IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anastasiia Chernova
18.11.2024

Outline

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

Executive Summary

- Summary of methodologies
  - SpaceX Data Wranglin
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Exploratory Data Analysis using SQL
  - SpaceX Machine Learning Landing Prediction
  - SpaceX EDA DataViz using Python Pandas and Matplotlib
  - SpaceX Launch Sites Analysis with Folium-Interactive Visual Analytics and PlotyDash
- Summary of all results
  - EDA results
  - Predictive Analysis (classification)
  - Interactive Visual Analytics and Dashbords

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. 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 SpaceX for a rocket launch.

- Problems you want to find answers

In this work we will predict if the Falcon 9 first stage will successfully, using data from Falcon 9 rocket launchers advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

The data from SpaceX were collected from the following resources:
SpaceX API (https://api.spacexdata.com/)
WebScraping (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

- Perform data wrangling

Collected data were enriched by cresting a landing outcome lable based on outcome data after summarizing and analizyng 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

Data wew collected until this step wewrw normalithed , divided in training and test dataset and evaluated by four different classification models

Data Collection

- Describe how data sets were collected.

○ At first, data were collected using SpaceX API (RESTful API) by making a get request to the SpaceX API. It was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.

○ To make the requested JSON results more consistent, the SpaceX launch data were requested and parsed using the GET request and then decoded the response content as a Json result which was converted into a Pandas data frame.

○ Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in a HTML: Using BeautifulSoup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas data frame.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/Anastasiia-Che/Lab-1-SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/jupyter-labs-spacex-data-collection-api.ipynb), as an external reference and peer-review purpose

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for thi

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS03
```

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

```
response.status_code
```

```
200
```

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

```
# 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

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- GitHub URL of the completed web scraping notebook: https://github.com/Anastasiia-Che/Web-scraping-Falcon-9-and-Falcon-Heavy-Launches-Records-from-Wikipedia/blob/main/jupyter-labs-webscraping%20(1).ipynb

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, as an HTTP response.

```
response = requests.get(static_url).text
```

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

```
soup = BeautifulSoup(response, 'html.parser')
```

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

```
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSou`
reference link towards the end of this lab

```
html_tables = soup.find_all("table")
print(html_tables)
```

Data Wrangling

- After obtaining and creating a Pandas DF from the collected data, data were filtered to only keep the Falcon 9 launches, then dealt with the missing data values in the LandingPad andbPayloadMass columns. Missing data values were replaced using mean value of column.
- Performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- GitHub URL of completed data wrangling related notebooks:

https://github.com/Anastasiia-Che/Lab-2-Data-wrangling/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`
it's one. Then assign it to the variable `landing_class`:

```
landing_class = []

for i in df["Outcome"]:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

print(landing_class)
```

```
0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1
, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
, 1, 1, 1, 1, 1, 1, 1]
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first st
land successfully; one means the first stage landed Successfully

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0

EDA with Data Visualization

- Data Analysis and Feature Engineering were performed using Pandas and Matplotlib:
- Exploratory Data Analysis
- Preparing Data Feature Engineering

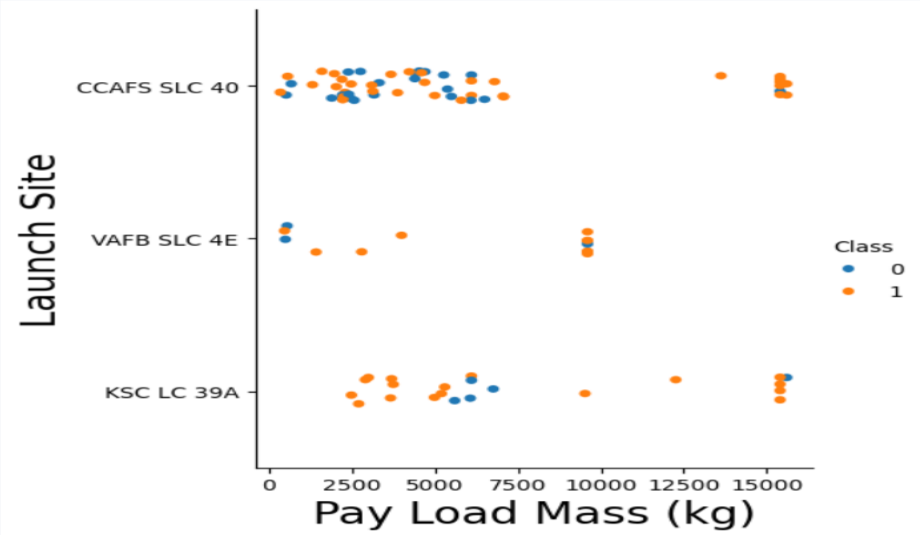
- To Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit type, Payload and Orbit type were used scatter plots.

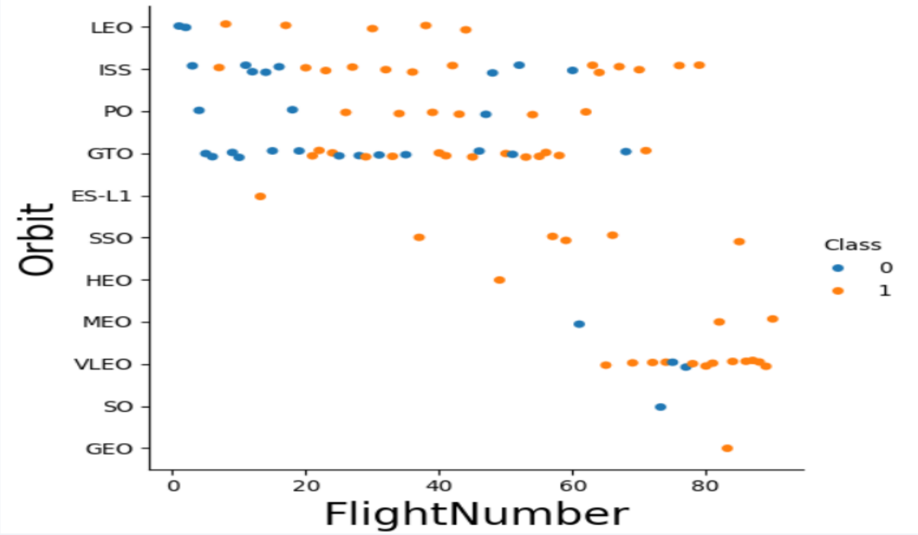
- To Visualize the relationship between success rate of each orbit type was used Bar chart.

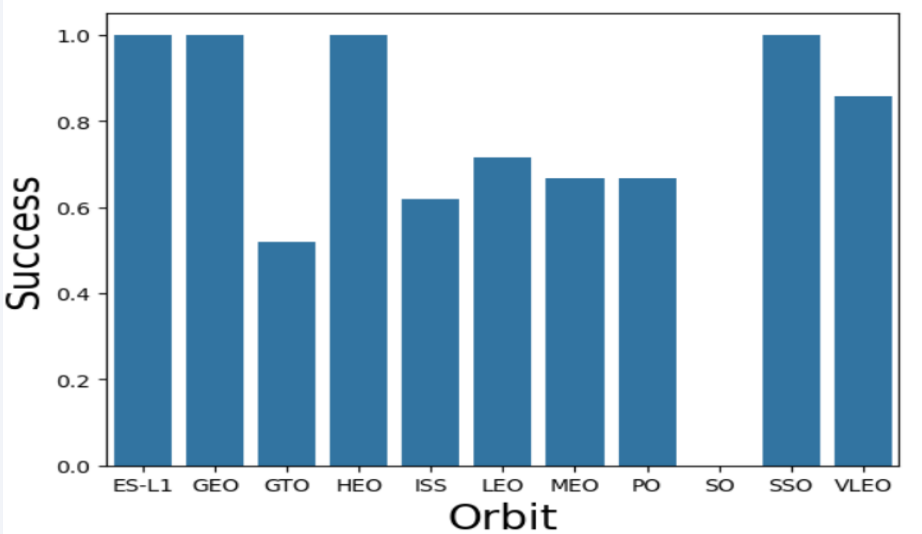
- To Visualize the launch success yearly trend was used Line plot.

