Winning Space Race with Data Science

Akash Kumar
31-Jan-2022

GIT Hub

Outline

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

Executive Summary

- SpaceX reusable rocket launch data was collected from different sources like web API and Web scrapping
- Data Exploration was done using Pandas, NumPy, SQL, visualization, Folium maps.
- Data preprocessing was done to fix missing values and data normalization and feature engineering was done for better results.
- ML model trained and optimized using Grid Search CV.
- Four models used were Logistics, Decision Tree, Support Vector Machine, K Nearest Neighbours
- All models achieved similar accuracy of 83%.

Introduction

Background

- Space exploration is the new big thing for Human Race, But........
- Space travel is highly expensive and SpaceX is the cheapest option available for around $62 million vs competition at $150+ million.
- Space Y want to compete with SpaceX and to do that they have to reuse there rocket like SpaceX do.

Problem

- Space Y has given us the task to analyze the Space X launch data to find information like
- Launch success rate, Price, Location wise success rate, Booster used for launch etc.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through SpaceX API and web scraping
- Perform data wrangling
  - Collected data was merged and calculated success rate Lauch site wise.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - 4 model were build and tuned using Grid search to achieve similar accuracy of 83%.

Data Collection

- Data was collected by calling the GET function on Space X API
- Used `.json_normalize()` function to normalize the received data
- `SpaceX API Data Columns:`

`Flight Number, Date, Booster, Launch Site, Outcome, Orbit, Payload, Latitude, Longitude, etc.`

- Data collected from SpaceX Wikipedia page using `BeautifulSoup()` function
- `SpaceX Wikipedia Data Columns:`

`Flight No., Launch site, Date, Time, Booster version, etc.`

Data Collection – SpaceX API

- Data collection done with SpaceX REST API call using GET method with URL (https://api.spacexdata.com/v4/launches/past).
- GitHub URL of the completed Jupyter Notebook.

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

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

Check the content of the response

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


b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":
/3c/0e/T8iJcSN3_o.png","large":"https://images2.imgbox.com/40/e3/GypSkayF_o.png"},
overy":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https:

Data Collection - Scraping

- Web scraping was done using `BeautifulSoup() method.`
- GitHub URL of the completed web scraping Jupyter Notebook

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

```
In [6]:
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html5lib')
```

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

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

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- Created a landing outcome label from Outcome column
- Using the `Outcome column`, created a list where the element is 0 if the corresponding row in `Outcome` is in the set `bad_outcome`, otherwise, it's 1. Then assigned it to the variable `landing_class`.
- GitHub URL of completed data wrangling related Notebooks.

```
In [14]: # landing_class = 0 if bad_outcome
         # landing_class = 1 otherwise
         landing_class = []
         for i in df["Outcome"]:
             if i in bad_outcomes:
                 landing_class.append(0)
             else:
                 landing_class.append(1)

In [16]: landing_class[:5]

Out[16]: [0, 0, 0, 0, 0]
```

EDA with Data Visualization

- Plotted graphs of FlightNumber vs. PayloadMass, FlightNumber vs LaunchSite, SuccessRate vs OrbitType, Payload vs OrbitType, etc.
- Graphs like scatter plot, Bar plot, line chart gave us an idea of relation between different variables.
- GitHub URL of your completed EDA with data visualization Notebook.

EDA with SQL

- Loaded data into the IBM DB2 database from a CSV file.
- Queried database using SQL magic function (%sql), that gave us a better understanding of the data.
- Add the GitHub URL of your completed EDA with SQL Notebook.

```
In [25]: %sql select min(DATE) from spacexdataset where "Landing _Outcome" like 'Success%';

 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.

Out[25]:        1
         2015-12-22
```

Build an Interactive Map with Folium

- Build interactive maps using Folium, with objects like markers, circles, lines, marker cluster etc.
- These objects were added to visualize launch site location on map, proximity from railway line, coastline and other key locations.
- This allow us to understand why a launch site was selected.
- GitHub URL of your completed interactive map with Folium map Notebook.

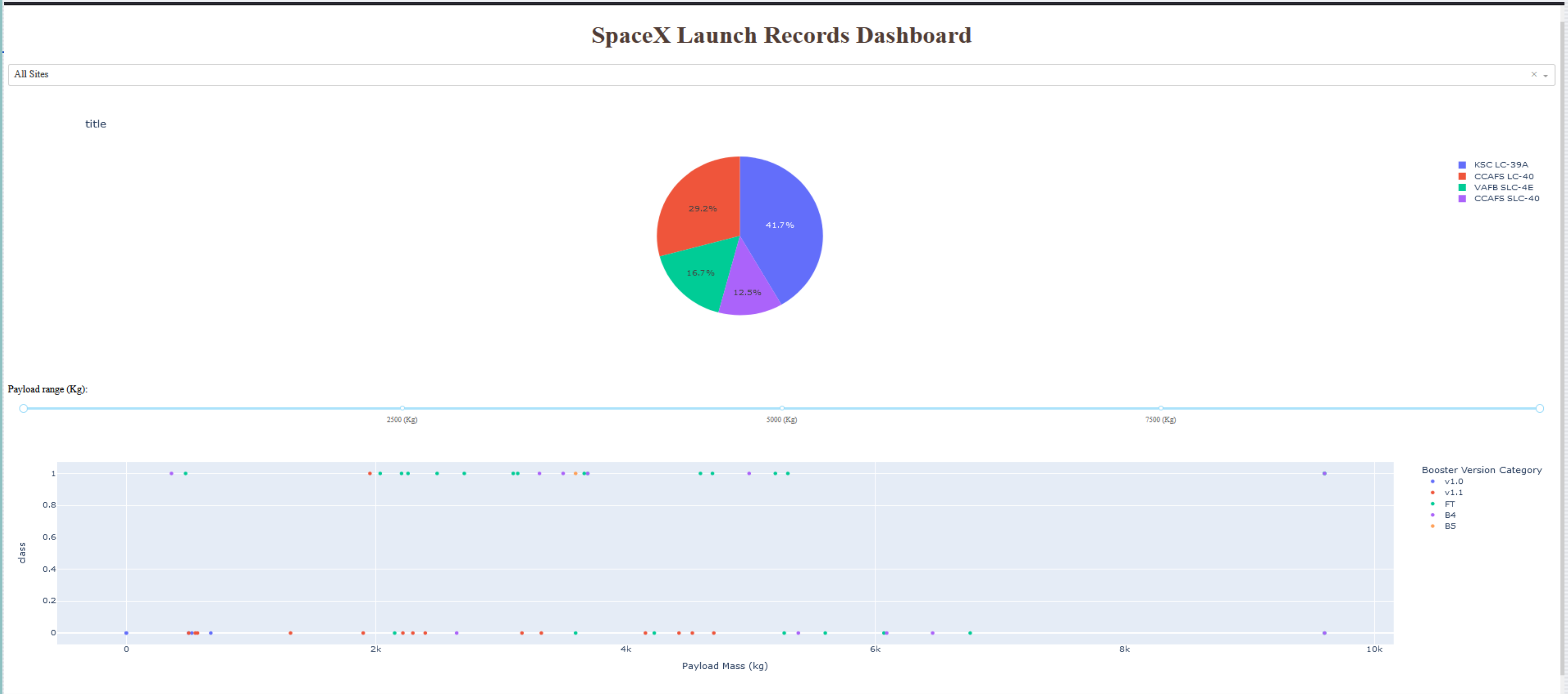
Build a Dashboard with Plotly Dash

- Created a Dashboard with Pie chart and scatter plot of number of successful launches and Payload weight that was launched.
- This helped us understand the relation between the weight and success rate of the launch.
- Scatterplot show us the details that which version of booster was used for a given weight.
- GitHub URL of your completed Plotly Dash lab Python File.

