IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shawna Lynn Hobbs
9/19/2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- **Summary of all results**
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the 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 land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Requests.get() was used to retrieve the SpaceX API data. The data was cleaned and formatted using data wrangling techniques.
- Notebook used:
  - [SpaceX Api]()

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/datasets/API_call_spacex_api.json'
```

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

```
[10]: response = requests.get(static_json_url)
      response.status_code
[10]: 200
```

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

```
[11]: # Use json_normalize meethod 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

- Falcon 9 launch records were webscraped with BeautifulSoup.
- Table was parsed and converted it into a pandas dataframe.
- NoteBook used:
  - [Webscraping]()

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

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

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

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

...

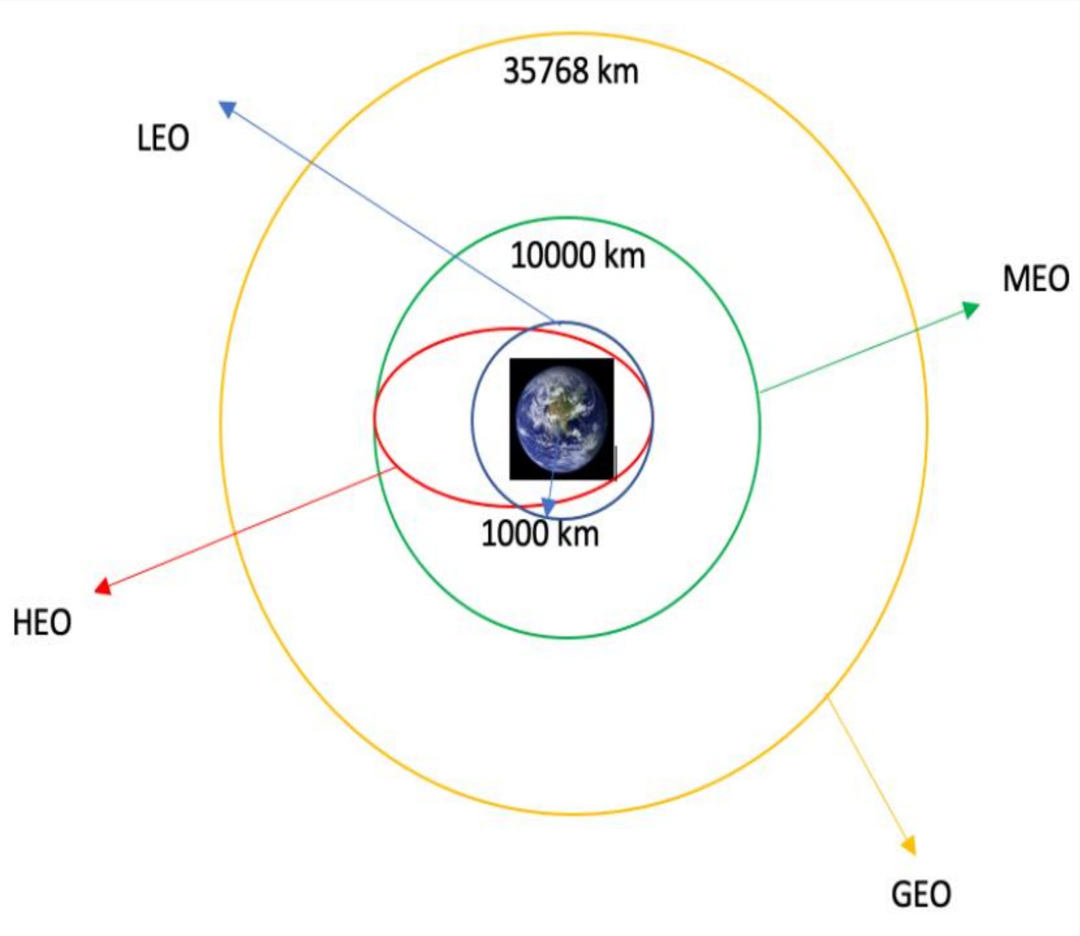
Print the page title to verify if the `BeautifulSoup` object was created properly

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

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

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site
- Also Calculated the number and occurrence of each orbits
- Landing outcome label from outcome column was created and the results Converted to csv.
- Notebook
  - [Data Wrangling]()



