IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Johnson Chishimba

23/05/2022

Outline

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

Executive Summary

- Summary of methodologies
  - ✓ Data collection through API
  - ✓ Data collection with web scrapping
  - ✓ Data wrangling/cleaning
  - ✓ Exploratory data analysis with SQL
  - ✓ Exploratory data analysis with folium
  - ✓ Prediction with machine leaning
- Summary of all results
  - ✓ Results for exploratory data analytics
  - ✓ Interactive analytics results (with screen shots)
  - ✓ Predictive analysis results

Introduction

- Project background and context

  SpaceX advertised 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 saving is because SpaceX can reuse the first stage. Hence, determining if the first stage will land, can help us determine the cost of a launch. This information can be used if an alternative company wants to bid against SpaceX for a rocket launch. Therefore, the goal of this project is to create a machine learning pipeline that can be used to predict if the first stage will land successfully.

- Problems that need answers

  ✓ What operating conditions are required for a successful rocket landing?

  ✓ What environment conditions influence a successful rocket landing?

  ✓ What features from the data affect the success rate of rocket landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Features to be used for future landing success predictions were selected.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models were developed in Jupyter notebook, the accuracy of each model was determined. The model with the highest accuracy was adopted.

Data Collection

- Describe how data sets were collected.
  - ✓ Data collection was done using the get request to the SpaceX API.
  - ✓ Then the response content was decoded as the json using .json() function call. Then it was turned into a pandas dataframe using .json_normalize().
  - ✓ After this, data wrangling was performed to check for missing values and then filling in of missing values where necessary was done.
  - ✓ Web scrapping from Wikipedia for Falcon 9 launch records was done using BeautifulSoup.

Data Collection – SpaceX API

- The get request was used to the SpaceX API to collect the data. Then data wrangling was done as well as formatting were necessary.

- Below is the GitHub URL: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/DATA%20COLLECTION.ipynb

In [96]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-
_spacex_api.json'
```

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

In [97]:
```
response.status_code
```

Out[97]:
```
200
```

Now we decode the response content as a Json using `.json()` a

In [98]:
```
# Use json_normalize meethod to convert the json resu
data = response.json()
pd.json_normalize(data)
```

Out[98]:

static_fire_date_utc	static_fire_date_unix	net	window

Data Collection - Scraping

- Web scraping was performed to collect falcon 9 historical launch records from wikipedia using BeautifulSoup.
- The GitHub URL is given below: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/DATA%20COLLECTION%20WITH%20WEB%20SCRAPING.ipynb

In [9]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falc
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

In [10]:
```
# 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`