Predictive Analysis (Classification)

- Extracted most use full features from the dataset like payload, booster, launch site, etc.
- Normalized the data using Fit() ,Transform() and StandardScaler() method from SKlearn library.
- Split the data into train and test set and used GridSearchCV to find the optimal parameters.
- Created confusion matrix and Bar plot for all 4 models to compare results.
- GitHub URL of your completed predictive analysis lab Notebook.

Results



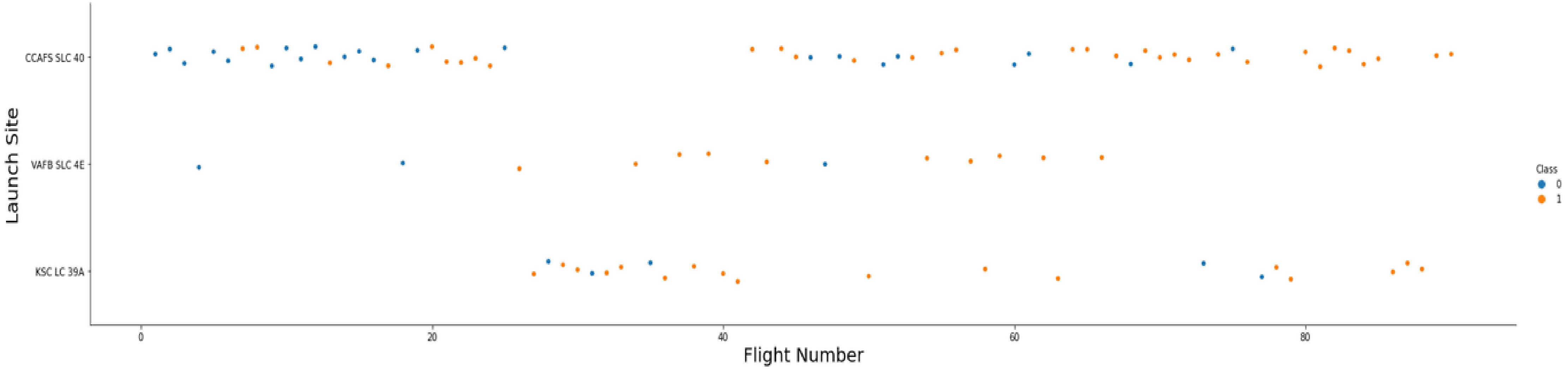
- Preview of the Plotly Dashboard created with slider for payload and site selection dropdown.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

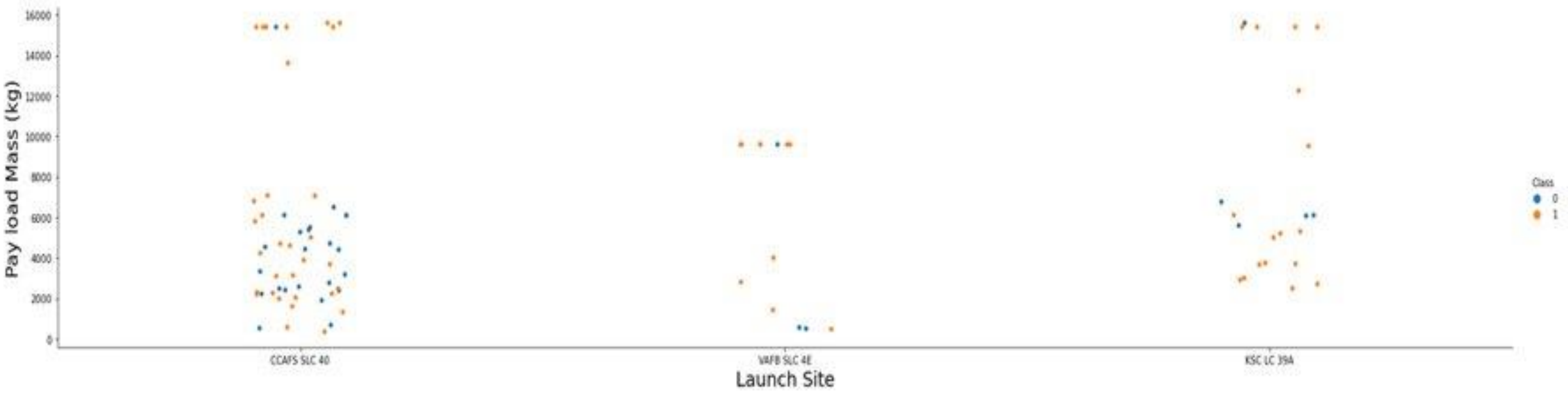
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



- This scatter plot show us the how many landings were successfully from a given site, as we can see CCAFS has maximum number of launches

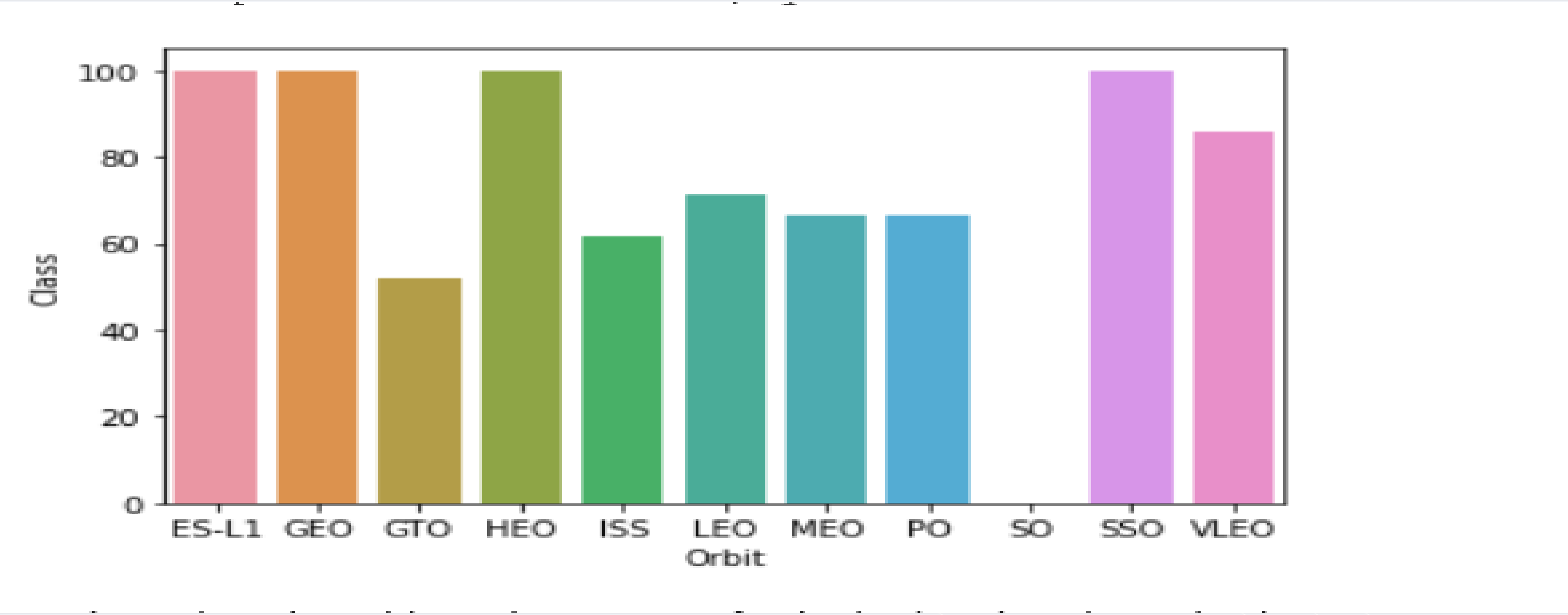
Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



