IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oleksandr Bondarenko
24 Aug 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 and Plotly Dash
  - 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. Missed values of PayloadMass column were filled with its mean
- 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

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

- This is the GitHub URL of the completed SpaceX API calls notebook https://github.com/Bondarenko-Alex/DS_Cert/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

1. Request a Rocket Launch data with requests.get() method
2. Create a dataframe from response JSON with Pandas json_normalize() method
3. Make additional requests to extract more data about BoosterVersion, LaunchSite, PayloadData and CoreData
4. Assembly all infos into one dataframe
5. Filter dataframe to only Falcon 9 launches
6. Replace NaN in PayloadMass column with mean value
7. Save dataset to future processing

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.
- The GitHub URL of the completed web scraping notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/jupyter-labs-webscraping.ipynb

1. Request a Falcon 9 Launch HTML page with requests.get() method
2. Create a BeautifulSoup object from HTML response
3. Parse HTML tables to extract column names and data to dictionary
4. Create dataframe from dictionary with parsed info
5. Save the dataframe to further processing

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 orbits
- We used one
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/labs-jupyter-spacex-Data_Wrangling.ipynb

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- The GitHub URL of my completed EDA with data visualization notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - 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 link to the notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/jupyter-labs-eda-sql-coursera.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 assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- The GitHub URL of our completed interactive map with Folium is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

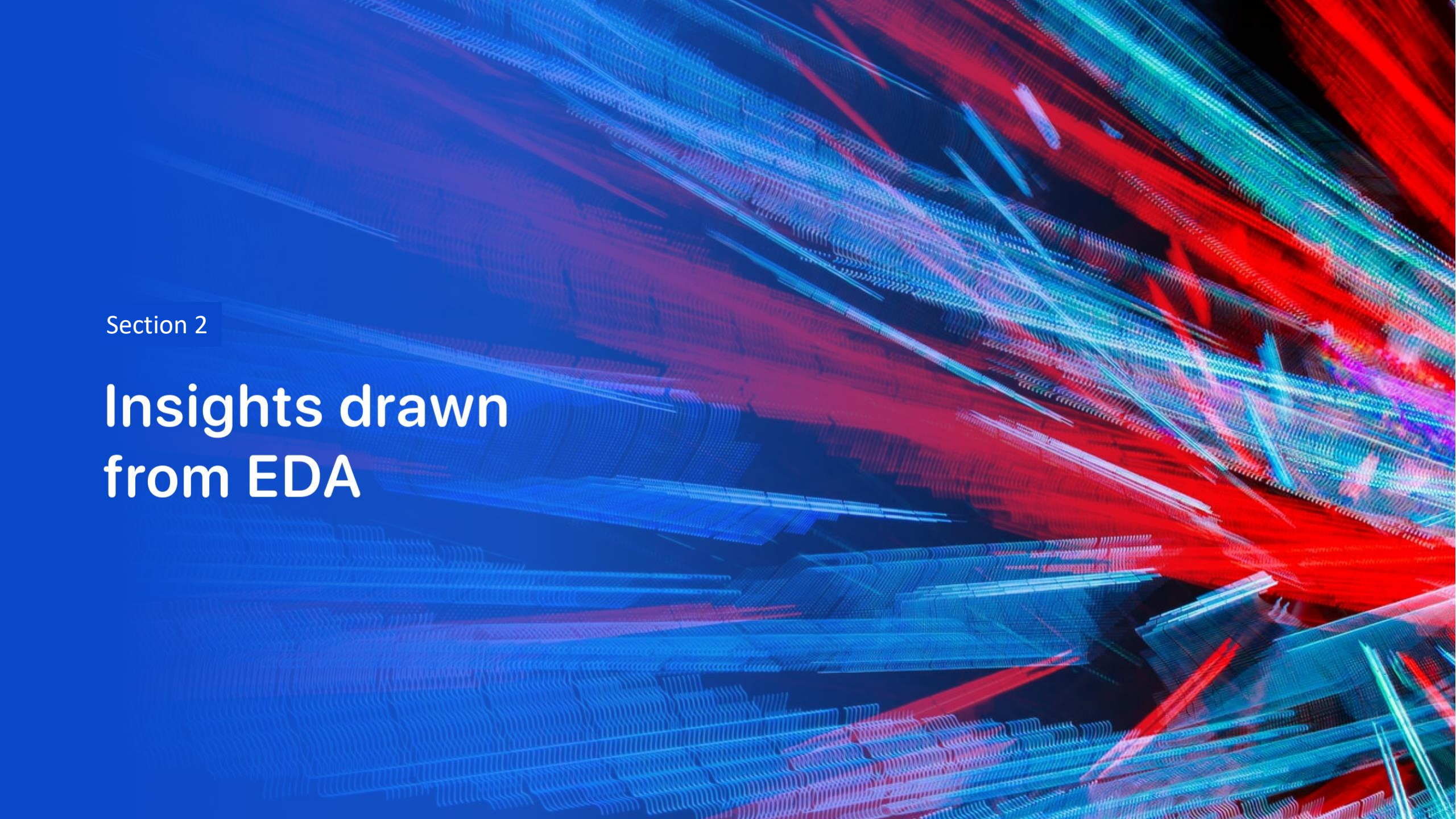
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is https://github.com/Bondarenko-Alex/DS_Cert/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