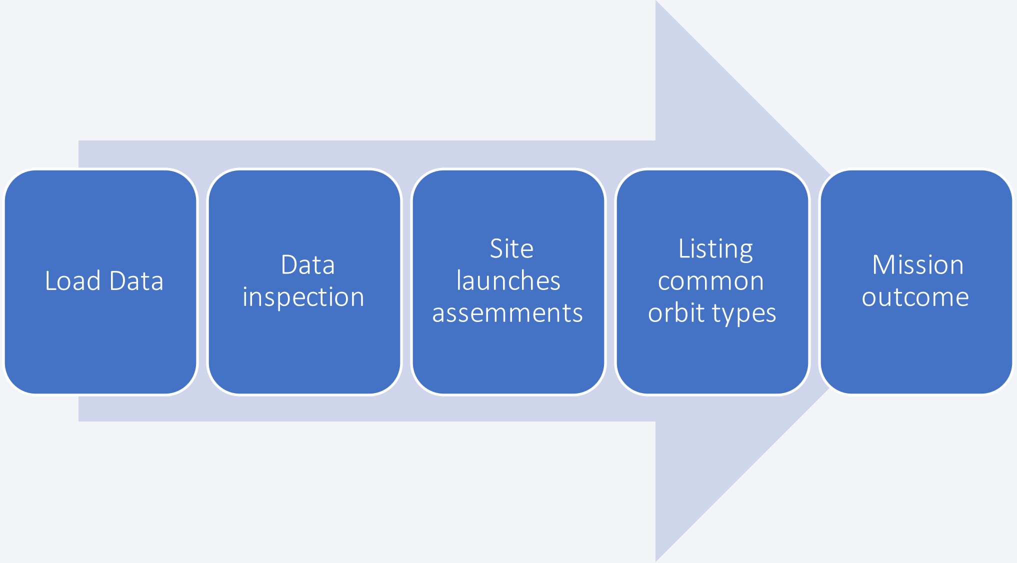
In [11]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a respons
soup = BeautifulSoup(response.content, "html.parser")
```

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

In [12]:
```
# Use soup.title attribute
soup.title
```

Data Wrangling

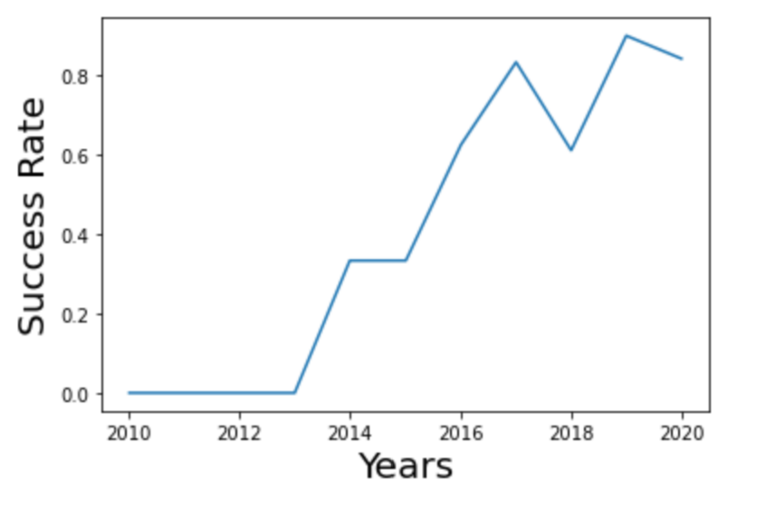
- Exploratory data analysis was conducted and the labels for training supervised models were identified.
- Data wrangling process was done as shown in the flow chart below.

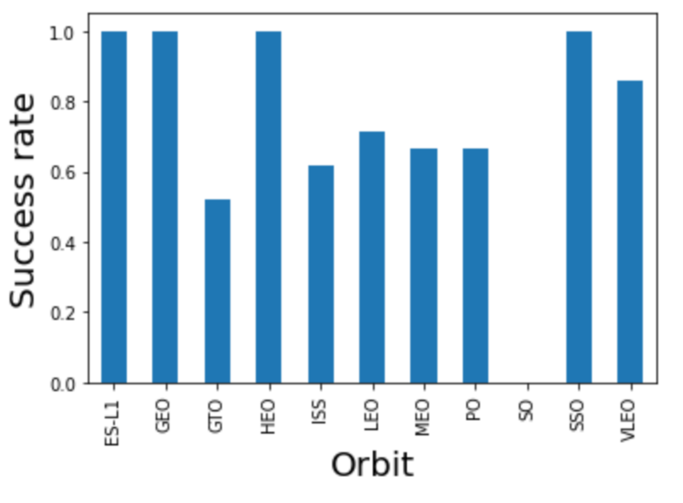


- The GitHub URL is as follows: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/DATA%20COLLECTION%20WITH%20WEB%20SCRAPPING.ipynb

EDA with Data Visualization

- Data visualization was conducted by showing the relationship between Flight Number and Launch Site, Payload and Launch Site, success rate of each orbit type, Flight Number and Orbit type, Payload and Orbit type and the launch success yearly trend.





- The GitHub URL is as follows: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/DATA%20COLLECTION%20WITH%20WEB%20SCRAPPING.ipynb

EDA with SQL

- A SQL extension was loaded, and a connection with a database was established.
- The following queries were written to get insight from the data:
  - ✓ The names of unique launch sites in the space mission.
  - ✓ The total payload mass carried by boosters launched by NASA.
  - ✓ 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 GitHub URL is as follows: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/jupyter-labs-eda-sql-edx_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- All Launch sites were marked and map objects such as circles, markers and lines were added to show the successful and failed launches for each site on a folium map.
- To show the successful and failed launches, launch outcomes feature was created whereby class 1 signified successful launch while o signified failed launches.
- The color label marker cluster was used to indicate which launch sites had a higher success rate.
- The following is a GitHub URL of my completed interactive map with Folium map: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

- Interactive dash board was built using Plotly dash.
- Total launches by different sites where plotted using the pie charts.
- Scatter plots were done to show the relationship between outcome and payload mass (Kg) for different booster version.
