IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Eish Kapoor
27th August' 23

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data Wrangling
- Exploring data analysis with SQL
- EDA with Data visualization
- Build an interactive map with folium
- Building Dashboard with ploty
- Predicitve analysis

- ***Summary of all results***
  - Exploratory data anaylsis result
  - Interactive analysis
  - Predicitive analysis result
  - Visualization and great for decsion making

Introduction

- ***Project background and context***

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. In this analysis we will focus on the study of the first stage of the Space X falcon 9 Rocket in order to obtain conclusion that allow us to make cost projection. It is important to know if the rocket will land successfully or not because the will cost company much more resources.

- ***Problem that needs answers***

1. Will the rocket land successfully?
2. What is the accuracy of successful landing?
3. Parameters that can determine success rate of landing
4. Determine the cost of future launches

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:	Data was collected using SpaceX API and web scraping from Wikipedia
Perform data wrangling	Data was transferred and one hot encoded was apply later on, to calculate no of launches, occurrence of each orbit and many more
Perform exploratory data analysis (EDA) using visualization and SQL	Exploring new patterns in data with visualization with help of graph like scatter.
Perform interactive visual analytics using Folium and Plotly Dash	Dash and folium were used to achieve this goal
Perform predictive analysis using classification models	Machine learning model were use to achieve this goal

Data Collection

- Data was collected from previous SpaceX mission and wikipedia pages and below processes were obtained:

  1. Data collection was done using get request to spaceX API.
  2. We also performed some basic formatting, Converted it into a data frame with the help of .JSON()
  3. Then cleaned the data, checked for missing values and fill in the missing values, if needed.(Pre-processing)
  4. We performed web scraping to collect Falcon 9 launch data.
  5. The objective was to extract the launch record as HTML table and convert it into pandas data frame.

Data Collection – 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.js
```

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

```
[10]: 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

```
[16]: # Get the head of the dataframe
      data.head(5)
```

```
•[85]: data_falcon9 = df[df['BoosterVersion'] == 'Falcon 9']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
[87]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
      data_falcon9
```

- We used the get request to the spaceX API to collect data, clean the data and did some basic formatting.

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- It can be seen in details in the following link : https://github.com/Eishkapoor12/spacey/blob/man/jupyter-labs-webscrapng.ipynb

Request

- Request Falcon 9 Launch from Wikipedia page via its URL.

