IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gyosmen-Hashims Shittu
September, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API
  - Data collection with web scrapping
  - Data wrangling
  - Exploratory data analysis (EDA) with SQL
  - EDA with data visualization
  - Machine learning prediction
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analytics

Introduction

- Project background and context

  SpaceX advertises that it's Falcon 9 rocket launches with a cost of 62 million dollars, a cost lower than other providers who put the cost upward of 165 million dollars. 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 then determine the cost of a launch. The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will make a successful landing?
  - What are the interactions between these factors that determine the success of a landing?
  - What are the operating conditions that are required to be in place to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using the SpaceX API and web scraping
- Perform data wrangling
  - One hot encoding was applied to categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data used in this project were collected from the following sources:
  - SpaceX API to request SpaceX data
  - Web scraping to collect data for Falcon 9 launch records from a Wikipedia web page

Data Collection – SpaceX API

- We requested for data from the SpaceX API, cleaned the resulting data, performed data wrangling and formatting

- The following link contains the notebook that shows this process: https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/Spacex_data_collection_api.ipynb

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

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

Check the content of the response

```
[8]  print(response.content)

     b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small
```

To make the requested JSON results more consistent, we will use the following static response object for this project:

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

[10] response.status_code

     200

[11] # Use json_normalize method to convert the json result into a dataframe
     data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

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

Data Collection - Scraping

- Using webscaping, we were about to get data on Falcon 9 Launch records which we then converted into pandas dataframe

- The following link contains the notebook that shows this process: https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/Data_collection_with_webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

```
[5]  # use requests.get() method with the provided static_url
     html_data = requests.get(static_url)

     # assign the response to a object
     html_data.status_code

     200
```

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

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

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

```
[7]  # Use soup.title attribute
     soup.title

     <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

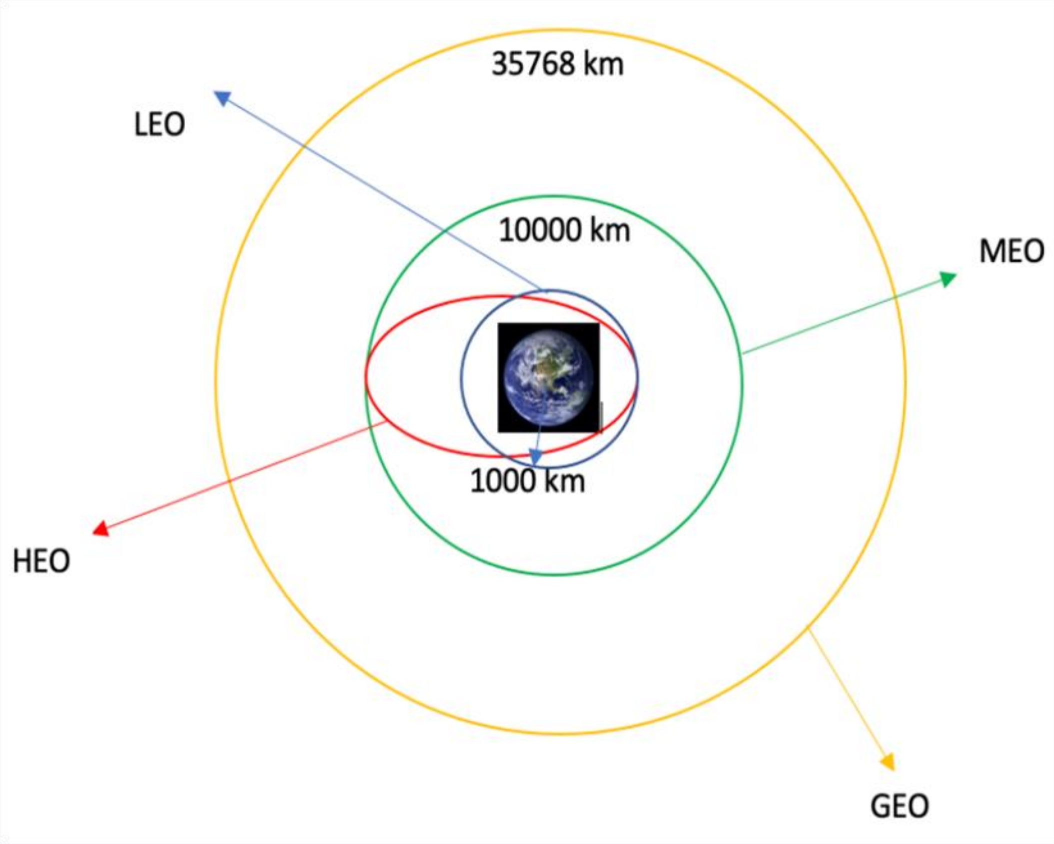
```
[8]  # Use the find_all function in the BeautifulSoup object, with element type `table`
     # Assign the result to a list called `html_tables`
     html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

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

Data Wrangling

- We determined the best training labels after exploratory data analysis
- We then calculated the number of launches at each site and the number and occurrence of each orbit
- The following link contains the notebook that shows this process: https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/Spacex_Data_wrangling.ipynb



EDA with Data Visualization

- We explored the data by creating the following visualizations:
  - Scatter plots to determine if there is a relationship between the Launch Site and Payload, Launch Site and Flight Number, Orbit Type and Flight Number, Orbit Type and Payload and if these had any bearing on the success
  - Bar chat to get a better picture of the success rate by class of each orbit
  - Line plot to view the trend of the yearly success rate
- The following link contains the notebook that shows this process: https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/EDA_Dataviz.ipynb