- The GitHub URL of the completed Plotly Dash lab is as follows: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data was loaded using numpy and pandas, the data was transformed and split into training and testing data.
- Different machine learning models were developed and hyperparameters were tuned using GridSearchCV.
- Accuracy was used as the metric for our model. The model was improved using feature engineering algorithm tuning.
- Then, the best performing classification model was selected.
- The GitHub URL shows the completed predictive analysis lab: https://github.com/Mapalo90/SpaceX-Rocket-Launch/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

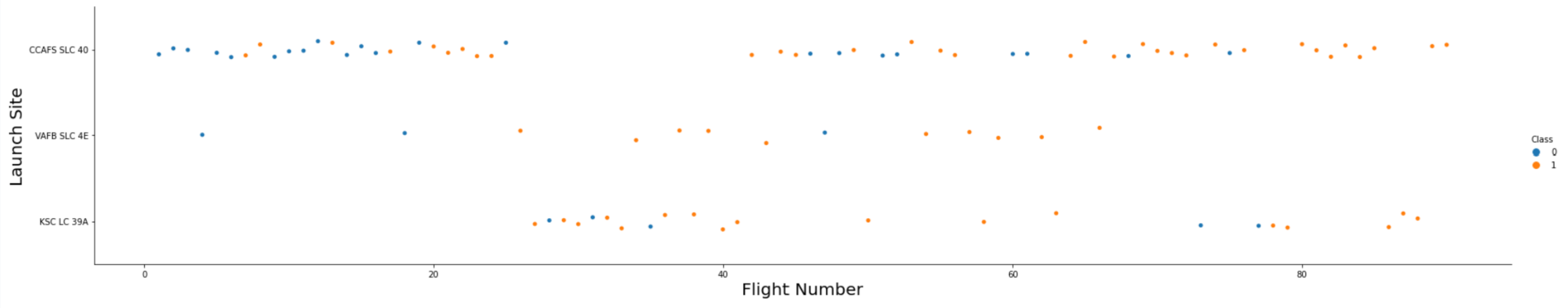
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

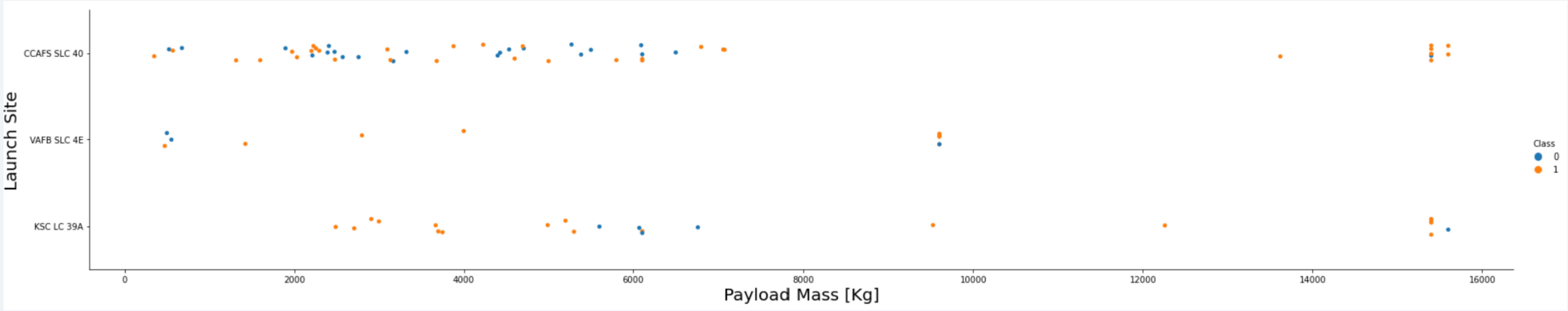
- The scatter plot of Flight Number vs. Launch Site is as shown below:



- The graph shows that the more the Flight Number increases, the higher the launch success rates.

Payload vs. Launch Site

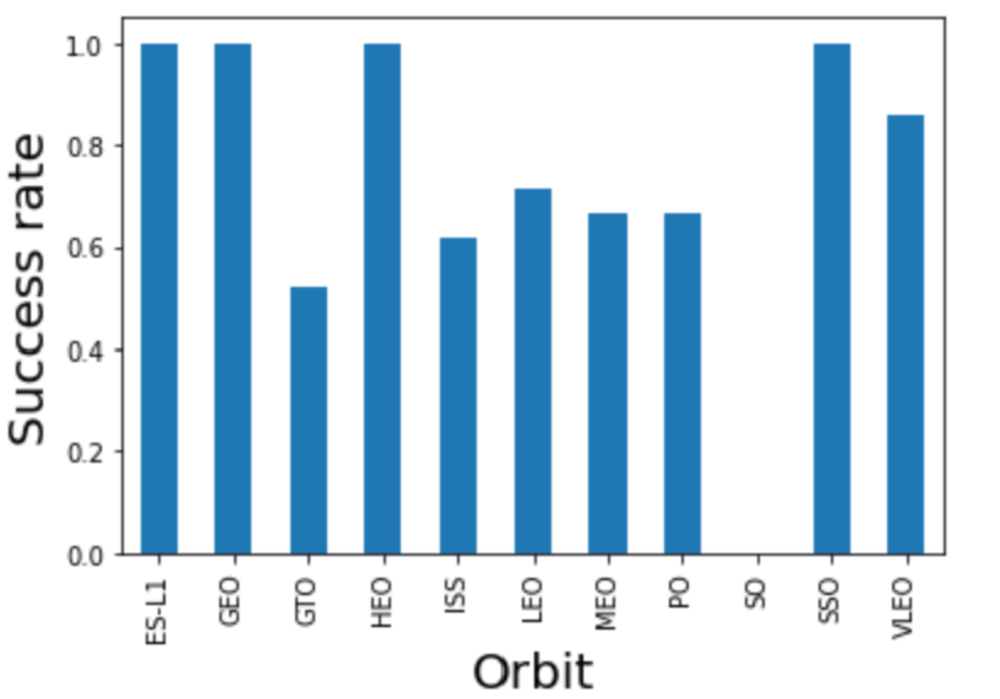
- Below is a scatter plot showing Payload vs. Launch Site



- There is no clear relationship between the success rates and the payload mass

Success Rate vs. Orbit Type

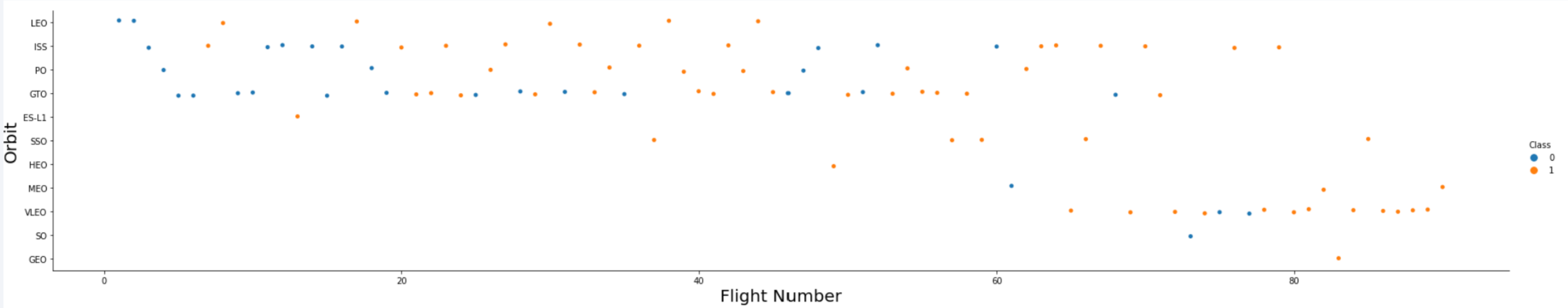
- A bar chart for the success rate of each orbit type is shown below:



- It can be seen that ES-L1, GEO, HEO and SSO had higher success rates.

Flight Number vs. Orbit Type

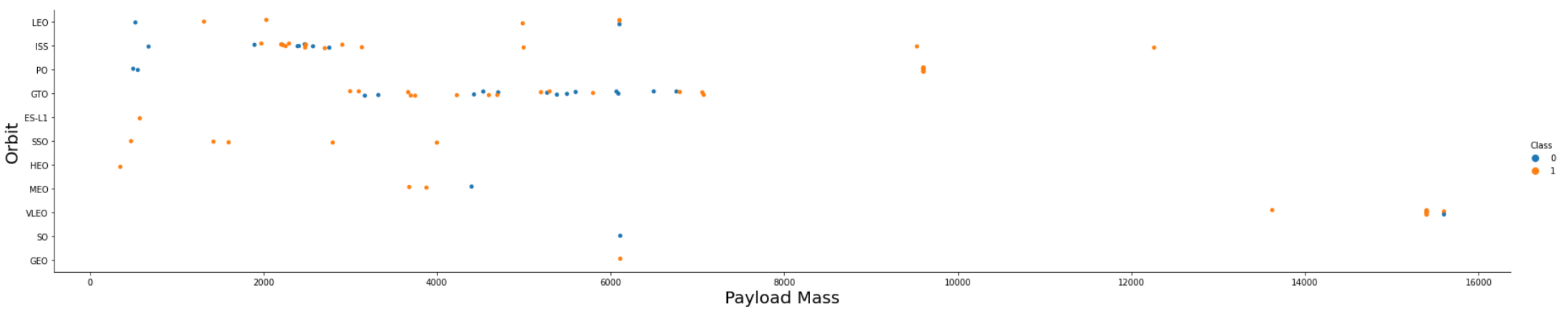
- Below is a scatter point of Flight number vs. Orbit type



- The success rate from LEO increased with Flight Number while GEO showed no relationship with Flight Number.

Payload vs. Orbit Type

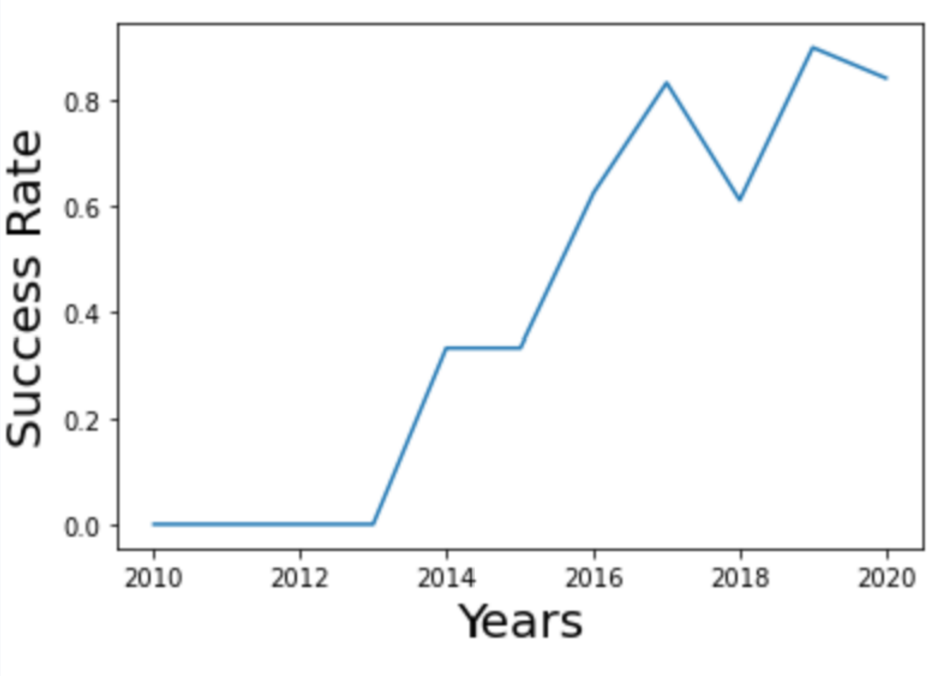
- A scatter point of payload vs. orbit type is shown below



- A relationship can be seen between increasing Payload Mass and Orbit type for Orbit types such as LEO, ISS, PO but no clear relationship can be seen for Orbit types such as GTO and those under it.

Launch Success Yearly Trend

- The figure below shows a line chart of yearly average success rates.



- It can be seen that the success rate started to increase after 2013 till 2020.

All Launch Site Names

- The key word DISTINCT was used to unique launch sites from SpaceX data.

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

In [10]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SpaceXTBL
```