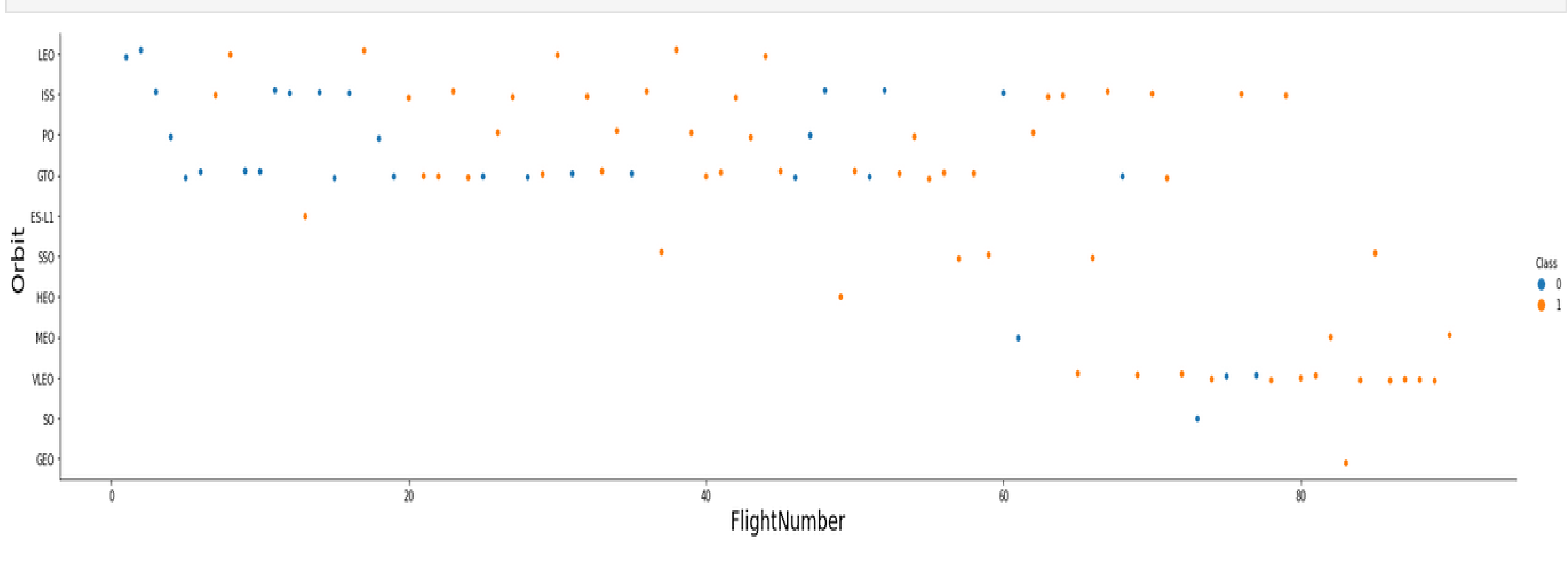
- This scatter plot show us the result (success or failure) of different landings on different launch site

Success Rate vs. Orbit Type



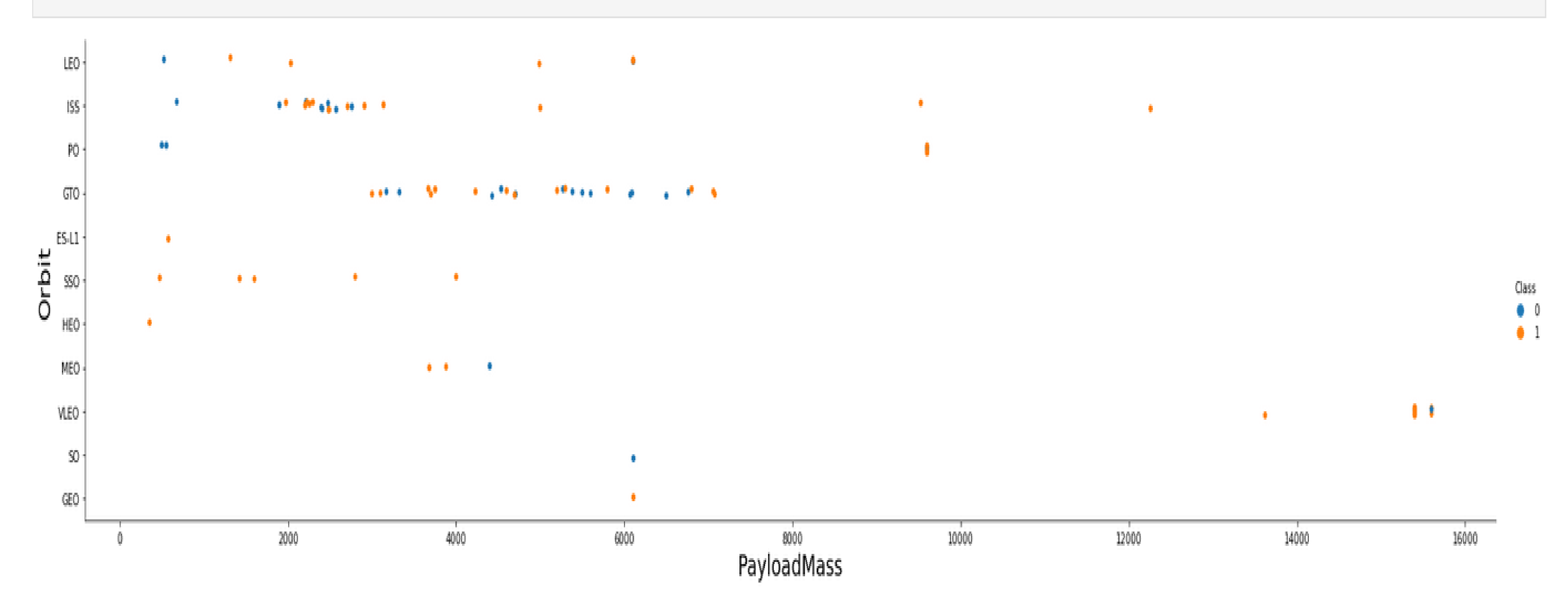
- This bar chart show us success rate of when the rockets was launched to a given orbit.