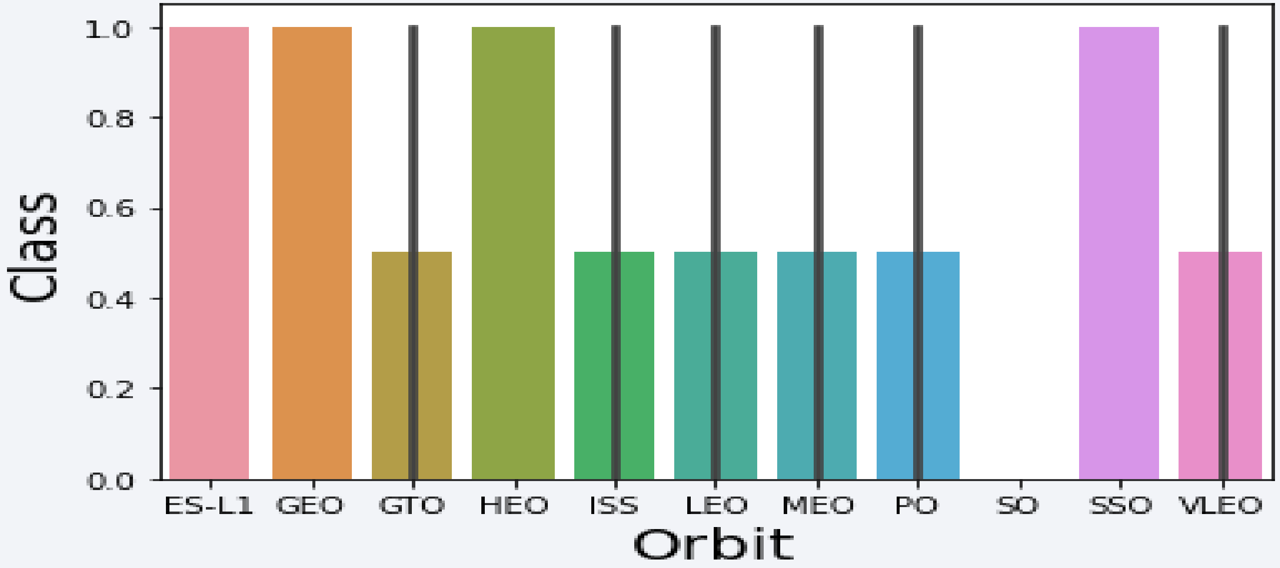
EDA with Data Visualization

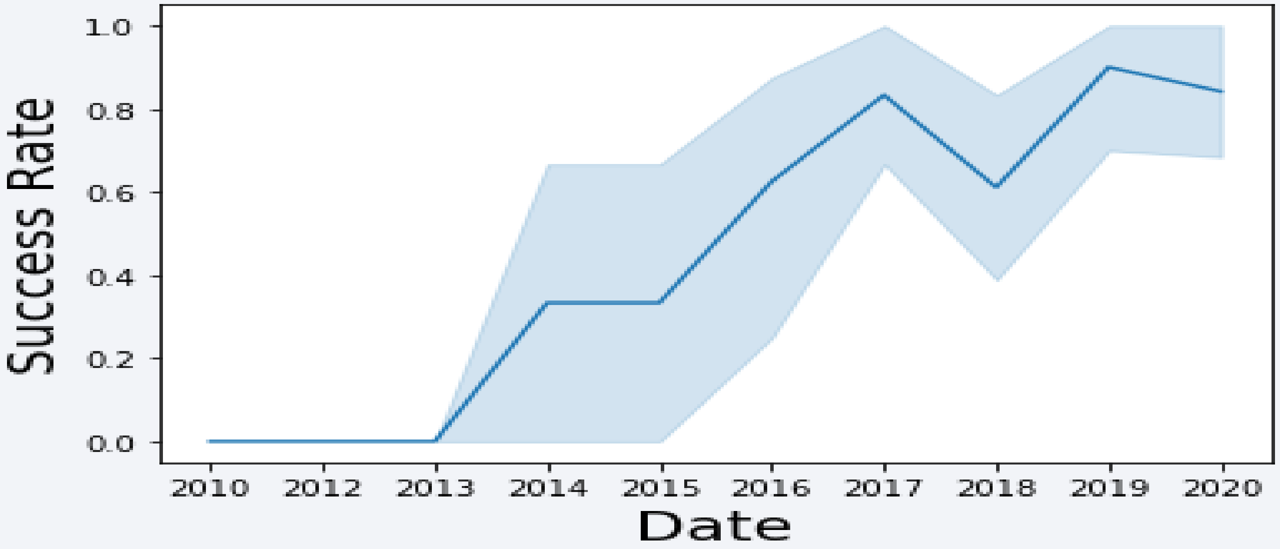
The data was explored visually to find the following relationships:

- Flight Number and Launch Site
- Payload and Launch Site
- Success rate of each orbit type
- Flight Number and Orbit Type
- Launch Success Yearly Trend.

Note Book:

[EDA Visulization]





EDA with SQL

The SpaceX dataset was loaded into a PostgreSQL database without leaving the jupyter notebook and EDA with SQL was applied to get insight from the data.

Queries used to find out for instance:

- Unique Launch site names
- Total payload mass carried
- Average payload mass carried
- Total number of successful outcomes as well as failures
- Failed landing outcomes, Booster Version, and Launch Site Names.

Notebook used:

- EDA With SQL

Build an Interactive Map with Folium

- All launch sites were marked with the following to determine the success or failure of launches for each site, adding map objects such as :
  - Markers
  - Circles
  - Lines
- The feature launch outcomes were marked as a class:
  - 0 for Failure
  - 1 for Success
- Using the color-labeled marker clusters, launch sites with relatively high success rate were noted.
- The distances between a launch site to its proximities were calculated.
- Finally, questions below were answered:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

Notebook Used:

- Folium Map

Build a Dashboard with Plotly Dash

- An interactive dashboard was built using Plotly dash
- Pie charts were Plotted to show the total launches by a certain sites
- Scatter graph was plotted showing the relationship of Outcome and Payload Mass (Kg).

Notebook used:

- Plotly Dashboard
- Note: Issues using skills lab site, this was completed using a standard .py file)

Predictive Analysis (Classification)

- The data was loaded using numpy and pandas, transformed the data, split the data into training and testing using test_train_slipt().
- Different machine learning models were built and tuned to different hyperparameters using GridSearchCV.
- Accuracy was used as the metric for the model, which was then improved using feature engineering and algorithm tuning.
- The best performing classification model was then found.

Notebook used:

- Machine Learning Prediction

Results

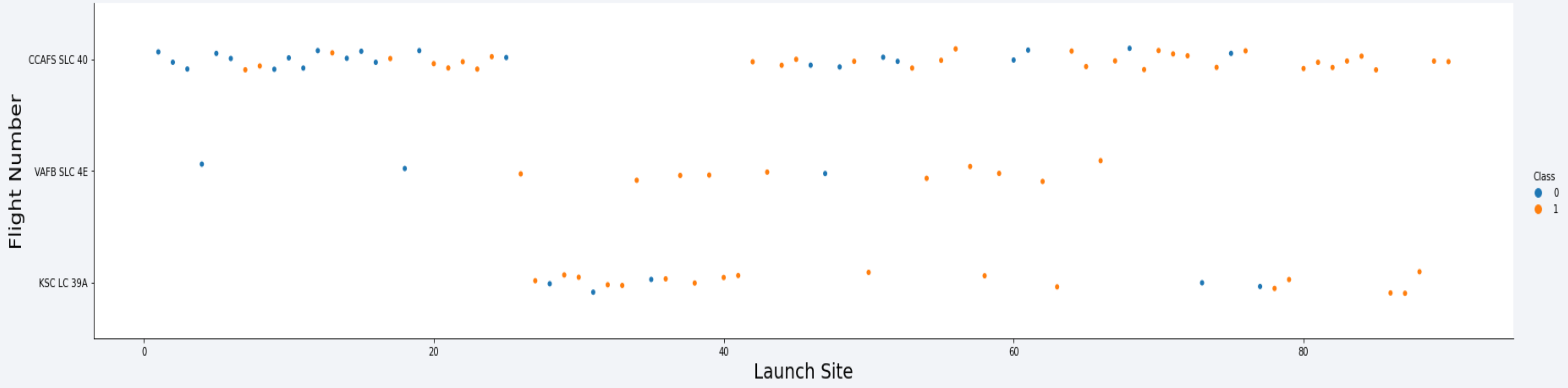
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