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

Out[10]:

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

Launch Site Names Begin with 'KSC'

- The query below was used to find 5 records where launch sites' names start with `KSC`

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

In [70]:

```
%sql SELECT * FROM SpaceXTBL WHERE LAUNCH_SITE LIKE 'KSC%' LIMIT 5
```

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

Out[70]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- The calculated total payload carried by boosters from NASA was 48213 Kg

```
In [71]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SpaceXTBL WHERE Customer LIKE 'NASA (CRS)%'

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

Out[71]:

TOTAL_PAYLOAD_MASS
48213

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was calculated as 2928.4 Kg. WHERE was used to locate only the F9 v1.1 data.

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

```
In [72]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SpaceXTBL WHERE Booster_Version = 'F9 v1.1'

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

Out[72]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The dates of the first successful landing outcome in drone ship were queried as below:

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

In [23]:
```
%sql SELECT Date FROM SpaceXTBL WHERE "Landing _Outcome" = 'Success (drone ship)'
```

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

Out[23]:

Date
08-04-2016
06-05-2016
27-05-2016
14-08-2016
14-01-2017
30-03-2017
23-06-2017
25-06-2017
24-08-2017
09-10-2017
11-10-2017
30-10-2017
18-04-2018
11-05-2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- The list of names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 was

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

```
In [24]: %sql SELECT Booster_Version FROM SpaceXTBL WHERE "Landing _Outcome" = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

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

Out[24]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- The calculated the total number of successful and failure mission outcomes are as shown below. GROUP BY was used to ensure that COUNT was categorical.

In [13]:

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

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

Out[13]:

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

Boosters Carried Maximum Payload

- A list of names of the booster which have carried the maximum payload mass was queried as below. MAX was used in the subquery in order to locate the max payload mass with reference to the booster version.

Task 8

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

In [14]:
```
%sql SELECT Booster_Version FROM SpaceXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SpaceXTBL)
```

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

Out[14]:

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

- A list of records displayed below shows the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017. It was queried as below.