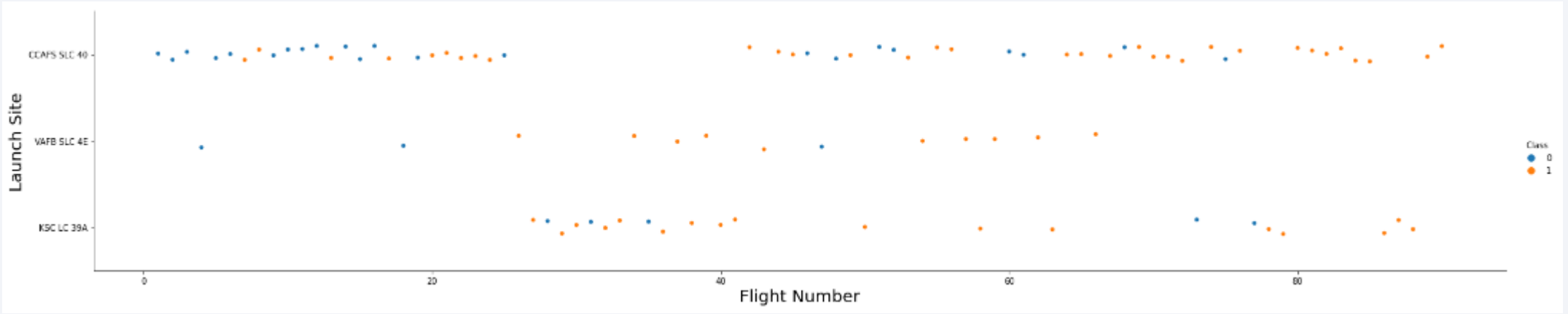
Results

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


Section 2
Insights drawn from EDA

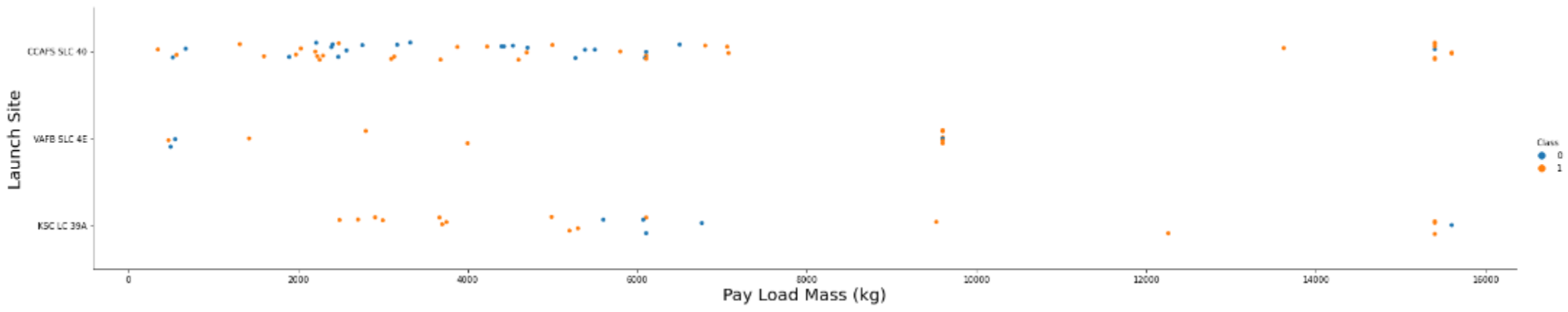
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



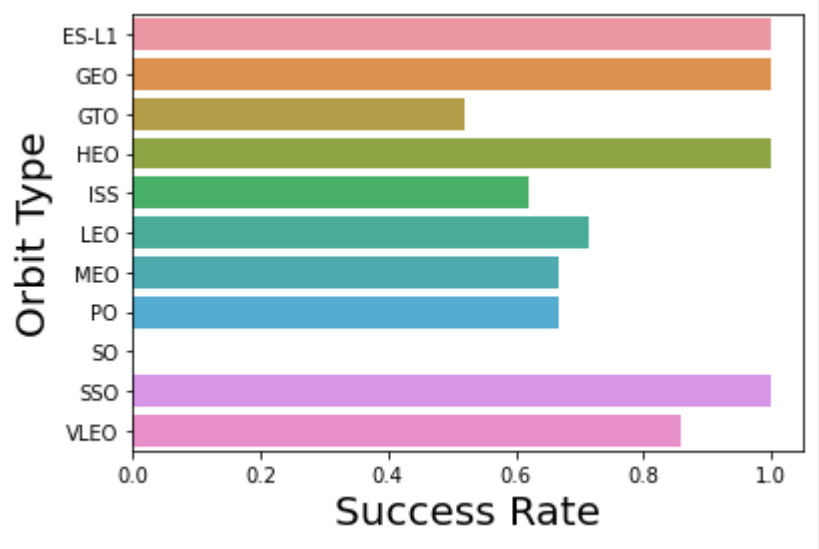
Payload vs. Launch Site

- When we observe Payload Vs. Launch Site scatter point chart we find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).
- We also find that the larger the payload, the more often the landing is successful.



