IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raül García Centelles
2 February 2025

Outline

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

Executive Summary

- Summary of methodologies
  - - SpaceX Data Collection via SpaceX API
  - - SpaceX Data Collection with Web Scraping
  - - SpaceX Data Wrangling
  - -SpaceX Exploratory Data Analysis via SQL
  - -SpaceX EDA Data Visualization with Pandas and Matplotlib
  - -SpaceX Launch Sites Analysis via Folium and Ploty Dash
  - -SpaceX Machine Learning Landind Prediction
- Summary of all results
  - -EDA results
  - -Interactive Visual Analytics and DashBoards
  - -Predictive Analysis (Clasification)

Introduction

- Project background and context

  SpaceX advirtises Falcon Rocket 9 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 lunch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of the project is to crate a machine learning pipeline to predict if the first stage will land succesfully.

- Problems you want to find answers
  - -Which factors determine if the rocket will land successfully?
  - -The integration of different features that determine the successful rate of a landing.
  - -Operating conditions needed to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Web Scrapping from Wikipedia.
- Perform data wrangling
  - One-hot encoding applied to categorical fetures.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics via Folium andPlotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating classification models.

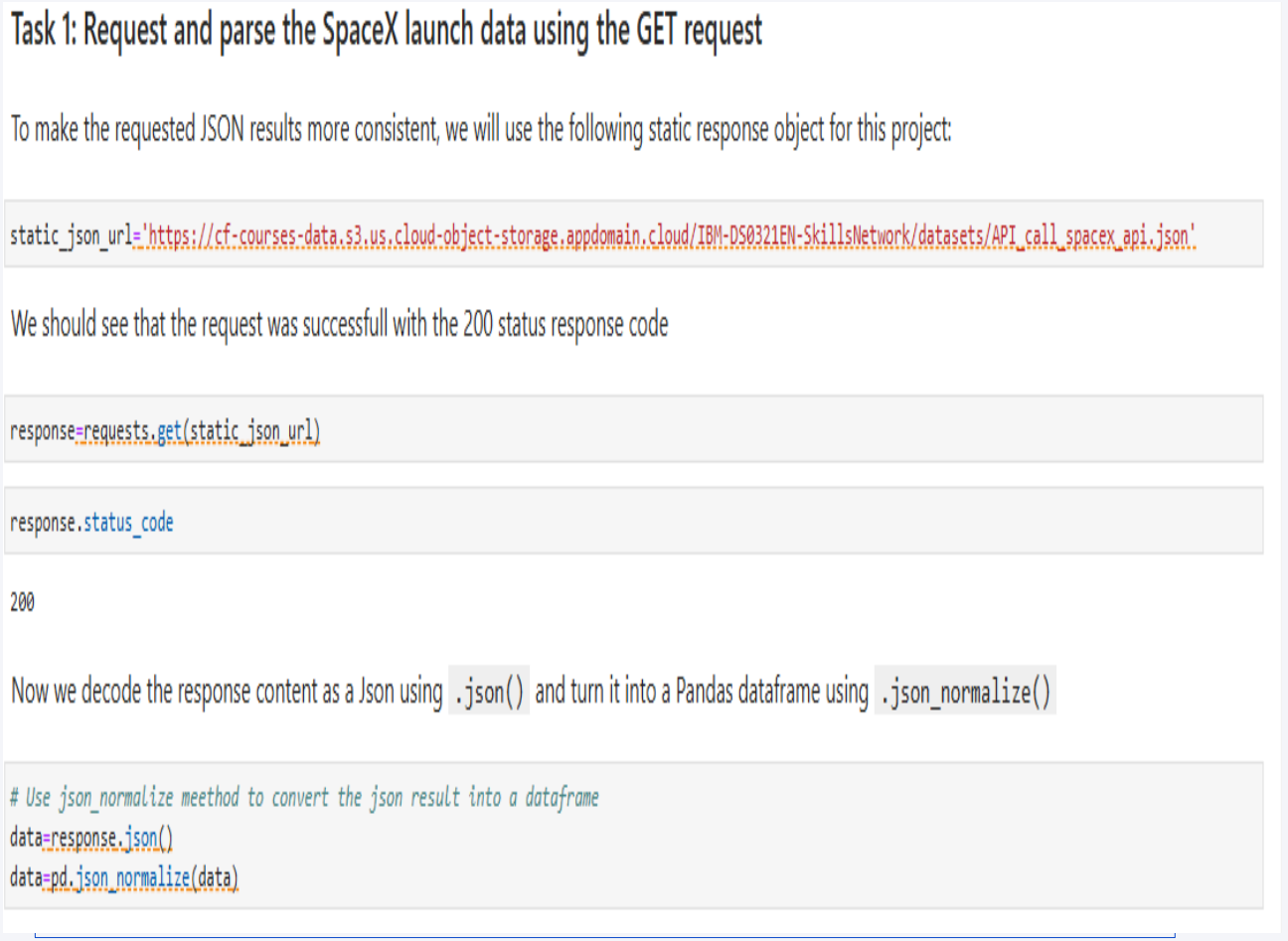
Data Collection

Description of SpaceX Falcon Data Collection:

- Data was first collected from SpaceX API by making a request to the SPACEX API. This was done by defining a series helper functions thtat 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 Space X API url.
- Secondly, the SpaceX launch data was requested and parsed using the GET request and the decoded the response content as a json result wich was then converted Into a Pandas data frame.
- Finally, Falcon 9 historical launch records data was collected via web scraping from the Wikipedia page: “List of Falcon 9 and Falcon Heavy Launches”. Using beautiful Soup and request Libraries, Falcon 9 launch HTML table records is extracted from the Wikipedia page. The table was converted into a Pandas data frame.

Data Collection – SpaceX API

- Data collected using SpaceX API by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a Json result which was then converted into a Pandas data frame.
- The GitHub URL of the completed SpaceX API calls notebook is: https://github.com/RaulGarciaCent/Module-9/blob/main/M1_1%20SpaceX%20Data-Collection%20Api.ipynb



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 this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

```
response=requests.get(static_json_url)
```

```
response.status_code
```

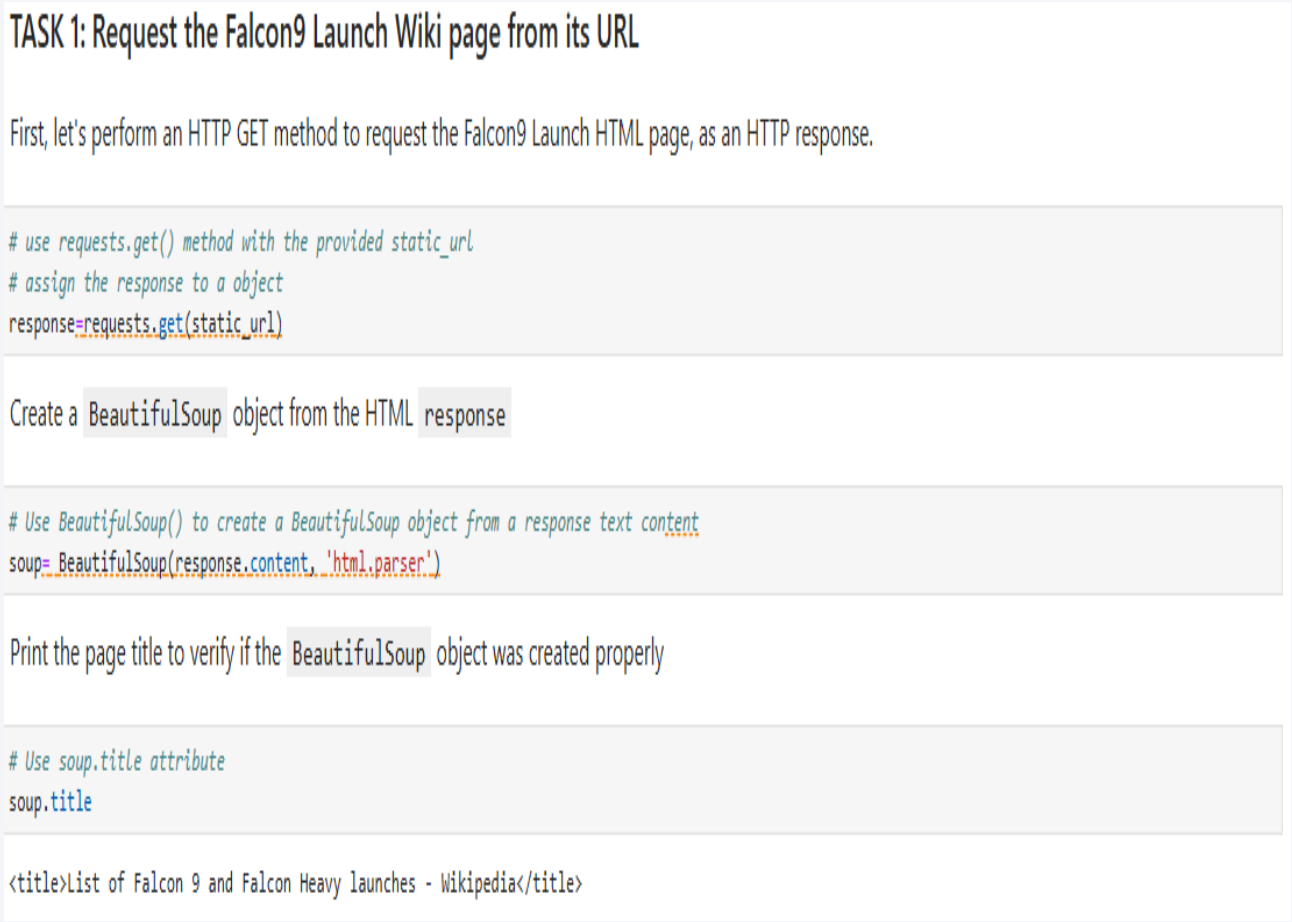
```
200
```

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

```
# Use json_normalize meethod to convert the json result into a dataframe
data=response.json()
data=pd.json_normalize(data)
```

Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using BeautifulSoup and request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.
- The the GitHub URL of the completed web scraping notebook, is: https://github.com/RaulGarciaCent/Module-9/blob/main/M1_2%20SpaceX%20Webscraping.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.

```
# 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`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup= BeautifulSoup(response.content, 'html.parser')
```

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

```
# Use soup.title attribute
soup.title
```

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

Data Wrangling