```
In [26]: %sql SELECT substr(Date, 4, 2), "Landing _Outcome", Booster_Version, Launch_Site FROM SpaceXTBL WHERE (subst

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

Out[26]:

substr(Date, 4, 2)	Landing _Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- A rank of the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order is shown below. LIKE "%S%" AND NOT LIKE "%F%" were used to focus on the successful outcomes and not the failures

In [46]:
```
%%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") AS LAUNCH_COUNT FROM SpaceXT
WHERE (Date BETWEEN '04-06-2010' AND '20-03-2017' AND "Landing _Outcome" LIKE "%S%" AND
GROUP BY "Landing _Outcome" ORDER BY 'LAUNCH_COUNT' DESC
```

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

Out[46]:

Landing _Outcome	LAUNCH_COUNT
Success (ground pad)	6
Success (drone ship)	8
Success	20

Section 3

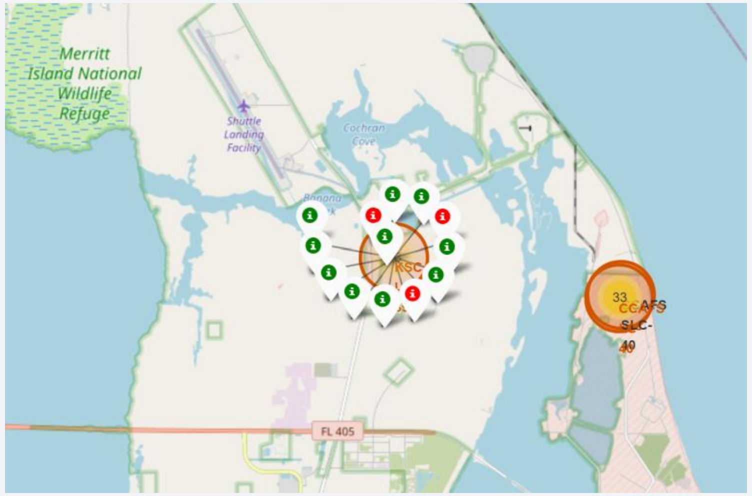
Launch Sites Proximities Analysis

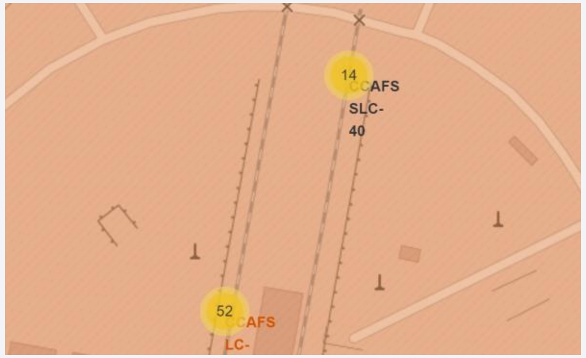
Map showing all launch sites

- It can be seen that most successful launches occurred at the coasts of Florida and California in America

