IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Carlos Guzman

12-30-2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with API and via Web Scrapping
  - Data Wrangling
  - EDA with SQL, Python and Data Visualization
  - Interactive Map Visualization with Folium
  - Interactive Dashboard with Dash and Plotly
  - Predictive Analysis with Machine Learning
- Summary of all results
  - EDA summary of results
  - Interactive analytics via Visualizations
  - Predictive Analysis with Machine Learning

Introduction

- Summary of methodologies

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.

- Summary of all results

Objective is to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Via SpaceX REST API
  - Web Scrapping with BeautifulSoup from Wikipedia
- Perform data wrangling
  - Replaced missing values with mean, and 0 for non-relevant fields, one hot encoding for data normalization
- Perform exploratory data analysis (EDA) using Data Visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using machine learning with classification models
  - Data validation, Model building, and Model evaluation for Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbours

Data Collection

- The data was collected using two methods in order to get an additional asset with more specific and also additional information to keep for further analysis:
  - Data collection of SpaceX Falcon 9 launch records was performed by making a get request to the SpaceX REST API and then extracting text data to store it into a pandas data frame.
  - Data collection of SpaceX for specific Falcon 9 launch records was performed by making web scraping to Wikipedia and then convert it to a pandas data frame.

Data Collection – SpaceX API

- Starting with the get request to the SpaceX REST API we get and check the response as well as status code.
- The content response was decoded using .json() and .json_normalize() to convert it to a pandas data frame.
- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

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

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

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork
```

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

In [10]:
```
response.status_code
```

Out[10]: 200

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

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response_alt = requests.get(static_json_url)
json_response = response_alt.json()
data = pd.json_normalize(json_response)
```

In [12]:
```
# Get the head of the dataframe
data.head()
```

Out[12]:

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[
								Successful first stage	

Data Collection - Scraping

- Data was collected by applying web scrapping to Wikipedia using BeautifulSoup.

- Data was parsed from tables and converted to a pandas data frame

- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/jupyter-labs-webscraping.ipynb

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

In [18]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated Universal Tim
e">UTC</a>)
```

In [24]:
```
df = pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

In [25]:
```
df.head()
```

Out[25]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43

Data Wrangling

- The exploratory data analysis was performed and dataset was cleaned using python and several libraries.
- Data types were fixed to make them compatible with ML algorithms.
- Null values were replaced.
- The data was filtered to only preserve Falcon 9 launches.
- Descriptive analysis to identify relevant features to calculate the dependent variable.

In [18]:

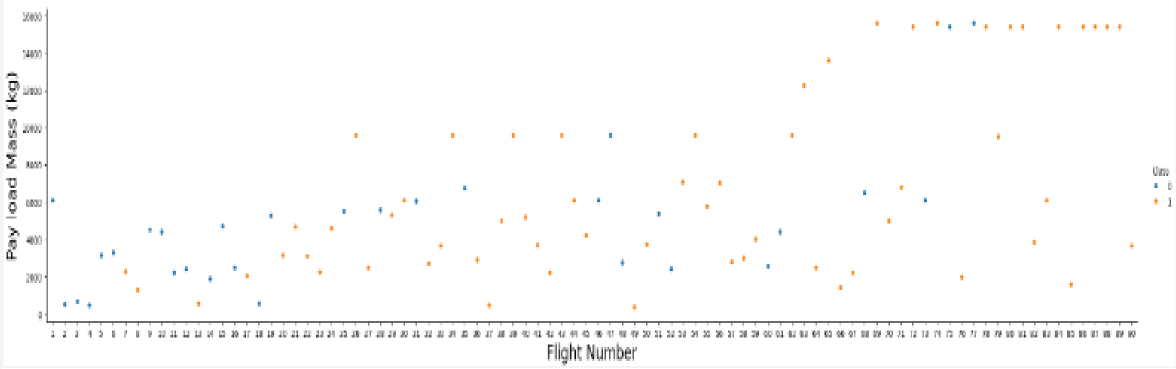
```
df.head(5)
```

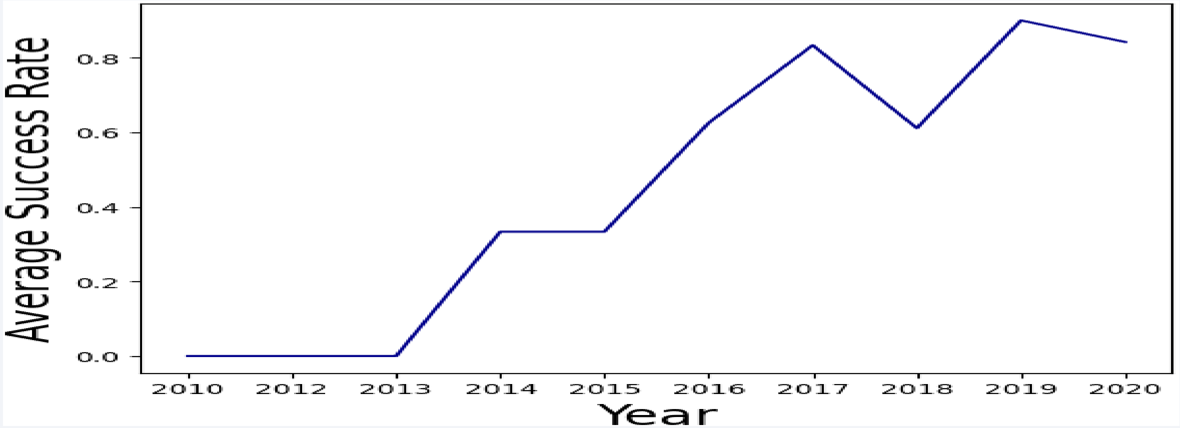
Out[18]:

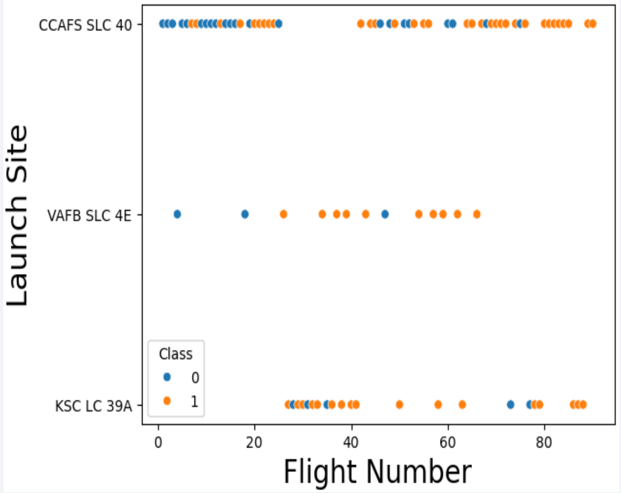
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False

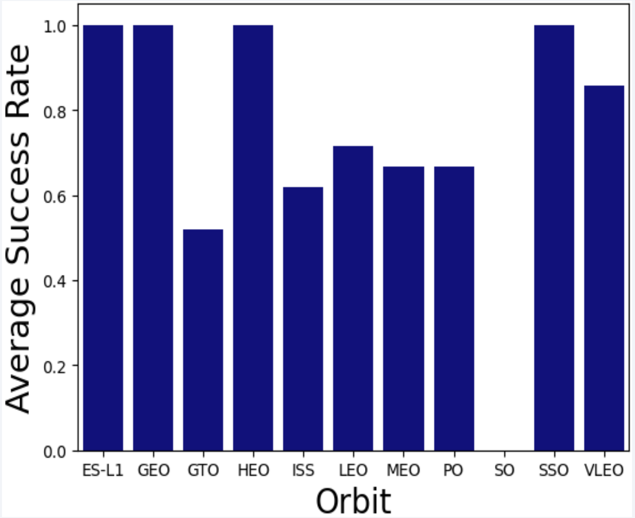
https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization







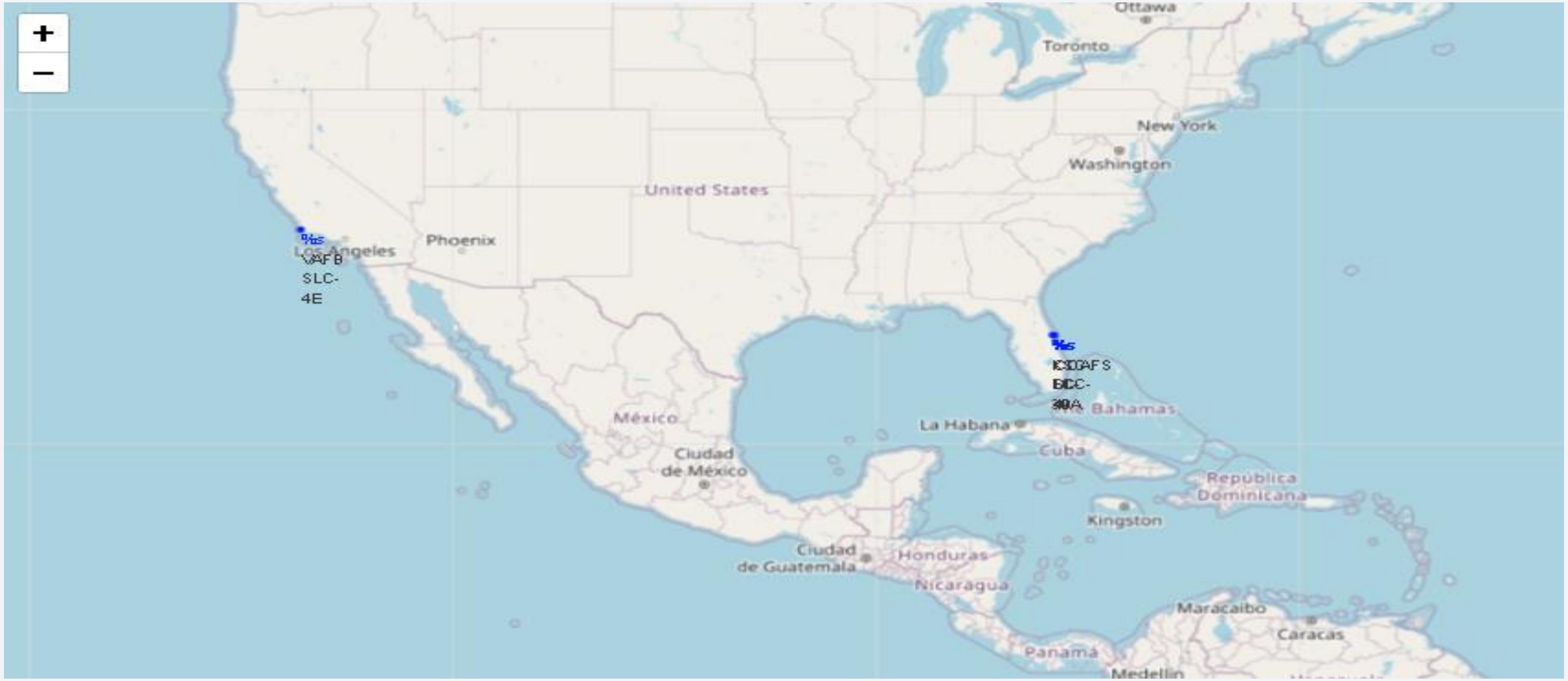


- Exploratory data analysis was performed by using data visualization with Matplotlib and Seaborn.
- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

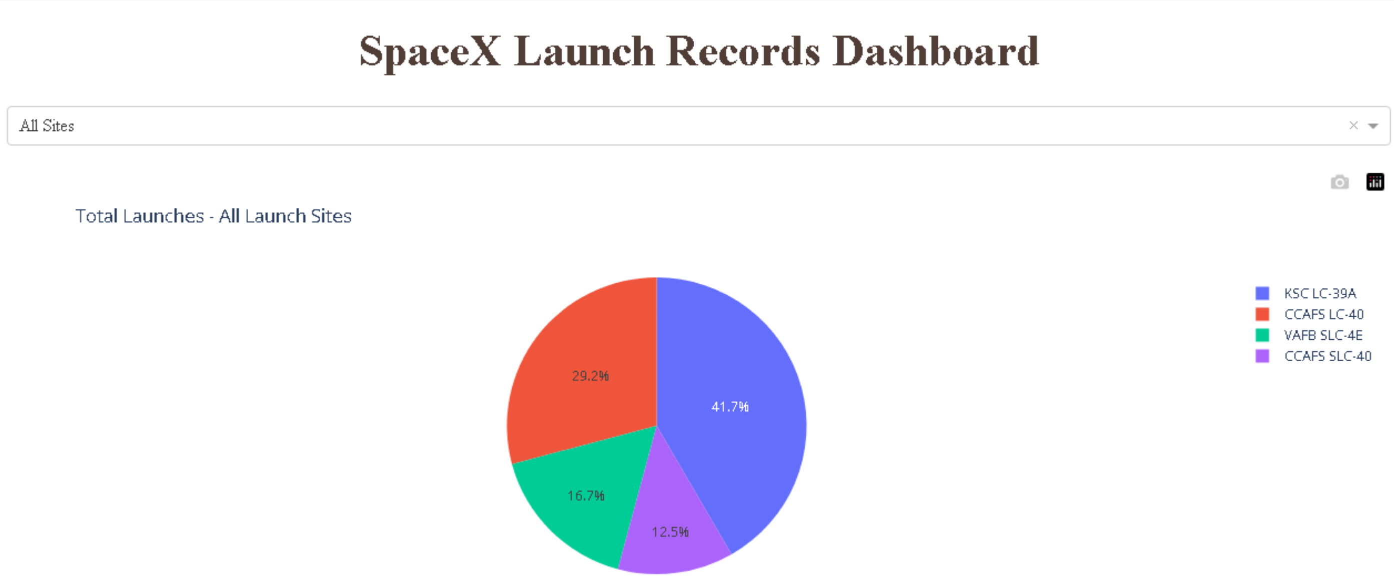
- Names of the unique launch sites in the space mission.
- Total payload mass carried by boosters launched by NASA (CRS).
- Average payload mass carried by booster version F9 v1.1.
- Date when the first successful landing outcome in ground pad was achieved.
- Names of boosters which have success in drone ship and have payload mass of 4000-6000.
- Total number of successful and failure mission outcomes.
- Names of booster_versions which have carried the maximum payload mass.
- 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.
- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium



- An interactive map was created using Folium in order to get insights about ideal locations for launch sites.
- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash



- A pie chart was created to show total launches for each launch site.
- A scatter plot was created to show payload mass carried by each booster version and also to check if the landing was successful or not.
- https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- The dataset was normalized and dummy variable for dependent variable was created.
- Columns were normalized and dataset was split into train and test datasets.
- GridSearchCV models were used to find the best hyperparameters.
- Created, trained and tested the models:
    - Logistic regression, Support Vector Machine, Decision Tree, K-Nearest Neighbours.
    - Performed Model Evaluation and compared models to get the model with highest accuracy.
    - https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

```
Y = data["Class"].to_numpy()
Y
```

```
array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1], dtype=int64)
```

```
# students get this
transform = preprocessing.StandardScaler()

X = transform.fit_transform(X)

X
```

```
array([[-1.71291154e+00, -1.94814463e-16, -6.53912840e-01, ...,
        -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
       [-1.67441914e+00, -1.19523159e+00, -6.53912840e-01, ...,
        -8.35531692e-01,  1.93309133e+00, -1.93309133e+00],
       [-1.63592675e+00, -1.16267307e+00, -6.53912840e-01, ...,
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 2)
```

we can see we only have 18 test samples.

```
Y_test.shape
```

```
(18,)
```

```
In [43]:
models = [logreg_cv, svm_cv, tree_cv, knn_cv]

labels = ['logreg_cv', 'svm_cv', 'tree_cv', 'knn_cv']

models_acc = {}

for model, label in zip(models, labels):

    models_acc[label] = model.score(X_test, Y_test)

for key, val in models_acc.items():
    print(key, ':', val)
```

```
logreg_cv : 0.8333333333333334
svm_cv : 0.8333333333333334
tree_cv : 0.8333333333333334
knn_cv : 0.8333333333333334
```

