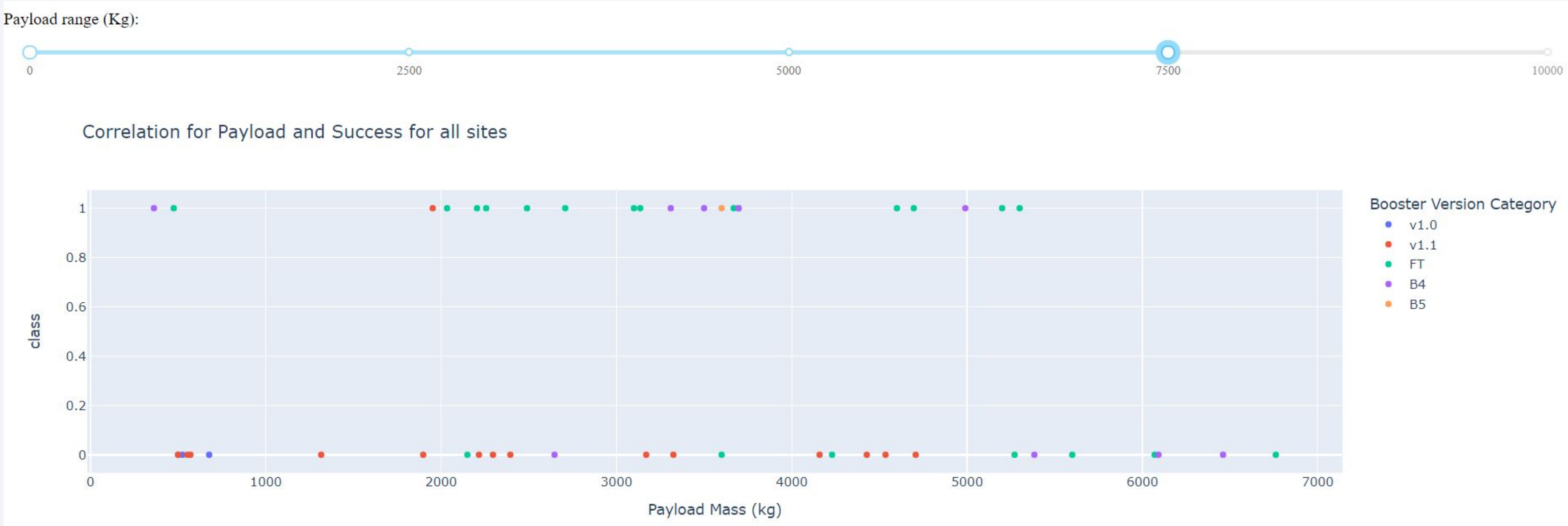
- From the pie chart, we can identity that site CCAFS LC-40 has the highest success rate of 46.4%

Dashboard - Launch Success for CCAFS LC-40



- From the pie chart, we can can see that launch site CCAFS LC-40 has a success rate of 73.1%