Colourful markers showing launch sites

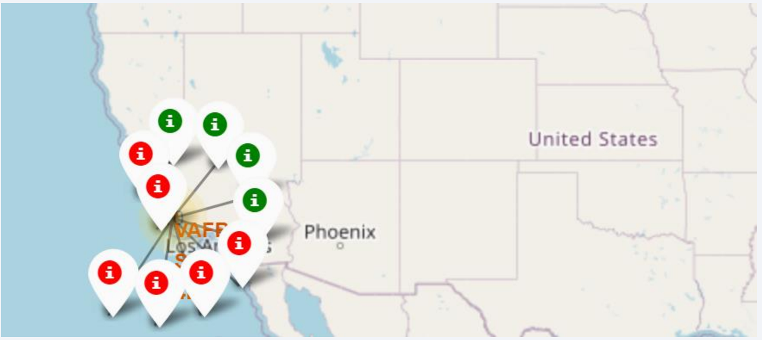
Florida



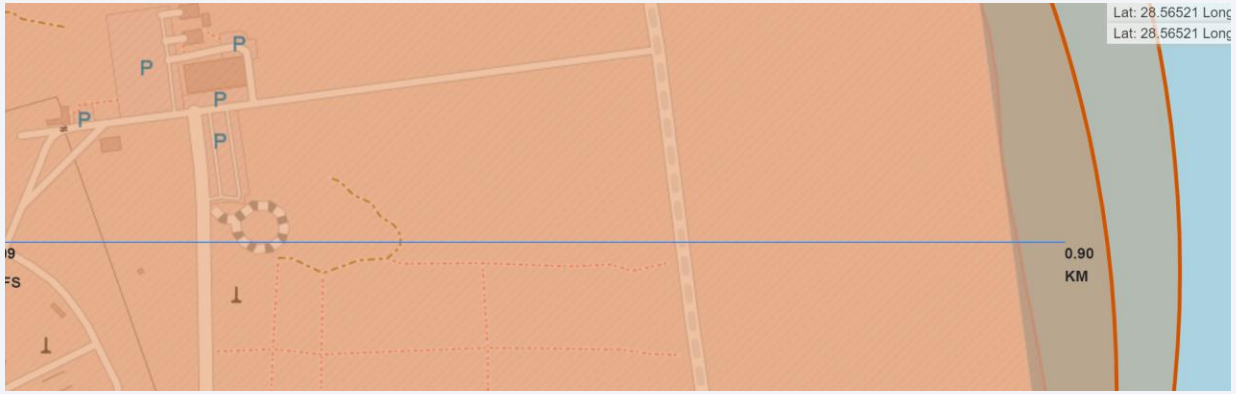


- Green shows successful launches
- Red shows failed launches

California



Distance of launch sites from landmarks



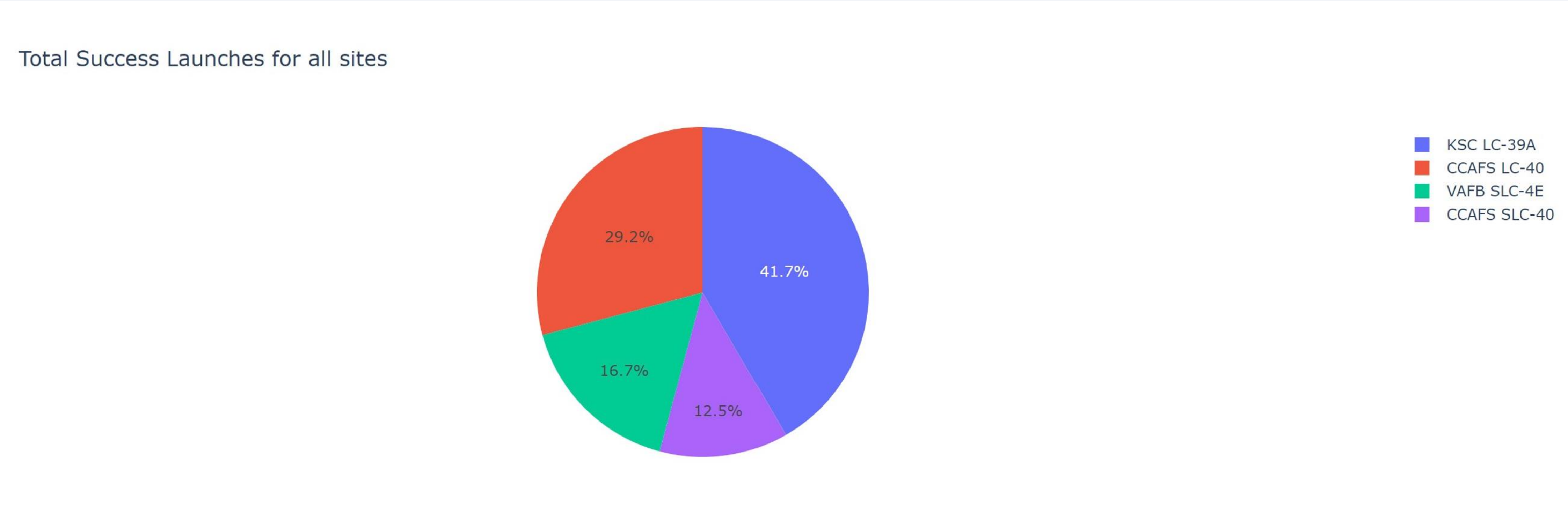
- Are launch sites in close proximity to rail lines? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coast lines? Yes
- Do launch sites keep certain distance away from main cities? Yes