Flight Number vs. Orbit Type



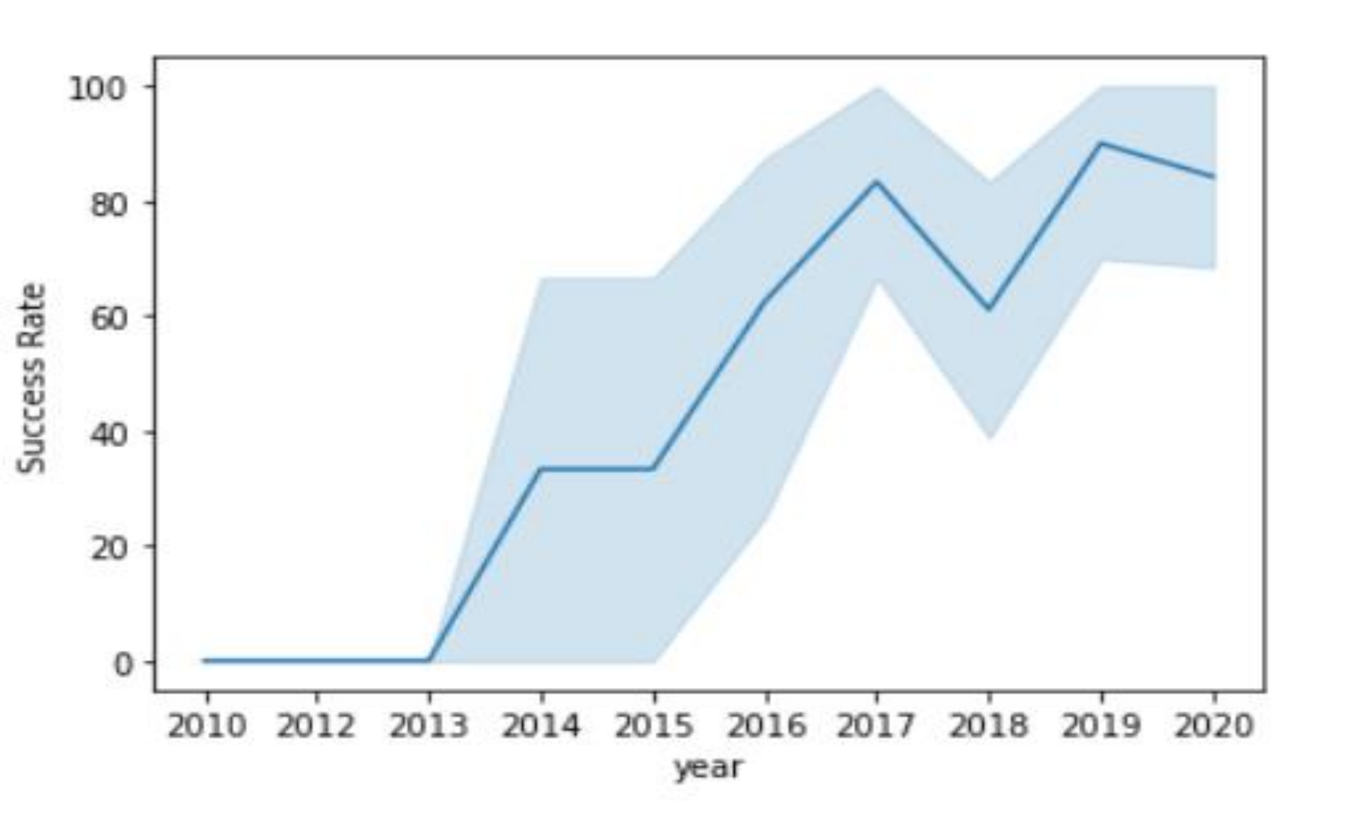
- This scatter plot show us how many successful and unsuccessful landing when a rocket was sent to a given orbit.

Payload vs. Orbit Type



- This scatter plot show us how much payload was sent to a given orbit and if the landing was successful or not.

Launch Success Yearly Trend



- This line char show us the increase in success rate of the different landings.

All Launch Site Names

- Used SQL query to find the name of all the distinct launch sites from the DB2 database.

```
%sql select distinct launch_site from SPACEXDATASET;
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.
Done.
```

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

Launch Site Names Begin with 'CCA'

- First 5 records where launch sites begin with 'CCA' using the like operator.

```
%sql select * from spacexdataset where launch_site like 'CCA%' limit 5;
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Total payload carried by boosters from NASA, data calculated using the sum(), group by() and like() functions.

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

```
%sql select sum(PAYLOAD_MASS__KG_) from spacexdataset where customer = 'NASA (CRS)' group by customer;
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

1
45596

Average Payload Mass by F9 v1.1

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

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

```
%sql select avg(PAYLOAD_MASS__KG_) from spacexdataset where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

1
2534

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad.

```
%sql select min(DATE) from spacexdataset where "Landing _Outcome" like 'Success%';
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select booster_version from spacexdataset where payload_mass__kg_ between 4000 and 6000 and "Landing _Outcome" = 'Success (drone shi
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculated the total number of successful and failure mission outcomes

```
%sql select count(*), mission_outcome from spacexdataset group by mission_outcome;
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

1	mission_outcome
1	Failure (in flight)
99	Success
1	Success (payload status unclear)

Boosters Carried Maximum Payload

- Variants of booster version F9 B5 B10xxx have carried the maximum payload mass and the list of all the boosters.