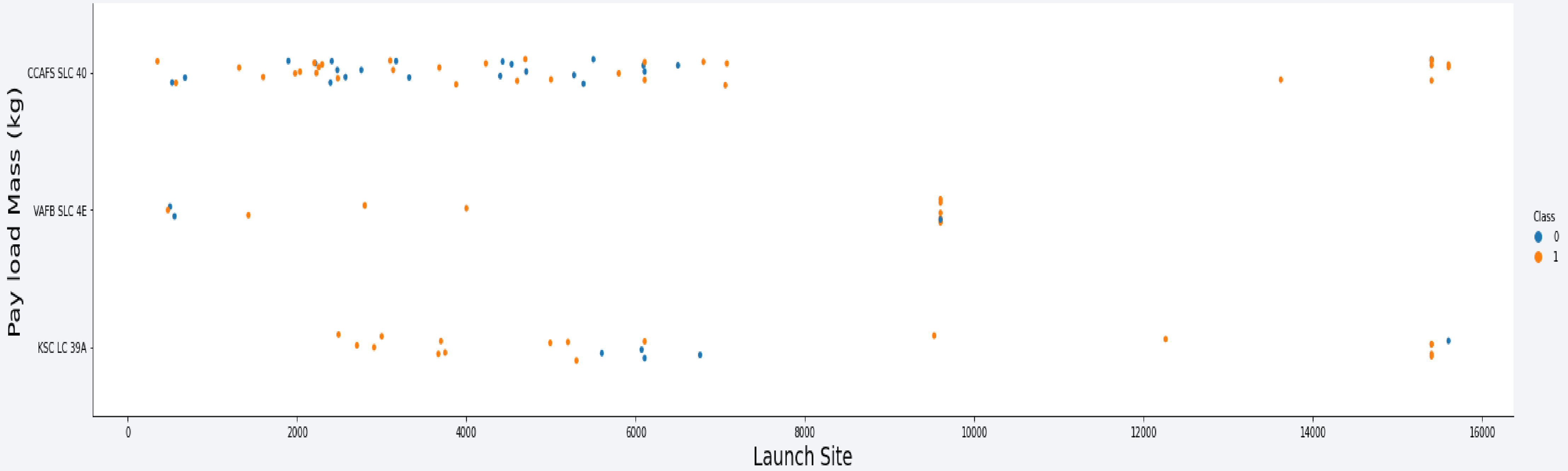
Flight Number vs. Launch Site

- The larger the flight amount at a launch site, the greater the success rate at a launch site was determined via the plot generated below.