Results

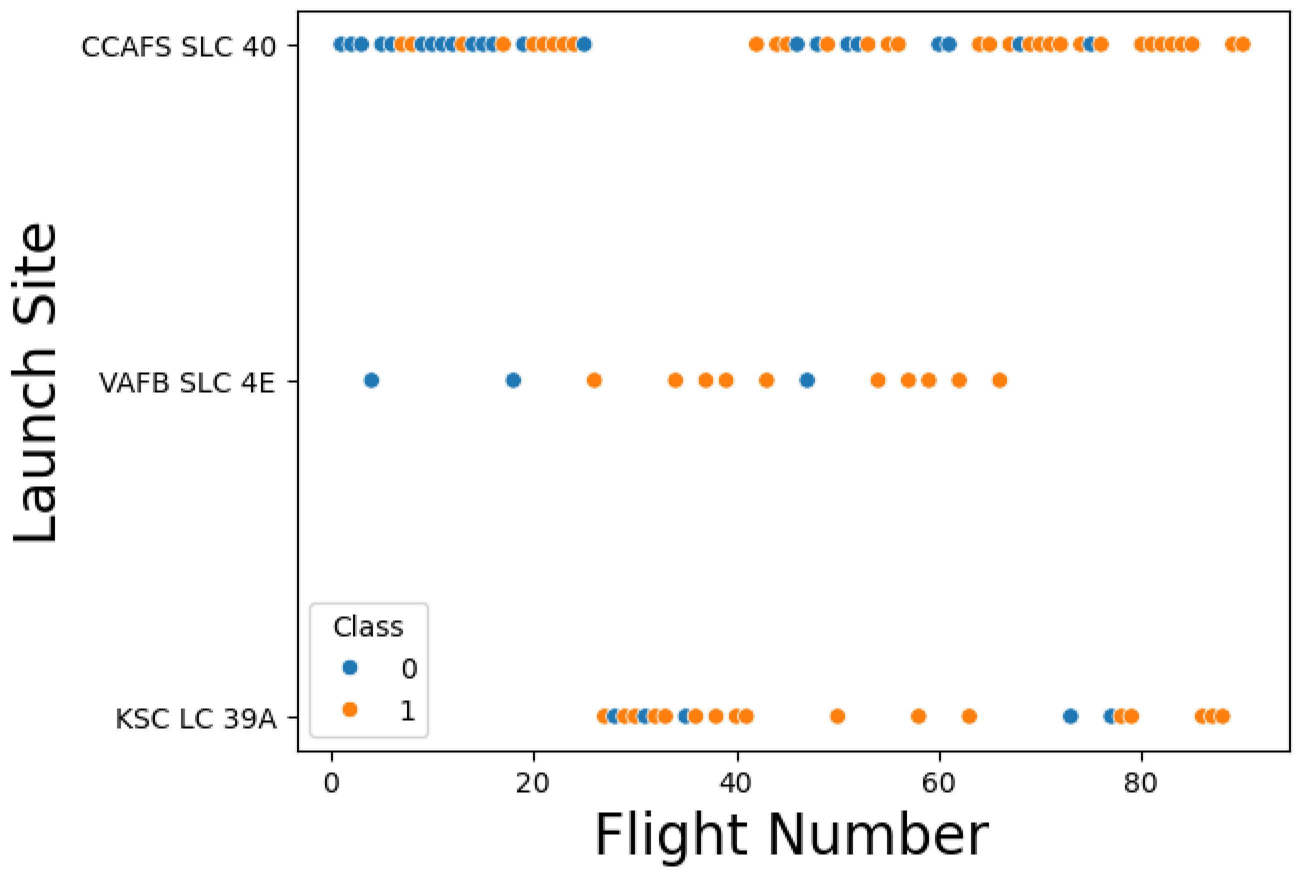
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

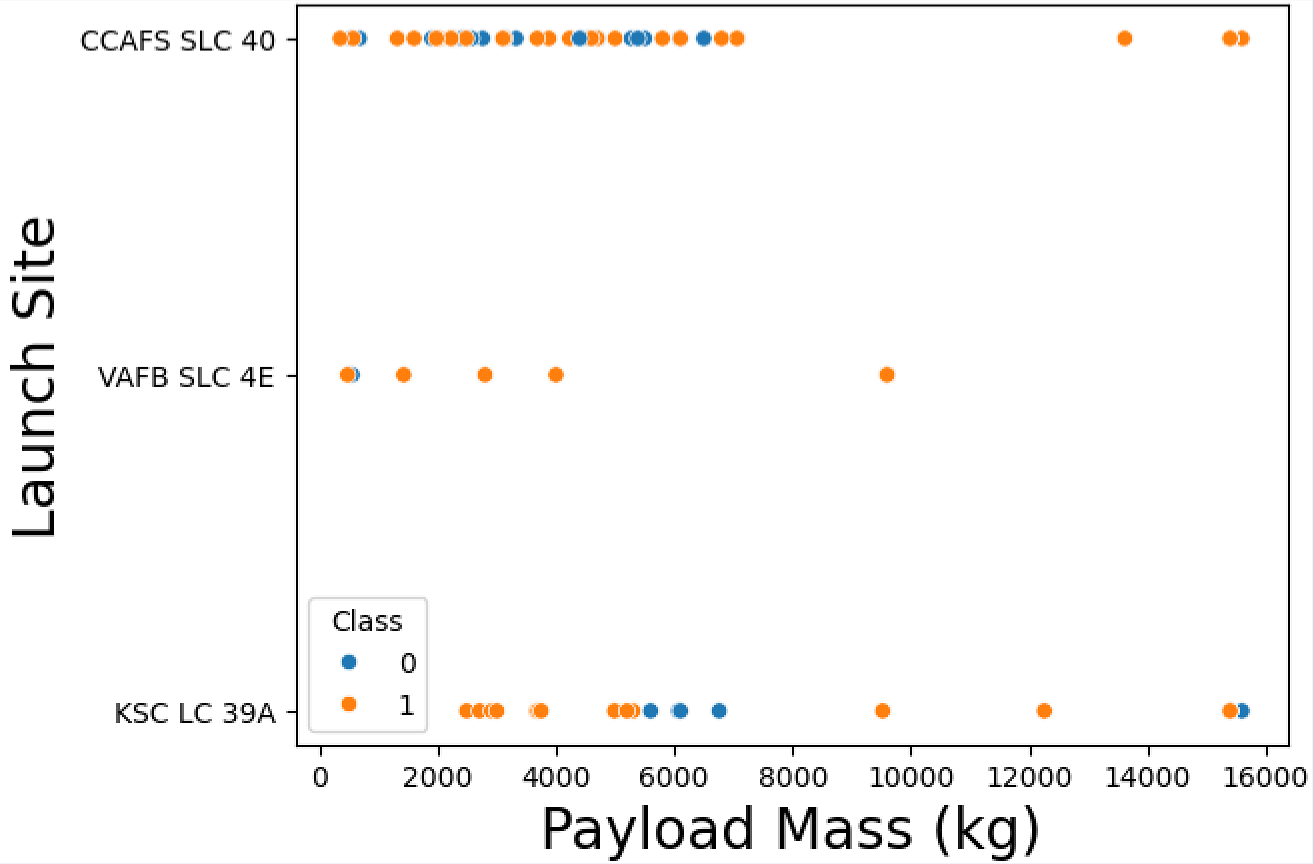
Flight Number vs. Launch Site

- CCAFS SLC-40 has the majority number of launches.
- It seems that KSC LC39A has many successful landings.