EDA with SQL

- We connected to the SQL database and ran queries in order to help us gain more insight about the data. The queries helped us to get insight about the following:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The following link contains the notebook that shows this process:

https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/EDA_SQL_COURSERA_SQLLITE.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We calculated the distances between a launch site to its proximities in order to answer questions such as:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?
- The following link contains the notebook that shows this process:

https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/Launch_site_location_with_folium.ipynb

Build a Dashboard with Plotly Dash

- We built a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- We added a pie chat to show the total successful launches by all sites and also to show the successful launches by the different sites.
- We also added a scatter plot to show the correlation between the payload and successful launches.
- The following link contains the notebook that shows this process:

https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We created a column for class, standardized the data and split the data into testing (20%) and training (80%) sets.
- We built Logistic Regression, Decision Tree, Support Vector Machine and K Nearest Neighbor machine learning models and tune different hyperparameters using GridSearchCV
- We used accuracy as the metric for evaluation, improved the model using feature engineering and algorithm tuning.
- We then found the best performing model and parameters.
- The following link contains the notebook that shows this process:

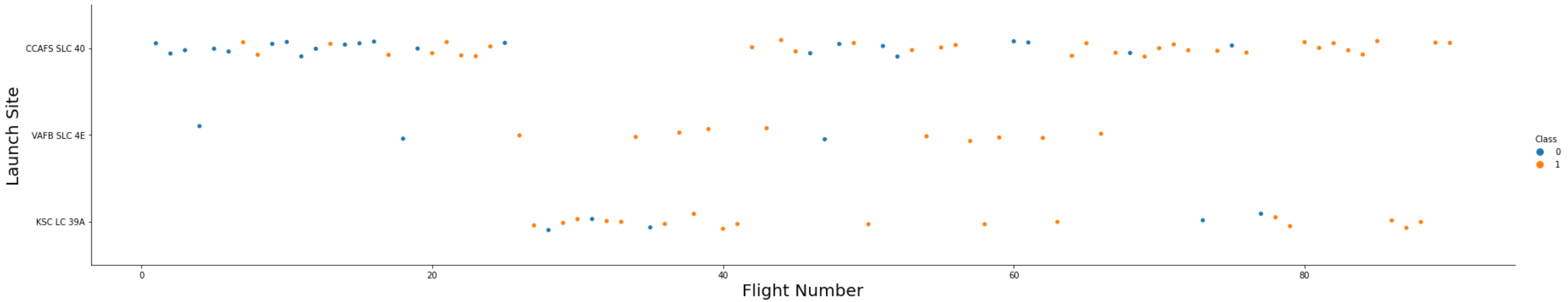
https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

Results

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

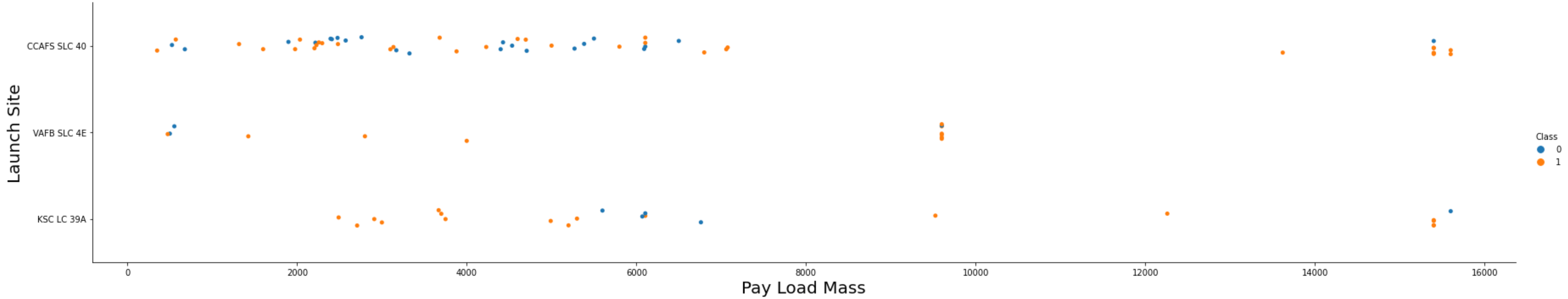
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- From the plot, we can see that the site CCAFS SLC 40 recorded more launches than any other launch site
- We also see that there are more successes with the larger flight numbers as opposed to the smaller flight numbers