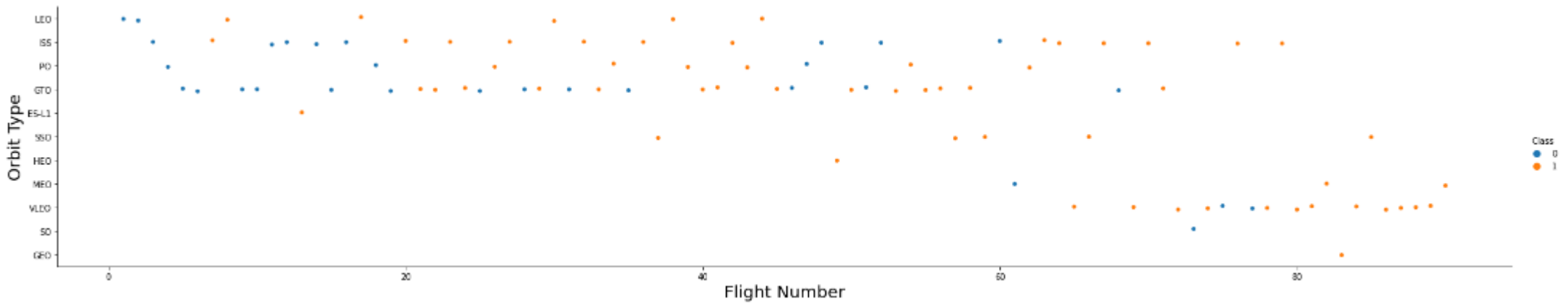
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



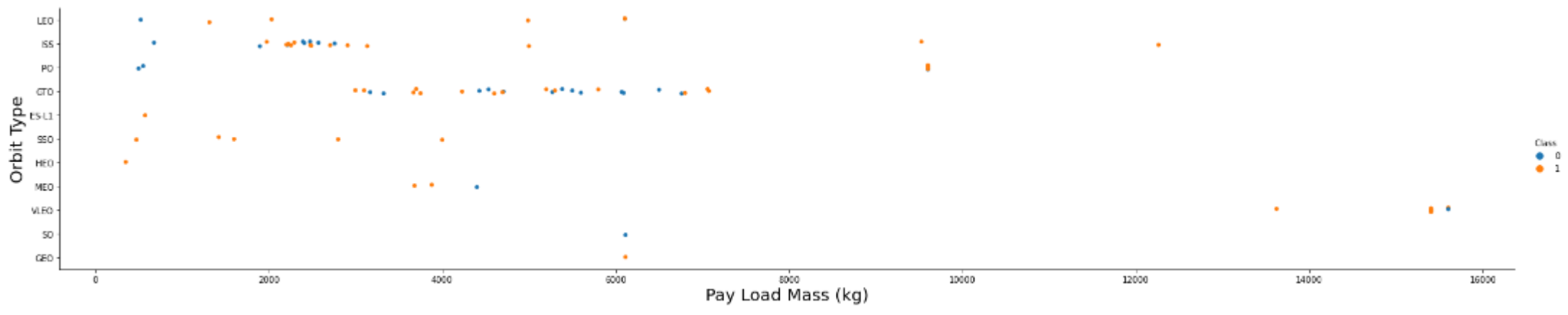
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe 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.



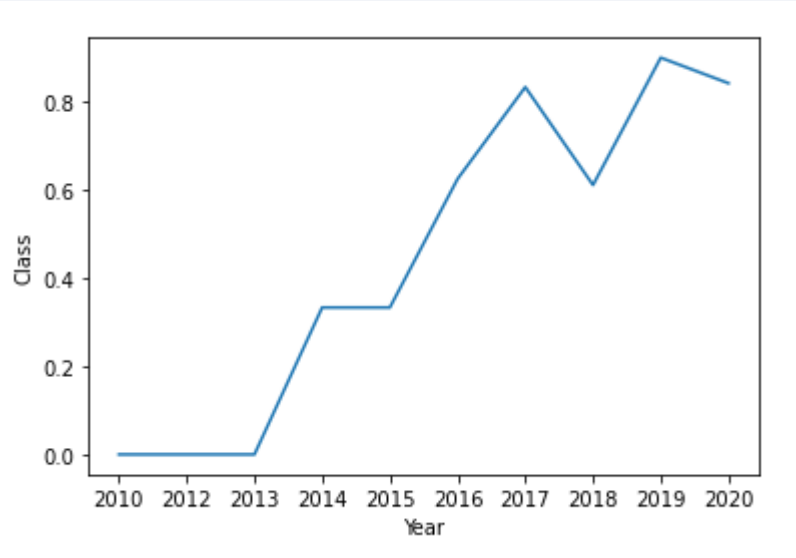
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.



Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- I use GROUP BY clause to show not only Launch Site Names but count of launches too.

Task 1

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

In [9]:
```
%sql select launch_site, count(*) from spacextbl group by launch_site
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Out[9]:

launch_site	2
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

Launch Site Names Begin with 'CCA'

- I use LIKE operator and LIMIT clause to find 5 records where launch sites begin with `CCA`

Task 2

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

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

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/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

- I used SUM aggregated function to calculate the total payload carried by boosters from NASA. I added also customer and launch count fields to output

Task 3

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

```
%sql select Customer, count(*) "Count", sum(payload_mass__kg_) "SUM Payload Mass" from spacextbl where customer = 'NASA (CRS)' group by customer
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

customer	Count	SUM Payload Mass
NASA (CRS)	20	45596

Average Payload Mass by F9 v1.1

- I used the AVG aggregated function to calculate the average payload mass carried by booster version F9 v1.1. I added also launch count field to output

Task 4

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

```
%sql select count(*) "Count", avg(payload_mass__kg_) "Average Payload Mass" from spacextbl where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Count	Average Payload Mass
15	2534

First Successful Ground Landing Date

- I used MIN aggregated function to find the dates of the first successful landing outcome on ground pad

```
%sql select min(DATE) "First Date" from spacextbl where landing__outcome ='Success (ground pad)'
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

First Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 I used BETWEEN operator

Task 6

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

```
%%sql
select booster_version, landing__outcome, payload_mass__kg_ from spacextbl
where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes I used GROUP BY clause with CASE operator to calculate mission status

Task 7

List the total number of successful and failure mission outcomes