Payload vs. Launch Site

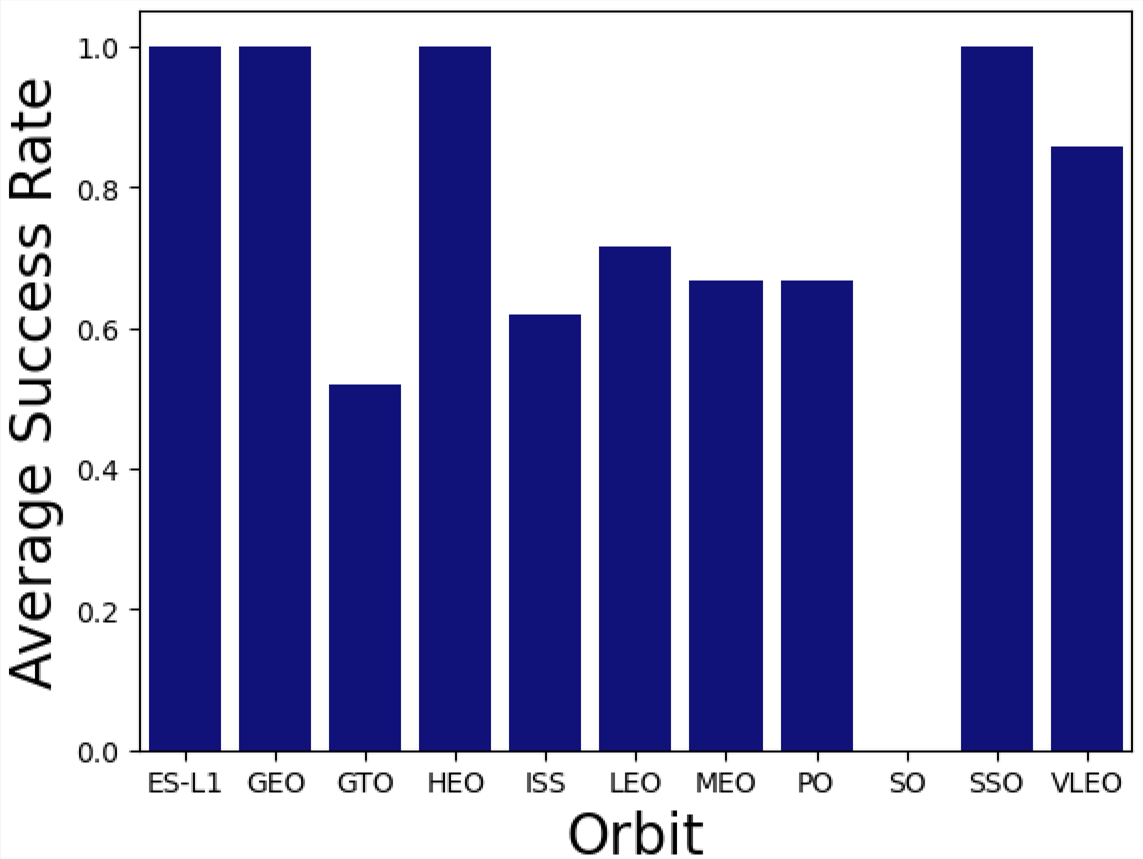
- CCAFS SLC-40 has the majority of its flights with lower payload mass.
- VAFB SLC-4E doesn't have flights for higher payload mass.
- KSC LC-39A has the majority of successful landings for higher payload mass.



Success Rate vs. Orbit Type

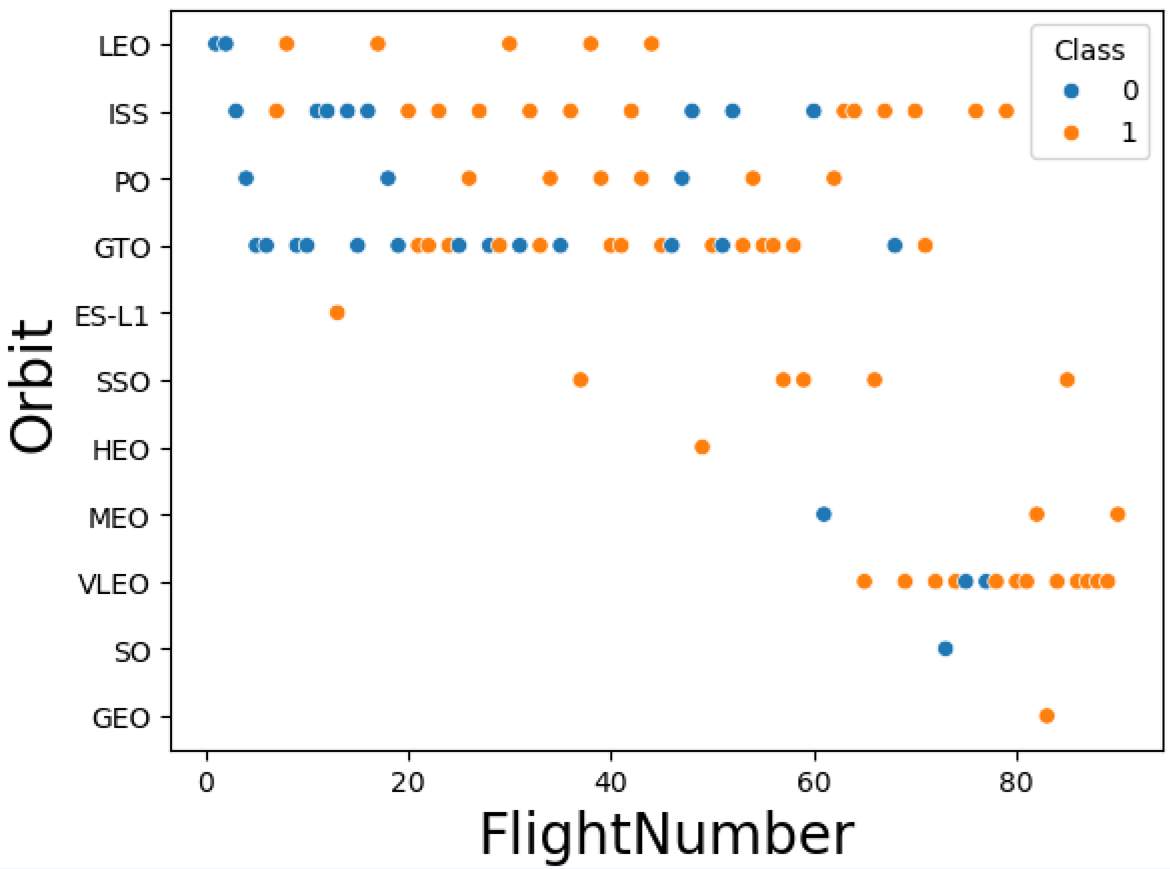
- ES-L1, GEO, HEO, SSO, and VLEO have the higher success rate of landings.

- SO shows a lower success rate of landings or not enough flights to calculate an accurate result.



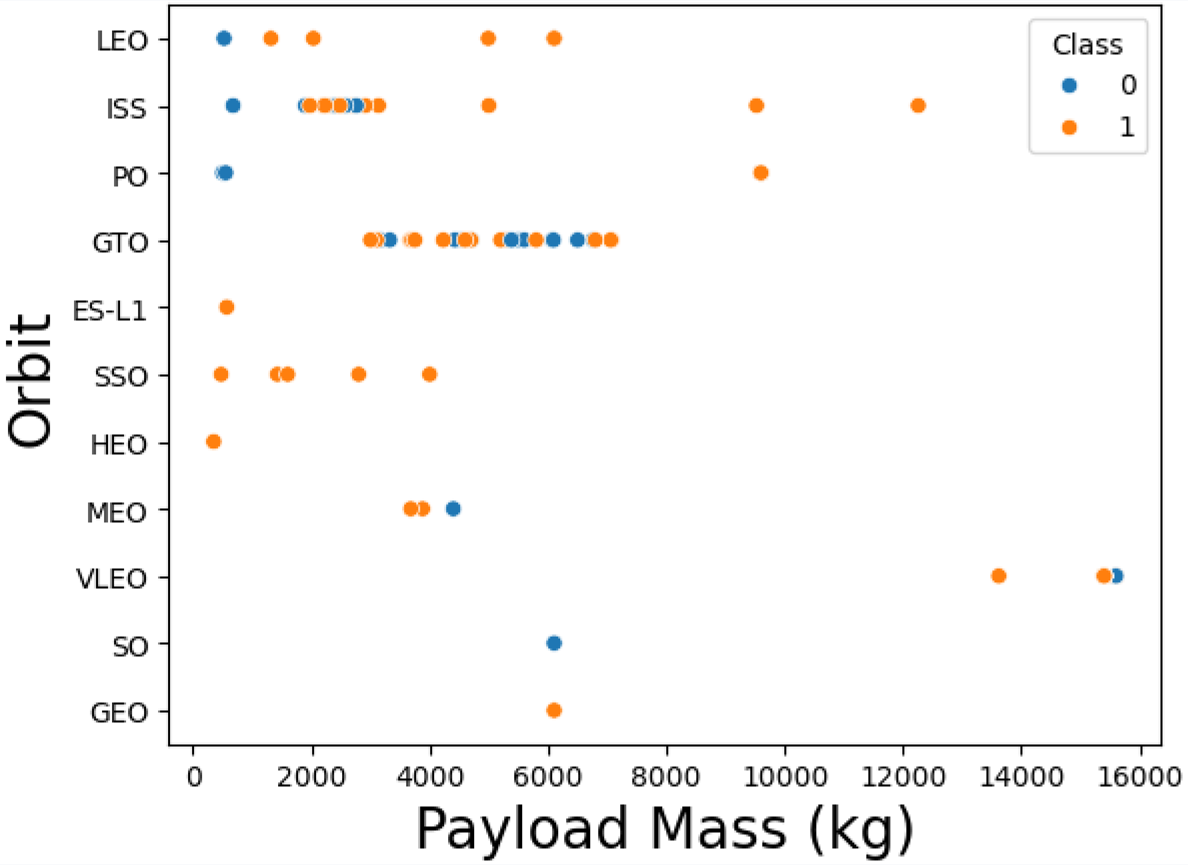
Flight Number vs. Orbit Type

- LEO, ISS, PO, and GTO have the higher count of flights.
- VLEO seems to have the higher count of flights and a considerable amount of them are successful.