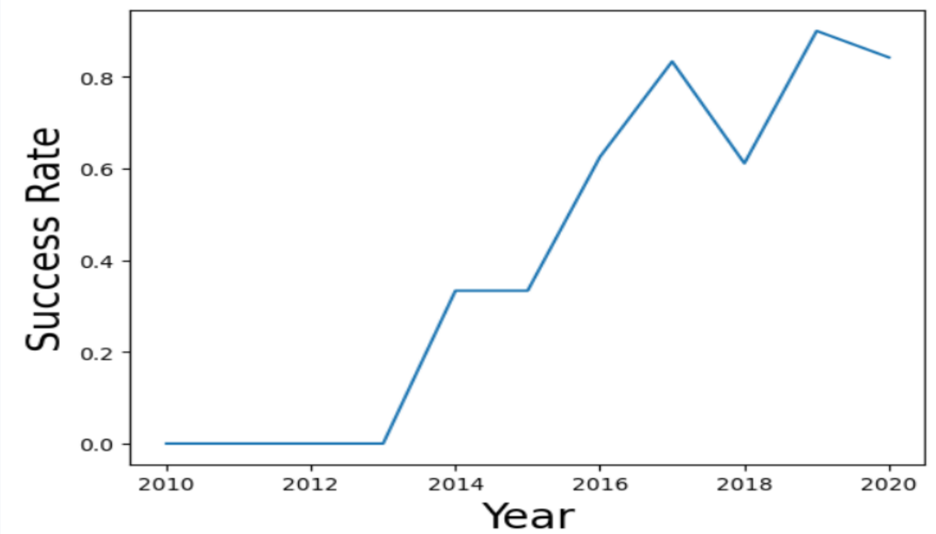
- GitHub URL of completed EDA with data visualization notebook: https://github.com/Anastasiia-Che/EDA-with-Visualization-Lab/blob/main/edadataviz%20(1).ipynb

EDA with Data Visualization: Bar Chart, Line Chart, Scatter Plot









EDA with SQL

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

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

```
%sql SELECT Launch_Site FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
```

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

```
%sql SELECT avg(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1';
```

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

```
%sql SELECT min(date) AS Early_Date from SPACEXTBL where Landing_Outcome LIKE 'Success (ground pad)'
```

- 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 DISTINCT Customer, Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome ='Success (drone ship)
```

EDA with SQL

- List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight
```

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

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

- 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.

```
%sql SELECT SUBSTR(Date,4,2) AS Month, Booster_Version, Launch_site FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure%drone
```