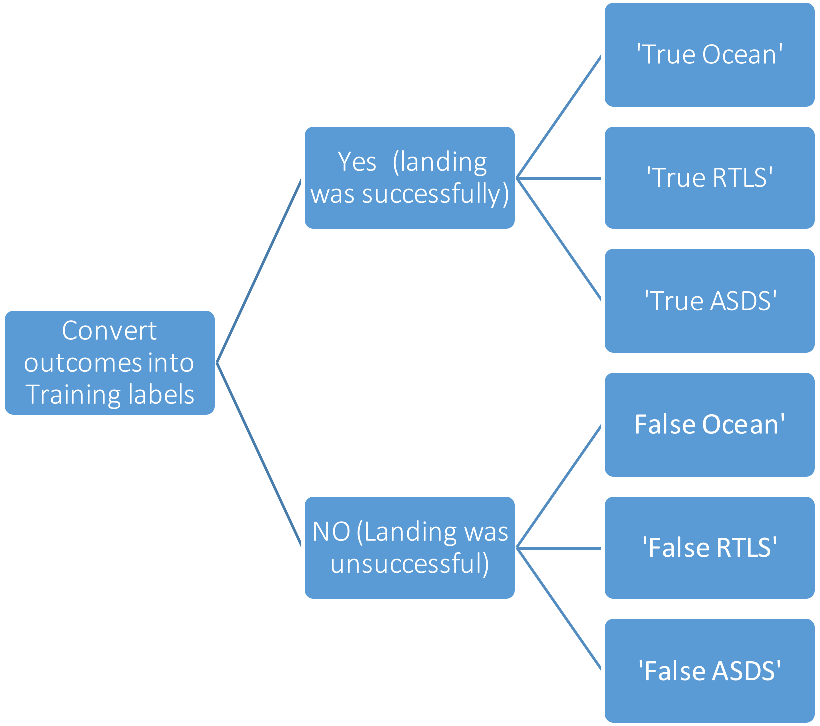
Extract

- Extract all the column names from HTML table

Modify

- Create a data frame by the launch HTML tables

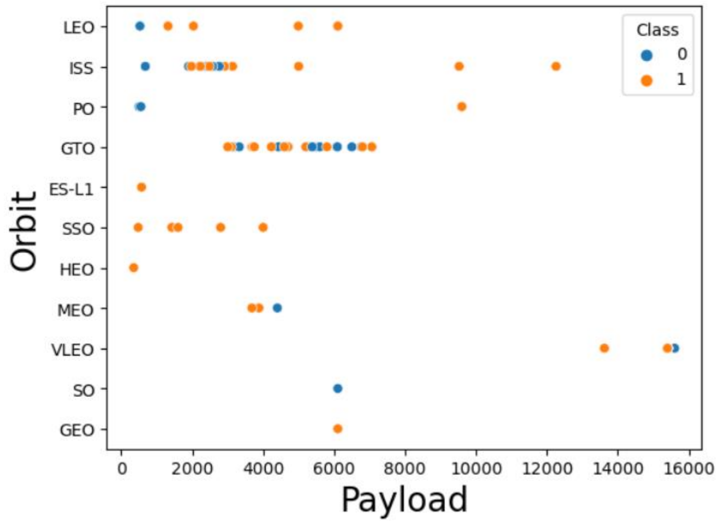
Data Wrangling



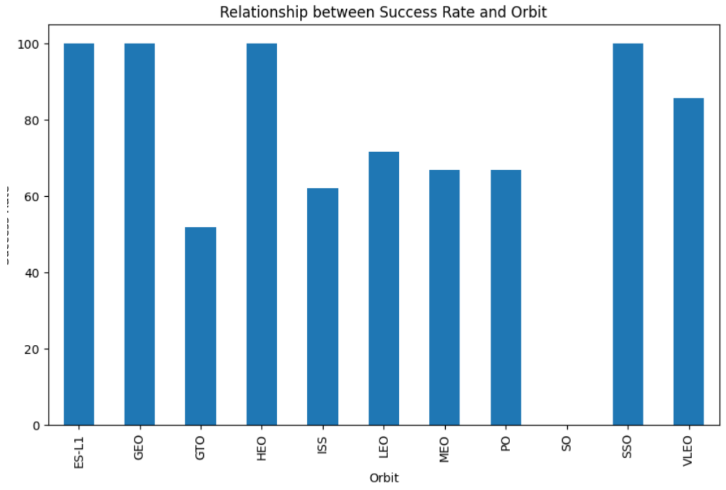
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site and the number and occurrence of each orbit.
- Then, we created landing outcome label from outcome result column and exported the result to csv.
- It can be seen in details in the following link: https://github.com/Eishkapoor12/spacey/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

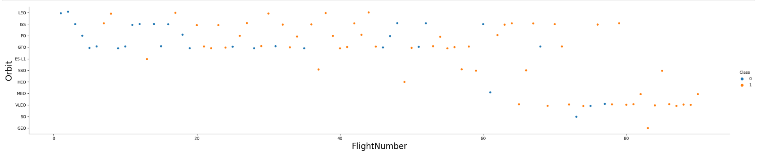
Link to this note book:https://github.com/Eishkapoor12/spacey/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb



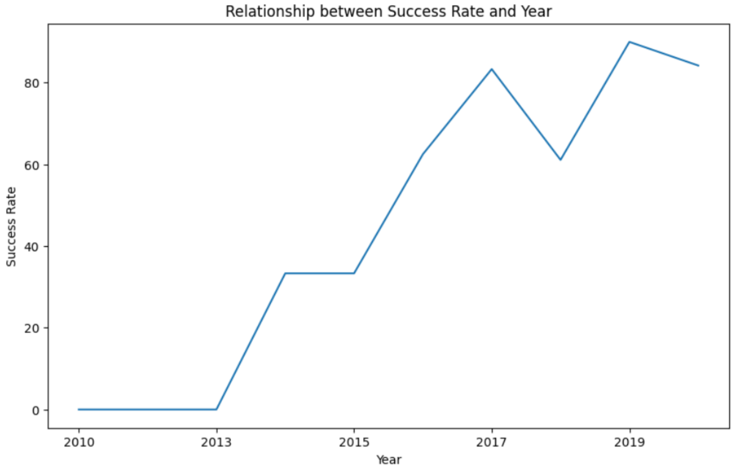
Scatter plot between Orbit and fight number



Bar chart between Orbit and success rate of each orbit



Categorial plot between fight number and pay load mass (Kg)



Line chart between year and success rate

EDA with SQL

- SQL queries performed:
  - Name of the unique launch site in the space mission.
  - Top 5 launch sites whose name begins with 'CCA'
  - Display the total payload mass carried by booster launched by NASA(CRS)
  - Display avg payload mass carried by booster version f9 v1.1
  - Total no of successful and failure mission outcome
  - Name of the booster which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List of booster version carried the maximum payload mass
  - Ranking the count of landing outcomes, like Failure (drone ship) or success (Ground pad)
- It can be seen in details in the following link: https://github.com/Eishkapoor12/spacey/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers, circle, lines and marker cluster were used with Folium maps.
- Indication of each element:
  - folium.marker() was used to create marks on maps
  - folium.Circle() was used to create circles
  - folium.Icon(O was used to create an iconon the map
  - Folium.Polyline() was use to create polynomial line between the points
  - MarkerCluster() was used to create a simplified map which contains several markers.
- We assigned the feature launch outcome to class 0 and 1, i.e., 0= failure and 1= success
- Link for this notebook:https://github.com/Eishkapoor12/spacey/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Dash and HTML component were used as they are the most important thing and almost everything depends on them, like the graph, dropdown and many more.
- Built an interactive dashboard with ploty dash
- Pandas were used to create a Dataframe.
- Ploty was used to create graphs.
  - Pie chart: showing total launches by certain sites
  - Scatter graph: Relation between outcome and payload mass (kg)
- Drop down was used for launch sites.