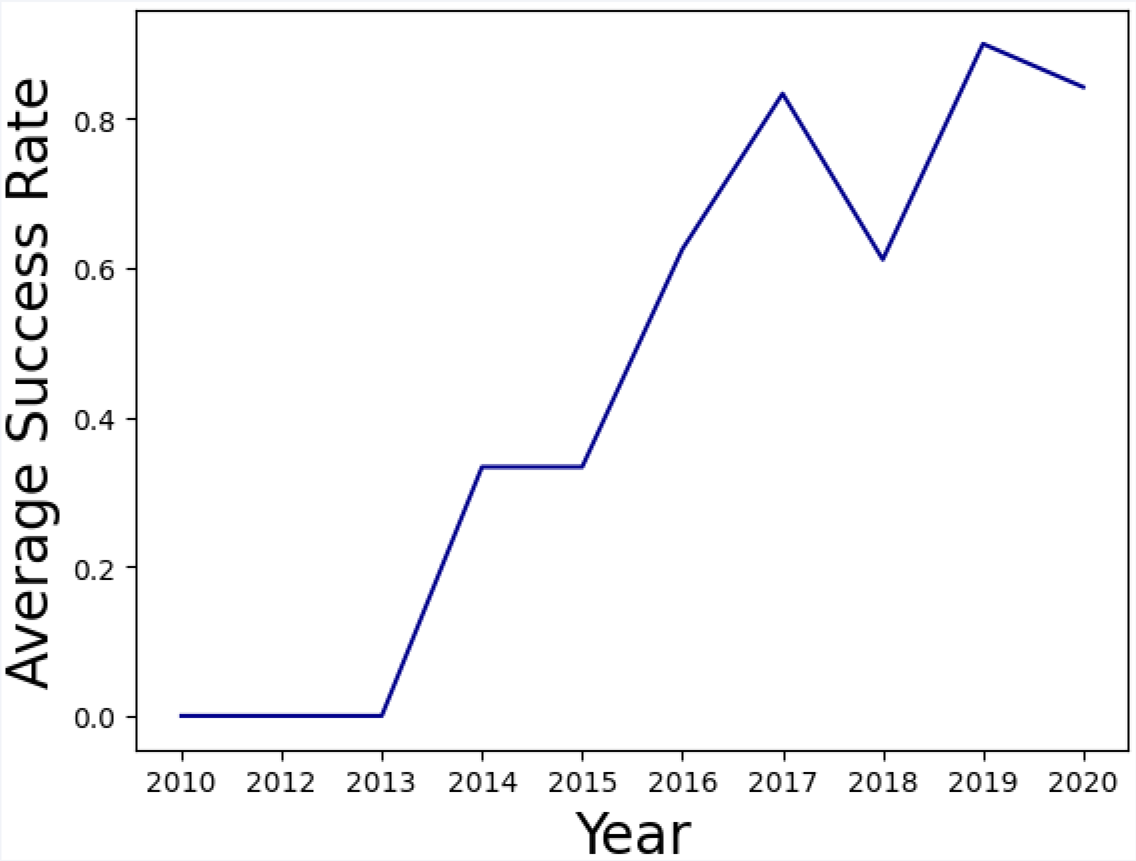
Payload vs. Orbit Type

- ISS, and GTO seems to be correlated to lower payload mass amounts.
- Ther amount of payload mass seems to variate for the rest of the orbits showing no correlation at all.



Launch Success Yearly Trend

- Since 2013 the success rate of landings increased.
- In 2015 increased again to a higher success rate.
- In 2020 it decreased.



All Launch Site Names

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

```
%%sql
SELECT
    DISTINCT("Launch_Site")
FROM
    SPACEXTABLE
```

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

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
    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

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,
    "Customer"
FROM
    SPACEXTABLE
WHERE
    "Customer" = 'NASA (CRS)'
GROUP BY "Customer"
```

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

total_payload_mass	Customer
45596	NASA (CRS)

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,
    "Booster_Version"
FROM
    SPACEXTABLE
WHERE
    "Booster_Version" = 'F9 v1.1'
GROUP BY "Booster_Version"
```

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

average_payload_mass	Booster_Version
2928.4	F9 v1.1

First Successful Ground Landing Date

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

```
%%sql
SELECT
    "Date",
    "Landing_Outcome"
FROM
    SPACEXTABLE
WHERE
    "Landing_Outcome" = 'Success (ground pad)'
ORDER BY "Date" ASC
LIMIT 1
```

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

Date	Landing_Outcome
2015-12-22	Success (ground pad)

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",
    "Landing_Outcome",
    "PAYLOAD_MASS__KG_"
FROM
    SPACEXTABLE
WHERE
    "Landing_Outcome" = 'Success (drone ship)' AND
    "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
GROUP BY "Booster_Version"
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS_KG_
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT
    "Mission_Outcome",
    COUNT("Mission_Outcome")
FROM
    SPACEXTABLE
GROUP BY "Mission_Outcome"
```

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

Mission_Outcome	COUNT("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

```
%%sql
SELECT
    "Booster_Version",
    "PAYLOAD_MASS__KG_"
FROM
    SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (SELECT
                                MIN("PAYLOAD_MASS__KG_")
                            FROM
                                SPACEXTABLE)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 v1.0 B0003	0
F9 v1.0 B0004	0

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.

```
%%sql
SELECT
    substr("Date", 0, 5) AS year,
    substr("Date", 6, 2) AS n_month,
    substr('JanFebMarAprMayJunJulAugSepOctNovDec', 1 + 3 * strftime('%m', date("Date")), -3) AS month,
    "Launch_Site",
    "Landing_Outcome",
    "Booster_Version"
FROM
    SPACEXTABLE
WHERE
    "Landing_Outcome" = 'Failure (drone ship)' AND
    substr("Date", 0, 5) = '2015'
```

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

year	n_month	month	Launch_Site	Landing_Outcome	Booster_Version
2015	01	Jan	CCAFS LC-40	Failure (drone ship)	F9 v1.1 B1012
2015	04	Apr	CCAFS LC-40	Failure (drone ship)	F9 v1.1 B1015

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.

```
%%sql
SELECT
    COUNT("Landing_Outcome") AS n_landing_outcomes,
    "Landing_Outcome"
FROM
    (SELECT
        "Date",
        "Landing_Outcome"
     FROM
        SPACEXTABLE
     WHERE
        ("Landing_Outcome" = 'Success (ground pad)' OR "Landing_Outcome" = 'Failure (drone ship)') AND
         ("Date" > '2010-06-04' AND "Date" < '2017-03-20'))
GROUP BY "Landing_Outcome"
ORDER BY n_landing_outcomes DESC
```

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

n_landing_outcomes	Landing_Outcome
5	Failure (drone ship)
3	Success (ground pad)