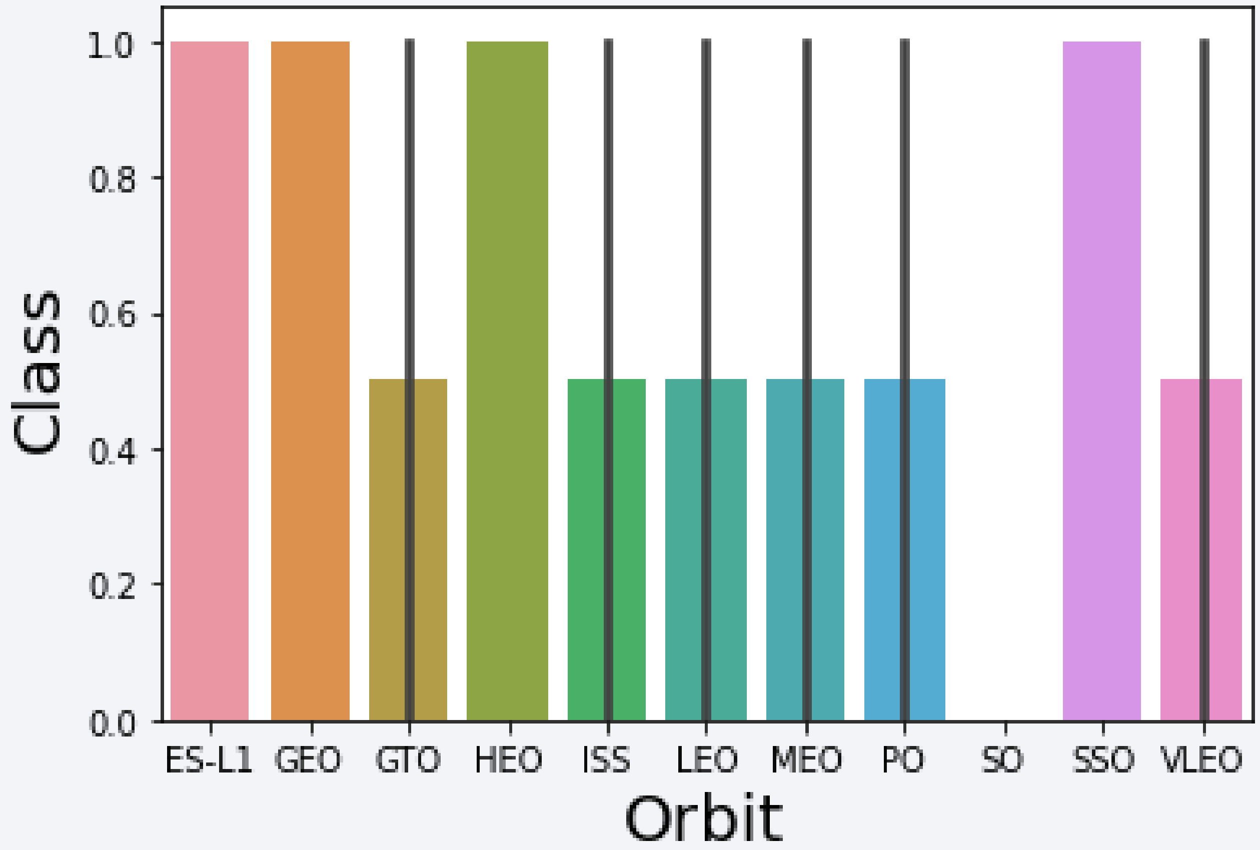
Payload vs. Launch Site

- The Greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



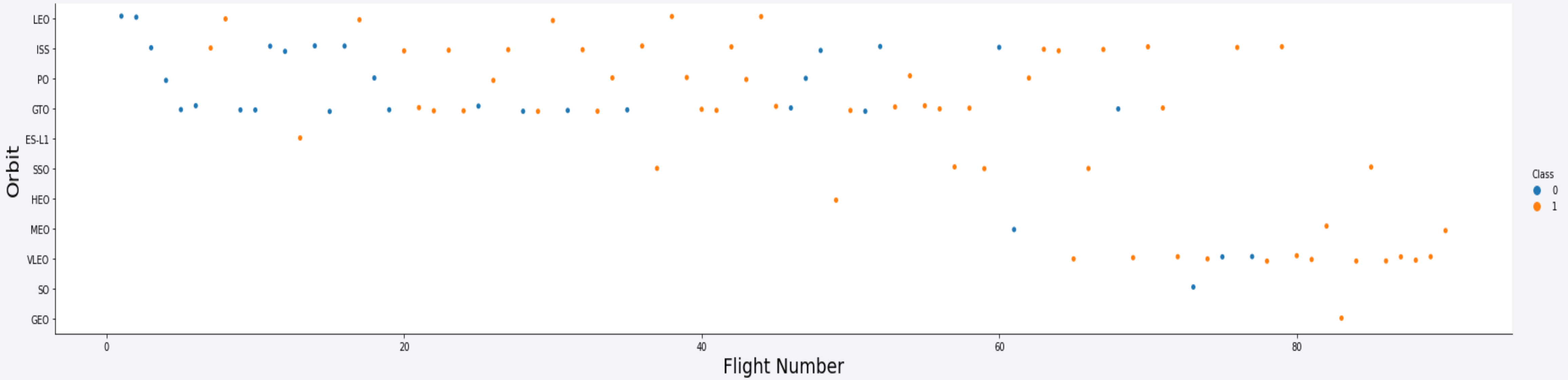
Success Rate vs. Orbit Type

- From the plot, it can be seen that ES-L1, GEO, HEO, SSO had the most success rate.



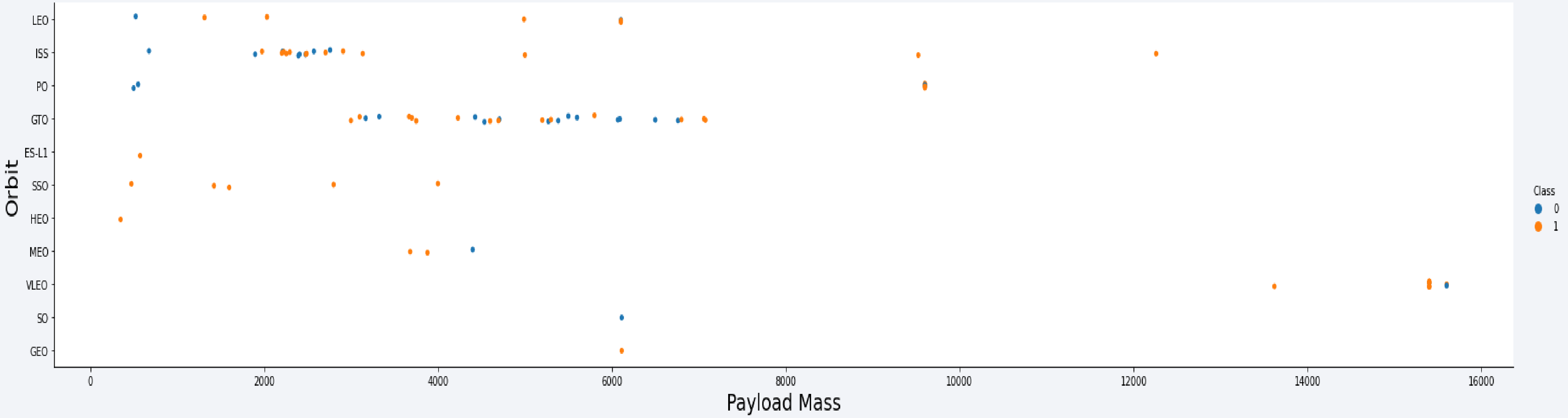
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. Observation concludes that in the LEO orbit, success is related to the number of flights, whereas in the GTO orbit, there is no relationship between flight number and the orbit.