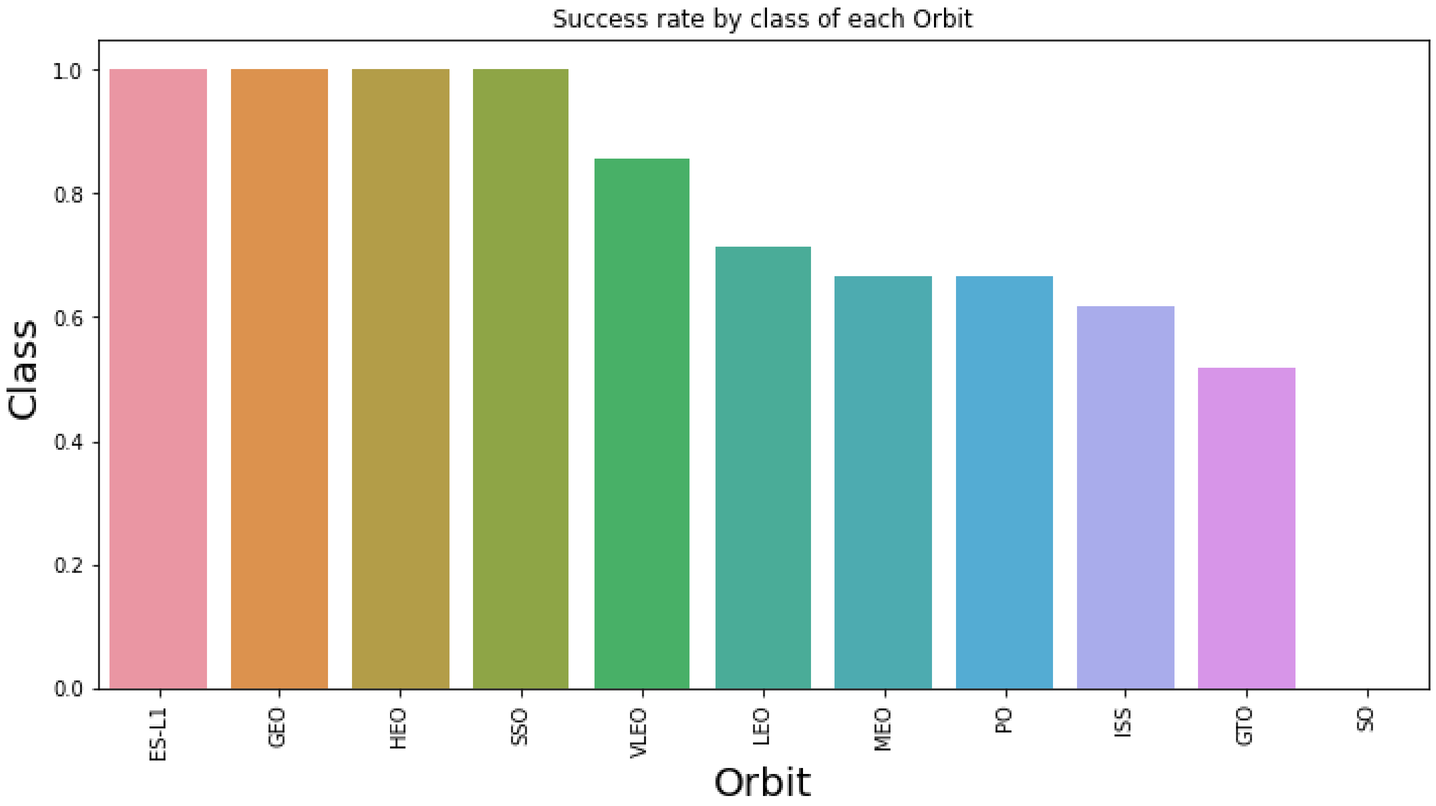


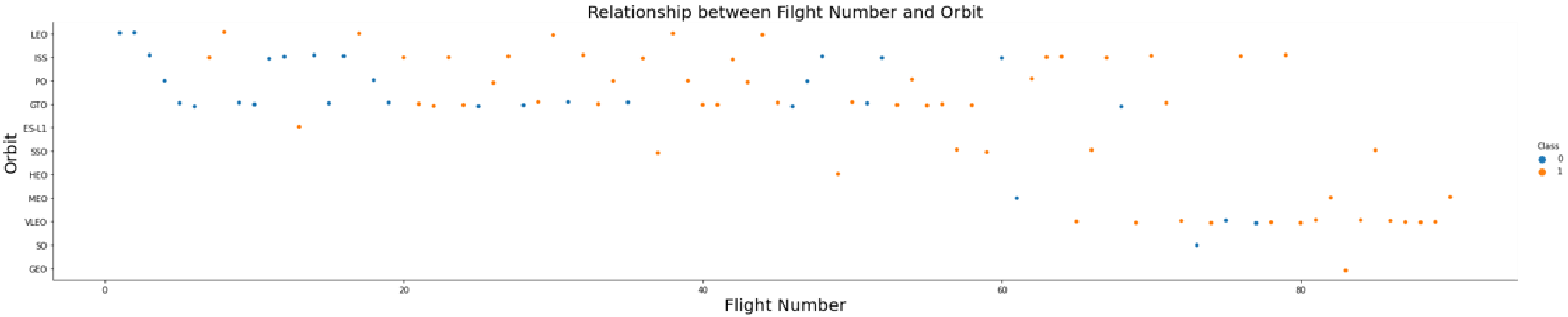
Payload vs. Launch Site

- Here, that though the number of launches are fewer at higher payloads, we have more success at higher payload.
- We can see however that with a payload mass of between 2000 and 4000, both VAFB SLC 4E and KSC LC 39A record great success as opposed to what we see from CCAFS SLC 40.

Success Rate vs. Orbit Type

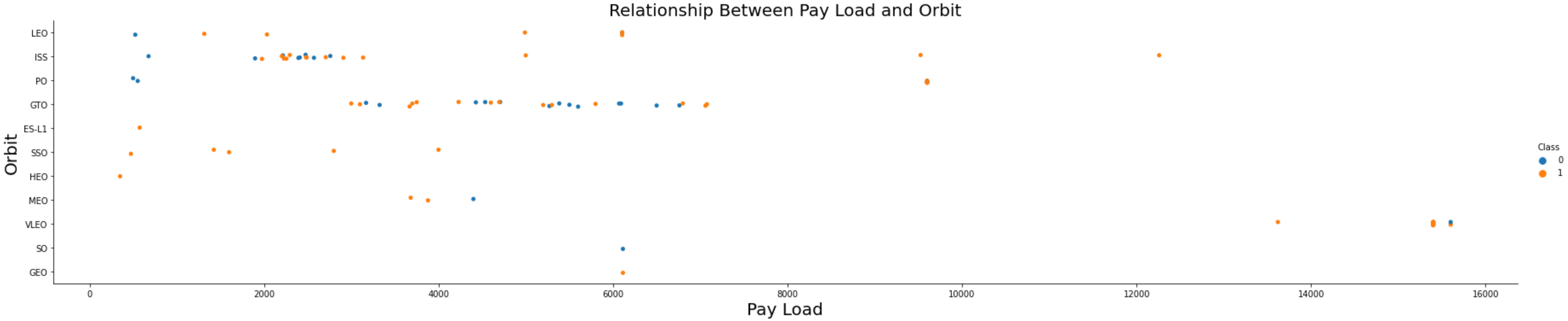
- From the plot, we can see that the orbits ES-L1, GEO, HEO, SSO and VLEO recorded the most success.