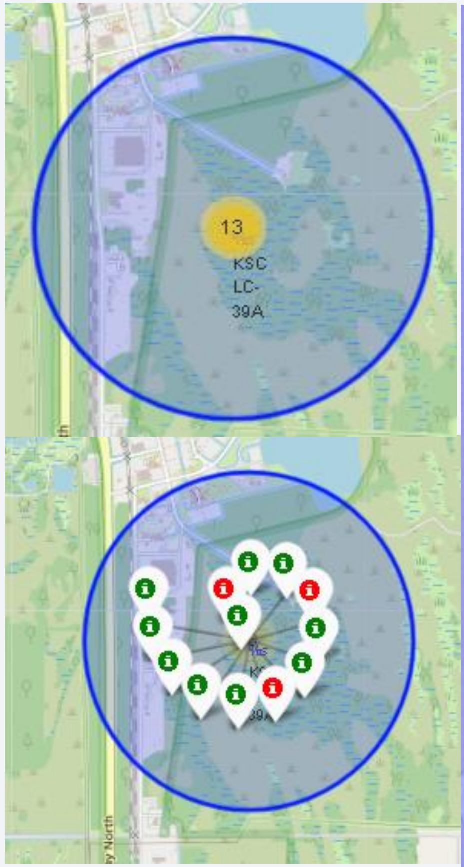
Section 3

Launch Sites Proximities Analysis

Launch Sites Locations

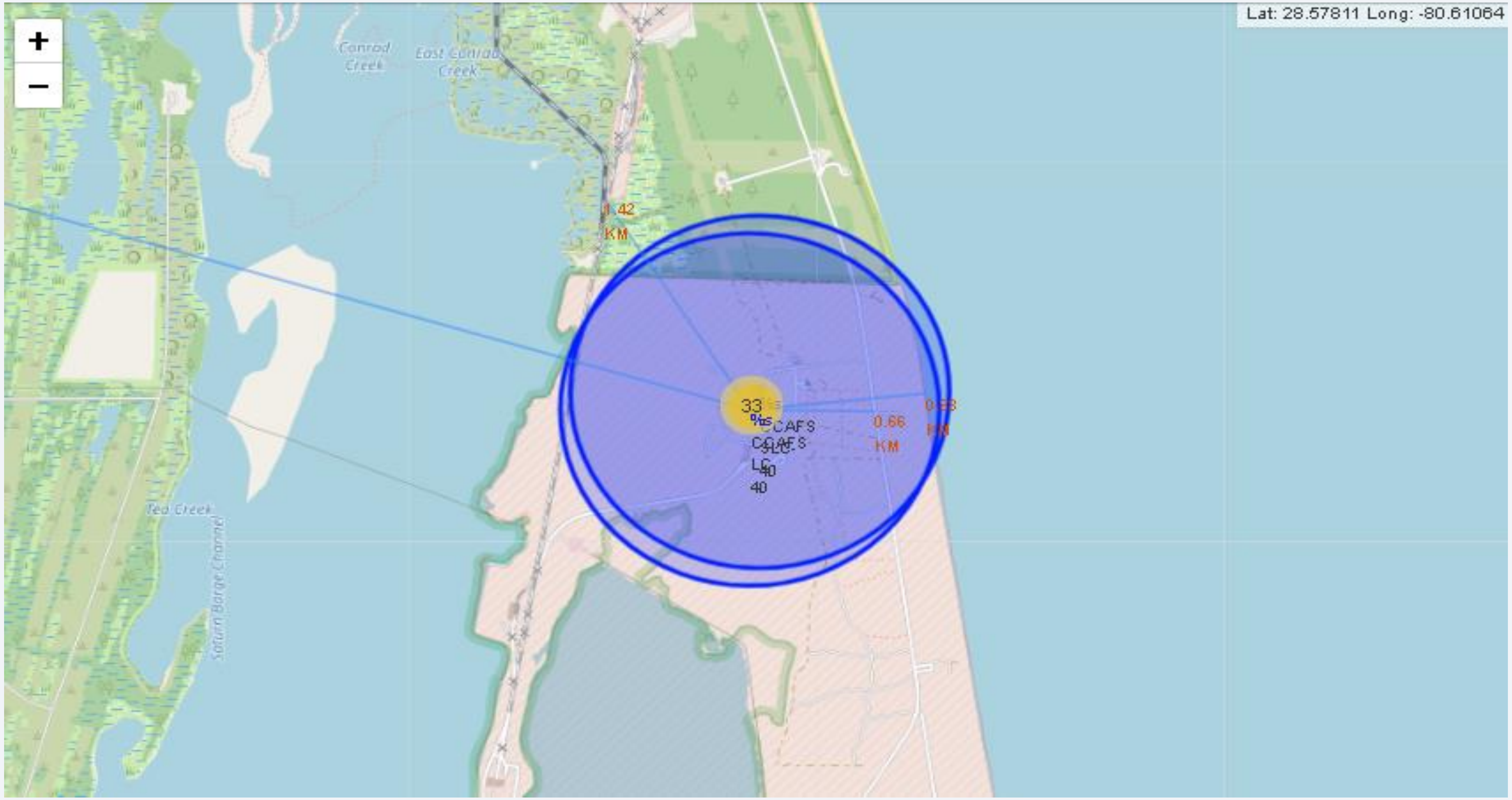
- All launch sites in very proximity to the coast.
- Launch sites have certain proximity to the Equator line.

Count of Successful/Unsuccessful Landing Outcomes for each Launch Site



- Count of launches labeled by successful or unsuccessful landing for each launch site.
- KCS LC-39A has the greater number of successful launches.

Launch Sites Proximities

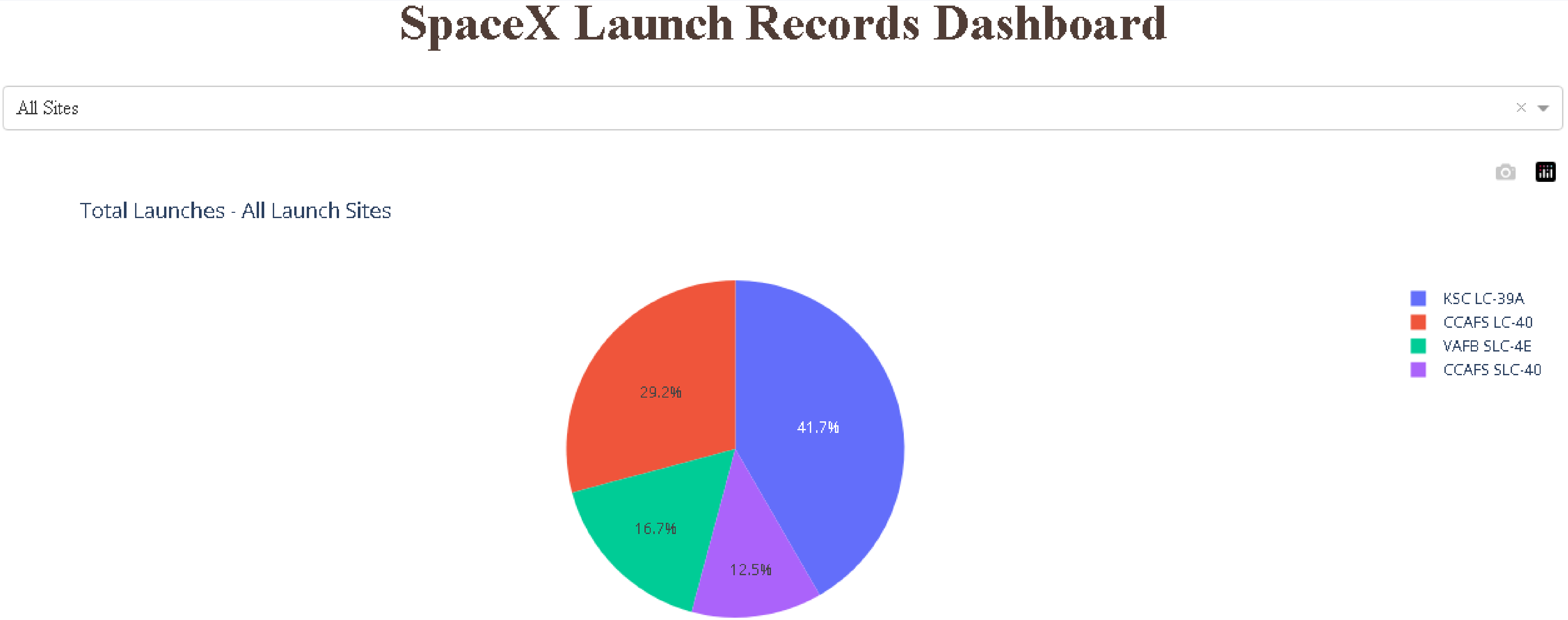


- CCAFS SLC-40 is the Launch Site that is nearest to a railway, highway, and city than the others.