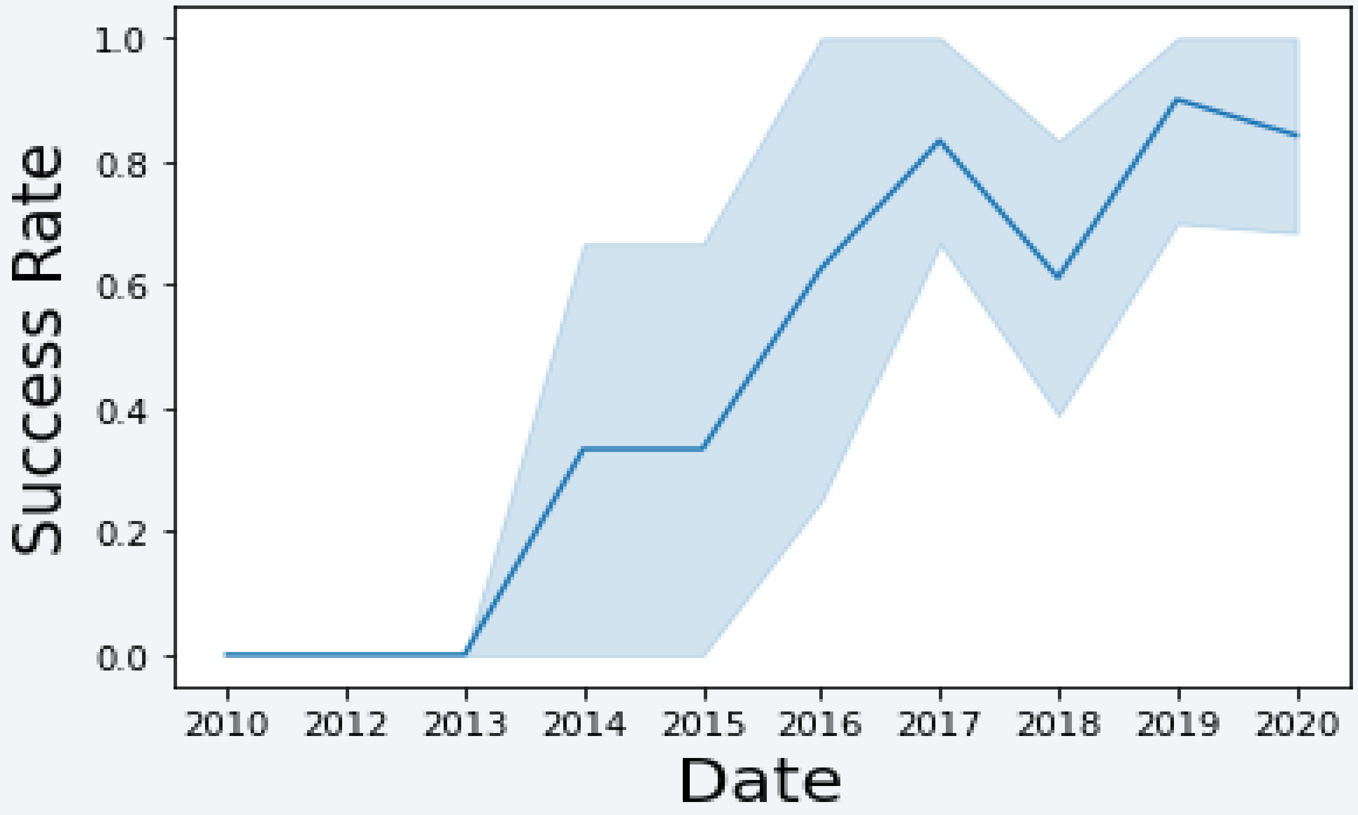
Payload vs. Orbit Type

- Observation dictates that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, observation concludes that success rate since 2013 kept on increasing until 2020.



All Launch Site Names

- Using the SpaceX data base, a Query was launched to find the unique names of the launch sites using **DISTINCT** Command

Task 1 ¶

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

```
[7]: %sql select distinct Launch_Site from SPACEXTBL
```

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

[7]:

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

Launch Site Names Begin with 'CCA'

- Using the query to the right, Five launch site records beginning with `CCA` were displayed.

```
[8]: %sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5

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

[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

Total Payload Mass

- Below is the results of calculating the overall payload mass of the boosters launched by NASA.

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

```
[10]: %sql select sum(payload_mass__kg_) from SPACEXTBL WHERE customer = 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.

[10]: sum(payload_mass_kg_)
                    45596
```

Average Payload Mass by F9 v1.1

- Calculated below is the average payload mass carried by booster version F9 v1.1

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

```
[11]: %sql select avg(payload_mass__kg_) from SPACEXTBL WHERE booster_version = 'F9 v1.1'

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

[11]:

avg(payload_mass__kg_)
2928.4

First Successful Ground Landing Date

- Below is the result of finding the dates of the first successful landing outcome on ground pad.

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

Hint:Use min function

```
[12]: %sql select min(DATE) from SPACEXTBL WHERE `Landing _Outcome` = 'Success (ground pad)'

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

[12]:

min(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- The boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are available below.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[13]: %sql select booster_version from SPACEXTBL where `Landing _Outcome` = 'Success (drone ship)'\
      and payload_mass__kg_ between 4000 and 6000

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

[13]:

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

Total Number of Successful and Failure Mission Outcomes

- The calculated total number of successful and failure mission outcomes is available in the screenshot below.

List the total number of successful and failure mission outcomes

```
[14]: %sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP BY mission_outcome
 * sqlite:///my_data1.db
Done.
```

[14]:

Mission_Outcome	count(mission_outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The subquery to the right was used to determine the boosters that have carried the maximum payload using **WHERE** and **MAX()**.

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

[15]:
```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL\
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

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

[15]:

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

2015 Launch Records

- The List below is of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 including dates.

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.

[61]:
```
%sql select booster_version, launch_site, `Landing _Outcome`, Date FROM SPACEXTBL\
WHERE `Landing _Outcome` LIKE 'Failure (drone ship)'\
AND Date BETWEEN '01-01-2015' AND '12-31-2015'
```

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

[61]:

Booster_Version	Launch_Site	Landing _Outcome	Date
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	10-01-2015
F9 FT B1020	CCAFS LC-40	Failure (drone ship)	04-03-2016

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

- Ranking 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 is previewed in the image to the right.

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

[84]:
```
%sql select count(`Landing _Outcome`), `Landing _Outcome`, Date from SPACEXTBL\
where Date between '04-06-2010' and '20-03-2017'\
group by `Landing _Outcome`\
order by count(`Landing _Outcome`) desc
```

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

[84]:

count(`Landing _Outcome`)	Landing _Outcome	Date
20	Success	07-08-2018
10	No attempt	08-10-2012
8	Success (drone ship)	08-04-2016
6	Success (ground pad)	18-07-2016
4	Failure (drone ship)	10-01-2015
3	Failure	05-12-2018
3	Controlled (ocean)	18-04-2014
2	Failure (parachute)	04-06-2010
1	No attempt	06-08-2019

Section 3

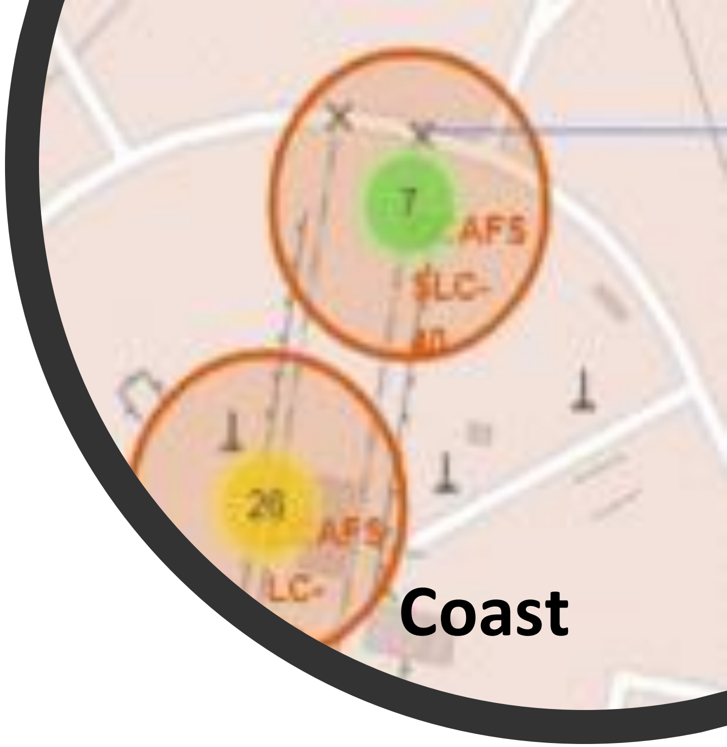
Launch Sites Proximities Analysis

Folium Generated Launch Site Map

- Located on the folium map presented here are the locations of the lauch sites centralized near Los Angels, California and near Orlando, Florida.

Color-Labeled Launch Site Outcomes

- The Four Screenshots on this slide show the color coordinates outcomes for the four sites various launches.