Flight Number vs. Orbit Type

- We can see from the plot that all orbit types show relatively more success as the flight number increases

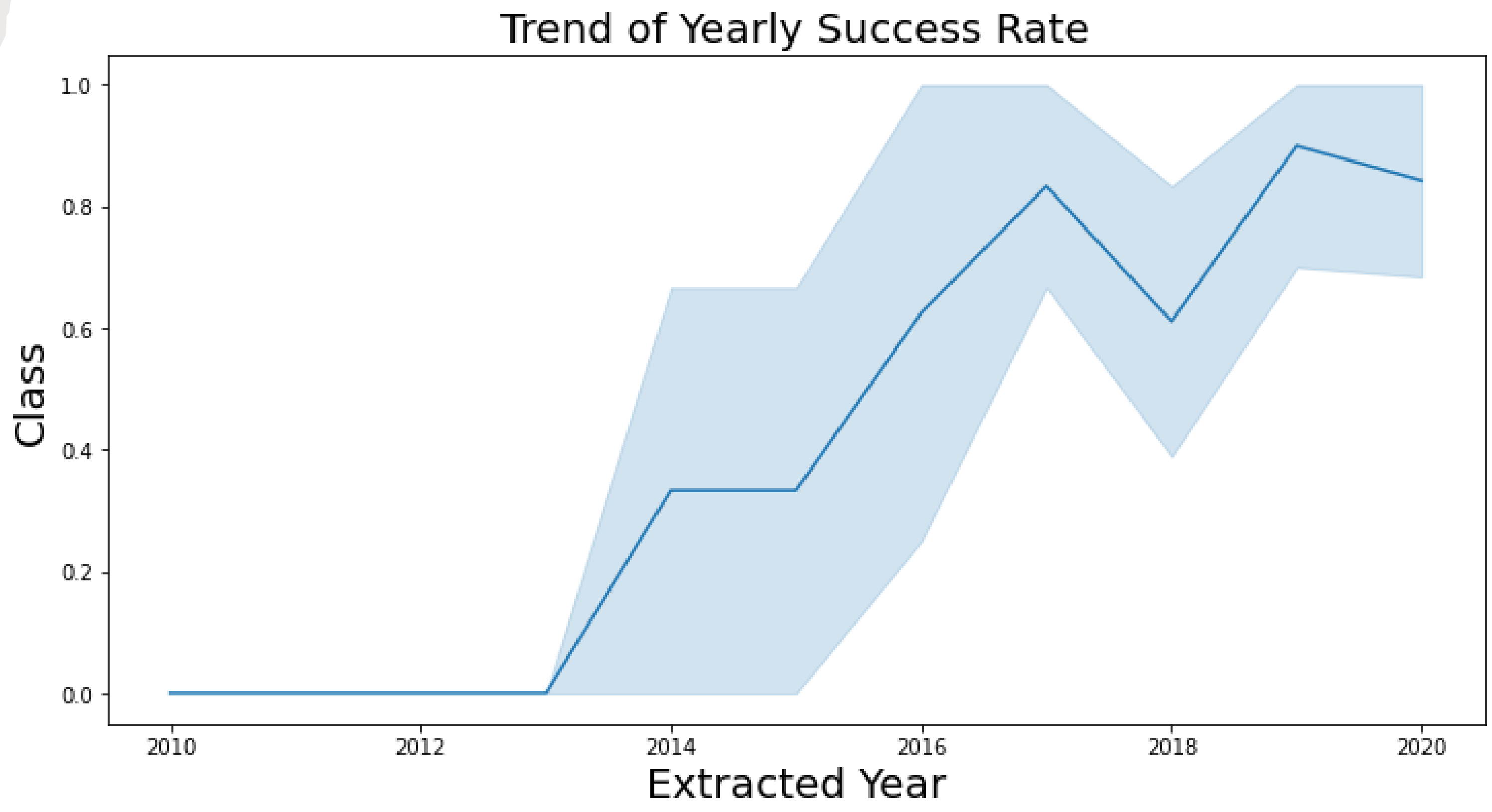


Payload vs. Orbit Type

- We can observe here that with heavier payloads, there are more successes recorded for PO, LEO and ISS orbits while with lighter payloads ES-L1, SSO, HEO and MEO show greater success

Launch Success Yearly Trend

- From the plot, we can see that there was a progressive increase in success rate from 2013 to 2020 with a dip in 2018.