Predictive Analysis (Classification)

Building the model

Create column for class

- Standardize date
- Split into test and train
- Build GridSearchCV model and fit in the data

Evaluating the Model

Calculate the accuracies

- **Plot the results**
- **Confusion matrix**

Finding **the optimal model**

Find the best model with highest accuracy

Choose the optimal model

Detail explanation in the link: https://github.com/Eishkapoor12/spacey/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

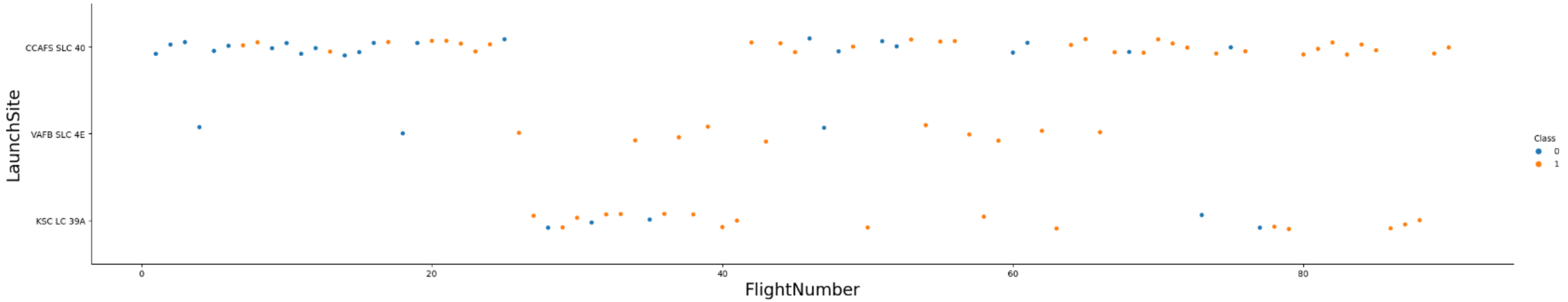
Results

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

Section 2

Insights drawn from EDA

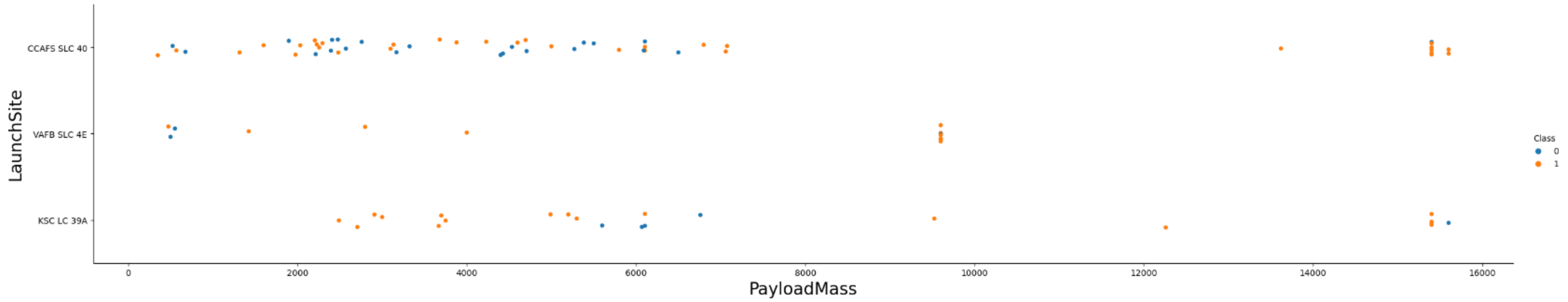
[7]:
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Flight Number vs. Launch Site