Section 4

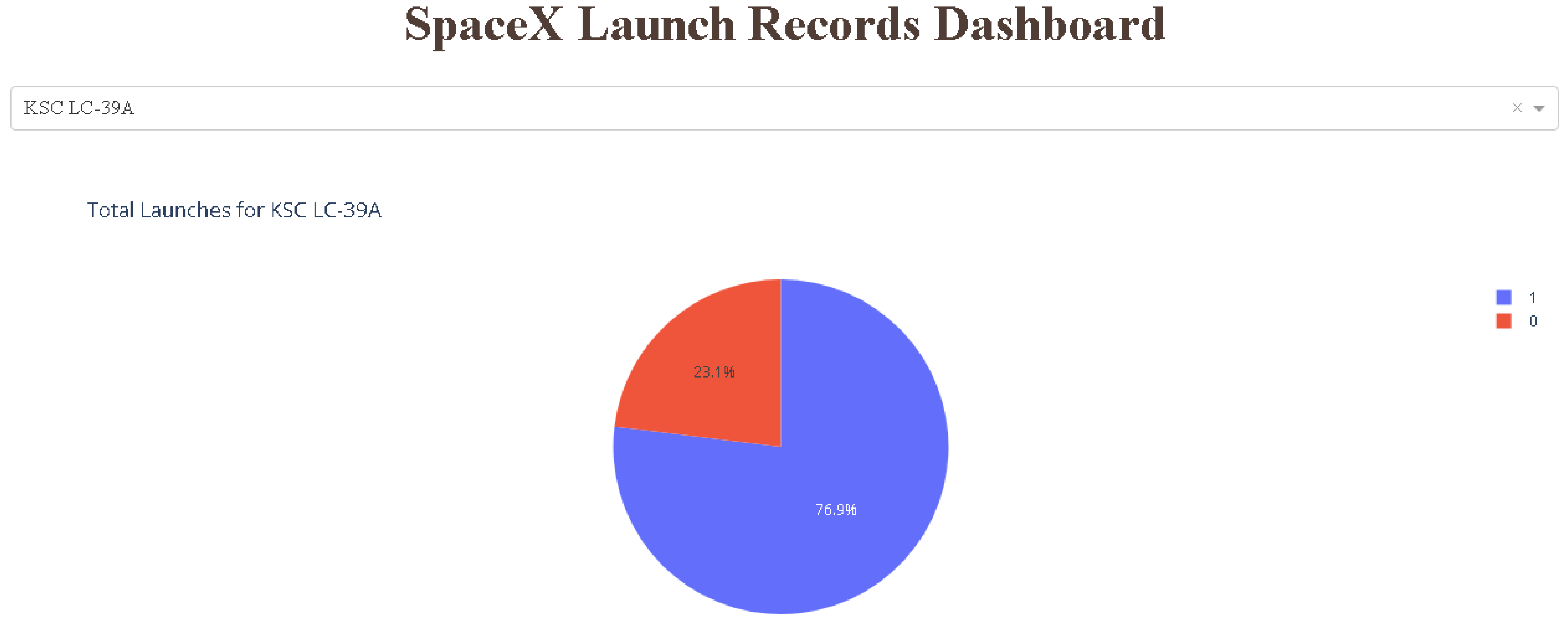
Build a Dashboard with Plotly Dash

Total Launches by Launch Site



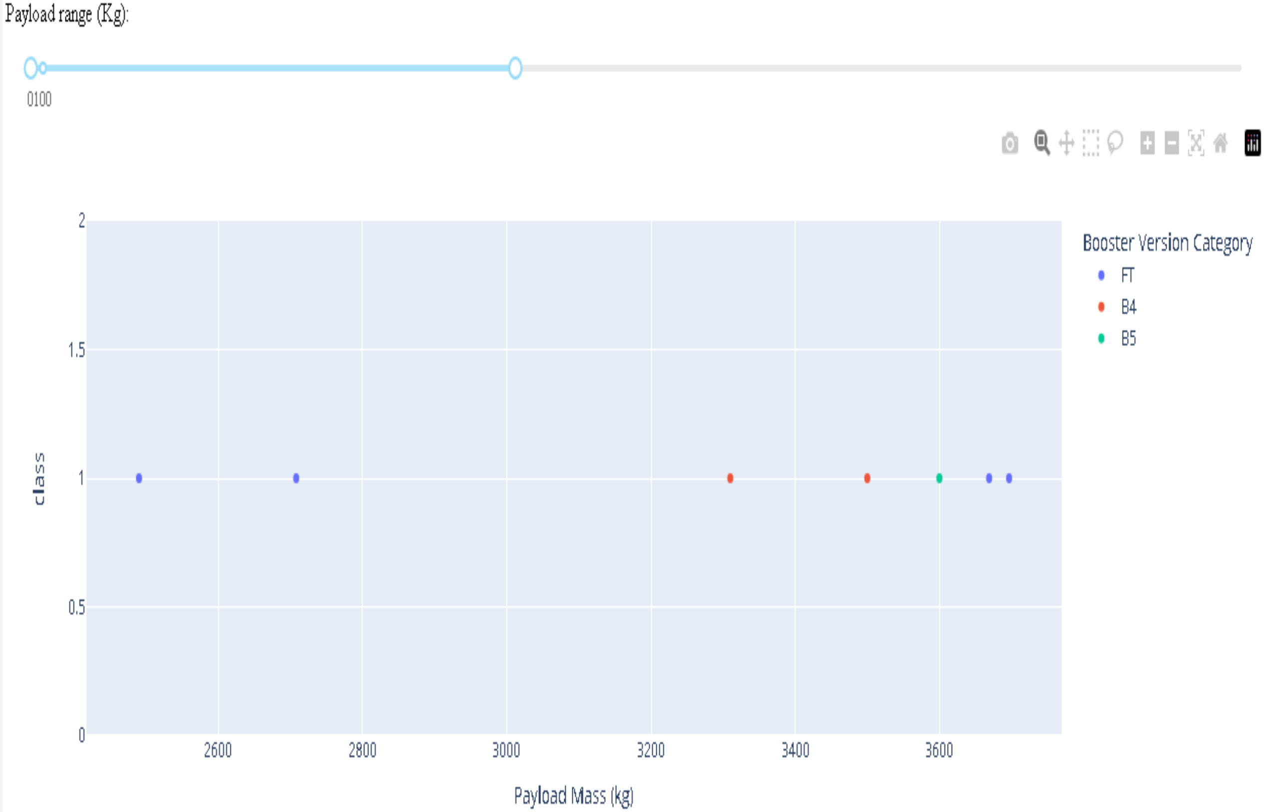
- Counts of launches for each site are compared.
- KSC LC-39A launch site has the majority of the total launches.

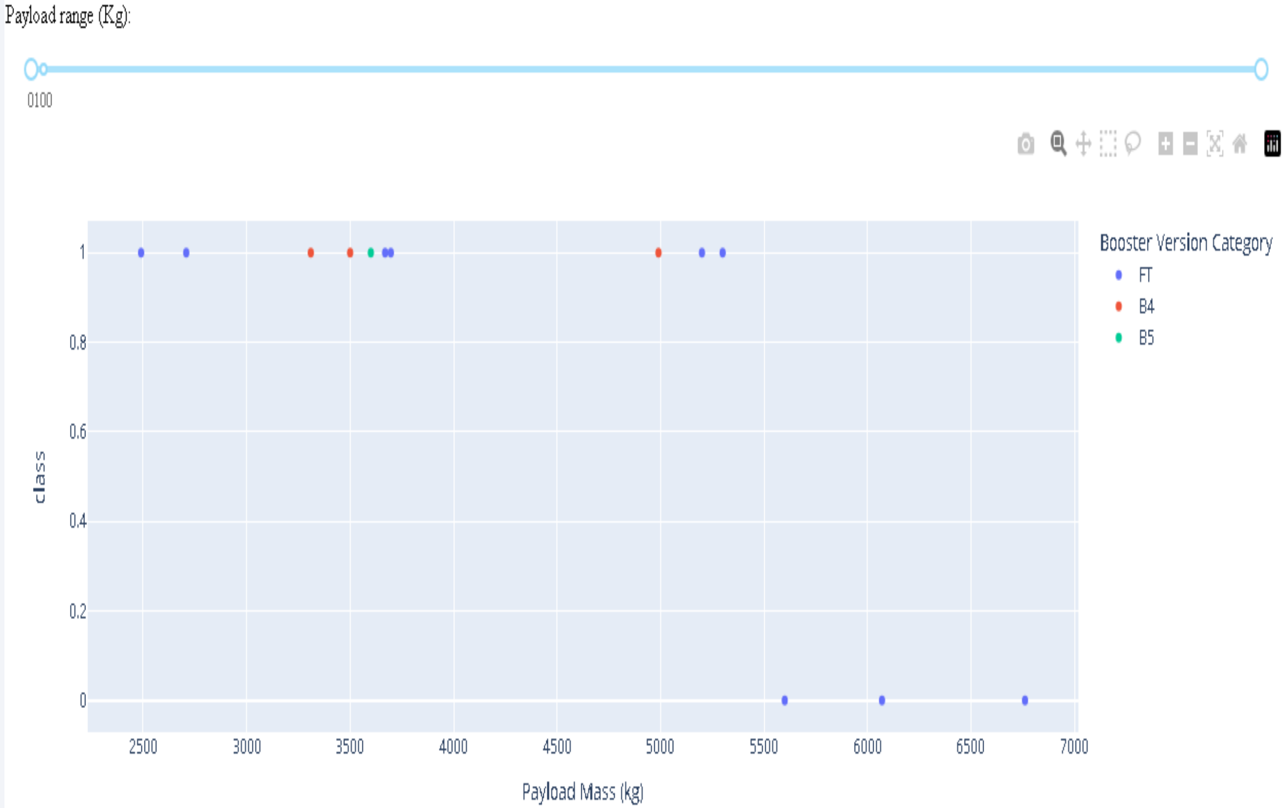
Launch Site with the Majority of Successful Launches



- KSC LC-39A has the greater ratio of successful launches.