Nearest Railway Station, Highway, Coast, Coastline and City

Distances to various Attributes

Section 4

Build a Dashboard with Plotly Dash

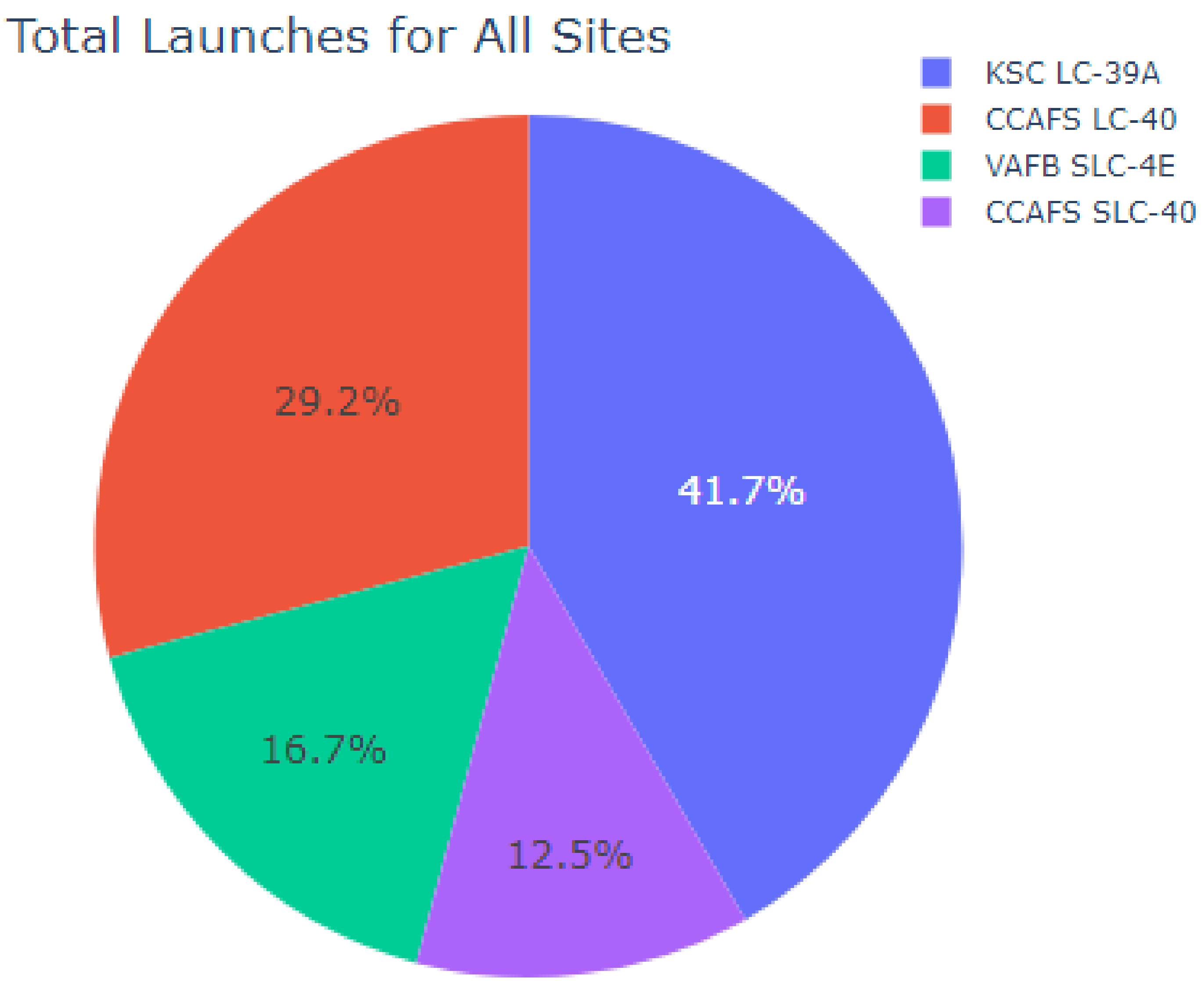
Launch Success Count (all sites)

The Graphic Show it is apparent that the Most successful Launch site is:

KSC LC-39A

While the least Successful launch site is:

CCAFS SLC-40



KSC LC-39A Launch Site Results

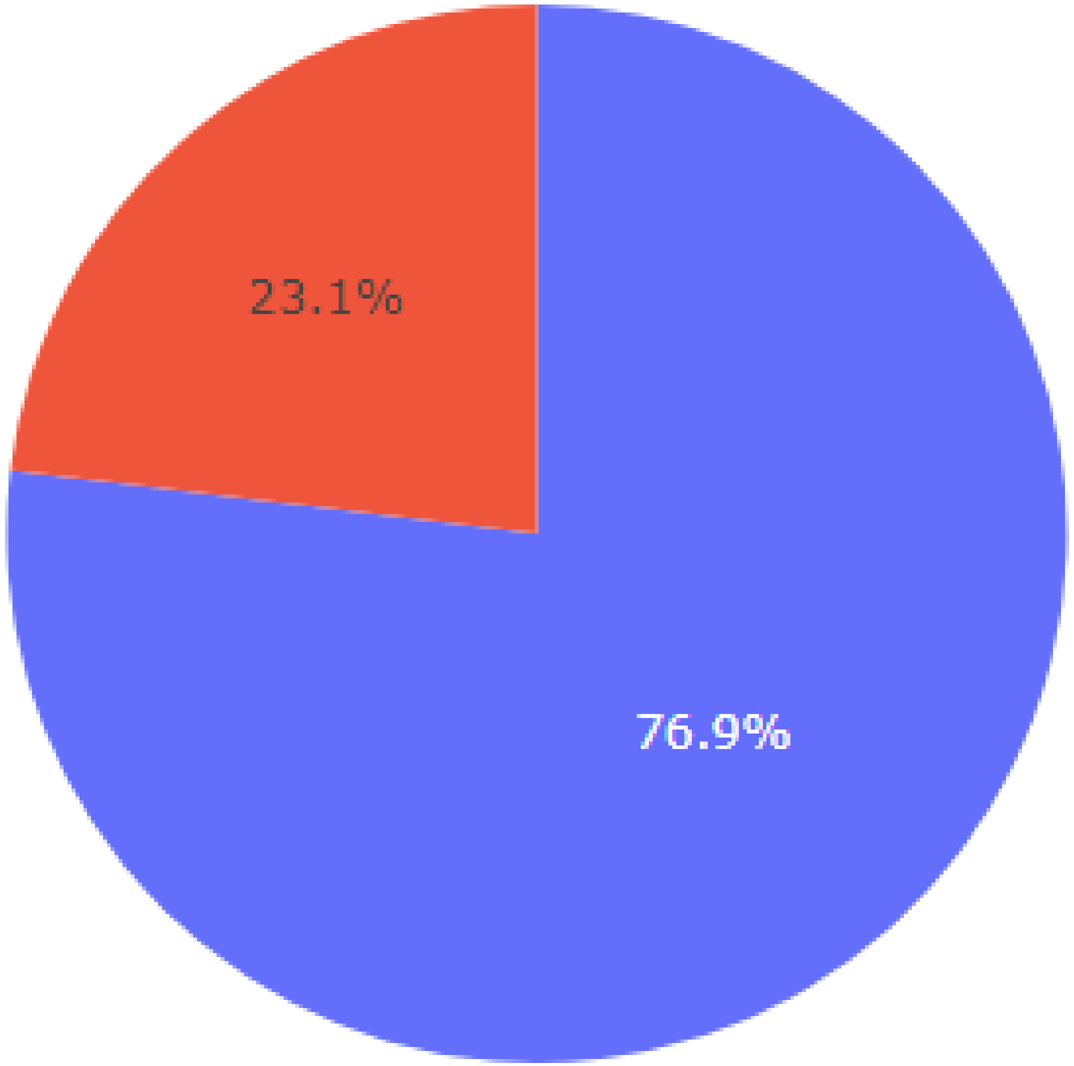
Depicted are the results of the KSC LC-39A Launches.