- With increase of fight number, the success rate increasing as well in the launch sites.
- From plot, you can observe that LEO orbit the success is related to the number fights, on the other hand there in GTO orbit seems to be no relationship between fight number

```
[13]: ### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

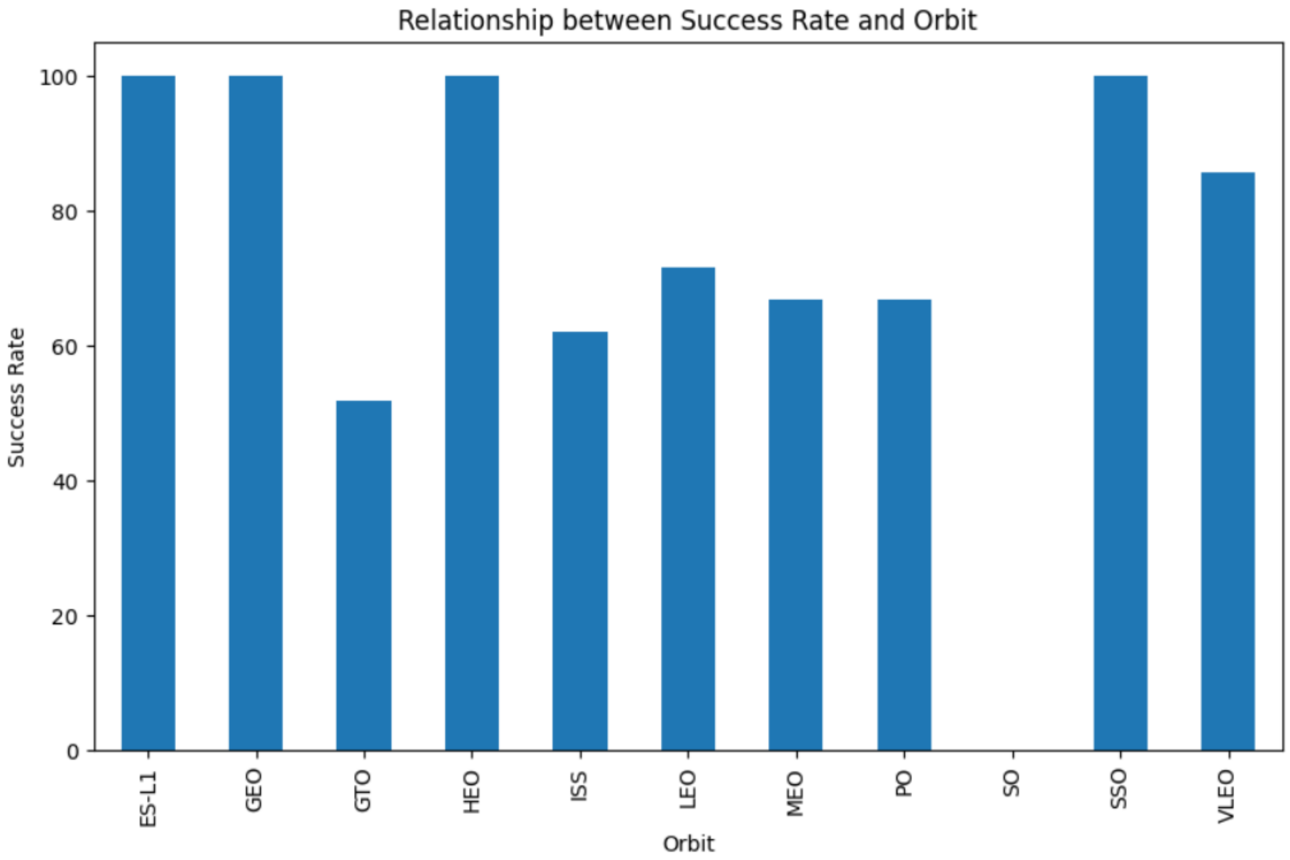


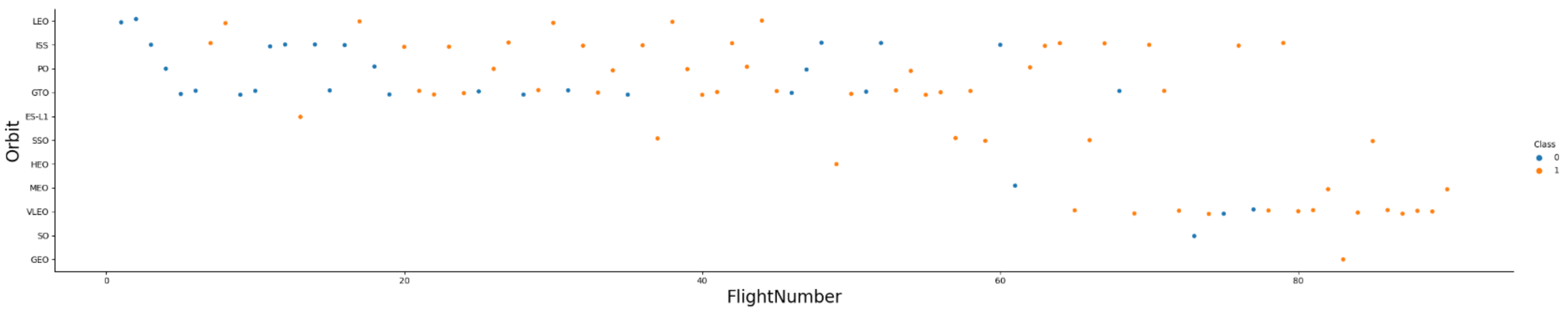
Payload vs. Launch Site

- With increase of payload mass the rate of success is increasing as well in the launch sites.
- In scatter plot you will find that VAFB-SLC launchsite have no rocket launched for heavypayload i.e., greater than 10000

Success Rate vs. Orbit Type

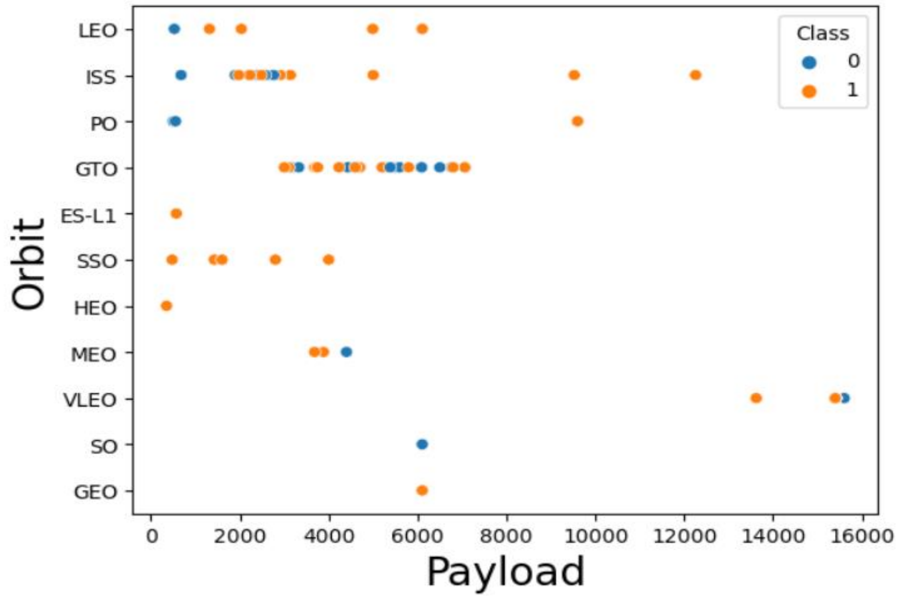
- 100% success rate:- ES-L1, GEO, HEO and SSO.
- 0% success :- SO





Flight Number vs. Orbit Type

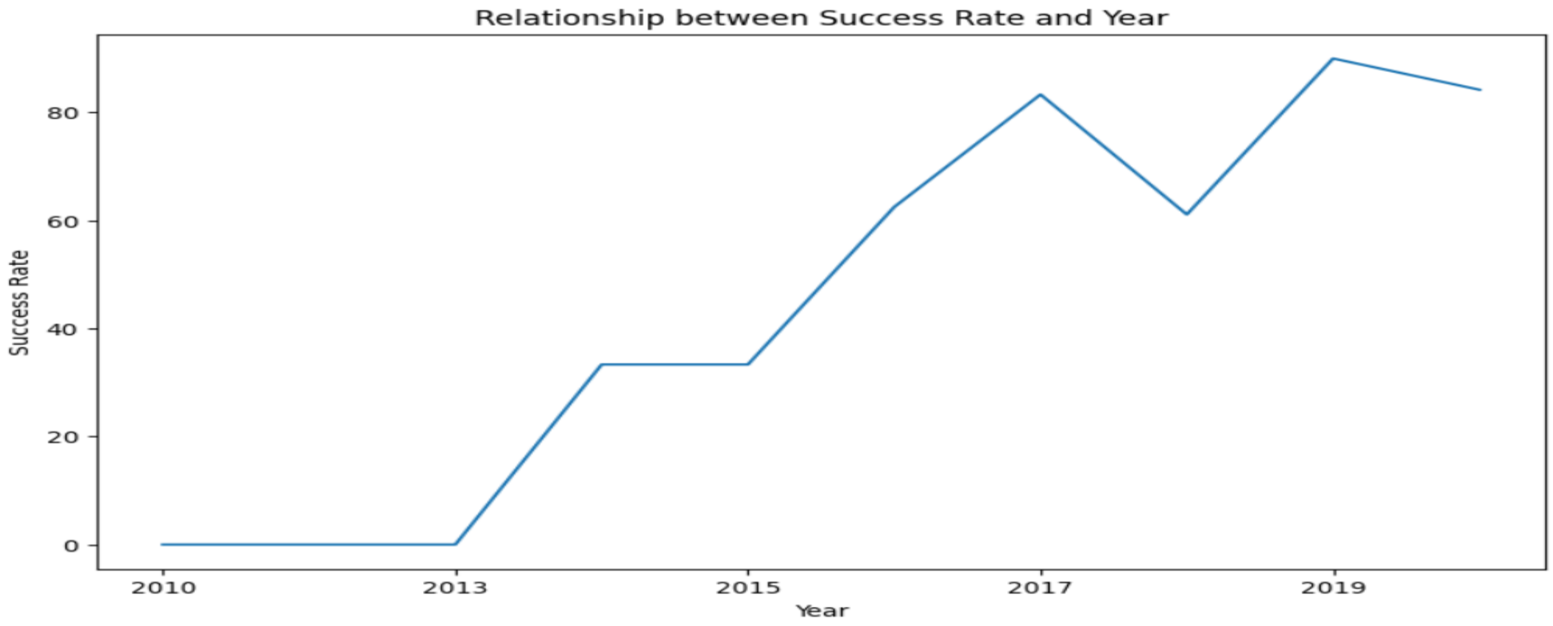
- The plot show Fight number vs Orbit number, you can observe that LEO success is related to the fight number whereas GTO has no relation between fight number and the orbit



Payload vs. Orbit Type

- We can observe that with heavy payload, the probability of successful landing are high for PO, LEO and ISS orbit.
- We can also see the effects of payload mass on orbit such as pay load mass between 3000 and 7000 is affecting GTO or Payload mass between 2000 and 3000 is affecting ISS.

Launch Success Yearly Trend



- We can observe that success rate had a massive increase since 2013 kept increasing till 2020.

All Launch Site Names

- The 'DISTINCT' is used to show the only unique launch sites from SpaceX data.

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- This query was used to display 5 launch sites begin with 'CCA'

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

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