```
%%sql
select case when landing__outcome like 'Success%' then 'Successful' else 'Failure' end "Status", count(*) "Count"
from spacextbl
group by case when landing__outcome like 'Success%' then 'Successful' else 'Failure' end
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Status	Count
Failure	40
Successful	61

Boosters Carried Maximum Payload

```
%sql select booster_version from spacextbl where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextbl)
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

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

- I used subquery to show lList the names of the booster_versions which have carried the maximum payload mass.

2015 Launch Records

- To list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 I used YEAR function and IN operator. I added landing__outcome column too.

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select booster_version, launch_site, year(date), landing__outcome "Year" from spacextbl
where landing__outcome in ('Failure (drone ship)', 'Precluded (drone ship)') and year(date) = 2015
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

booster_version	launch_site	3	Year
F9 v1.1 B1012	CCAFS LC-40	2015	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	2015	Failure (drone ship)
F9 v1.1 B1018	CCAFS LC-40	2015	Precluded (drone ship)

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

Task 10

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(*) "Count" from spacextbl
where date between '2010-06-04' and '2017-03-20'
group by landing__outcome order by 2 desc
```

```
 * ibm_db_sa://xsz30907:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

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

- To 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 I used BETWEEN operator and GROUP BY and ORDER BY clauses.

Section 3

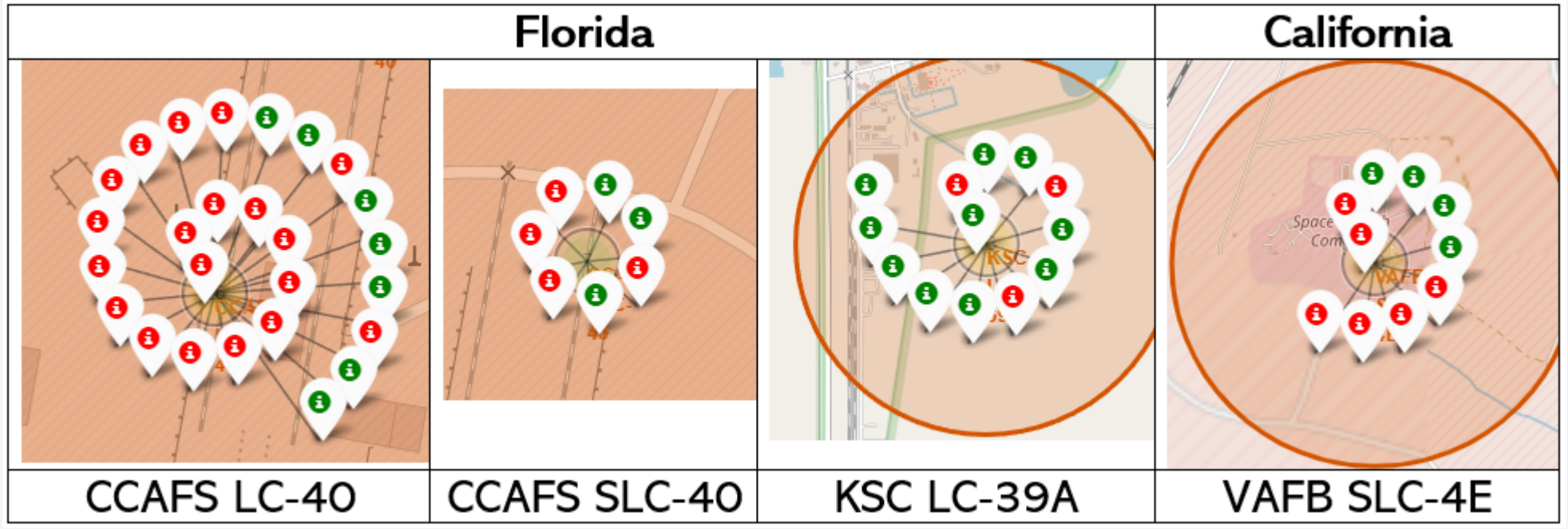
Launch Sites Proximities Analysis

All launch sites global map markers

- We can see that the SpaceX launch sites are in the USA coasts in Florida and California, as close to the equator as possible.

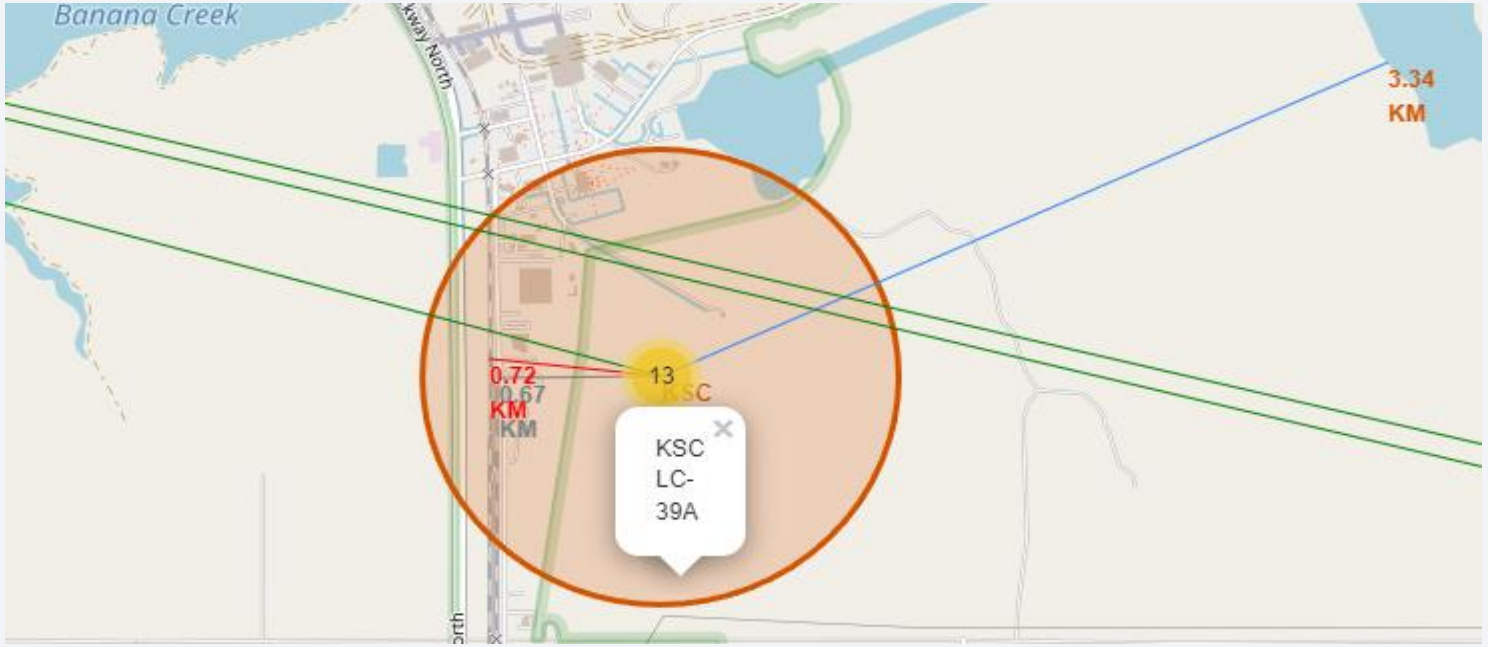
Markers showing launch sites with color labels

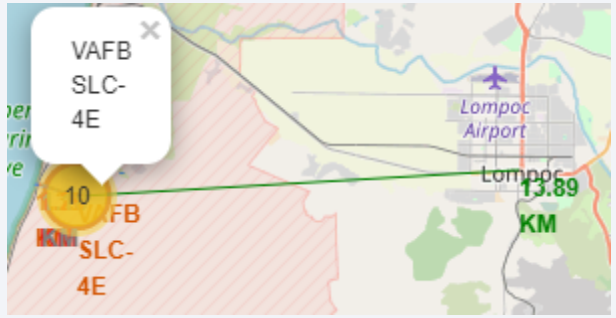
- Most launches were from Florida.
- CCAFS LC-40 and VAFB SLC-4E frequently used for testing purposes
- CCAFS LC-40 and CCAFS SLC-40 are nearby and the first is more often used



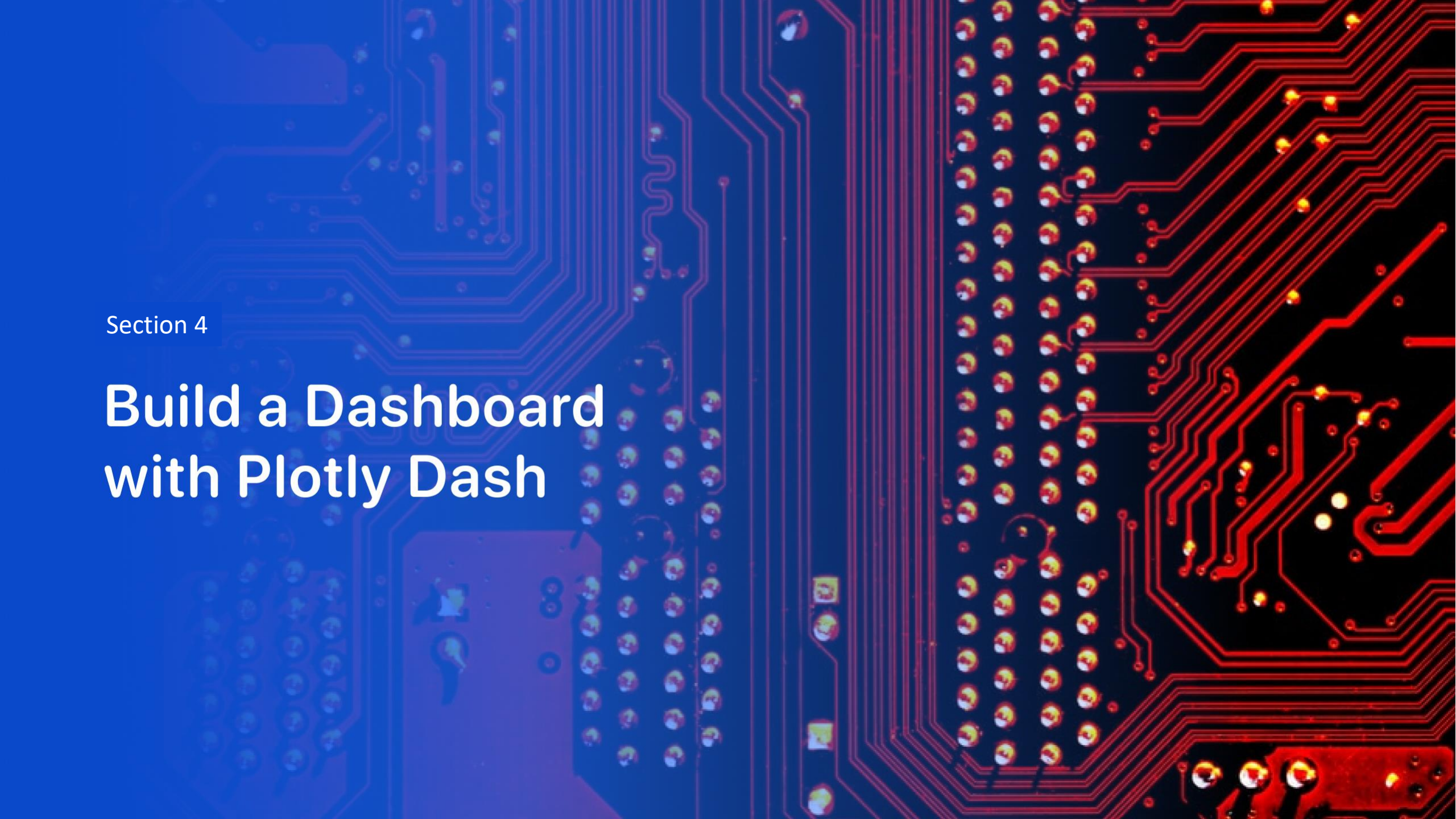
Launch Site distance to landmarks