Dashboard - Payload vs. Launch Outcomes for All Sites



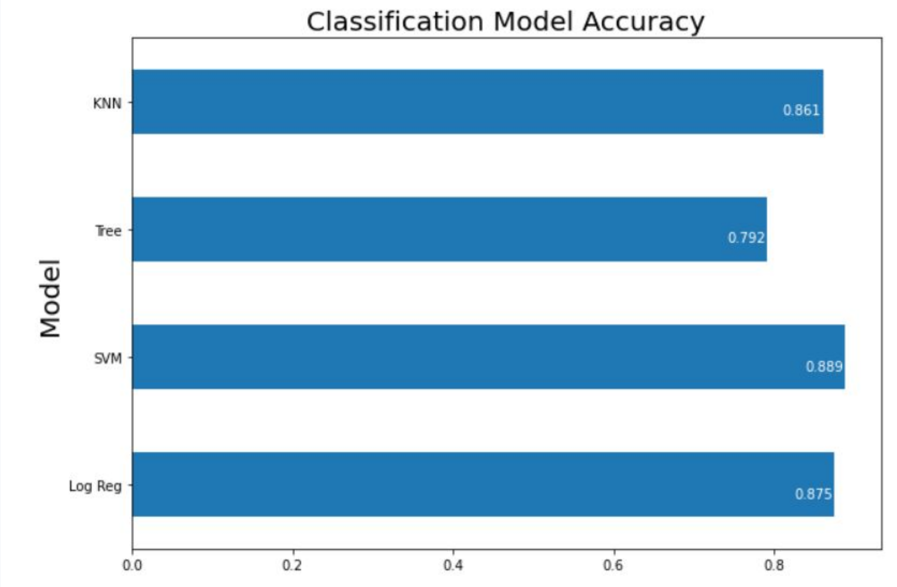
- From the scatter graph, we can identity that the FT booster has a relatively high success rate compared to the other boosters within the payload range of 2000 - 5500 kg

Section 6

Predictive Analysis (Classification)

Classification Accuracy

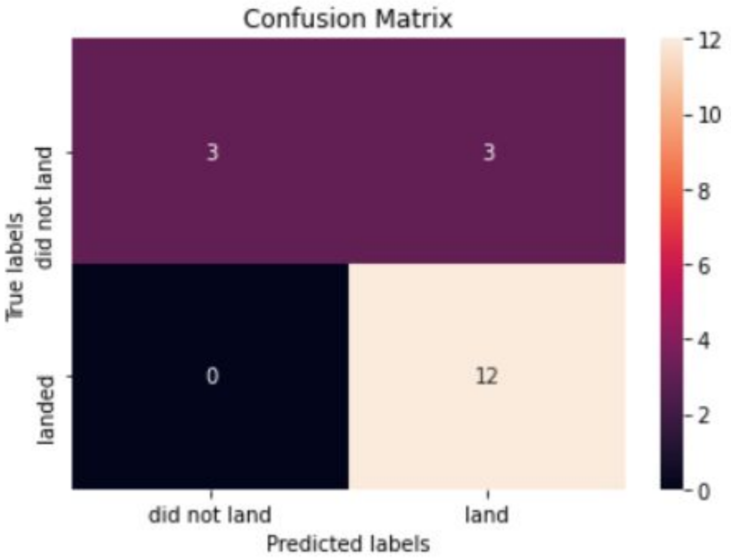
- From the bar chart, we can identify that the Support Vector Machine model has the highest classification accuracy of about 88.9%



Confusion Matrix

- Examining the confusion matrix, we see that Supply Vector Machine can distinguish between the different classes. We see that the major problem is false positives

In [25]:
```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Success rate appears to be related each site as the number of flights increase
- Success rate since 2013 has been increasing
- Orbits with high success rate: ES-L1, GEO, HEO, SSO, VLEO
- All launch sites are in very close proximity to the coast
- Site CCAFS LC-40 has the highest success rate out of all sites, with a success rate of of 73.1%
- FT booster has a relatively high success rate compared to the other boosters within the payload range of 2000 - 5500 kg
- Support Vector Machine model has the highest classification accuracy of about 88.9%

Appendix

```
In [31]: accuracy_data = {
             'Model': ['Log Reg', 'SVM', 'Tree', 'KNN'],
             'Best Score': [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_],
             'Accuracy': [logreg_cv.score(X_train, Y_train), svm_cv.score(X_train, Y_train), tree_cv.score(X_train, Y_train), knn_cv.score(X_train, Y_train)]
         }
         accuracy_df = pd.DataFrame(data=accuracy_data)
         accuracy_df.set_index('Model', inplace = True)
         accuracy_df
```

Out[31]:

Model	Best Score	Accuracy
Log Reg	0.846429	0.875000
SVM	0.848214	0.888889
Tree	0.903571	0.791667
KNN	0.848214	0.861111

Thank you!