```
[12]: %sql select sum(PAYLOAD_MASS__KG_) as 'NASA(CRS)' from SPACEXTBL where Customer = 'NASA (CRS)';

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

[12]:

NASA(CRS)
45596

Total Payload Mass

- Calculation of the total payload by using 'SUM' value.

```
[39]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where  Booster_Version = 'F9 v1.1' ;

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

avg(PAYLOAD_MASS__KG_)
2928.4

Average Payload Mass by F9 v1.1

- Calculation of the average by using 'AVG' function
- The result is 2928.4

```
[13]: %sql select min(Date) as 'First success' from SPACEXTBL where Landing_Outcome= 'Success (ground pad)';

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

[13]:

First success
2015-12-22

First Successful Ground Landing Date

- We get the first successful landing data by using 'MIN'.
- We can see that the first successful landing was on 22th dec'22

```
[46]: %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.
```

[46]:

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- The 'WHERE' clause to filter the booster which as landed successfully on drone ship.
- The 'AND' was used to dterminethe successful landing between mass greater than 4000 and less than 6000

```
[51]: %sql select Mission_Outcome,COUNT(Mission_Outcome) from SPACEXTBL group by Mission_Outcome ;
```

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

[51]:

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

Total Number of Successful and Failure Mission Outcomes

- The 'GROUP BY' clause is used to group success and failure.
- The 'COUNT' function is used to count the total no of mission outcome
- Success = 100
- Failure =1

```
[14]: %sql select DISTINCT Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_= (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
 * sqlite:///my_data1.db
Done.
```

[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

Boosters Carried Maximum Payload

- Booster that carried the maximum payload using a subquery was determined by the 'WHERE' clause.
- 'MAX' was used to get the maximum payload mass.

```
[17]: # -- %sql select substr(Date, 4, 2) as month, MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE, [LANDING_OUTCOME] from SPACEXTBL where [LANDING_OUTCOME] = 'Failure (
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL where Landing_Outcome LIKE 'Failure (drone ship)' and Date BETWEEN '2015-01-01' AND '2015-12-31';

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

[17]:

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

2015 Launch Records

- This query was a combination of 'LIKE', 'AND' and 'BETWEEN' which help us to filter for failed landing in drone ship, their version and year i.e., 2015.