```
%sql select booster_version,payload_mass__kg_ from spacexdataset where payload_mass__kg_ = (select max(payload_mass__kg_) from spacexdata
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, total 5 failed landings found.

```
%sql select "Landing _Outcome", booster_version,launch_site  from spacexdataset where "Landing _Outcome" = 'Failure (drone ship)';
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Landing _Outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
Failure (drone ship)	F9 FT B1020	CCAFS LC-40
Failure (drone ship)	F9 FT B1024	CCAFS LC-40

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

- Rank of 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

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 "Landing _Outcome",count("Landing _Outcome") as count, DENSE_RANK() OVER( order by count desc, "Landing _Outcome") as rank
from spacexdataset group by "Landing _Outcome";
```

```
 * ibm_db_sa://sws32683:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

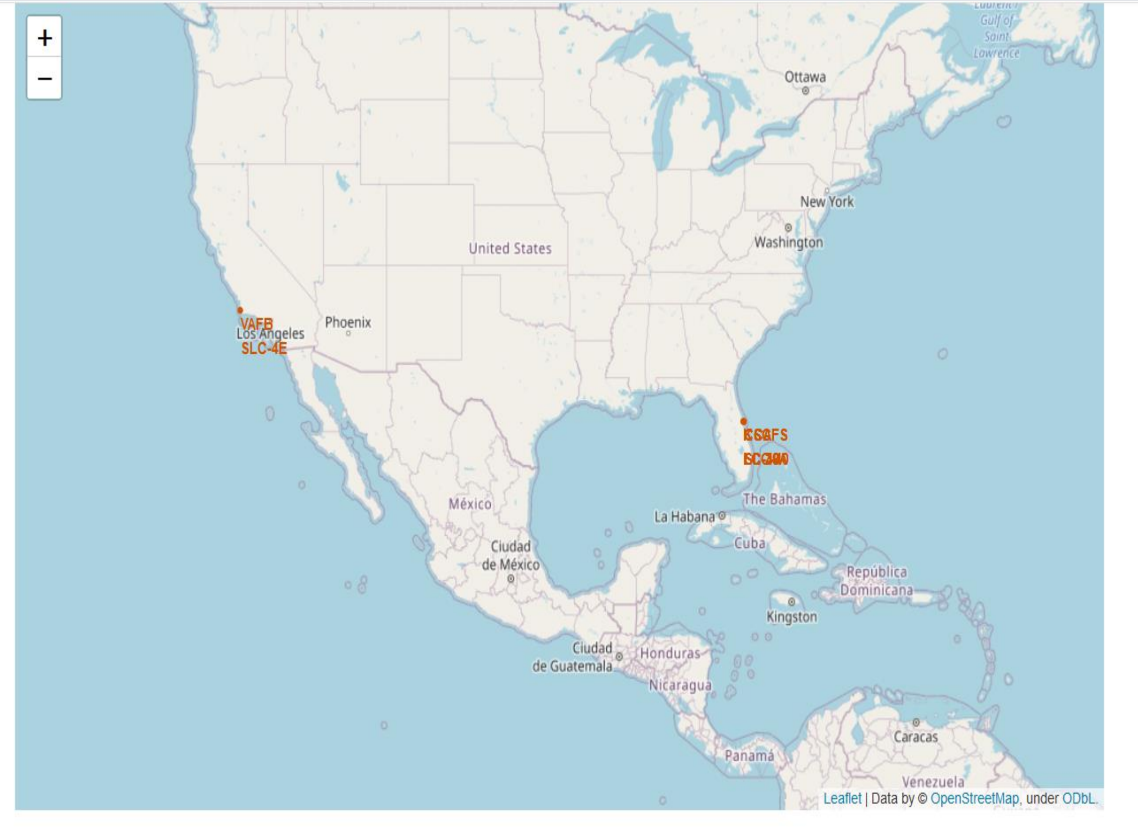
Landing _Outcome	COUNT	RANK
Success	38	1
No attempt	22	2
Success (drone ship)	14	3
Success (ground pad)	9	4
Controlled (ocean)	5	5
Failure (drone ship)	5	6
Failure	3	7
Failure (parachute)	2	8
Uncontrolled (ocean)	2	9
Precluded (drone ship)	1	10

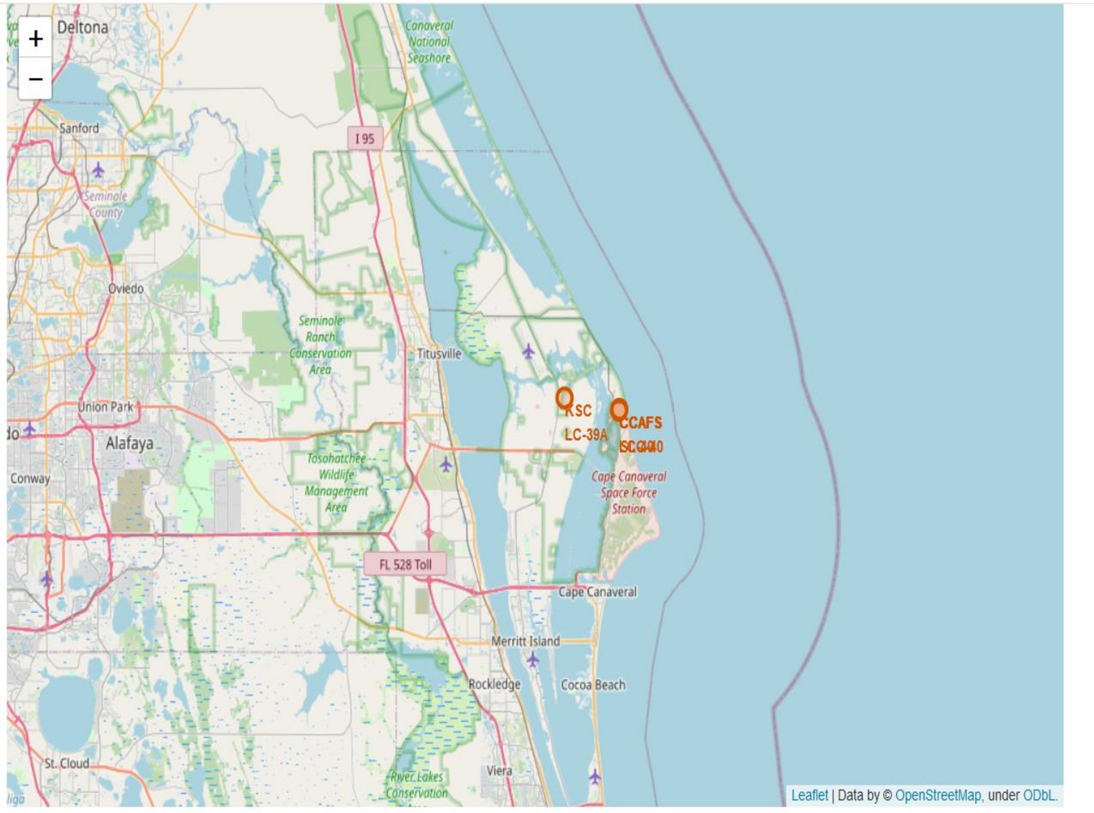
Section 3

Launch Sites Proximities Analysis

Launch Site locations

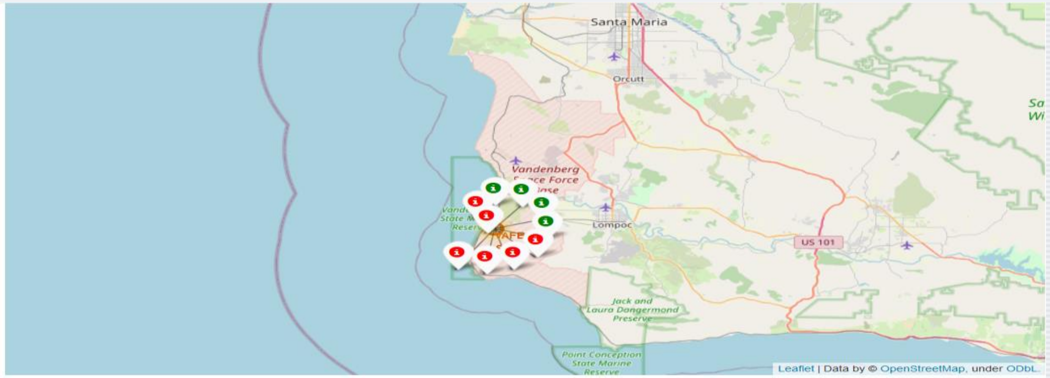
- Launch site locations, as we can see all the sites are close to ocean, left map show two locations since they are close to each other

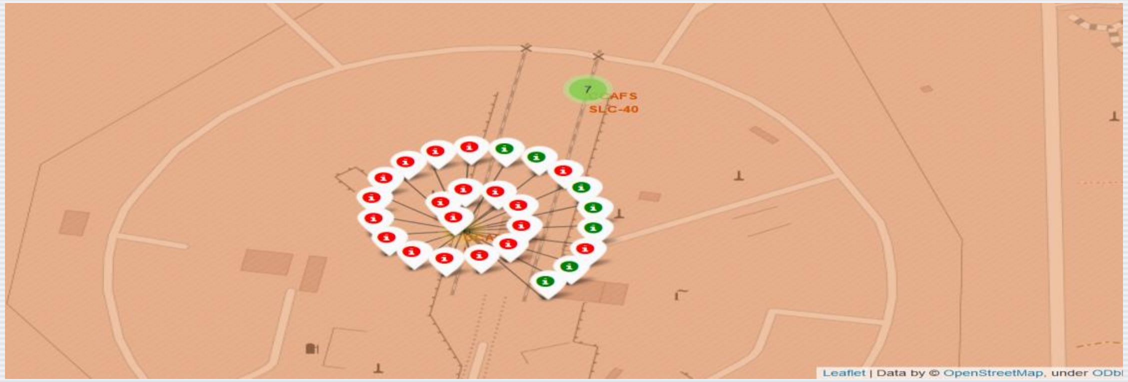


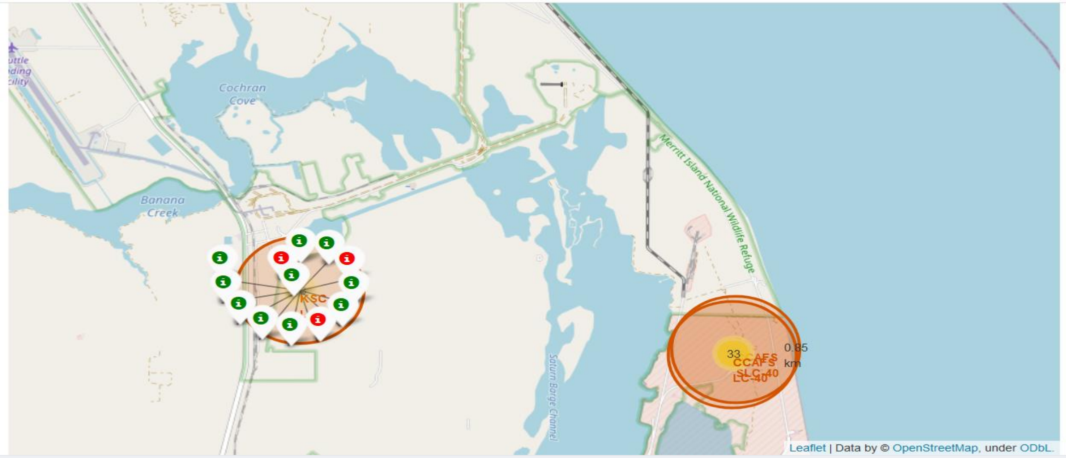


Launch Site Outcomes

- These map show markers with the outcome from the given launch site location.

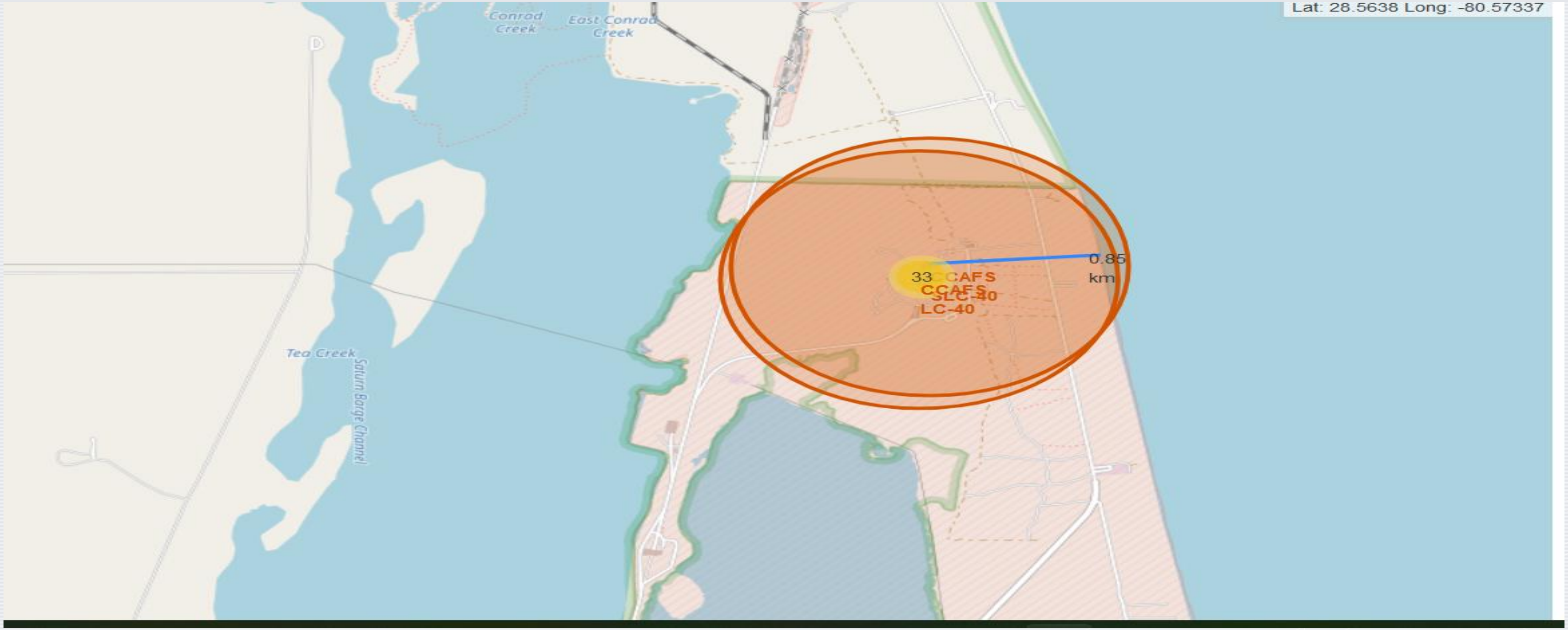






Proximity from key locations

- This map show a line with distance from the key locations.

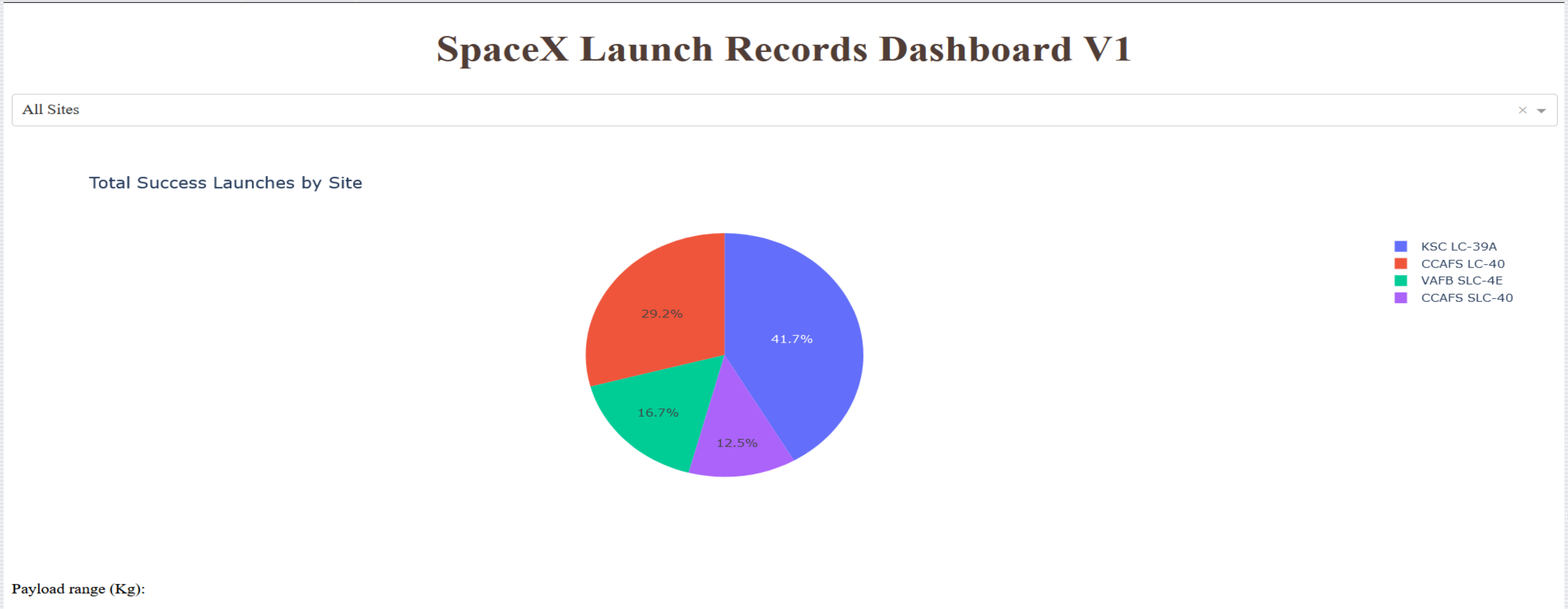


Section 4

Build a Dashboard with Plotly Dash

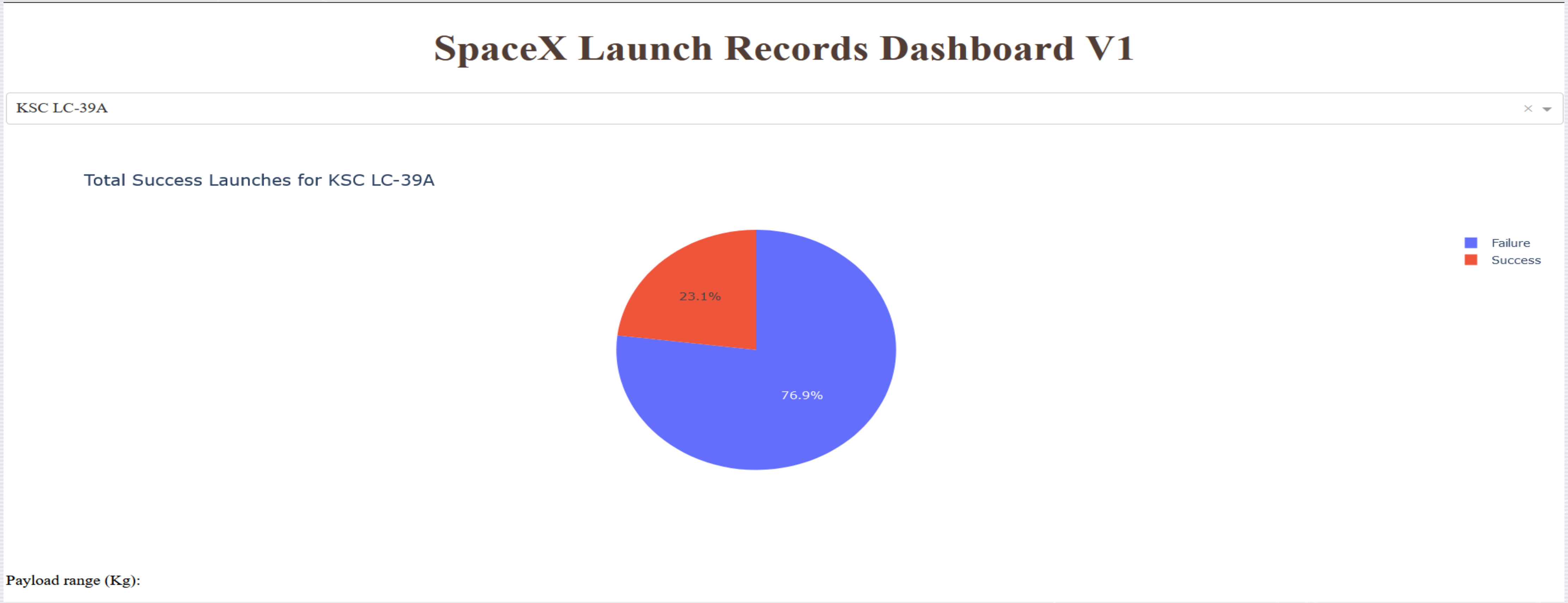