- 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(*) AS Numbers FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%' AND Date BETWEEN '04-
```

- The GitHub URL of completed EDA with SQL notebook: https://github.com/Anastasiia-Che/Assignment-SQL-Notebook/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.

- Markers indicate points like launch sites.
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center.
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site.
- Lines are used to indicate distances between two coordinates.

- Created a launch set outcomes (failure=0 or success=1).

- The GitHub URL of completed interactive map with Folium map: https://github.com/Anastasiia-Che/Hands-on-Lab-Interactive-Visual-Analytics-with-Folium-lab/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Built an interactive dashboard application with Plotly dash by:

- Adding a Launch Site Drop-down Input Component.
- Adding a callback function to render success-pie-chart based on selected site dropdown.
- Adding a Range Slider to Select Payload.
- Adding a callback function to render the success-payload-scatter-chart scatter plot.

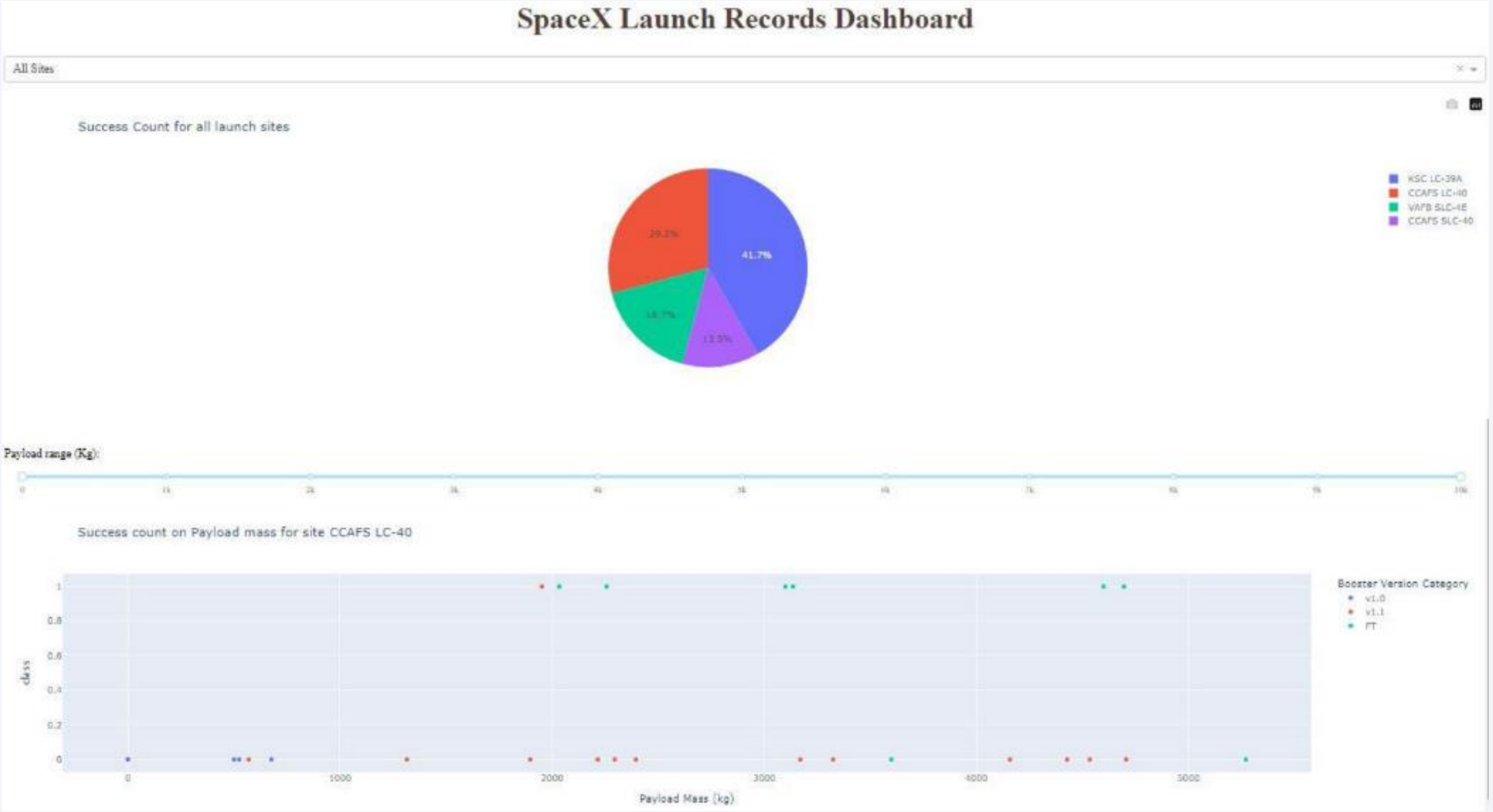
The following graphs and plots were used to visualize data

- Percentage of launches by site.
- Payload range.

This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is best place to launch according to payloads.

- The GitHub URL of completed Plotly Dash lab: https://github.com/Anastasiia-Che/Hands-on-Lab-Build-an-Interactive-Dashboard-with-Ploty-Dash