```
[21]: %sql select Landing_Outcome, count(Landing_Outcome) from SPACEXTBL where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
ORDER BY COUNT(Landing_Outcome) DESC;
```

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

[21]:

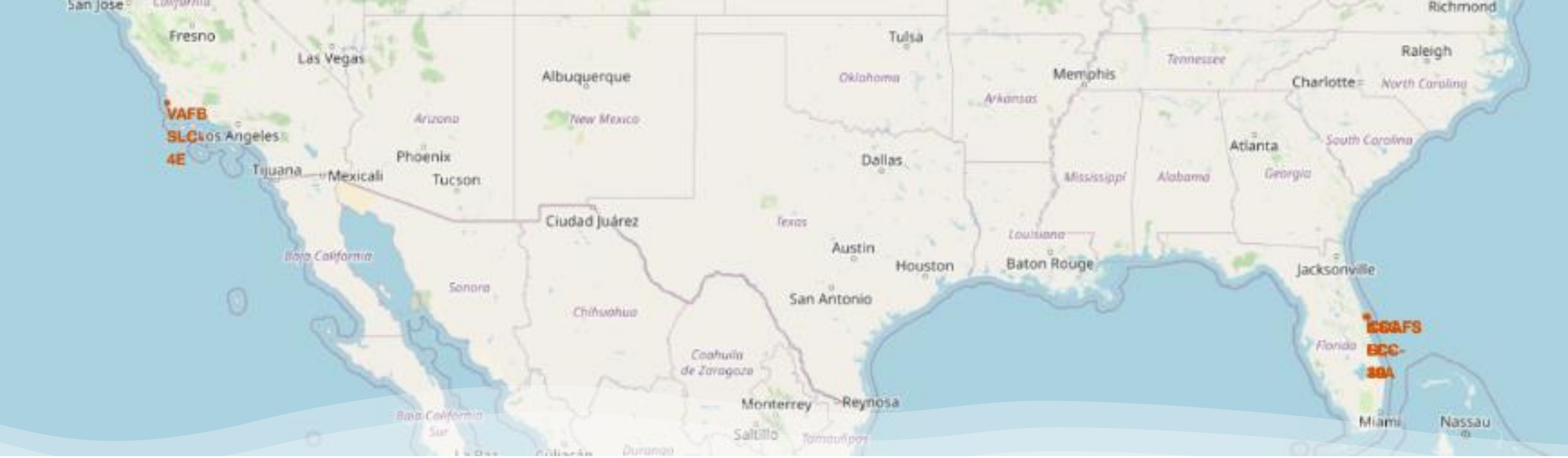
Landing_Outcome	count(Landing_Outcome)
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

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

- Landing outcome and 'COUNT' of landing outcomes from the data was selected.
- 'WHERE' clause was use to filter landing outcomes BETWEEN 2010-06-04 to 2010-03-20
- 'GROUP BY' clause was use to group the landinh outcomes.
- 'ORDER BY' clause to order outcomes in descending order.

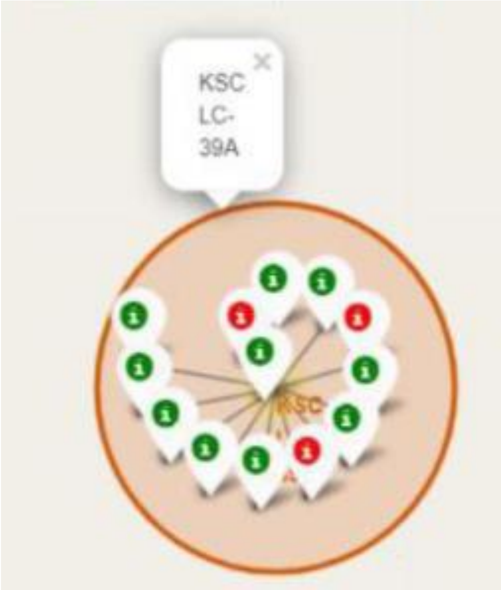
Section 3

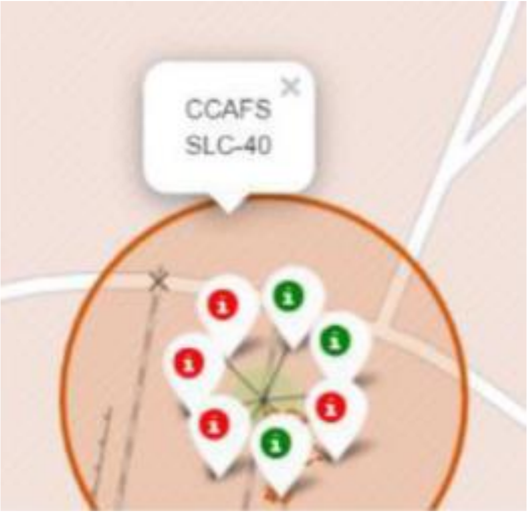
Launch Sites Proximities Analysis



Launch sites

All the launch sites are in USA, near Florida and California.





Launch site

Filtered by successful and failure

Green marker represents successful launches and *Red marker* shows Failure.

Distance from launch site.

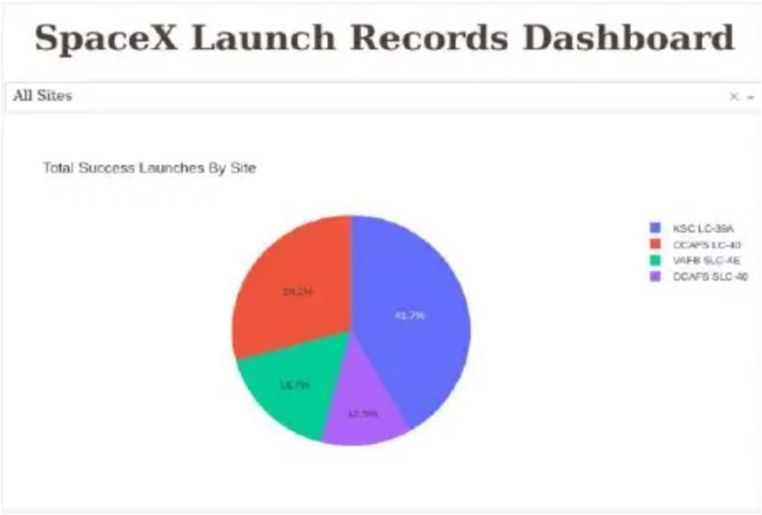
- Lauch sites does have a safe distance from railways, highways and coastline.

Section 4

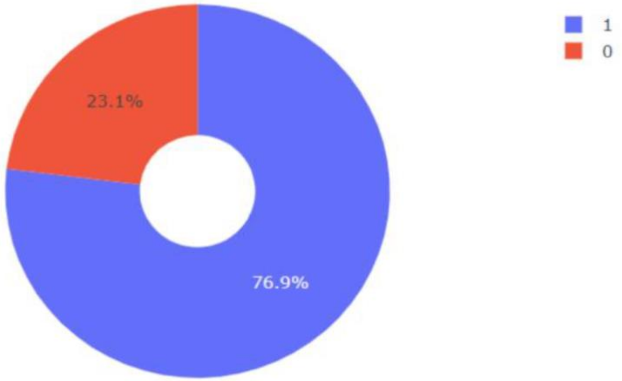
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

- We can see that KSC LC-39 had the most successful launches from all the other sites
- VAFB SLC-4E has the least number of stage 1 landings.