The Successes Denoted in:

BLUE

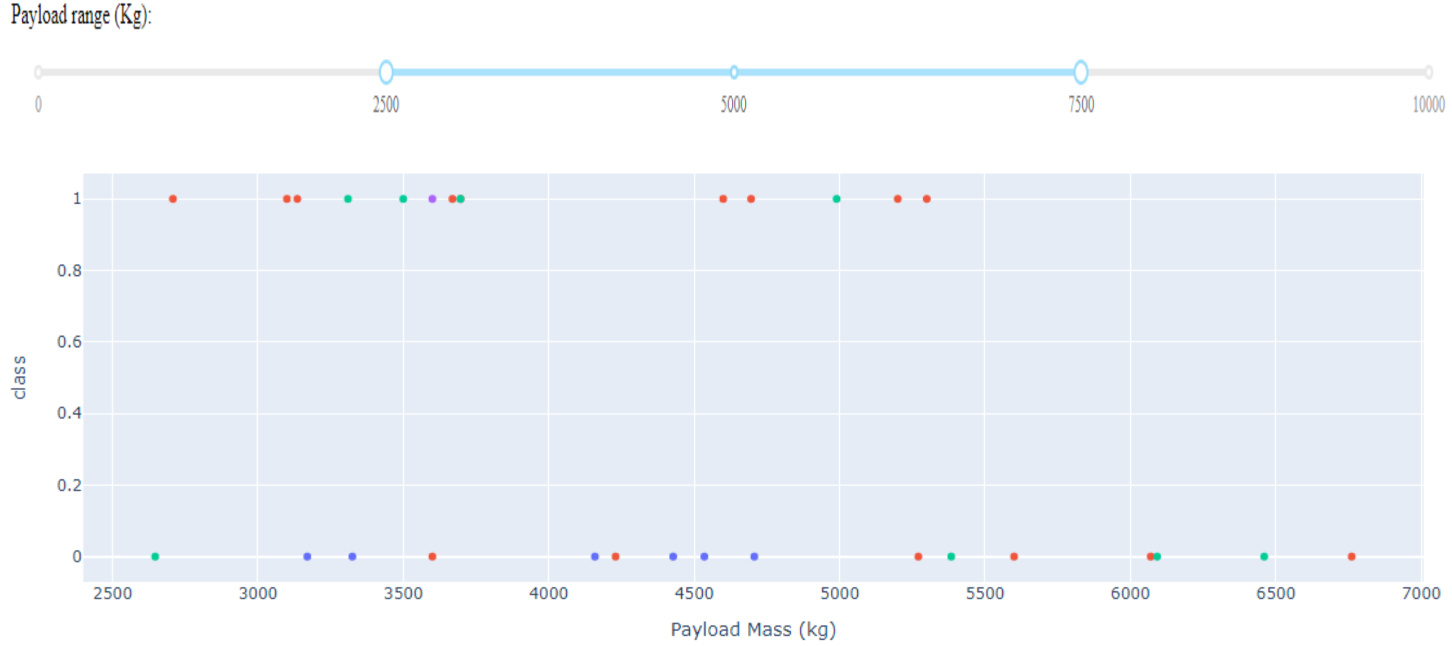
The Failures Denoted in:

RED



Payload Mass Vs. Launch Oustcome

The Top graph is set to the default payload range while the bottom is set to the range of 2,500 – 7,500