SpaceX Dash App



Predictive Analysis (Classification)

- To find the best ML method that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression were did next steps:

- Was created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.
- For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find best Hyperparameter.
- Next, after fitting the training set, was outputed GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params_and the accuracy on the validation data using the data attribute best_score
- Next, using the method score, was calculate the accuracy on the test data for each model and plotted a confussion matrix for each (using the test and predicted outcomes).

- The GitHub URL of completed predictive analysis lab: https://github.com/Anastasiia-Che/Space-X-Falcon-9-First-Stage-Landing-Prediction/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

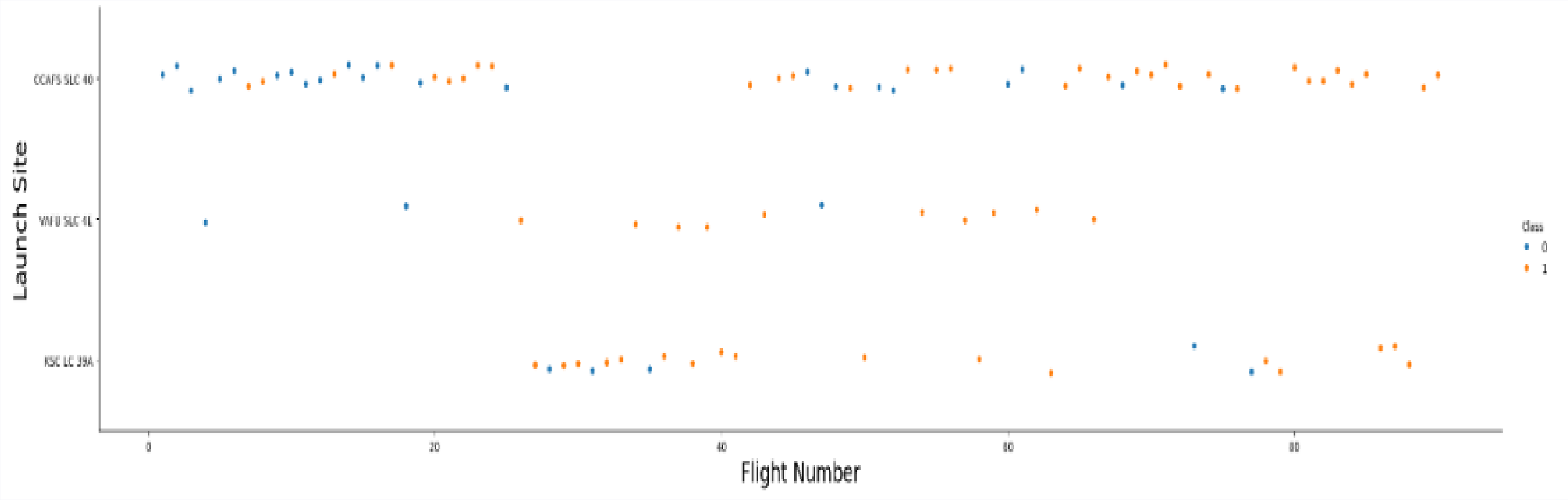
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

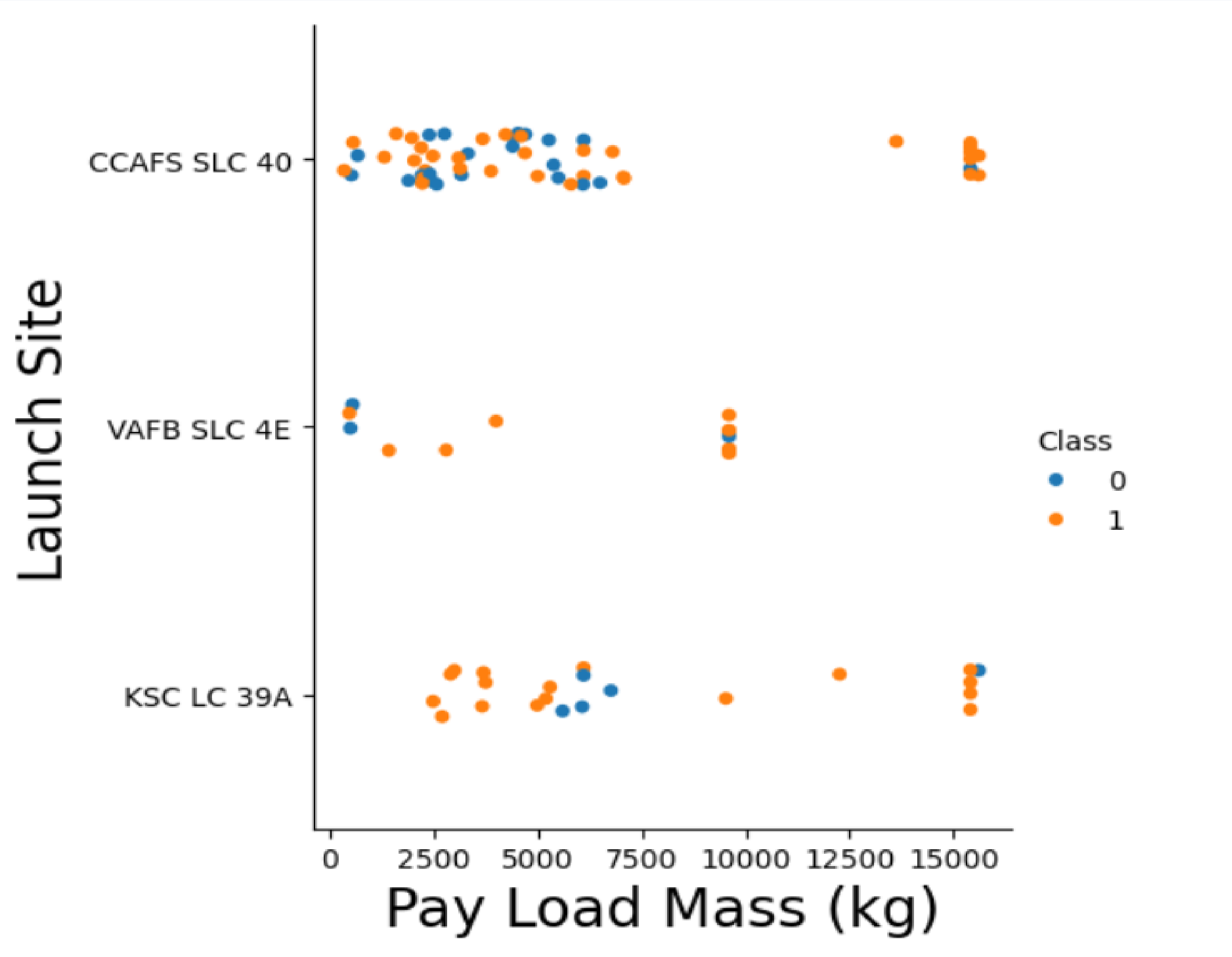
Flight Number vs. Launch Site

A scatter plot of Flight Number vs. Launch Site



Based on the plot above, it's possible to conclusion that the best launch site is CCAF5 SLC 40, where most of recent launches were successful.
In second place VAFB SLC 4E and third place KSC LC 39A.

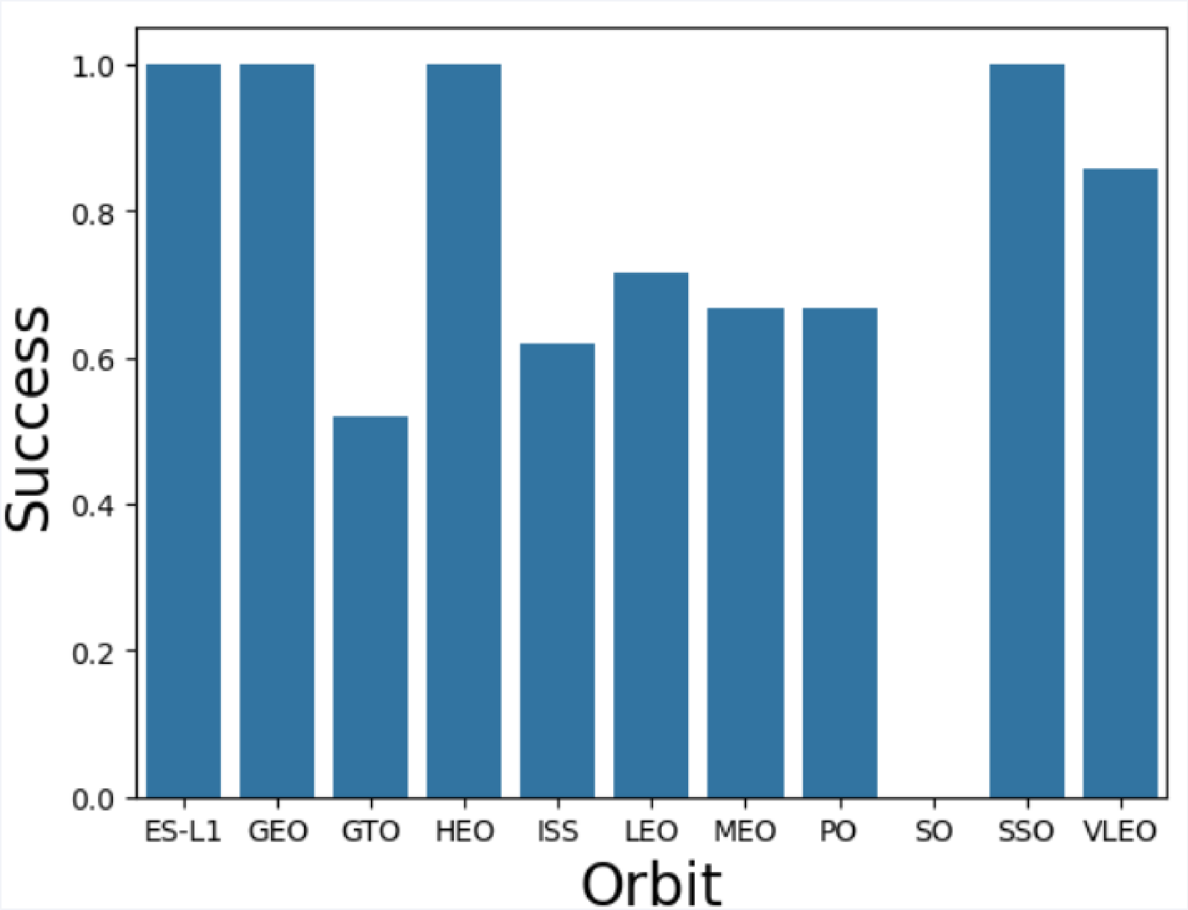
Payload vs. Launch Site



According to the data on the scatter plot of Payload vs. Launch Sit we can conclude:

- Payloads over 9,000kg have excellent success rate.
- Payloads over 12,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.

Success Rate vs. Orbit Type

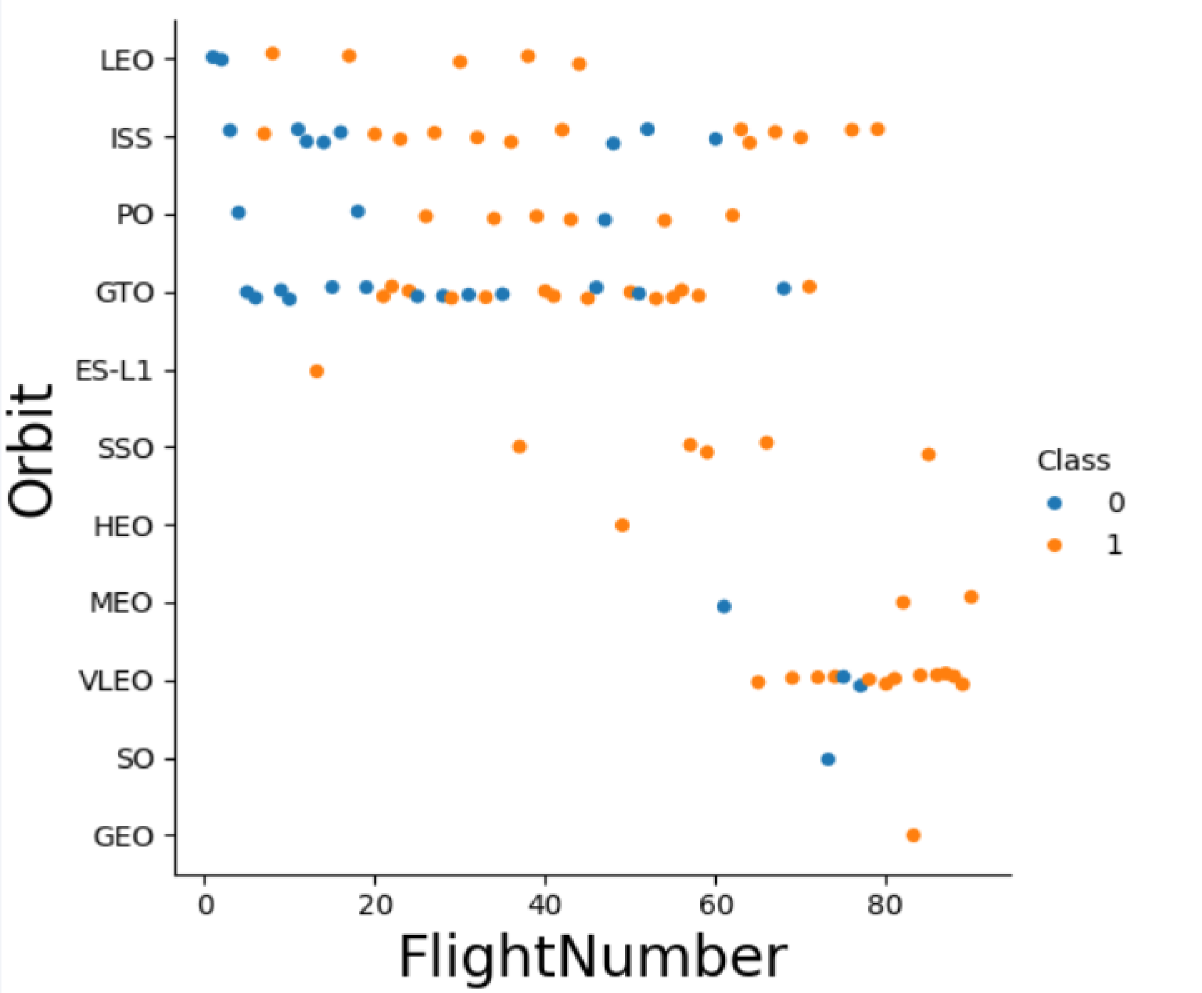


- The biggest success rates at 100% have next orbits:

- ES-L1
- GEO
- HEO
- SSO

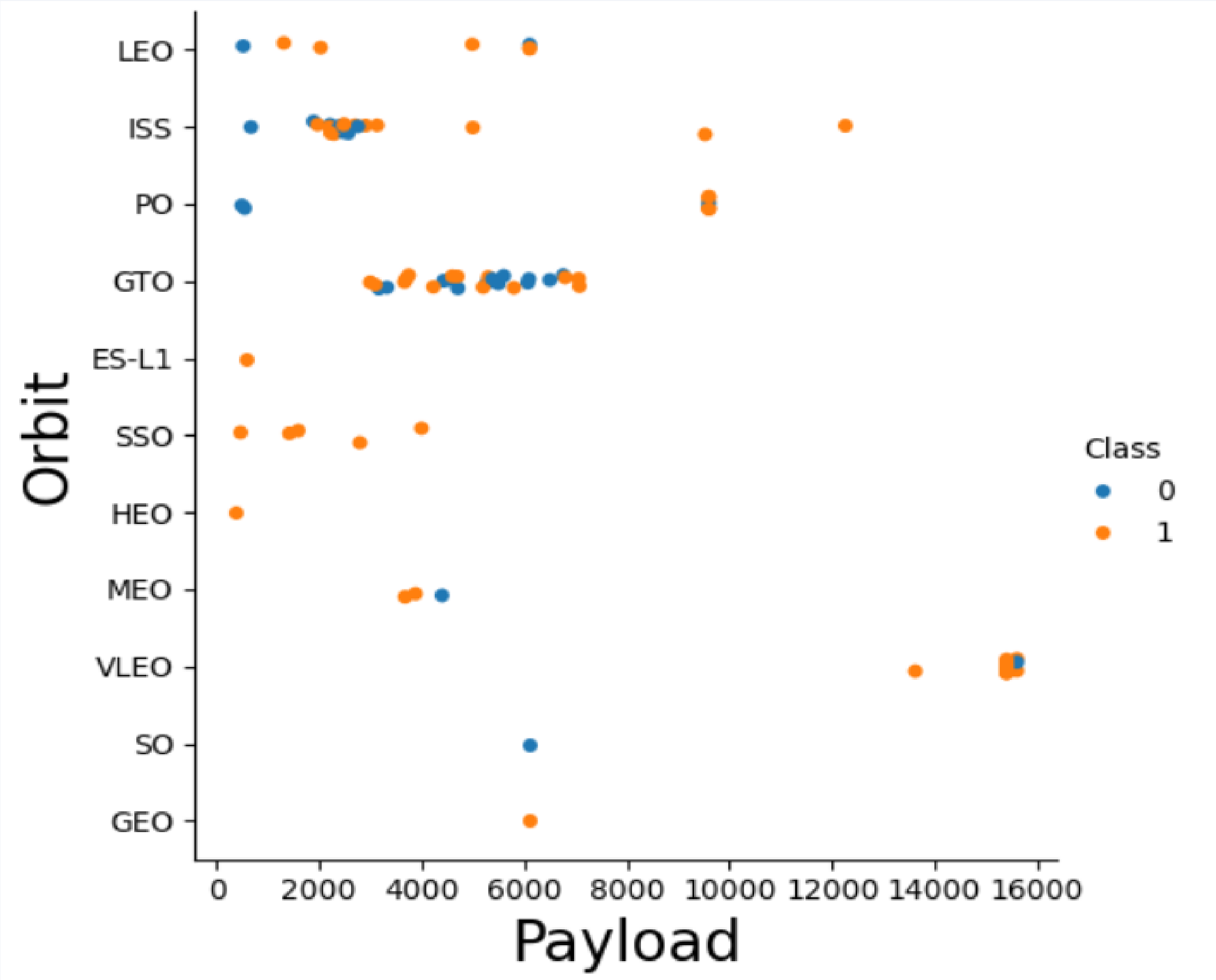
- SO orbit has the lowest success rate (0%)

Flight Number vs. Orbit Type



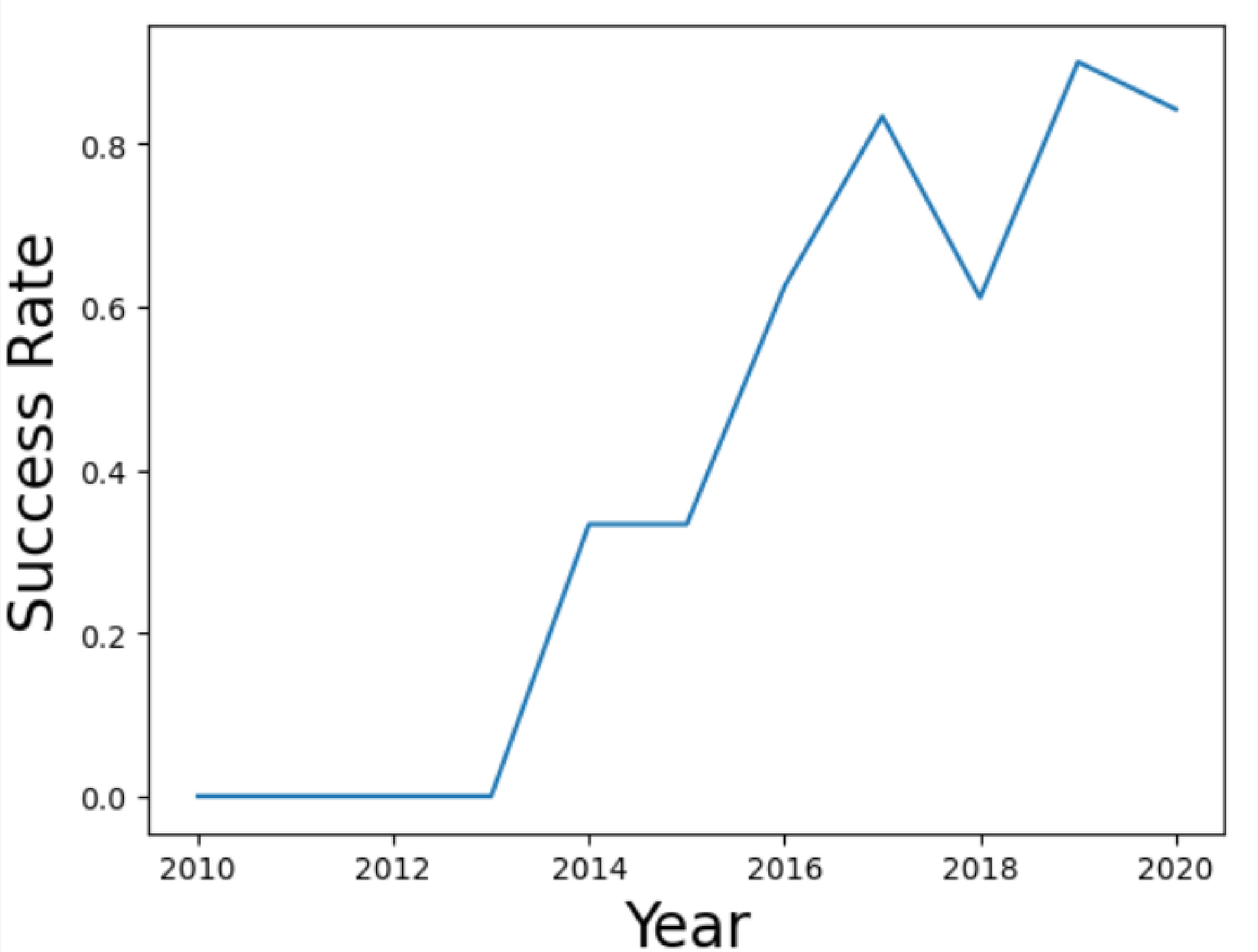
- Success rate improved over time to all orbits.
- VLEO orbit happened recent increase of its frequency

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- ISS orbit has the widest range of payload and a good rate of success;
- There are only few launches to the orbits SO and GEO

Launch Success Yearly Trend



- Since 2013 the success rate kept going up till 2020

All Launch Site Names

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

- Find the names of the unique launch sites
- Used ‘Select’ Distinct statement to return only the unique launch sites from the ‘Launch_Site’ column on the SPACEXTBL table

Launch Site Names 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;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOA
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	

- Used 'Like' command with '%' wildcard in 'Where' claise to select and display a table of 5 (used 'Limit 5'command) records where launch sites begin with \`CCA\`

Total Payload Mass

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTBL
```