All Launch Site Names

- From the query result, we can see there are only 4 Launch sites in the Space X data

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

In [7]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

Out[7]:

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'

In [8]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- From the above query result, we can see that Mission Success was recorded in the Launch sites beginning with 'CCA'

Total Payload Mass

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

In [9]:
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass
FROM SPACEXTBL
WHERE Customer IS 'NASA (CRS)';
```

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

Out[9]:

Total_Payload_Mass
45596

- The query result shows that the total payload mass recorded was 45,596kg

Average Payload Mass by F9 v1.1

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

In [10]:
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass
FROM SPACEXTBL
WHERE Booster_Version IS 'F9 v1.1';
```

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

Out[10]:

Avg_Payload_Mass
2928.4

- The query result shows the average payload mass carried by booster version F9 v1.1 was 2,928.4kg

First Successful Ground Landing Date

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

Hint:Use min function

In [11]:
```
%%sql
SELECT MIN(DATE) AS First_sucess
FROM SPACEXTBL
WHERE "Landing _Outcome" IS 'Success (ground pad)';
```

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

Out[11]:

First_sucess
01-05-2017

- The query result revealed that the first successful landing outcome on ground pad was recorded on the 1st of May, 2017

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 [12]:

```
%%sql
SELECT Booster_Version
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (drone ship)'
        AND PAYLOAD_MASS__KG_ > 4000
        AND PAYLOAD_MASS__KG_ < 6000;
```

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

Out[12]:

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

- The query result shows that there are 4 booster versions with the successful drone ship landing having a payload between 4000 and 6000 as listed in the screen shot above.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [13]:

```
%%sql
SELECT COUNT(MISSION_OUTCOME) AS Mission_Success, Mission_Failure
FROM SPACEXTBL, (SELECT COUNT(MISSION_OUTCOME) AS Mission_Failure
                 FROM SPACEXTBL
                 WHERE MISSION_OUTCOME LIKE 'Failure%')
WHERE MISSION_OUTCOME LIKE 'Success%';
```

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

Out[13]:

Mission_Success	Mission_Failure
100	1

- From the query result above, we see that there were 100 mission successes with one recorded mission failure

Boosters Carried Maximum Payload

- The query result lists the boosters 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 [14]:

```
%%sql
SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
                           FROM SPACEXTBL)
ORDER BY Booster_Version;
```

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

Out[14]:

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

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [15]:

```
%%sql
SELECT substr(Date, 4, 2) AS  Month, Booster_Version, Launch_site, "Landing _Outcome"
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE 'Failure (drone ship)'
        AND substr(Date,7,4)='2015';
```

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

Out[15]:

Month	Booster_Version	Launch_Site	Landing _Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- The query shows the list of the failed landing outcomes in drone ships, their booster versions, launch site names and month of the failed mission for in year 2015

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

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [16]:
```
%%sql
SELECT "Landing _Outcome", COUNT("Landing _Outcome") AS Count_of_outcome
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010'AND '20-03-2017'
GROUP BY "Landing _Outcome"
ORDER BY Count_of_outcome DESC;
```

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

Out[16]:

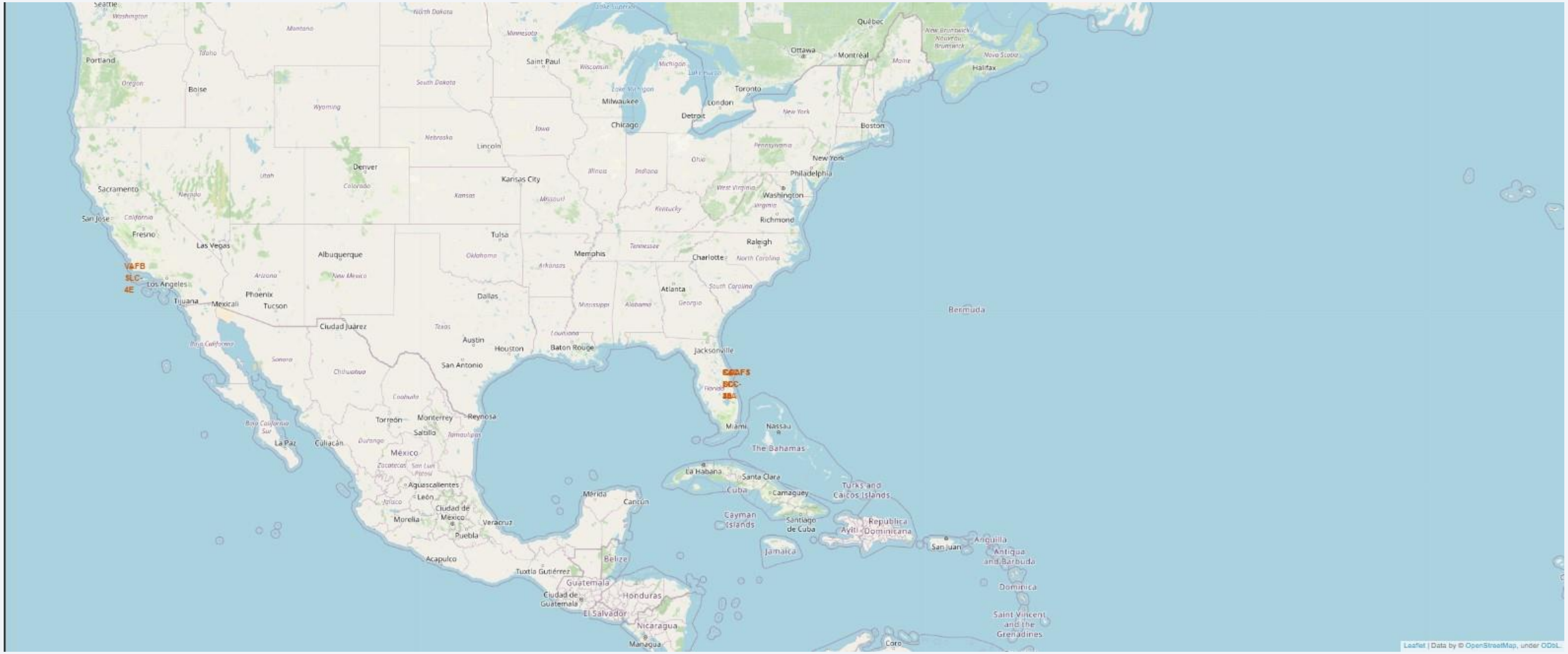
Landing _Outcome	Count_of_outcome
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Controlled (ocean)	3
Failure	3
Failure (parachute)	2
No attempt	1