Pie chart showing launch with highest launch ration of KSC LC39A

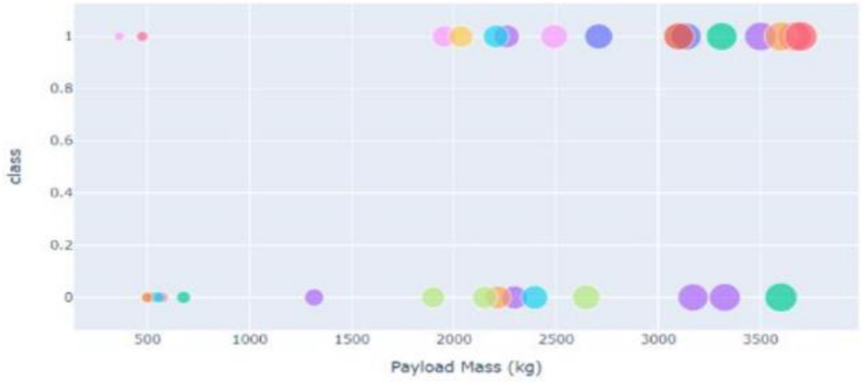


- KSC LC 39A have achieved highest of 76.9% success rate and 23.1% failure rate

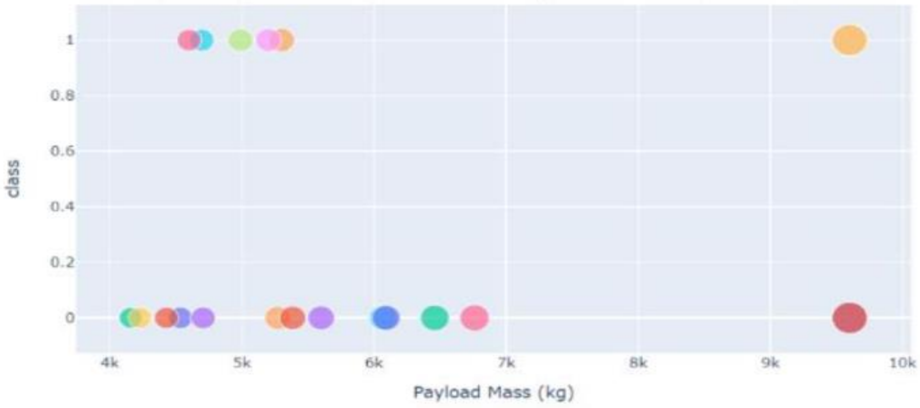
Payload vs Lauch Outcome (scatter plot)

This is a scatter plot of payload vs launch outcome for all sites with different payload selected in the range slider

Low Weighted Payload 0kg – 4000kg



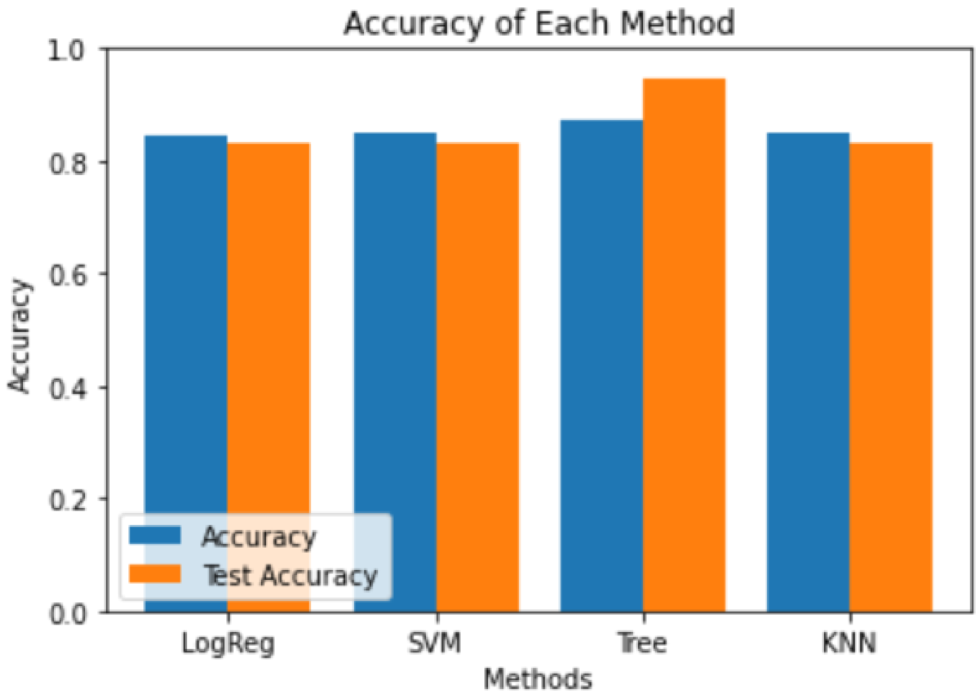
Heavy Weighted Payload 4000kg – 10000kg



We can see the success rate for low weighted payload is higher than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

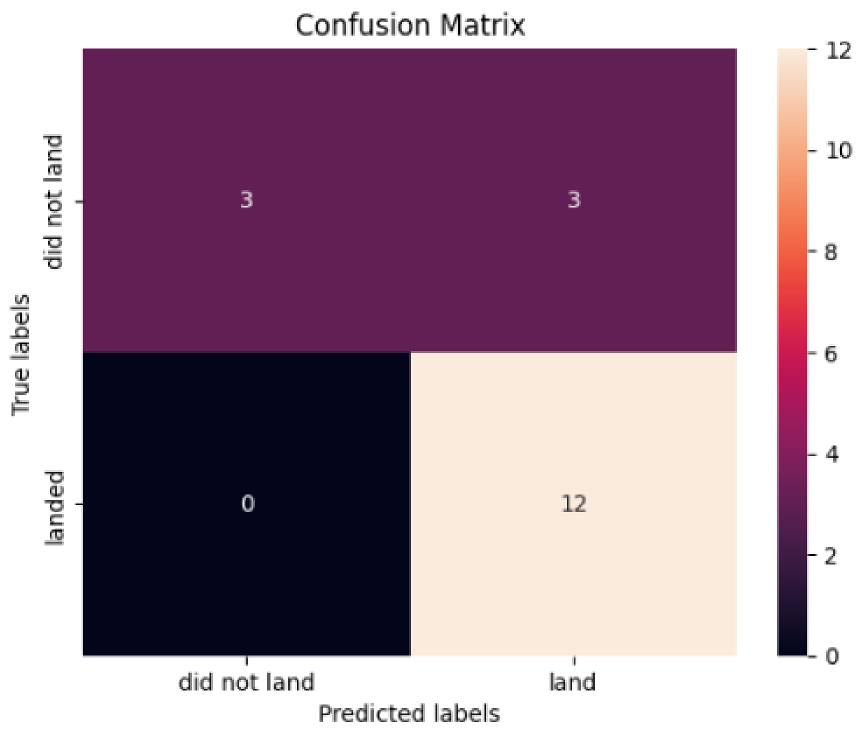


Classification Accuracy

- Four classification model were tested and their accuracies were plotted beside
- Decision Tree has the highest accuracy with almost 0.89 then the remaining with 0.84

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 positive .ie, is unsuccessful landing.



TRUE POSITIVE	12
True Negatives	5
False positive	1
False negatives	0

Conclusions

- We can conclude that:

1. The site with highest score was KSC LC-39A
2. The payload of 0kg to 5000kg was more diverse than 6000kg to 10000kg
3. Launch success rate started to increase 2013 to 2020
4. Decision tree was the optimal model with accuracy of 0.89
5. All model had atleast one false positive.
6. Logistic Regression is quite accurate whenit comes to prediction.

Thank you!