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

total_payload
45596

- Used the 'SUM()' function to return and display the total sum of PAYLOAD_MASS_KG for customer 'NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql SELECT avg(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1';
```

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

Avg_Payload
2928.4

- Used the ‘AVG()’ function to return and display average payload mass. Obtained the value of 2928,4

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql SELECT min(date) AS Early_Date from SPACEXTBL where Landing_Outcome LIKE 'Success (ground pad)'
```

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

Early_Date
2015-12-22

- Used 'MIN()' function to return the date when the first successful landing outcome in ground pad was achieve. It's happened on 22.12.2015

Successful Drone Ship Landing with Payload between 4000 and 6000

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 DISTINCT Customer, Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome ='Succes
```

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

Customer	Landing_Outcome	PAYLOAD_MASS__KG_
SKY Perfect JSAT Group	Success (drone ship)	4696
SKY Perfect JSAT Group	Success (drone ship)	4600
SES	Success (drone ship)	5300
SES EchoStar	Success (drone ship)	5200

- Used 'SELECT DISTINGT' statement to return and list the unique names

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight
```

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

count(MISSION_OUTCOME)
99

- Used the ‘Count’ statement to calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

Task 8

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

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

- There are the boosters which have carried the maximum payload mass registered in the dataset

2015 Launch Records

Task 9

List the records which will displav the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the mo

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mis
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