Section 4

Build a Dashboard with Plotly Dash

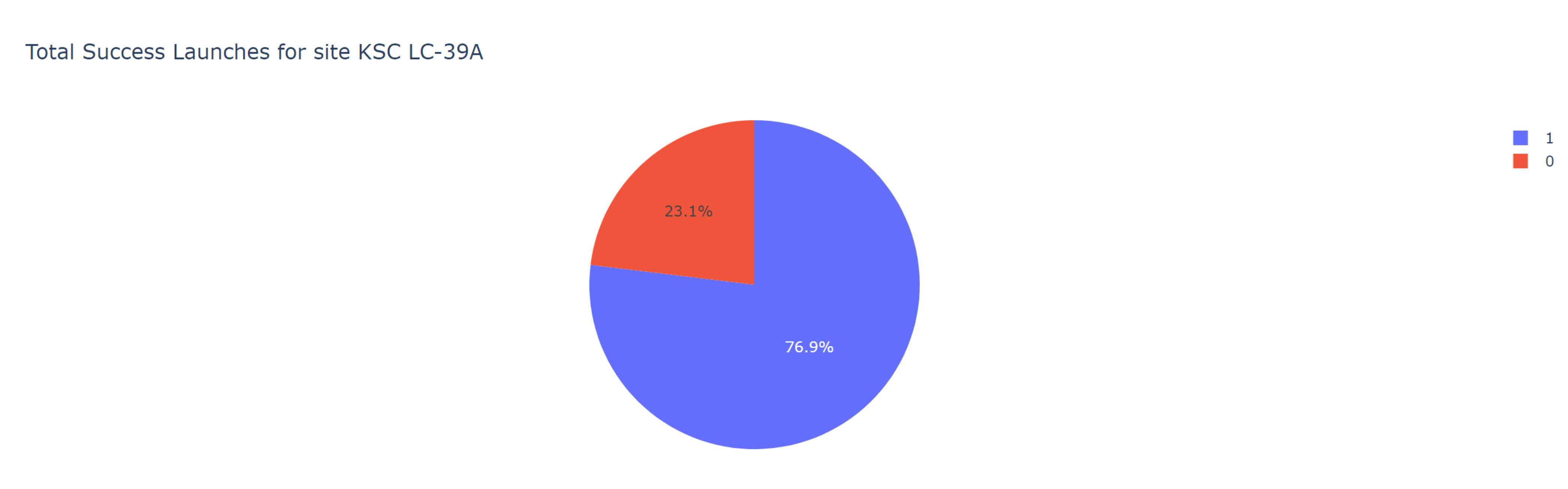
Success achieved by each launch site

- It can be seen that the KSC LC-39A had more successful launches



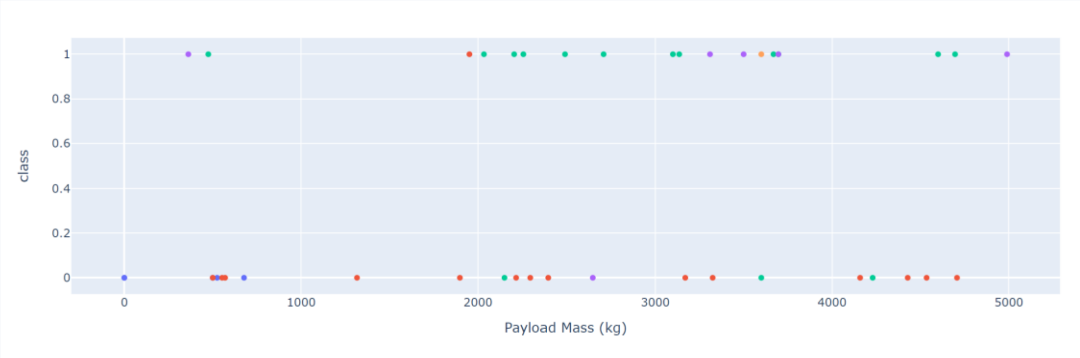
Launch Site Success Ratio at KSC LC-39A

- The success rate at KSC LC-39A was 76.9 % while the failure rate was 23.1 %.

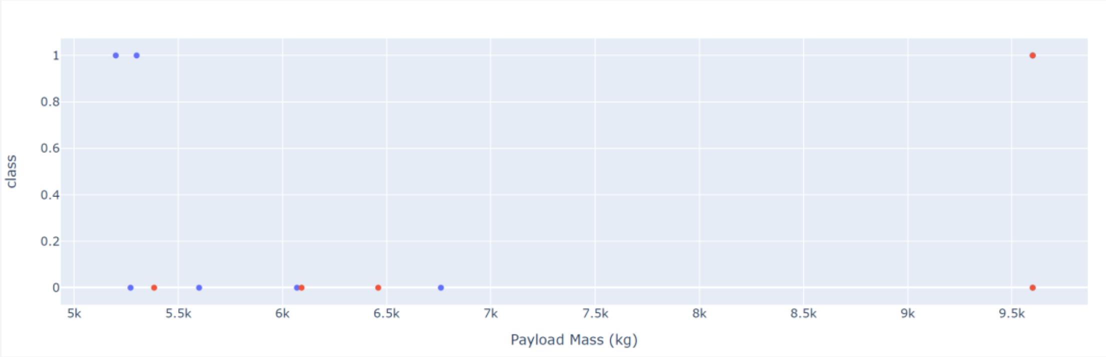


Relationship between payload mass and class

Analyzed in range of payload mass from 0 – 5000 Kg



Analyzed in range of payload mass from 5000 – 10000 Kg



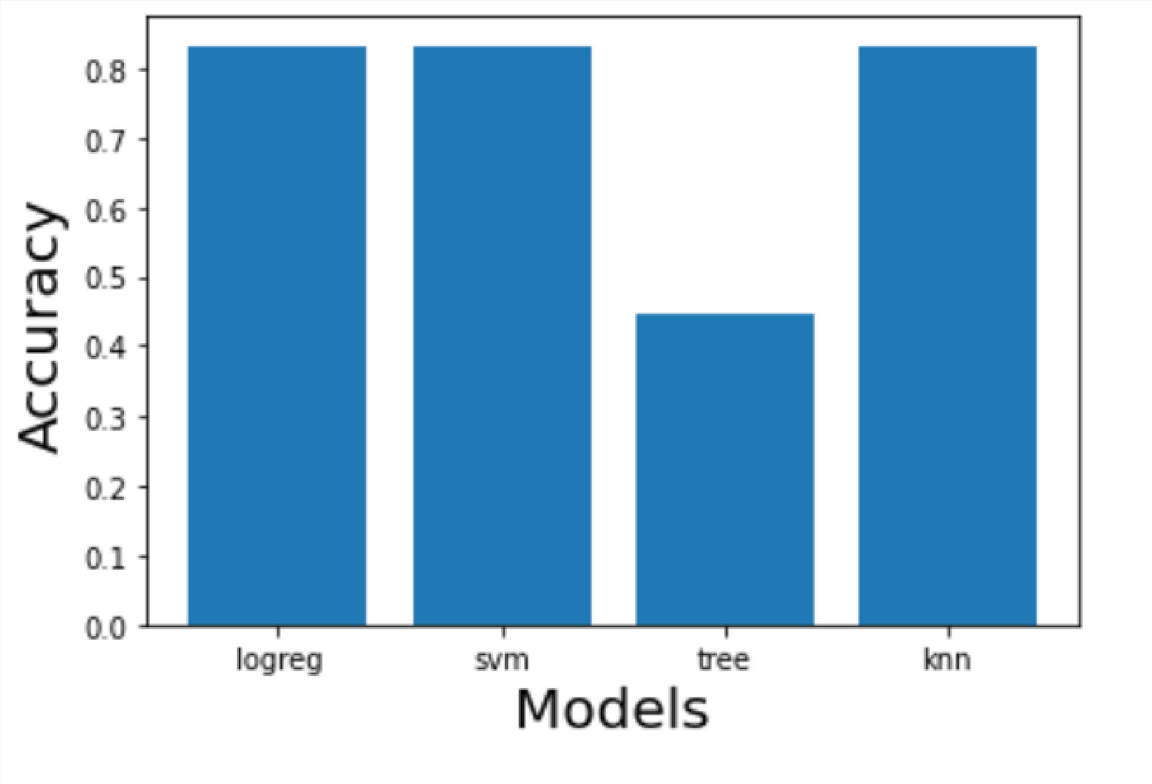
- This indicated that the lower the payload mass the higher the success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logreg, svm and knn had the highest accuracy of 0.83. Decision tree had the least accuracy of 0.44.