- The query result above shows the ranked count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Section 3

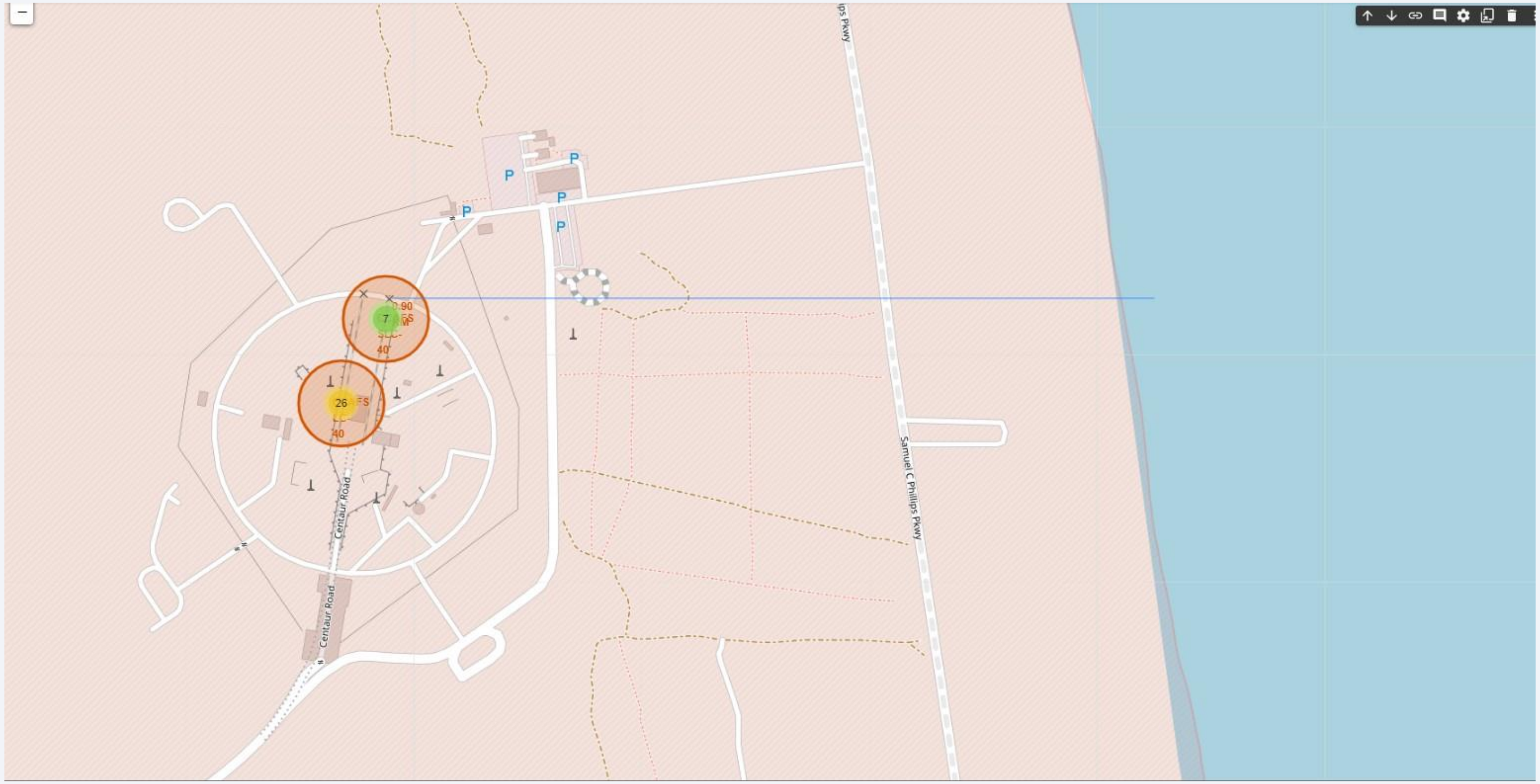
Launch Sites Proximities Analysis

All launch sites global map markers



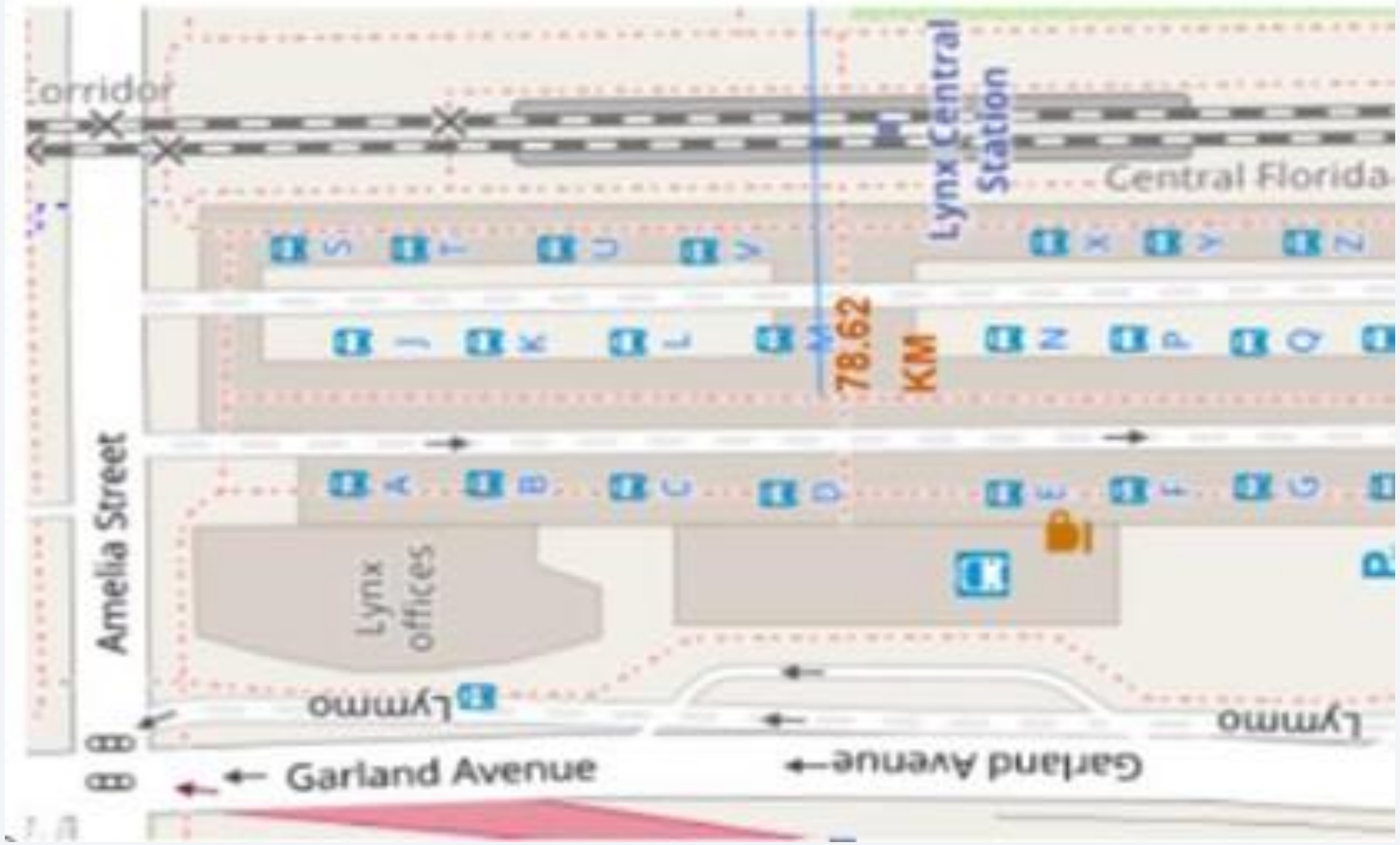
- The map above shows that SpaceX launch sites are in the United States of America coastal states Florida and California