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

Task 10

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

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;
```

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

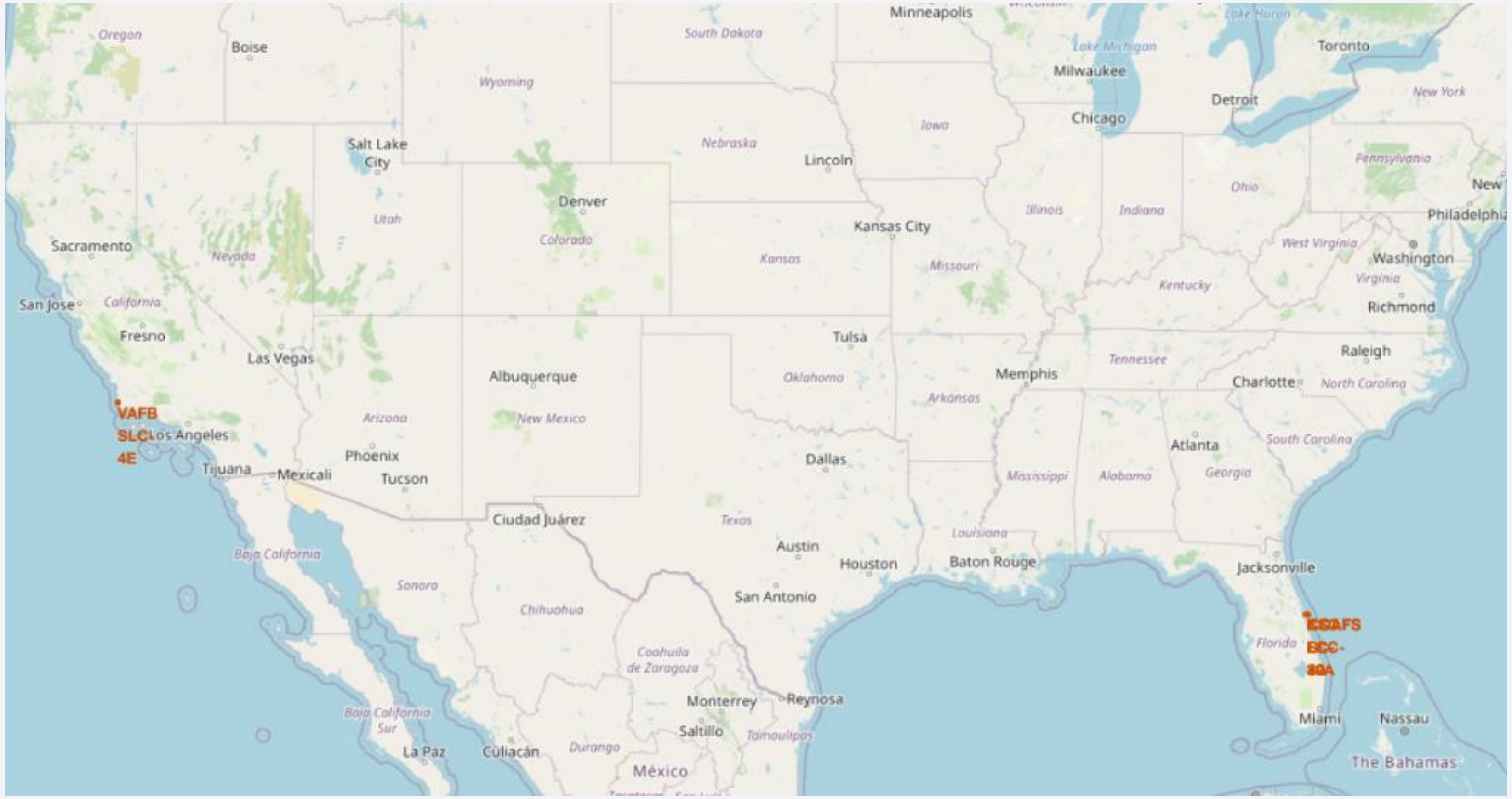
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)

- Result of ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20

Section 3

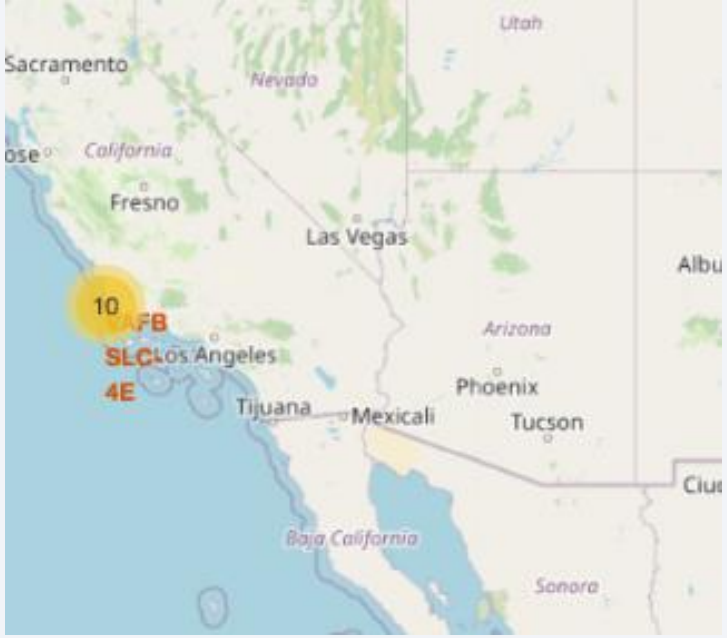
Launch Sites Proximities Analysis

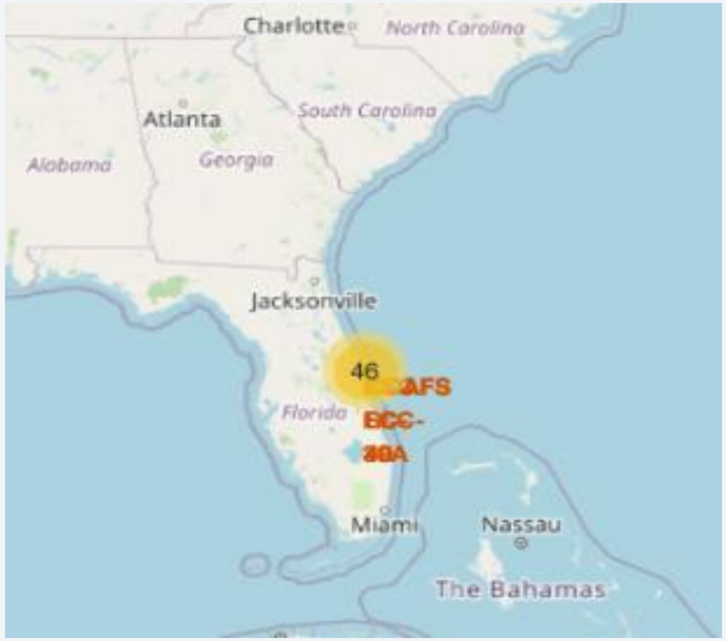
Markers of Launch Sites on Global Map



- According to the map, launch sites are near sea. Probably it's by safety and for convenience, but not too far from roads and railroads.

Launch Outcomes for Each Site on the Map





- Green marker used if a launch was successful
- Red marker used if a launch was failed

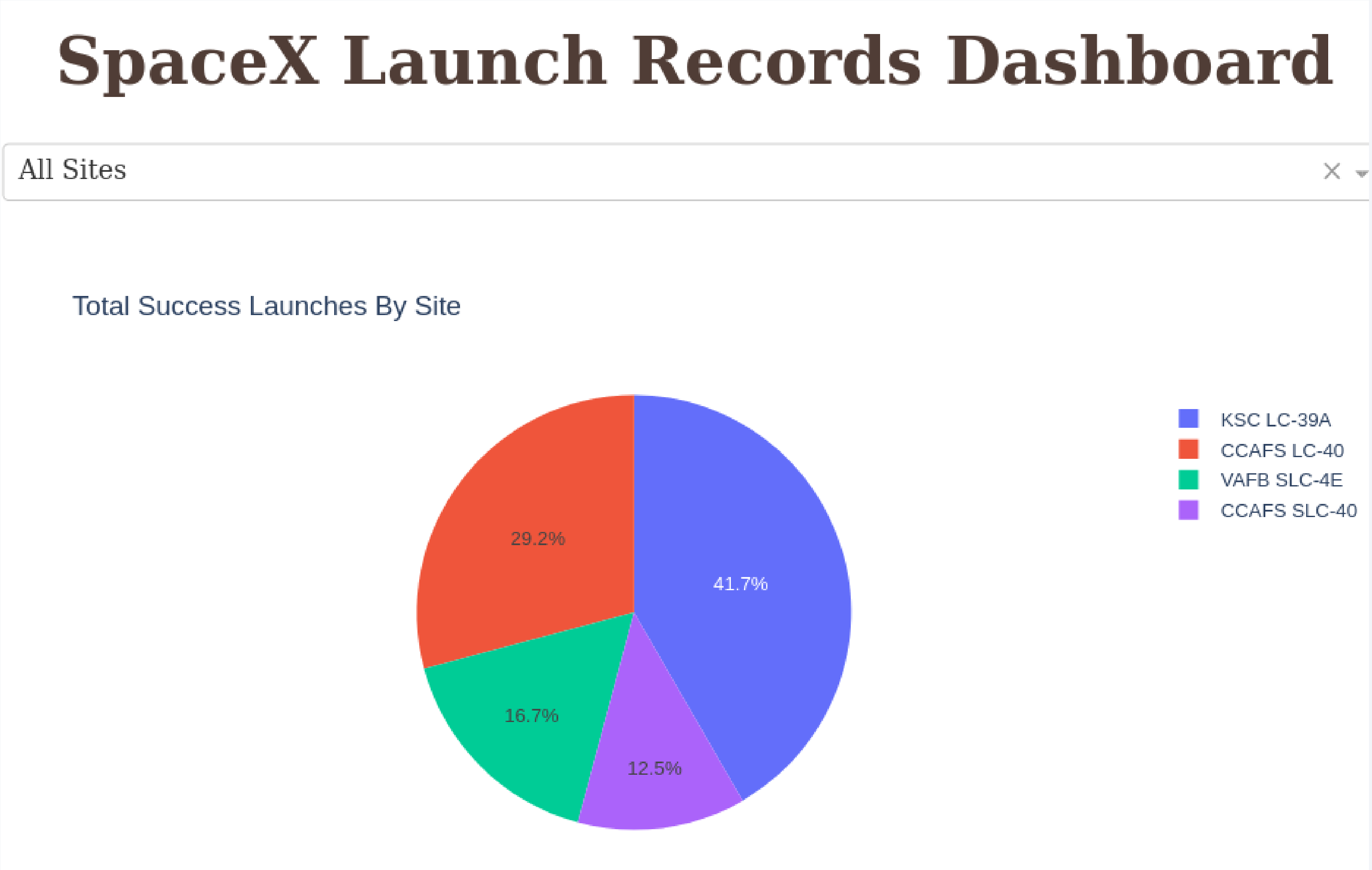
The Distance Between a Launch Site to Its Proximities

- The distances between the coastline point and the launch site is 86.7km

Section 4

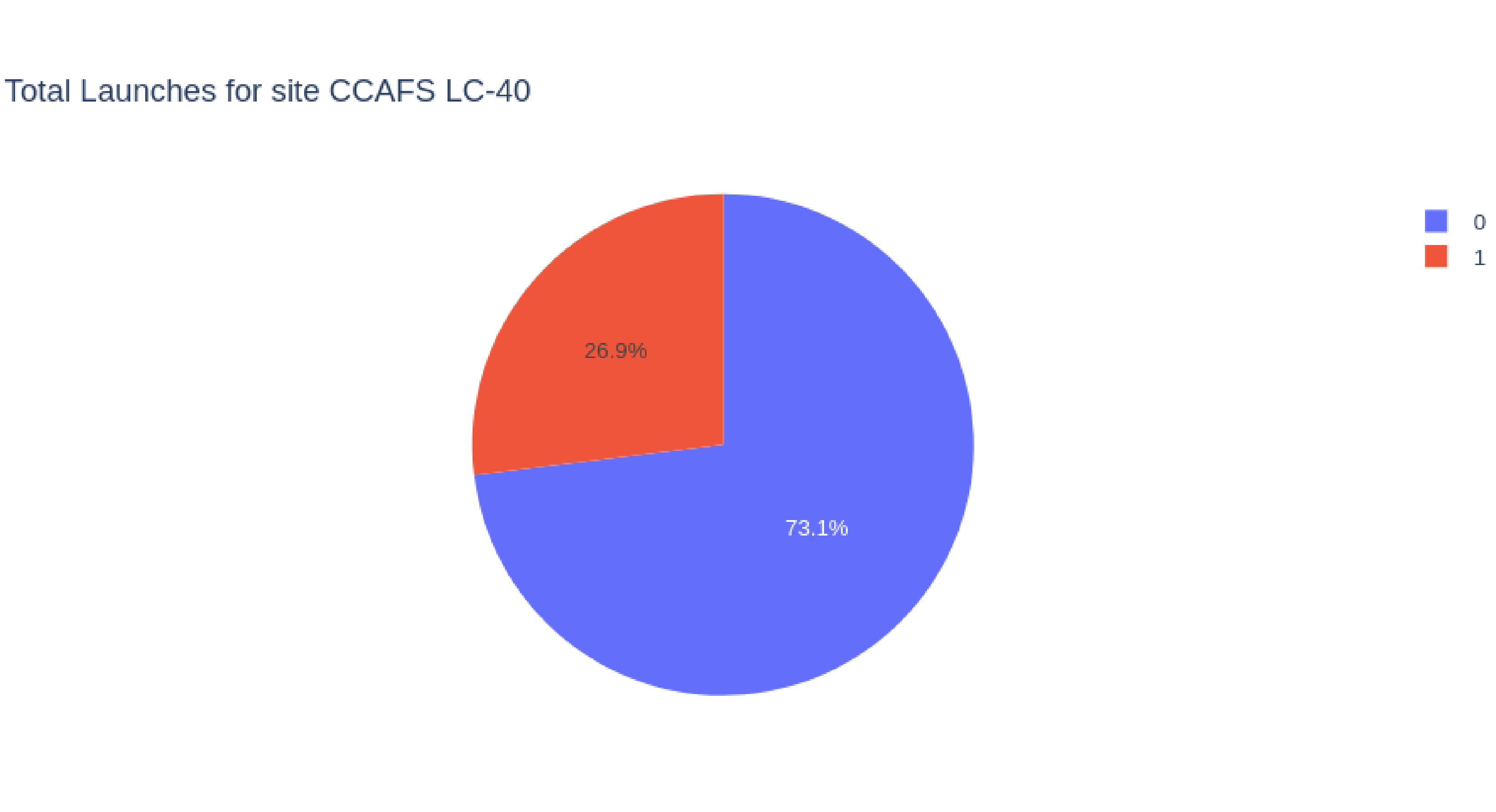
Build a Dashboard with Plotly Dash

Successful Launchers by Site



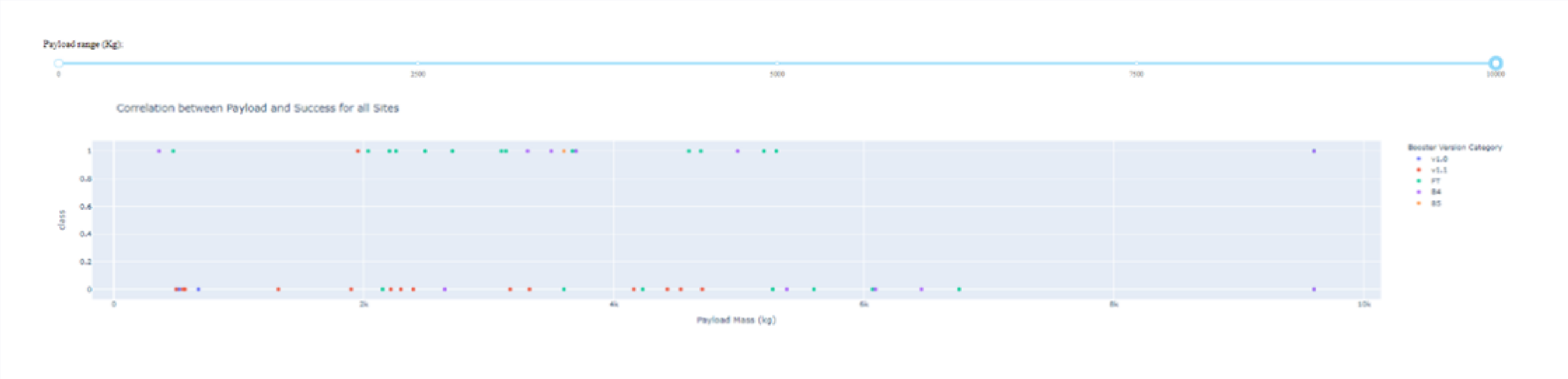
- Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Launch Success Ratio for CCAFS LC-40