Successful Launch from All sites

- Pie chart show successful landing on a given site, we can see site KSC LC has the maximum successful launches.



Highest successful landings site

- This chart show us that 77% landings on site KSC LC are successful, this indicate Space Y can also choose a similar type of landing site for their rockets.



Booster Version vs Payload vs Class

- Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slide, we can see that booster version FT is more successful lighter for payload .

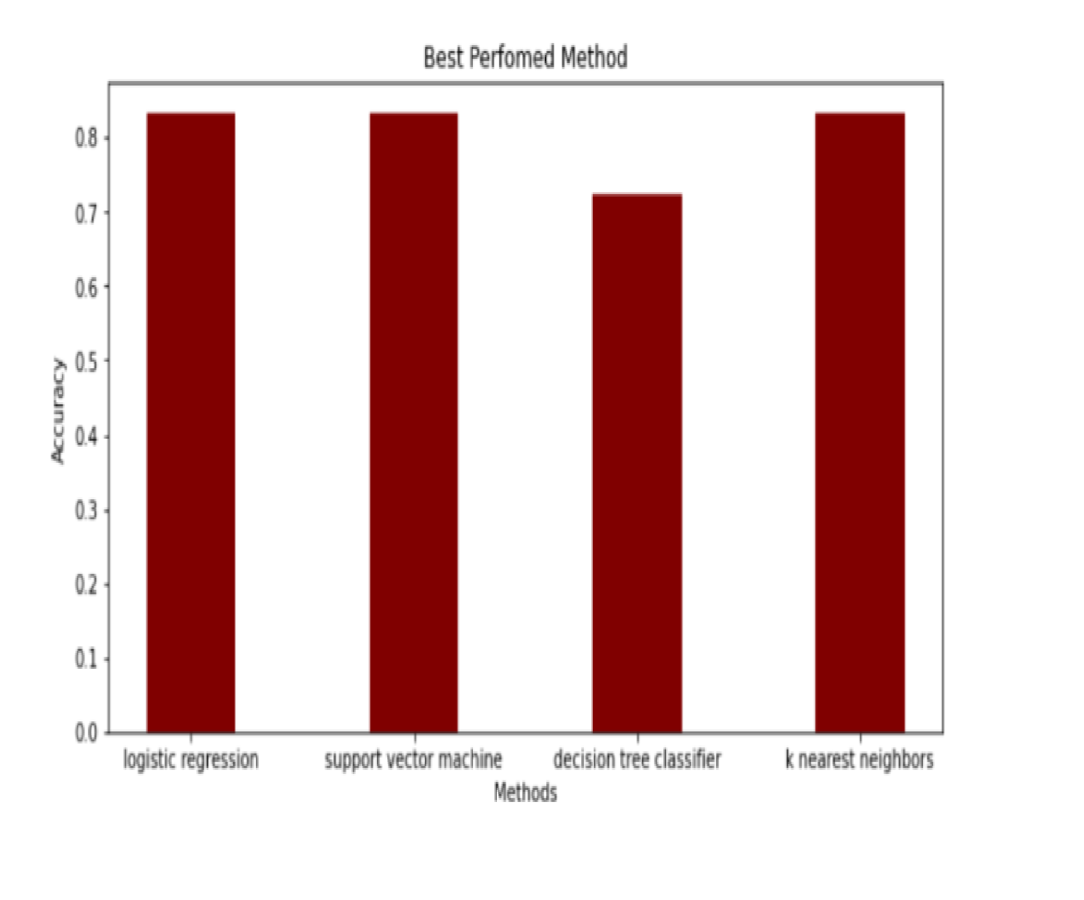
Section 5

Predictive Analysis (Classification)

Classification Accuracy

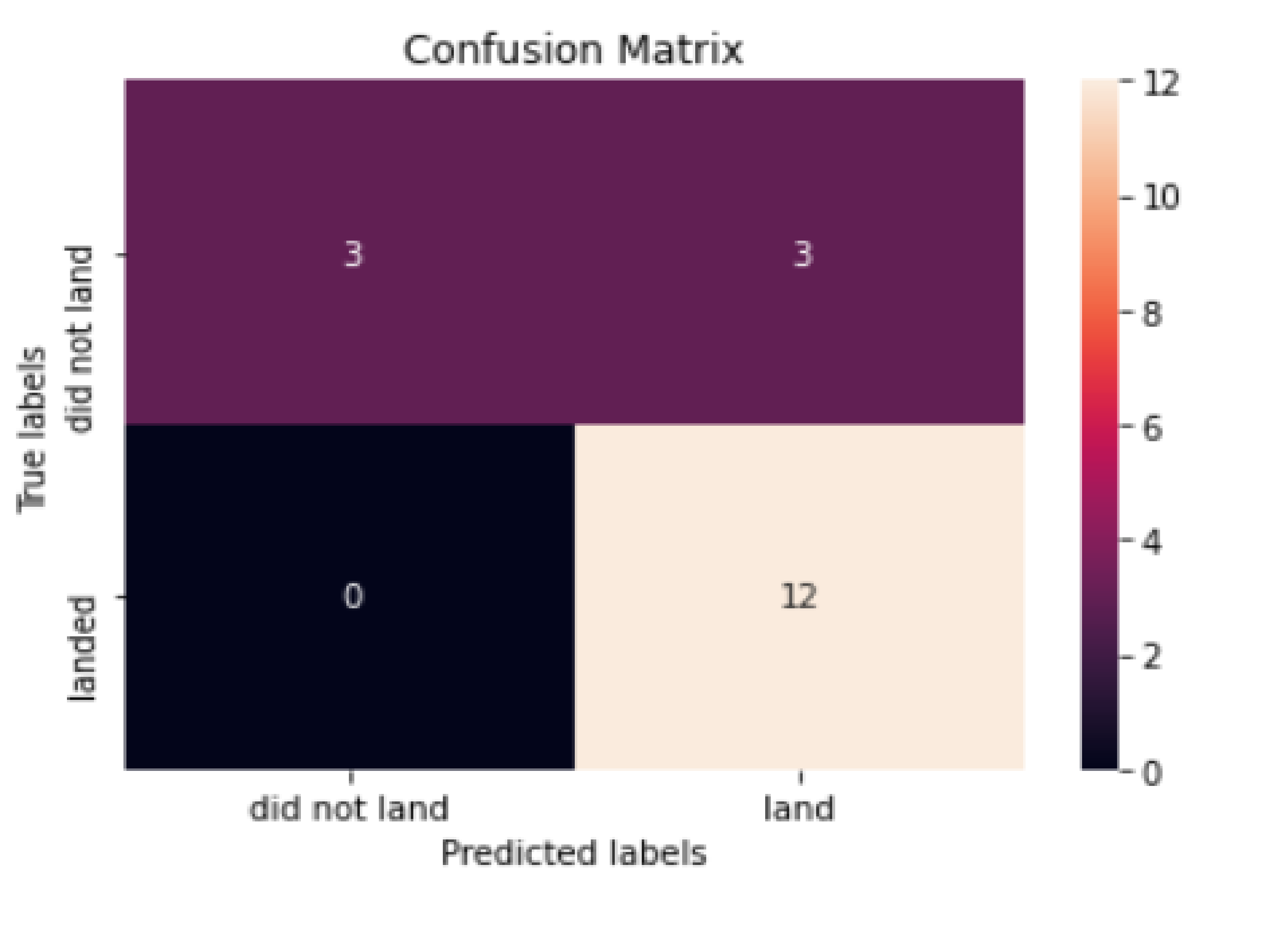
- Visualized the built model accuracy for all built classification models, in a bar chart

- We can see all the models perform similarly with an accuracy of 83%.



Confusion Matrix

- Since all the models performed the same on the data confusion matrix is also same, all the model
- Model predicted 12 successful landings when the label was successful landings
- Model predicted 3 unsuccessful landings when label was unsuccessful.



Conclusions

- SpaceY has to find a launch site close to ocean with close proximity to key locations like railway, highways and other key locations.
- Launch site should be close to Equator as the SapceX does.
- Allon Mask of SpaceY can use this model that we created to predict with relatively high accuracy whether a launch will have successful landing or not.
- Collecting more data can help increase the accuracy of the model raising it from 83% that we achieved from the given data.

Appendix

- GIT Hub Link : https://github.com/bluebird24/Data-Science-Capstone-Project/
- NumPy documentation: https://numpy.org/doc/
- SKlearn documentation: https://scikit-learn.org/stable/index.html
- Pandas documentation: https://pandas.pydata.org/pandas-docs/stable/
- Thanks to all the instructors of this course: https://www.coursera.org/professional-certificates/ibm-data-science

Thank you!