Closest Coastline to Launch Site in Florida



- The map shows the closest coastline and the distance between the coastline point and the launch site in Florida.

Closest Train Station to Launch Site

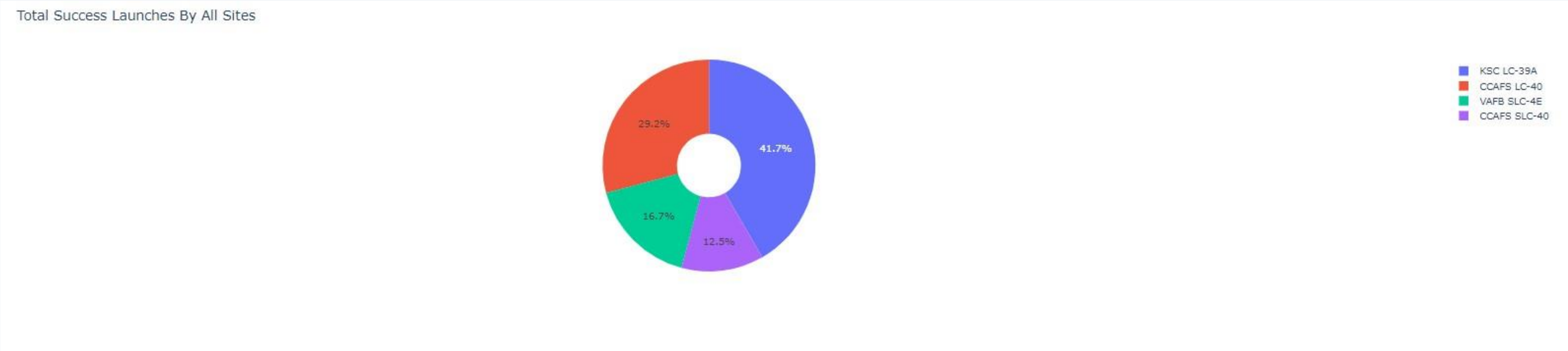


- The map above shows the distance to the train station from the closest launch site