Payload Mass Carried by Boosters





- FT Booster Version Category has the majority of successful launches for medium and high amount of payload mass.

Section 5

Predictive Analysis (Classification)

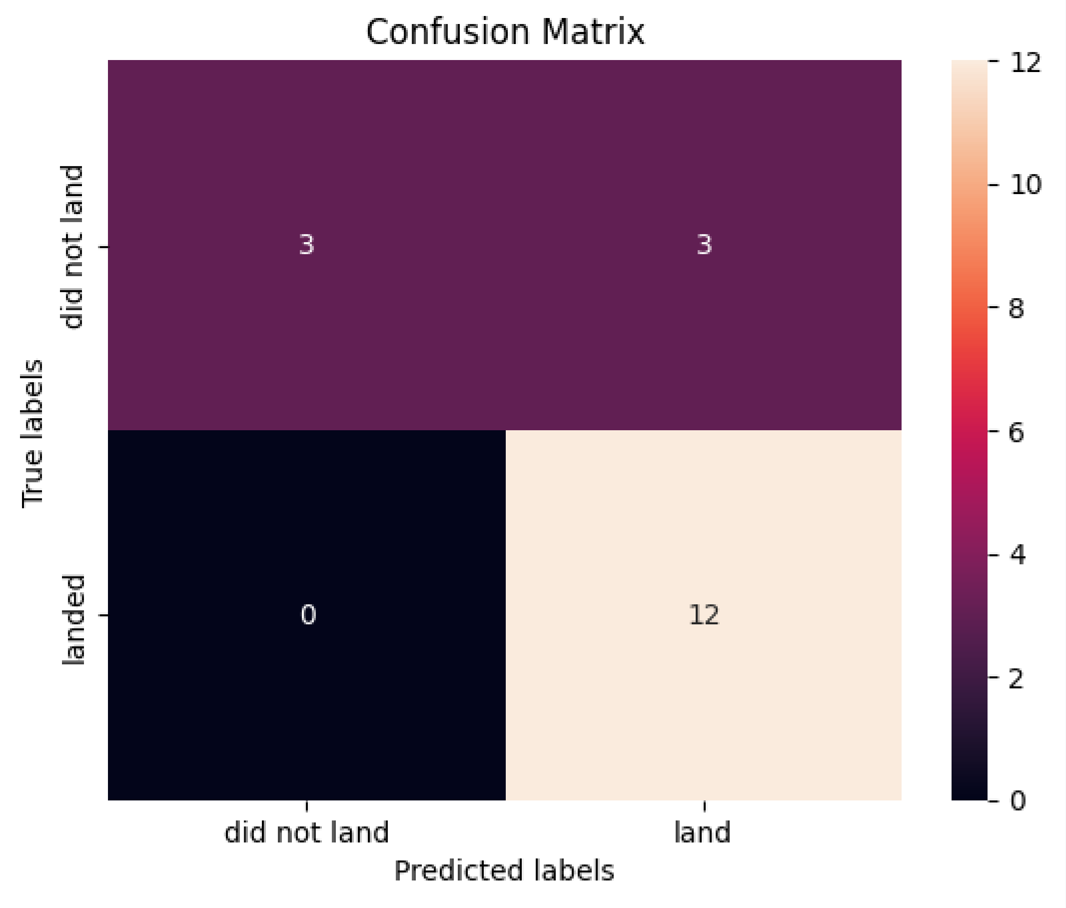
Classification Accuracy

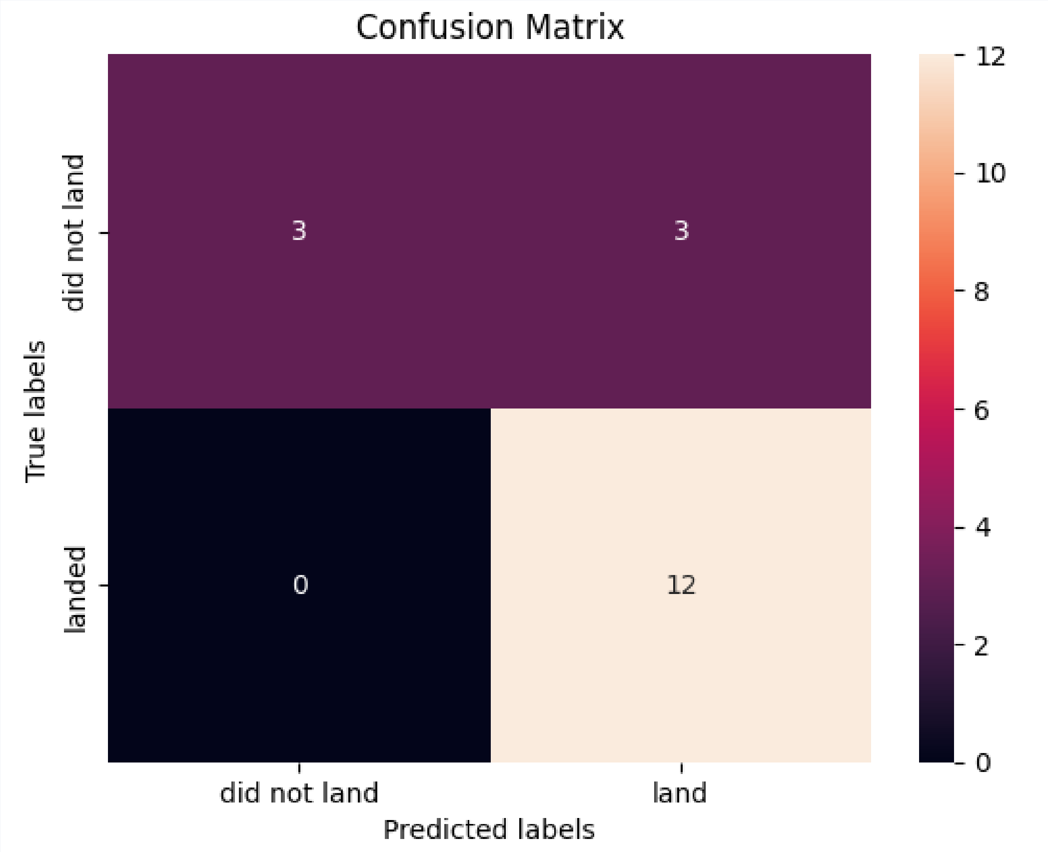
```
models = [logreg_cv, svm_cv, tree_cv, knn_cv]

labels = ['logreg_cv', 'svm_cv', 'tree_cv', 'knn_cv']

models_acc = {}

for model, label in zip(models, labels):

    models_acc[label] = model.score(X_test, Y_test)

for key, val in models_acc.items():
    print(key, ':', val)
```

```
logreg_cv : 0.8333333333333334
svm_cv : 0.8333333333333334
tree_cv : 0.8333333333333334
knn_cv : 0.8333333333333334
```

All created models have shown almost the same accuracy, however for the purposes of this dataset, decision trees or logistic regression may be a good fit.

Confusion Matrix





- All models are showing strong accuracy for predicting successful outcomes, the consistent problem is with the False Positives.

Conclusions

- Launches with payload mass higher than 10000 have a low success rate of landing.
- ES-L1, GEO, HEO, SSO, and VLEO Orbits have the higher success rate of landings.
- KSC LC-39A Launch Site has the majority count of successful landings.
- Almost all launch sites are near to the coast, railway, and relatively near cities depending on launch site's location.
- The best models determined by an accuracy of 83% and the characteristics of the dataset are Logistic Regression and K-Nearest Neighbours, also Support Vector Machines can be used for the predictive analysis.
- Yearly trend success rate shows that probably improvements on technology, lessons learned and trial and error are shaping the way to get a 100% success rate of landing in the future, as more data on those launches is generated, better models can be created.

Appendix

- Material, Notebooks, Python Code and Datasets used for analysis can be found on a github's repository in the following link:
    - https://github.com/carloseguzf/Applied_Capstone-IBM_DS-SpaceX_Launches

Thank you!