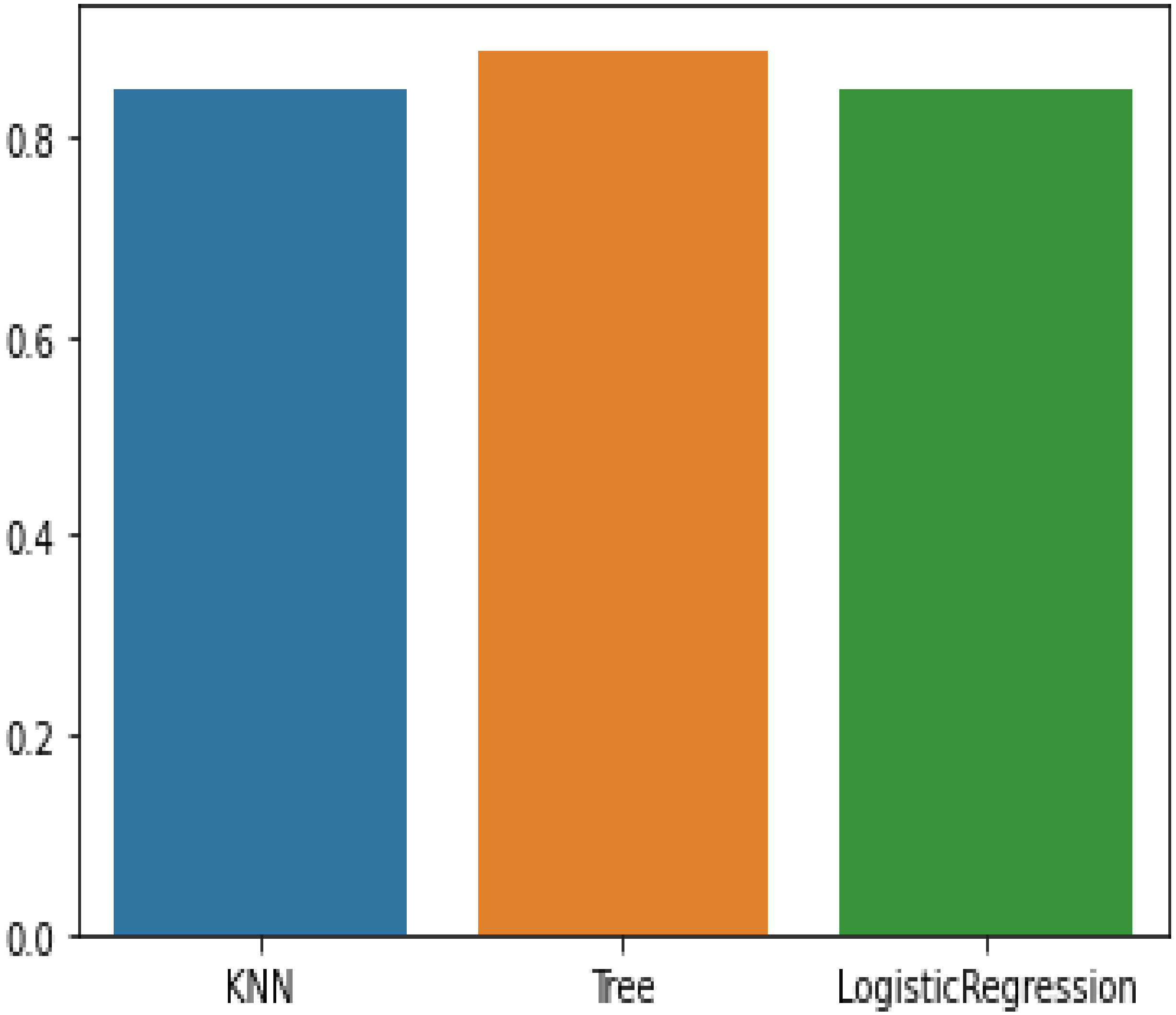


Section 5

Predictive Analysis (Classification)

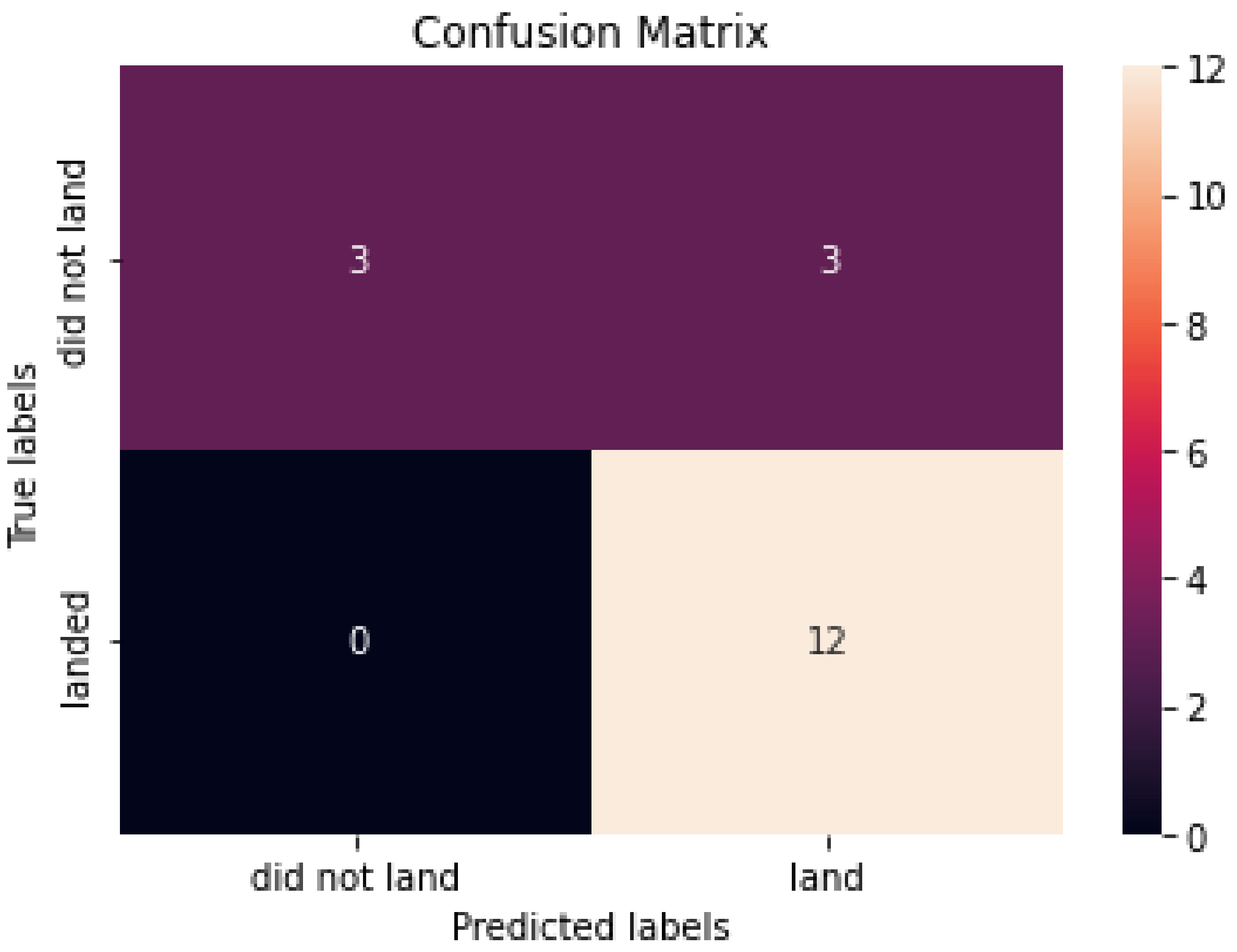
Classification Accuracy

As is show on this bar graph the accuracy of the models vary only slightly but it is clear that the Decision Tree model was the most accurate.



Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- It can be concluded that
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!