- Launch site CCAFS LC-40 had the second highest: success ratio of 73% success against 27% failed launches

Payload vs. Launch Outcome

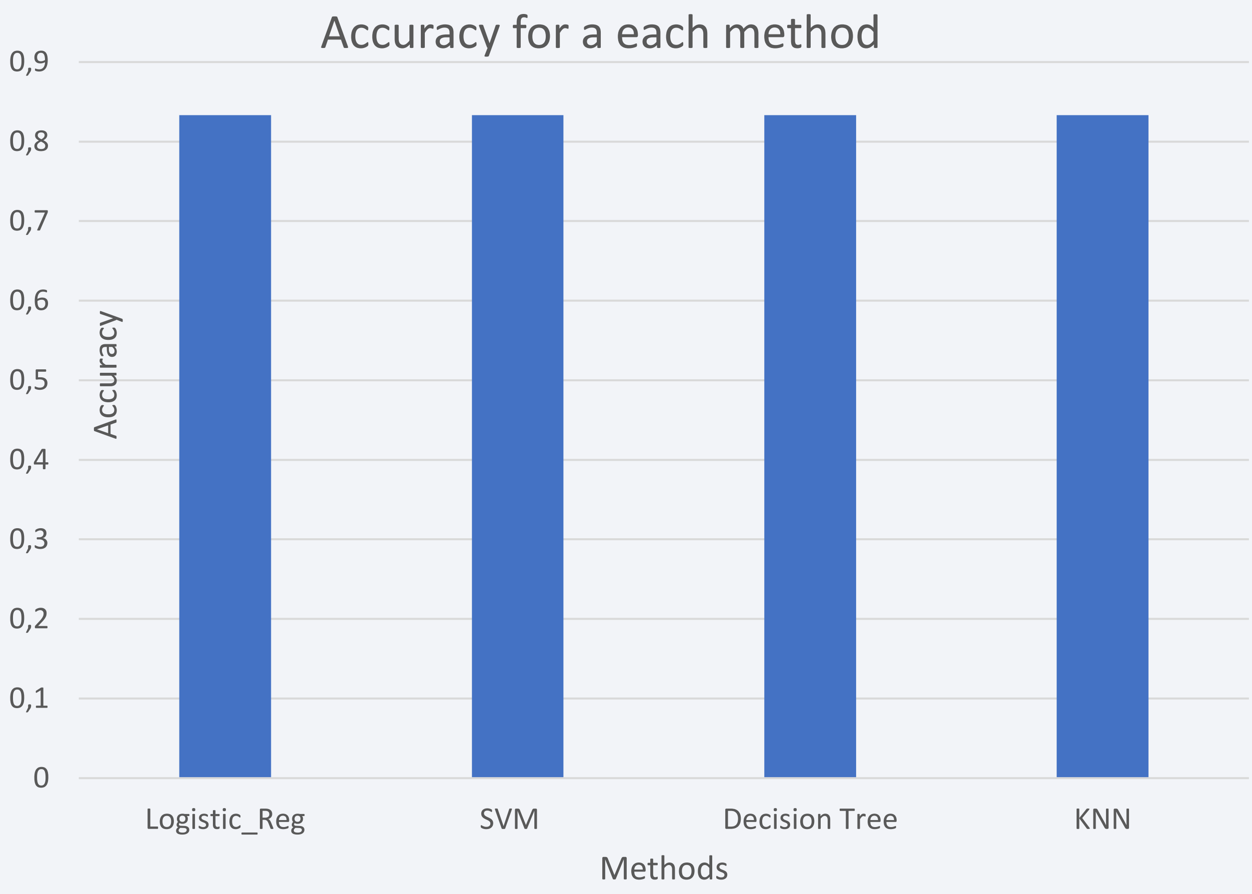


- For Launch site CCAFS LC-40 the booster version FT has the largest success rate from a payload mass of >2000kg

Section 5

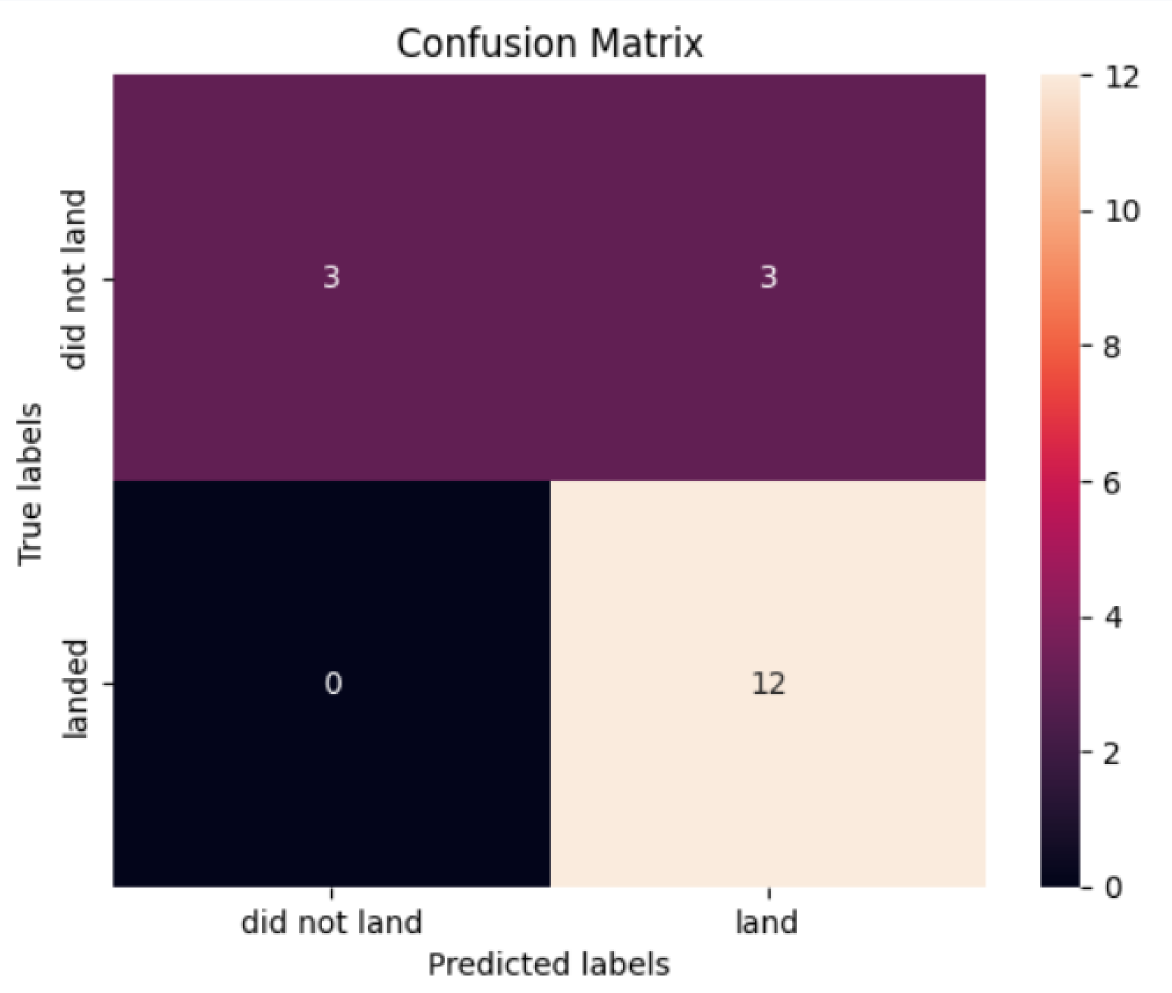
Predictive Analysis (Classification)

Classification Accuracy



- All the methods perform equally on the test data. They all have the same accuracy of 0.83333 on the test Data

Confusion Matrix



- All the four classification model had the same confusion matrixes and were able equally distinguish between the different classes. The main problem is false positives for all the models

Conclusions

- Different launch sites have different success rates. KSC LC-39A and VAFB SLC 4E has a success rate of 77%, while CCAFS LC-40, has a success rate of 60 %.
- With the flight number increases in each of the third launch sites, so does the success rate. For example, KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at -50%. Orbit SO has 0% success rate.
- 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.
- The success rate since 2013 kept increasing till 2020.

Thank you!