Section 4

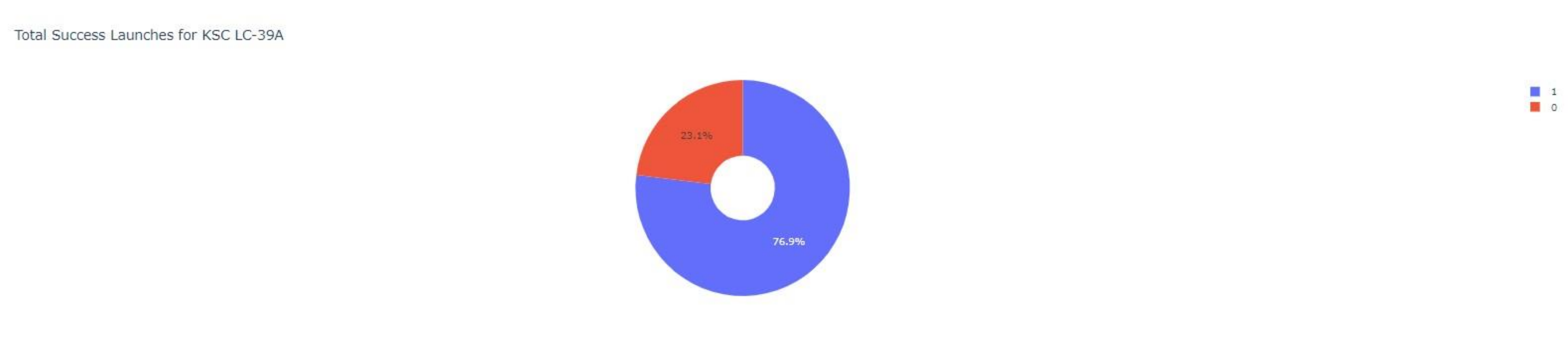
Build a Dashboard with Plotly Dash

Launch Success count for all launch sites



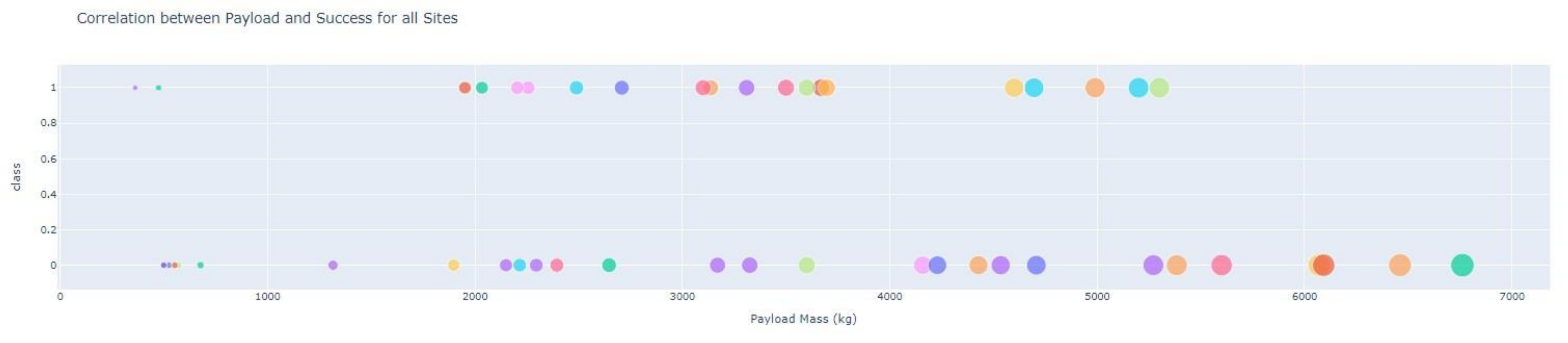
- Here we see that KSC LC-39A had the most successful launches of all sites

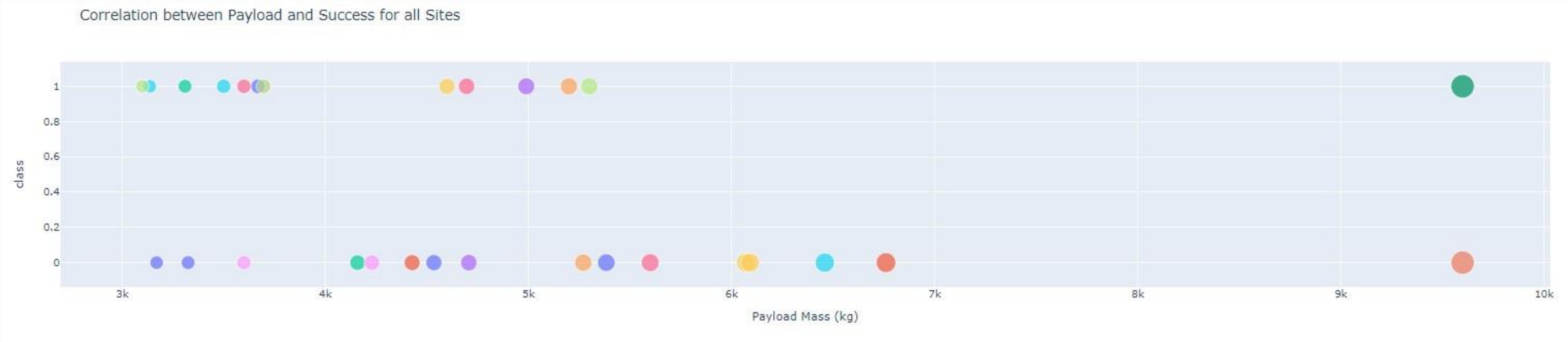
Launch site with highest launch success ratio



- KSC LC-39A achieved an overall success of 76.9% with only 23.1% failures.

Payload vs Launch for all sites





We see that there are more success recorded for lower payloads than higher payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

In [115…

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

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.875
Best params is : {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
```

Decision tree model has the highest classification accuracy

Confusion Matrix of Decision Tree

- The confusion matrix shows that the model correctly distinguishes the successful landing but has a 50% prediction for the unsuccessful landings I.e the false positive.

Conclusions

From our analysis, we can make the following conclusions:

- The large flight numbers result in more successes than the lower flight numbers at the launch sites
- There was a steady increase in the yearly trend of the success rate in the years reviewed
- The orbits ES-L1, GEO, HEO, and VLEO recorded the most successes.
- KSC LC-39A is the site with the most successful launches of any sites
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- The data for the app created in this project can be found here:

https://github.com/gmenshi4/IBM_Coursera_DS_repo/blob/main/spacex_launch_dash.csv

Thank you!