- All launch sites located near railways, highways and coast lines, it solves logistical problems.
- And keep certain distance away from towns to avoid casualties in a crash



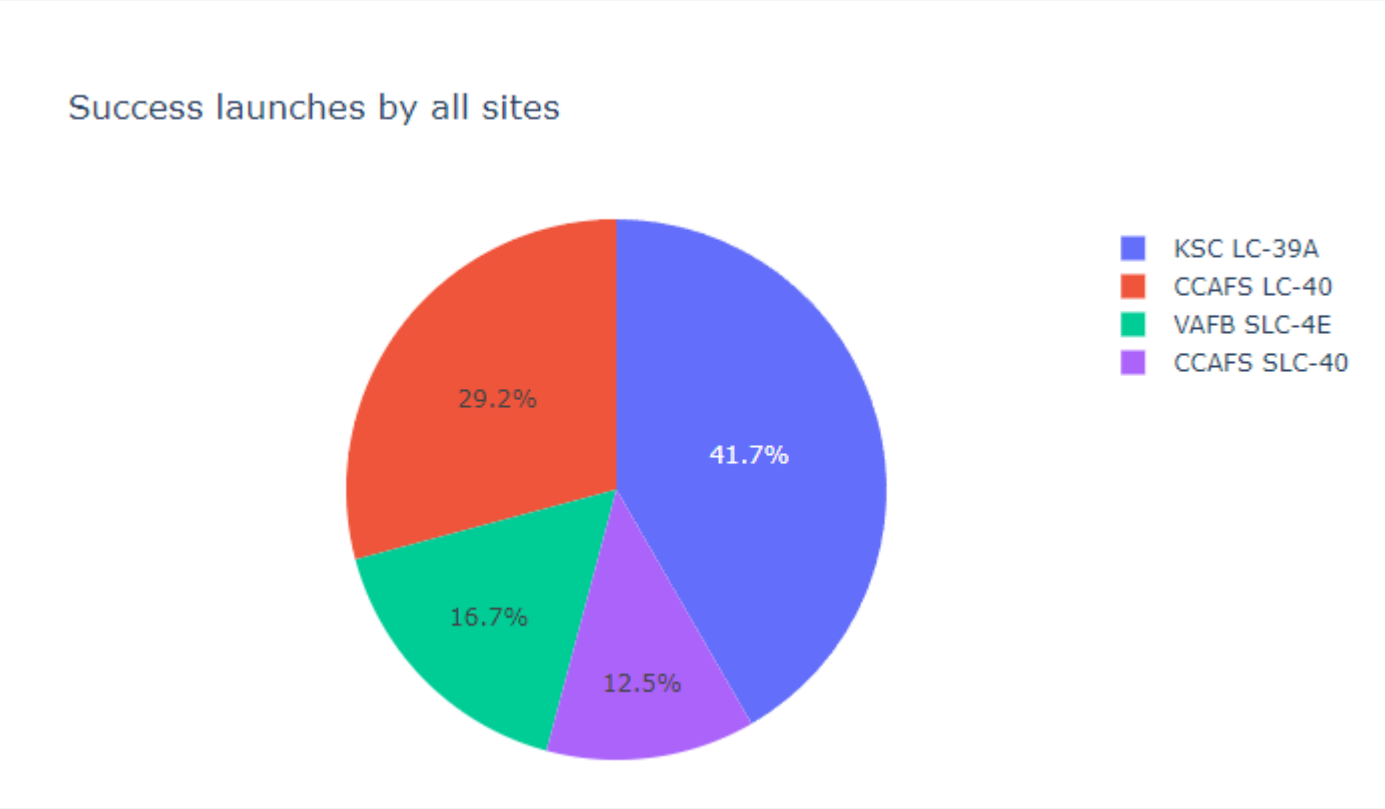


Site/Dist to	Coastline	Railway	Highway	Town (km)
CCAFS LC-40	0.930	1.338	0.306	23.223
CCAFS SLC-40	0.856	1.291	0.412	23.250
KSC LC-39A	3.341	0.718	0.673	16.342
VAFB SLC-4E	1.353	1.257	1.176	13.887


Section 4
Build a Dashboard with Plotly Dash

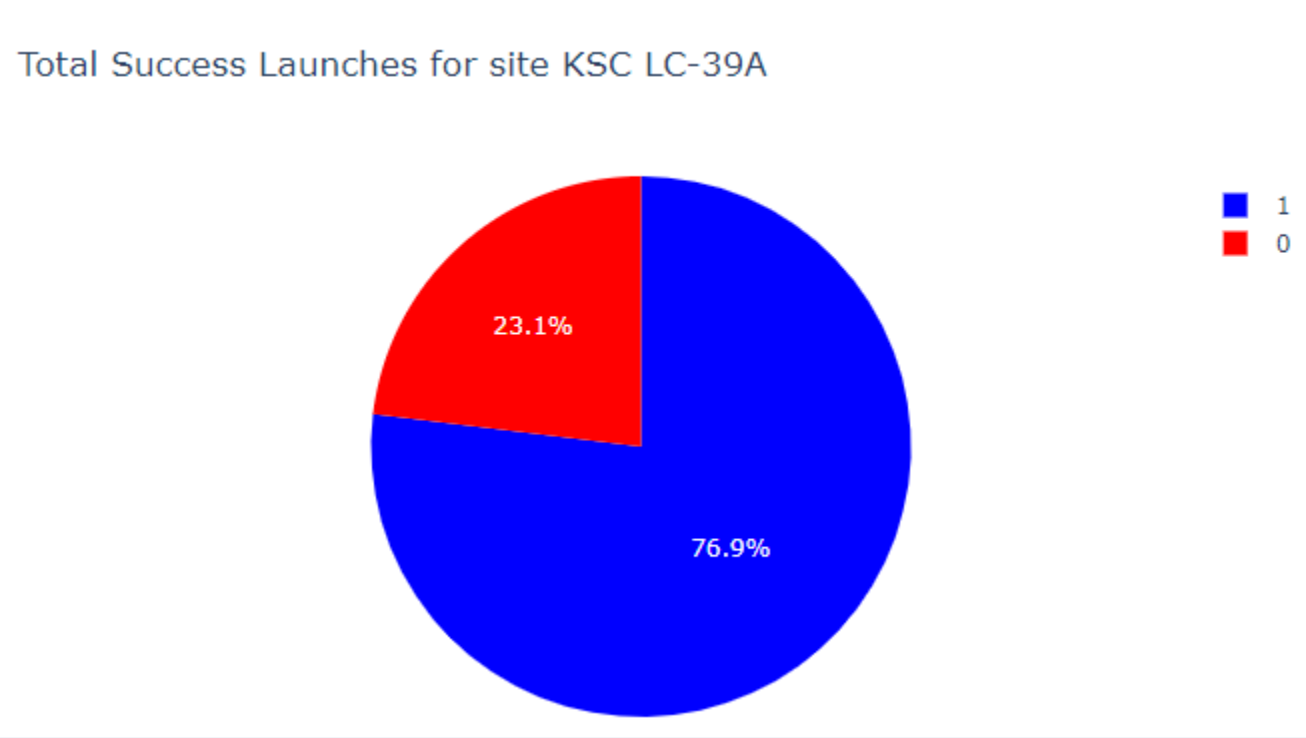
Pie chart showing the success percentage achieved by each launch site

- We see, that KSC LC-39A had the most successful launches from all sites



Launch site with highest launch success ratio

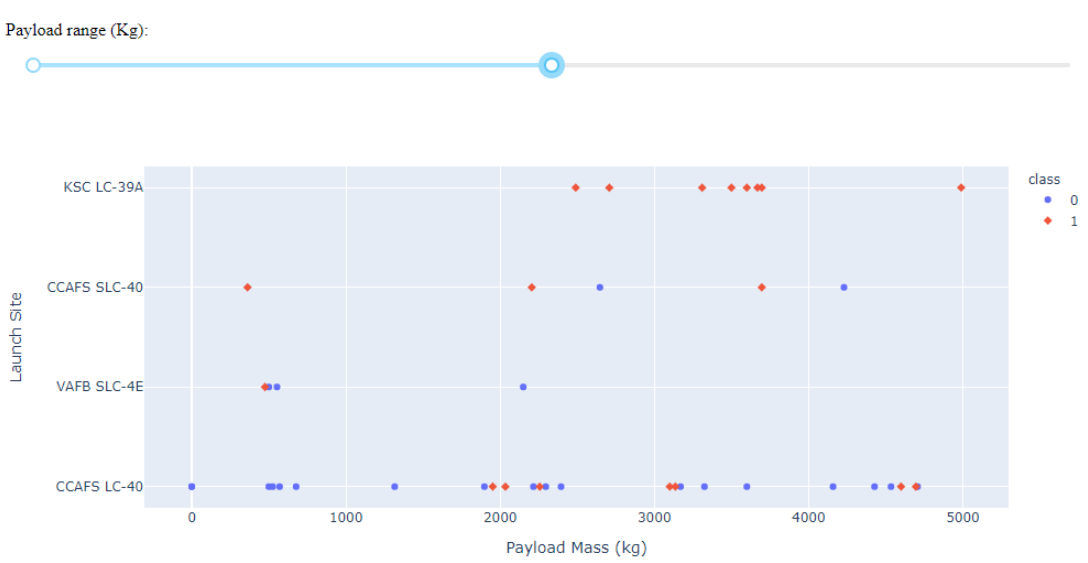
- KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate



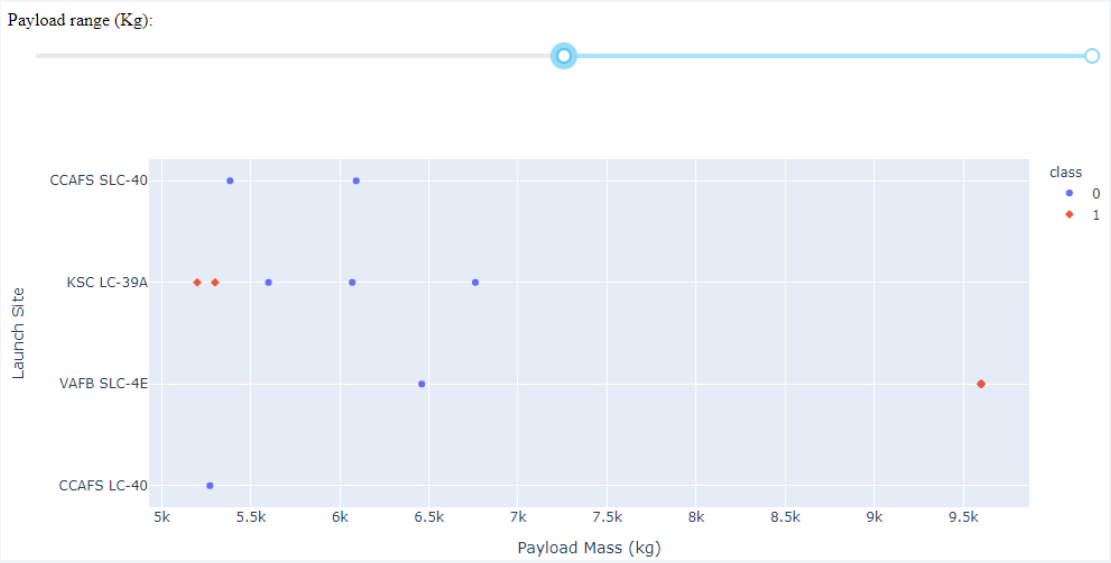
Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

- We can see the success rates for low weighted payloads are higher than the heavy weighted payloads

Low weighted Payloads



High weighted Payloads

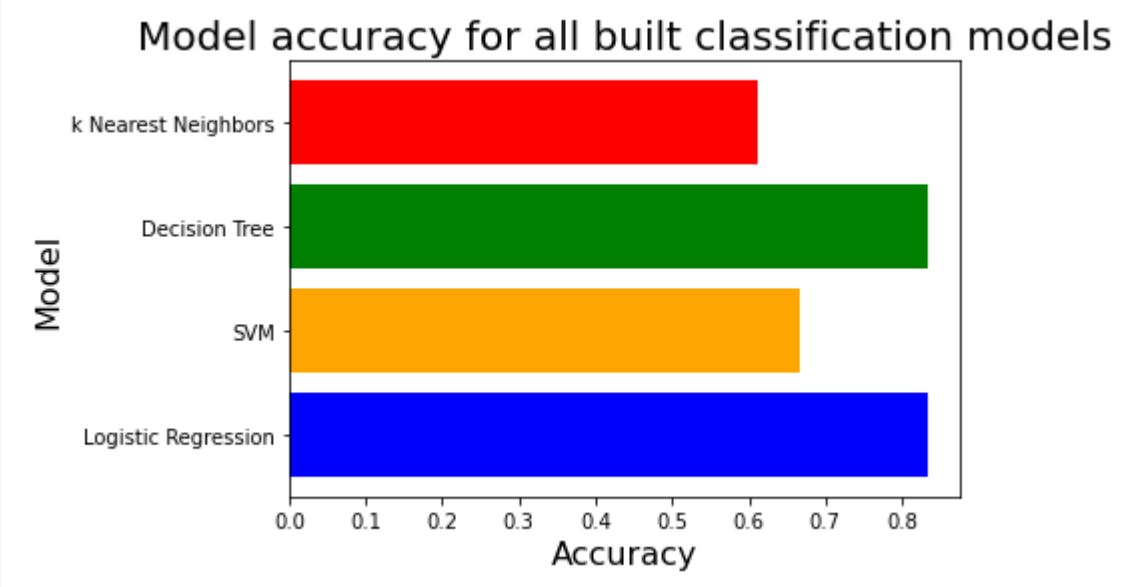


Section 5

Predictive Analysis (Classification)

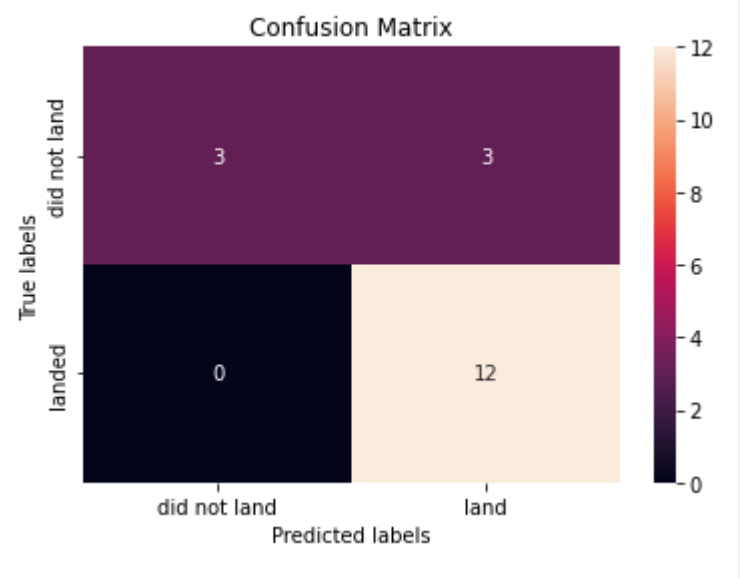
Classification Accuracy

- The highest classification accuracy on test dataset had Logistic Regression and Decision Tree Classificators.



Confusion Matrix

- Both of best models have the same confusion matrix.
- This matrix 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

We can conclude that:

- 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 Logistic Regression and Decision Tree classifiers are the best machine learning algorithmes for this task.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!