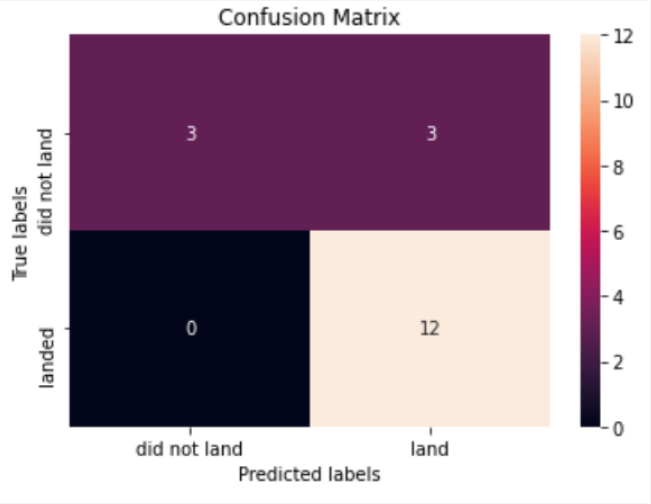

```
[76]: print('Accuracy for Logistic Regregression method:', logreg_cv.score(X_test, Y_test))
      print('Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
      print('Accuracy for Decision Tree Method:', tree_cv.score(X_test, Y_test))
      print('Accuracy for K nearest neighbors method:', knn_cv.score(X_test, Y_test))

      /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/sklearn/linear_model/base
      iltin `int`. To silence this warning, use `int` by itself. Doing this will not modify an
      e.g. `np.int64` or `np.int32` to specify the precision. If you wish to review your curre
      Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/relea
        indices = (scores > 0).astype(np.int)
      Accuracy for Logistic Regregression method: 0.8333333333333334
      Accuracy for Support Vector Machine method: 0.8333333333333334
      Accuracy for Decision Tree Method: 0.4444444444444444
      Accuracy for K nearest neighbors method: 0.8333333333333334
```

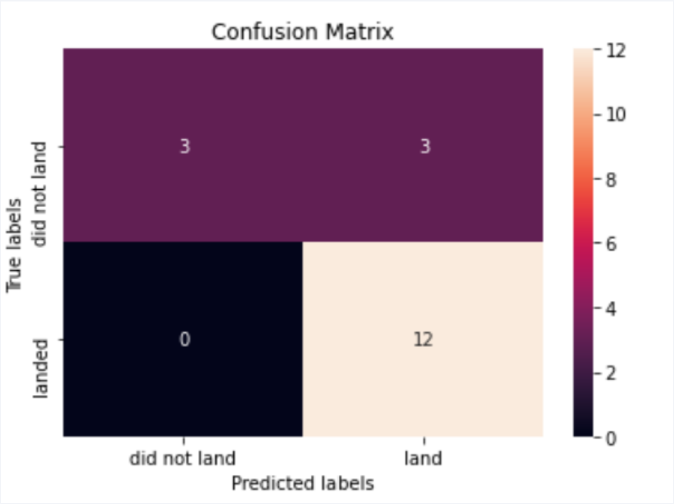
Confusion Matrix

- The confusion matrix of the three best performing models are as shown below. The confusion matrix shows that the models below can distinguish between different classes. But the issue is with the false positives where 3 cases of unsuccessful landing were classified as successful.

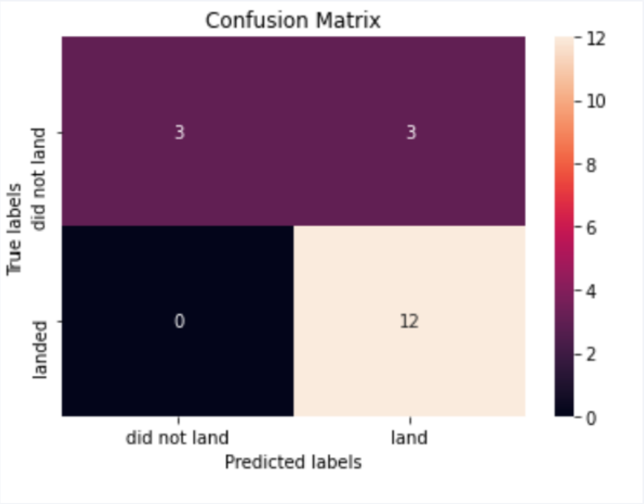
LOGREG



SVM



KNN



Conclusions

- It was observed that the larger the flight number, the higher the success rate was at the respective launch site.
- It can be seen that the launch success rate started in 2013, continued to rise till 2020.
- Orbits such as LEO, ISS, PO and GTO had the most success rate.
- The launch site KSC LC-39A had the most success rates.
- Logreg, svm and knn had the best accuracy of 0.83 and so any can be picked for making predictions.

Thank you!