- First, after obtaining and creating a Pandas DataFrame from the collected data, data was filtered using the '***BoosterVersion***' column to only keep the Falcon 9 launches, then dealt with the missing data values in the '***LandingPad***' and '***PayloadMass***' columns. For the '***PayloadMass***', missing data values were replaced using mean value of column.

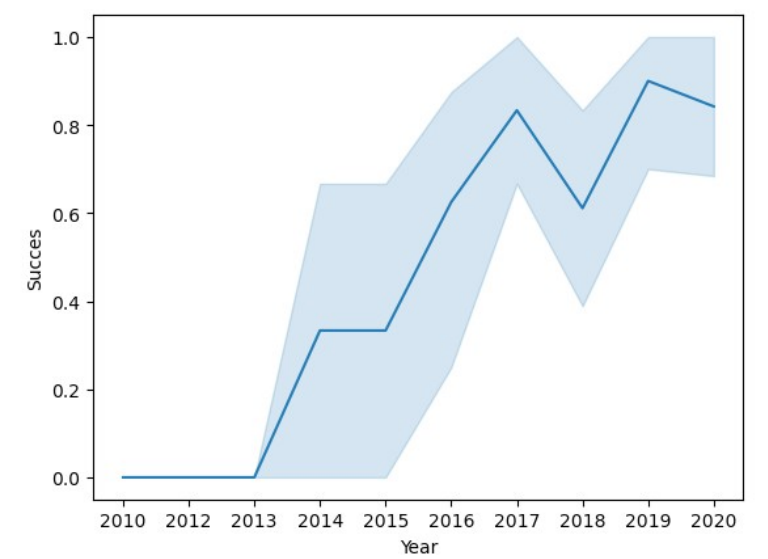
- Also performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- The GitHub URL of the completed data wrangling related notebooks is: https://github.com/RaulGarciaCent/Module-9/blob/main/M1_3%20SpaceX%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Data was explored 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 the completed EDA with data visualization notebook is: https://github.com/RaulGarciaCent/Module-9/blob/main/M2_2%20SpaceX%20EDA%20with%20Pandas%20and%20Matplotlib.ipynb



EDA with SQL

- SpaceX dataset was loaded into a PostgreSQL database.
- EDA with SQL was applied to get insight from the data. The following queries were wrote to find out for instance:
  - -Names of unique launch sites in the space mission.
  - -Total payload mass carried by boosters launched by NASA(CRS)
  - -Average payload mass carried by booster version F9 v1.1
  - -Total number of successful and failure mission outcomes
  - -Failed landing outcomes in drone ship, their booster version and launch site names.
- The GitHub URL of th completed EDA with SQL notebook is: https://github.com/RaulGarciaCent/Module-9/blob/main/M2_1%20SpaceX%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Launch sites were marked, and map objects such as markers, circles, lines were added to mark the success or failure of launches for each site on the folium map.

- The feature launch outcomes (failure or success) were assigned to class 0 and 1.i.e., 0 for failure, and 1 for success.

- Using the color-labeled marker clusters, launch sites which have relatively high success rate were identified.

- The distances between a launch site to its proximities were calculated.

-The GitHub URL of the completed interactive map with Folium map is:
https://github.com/RaulGarciaCent/Module-9/blob/main/M3_1%20SpaceX%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

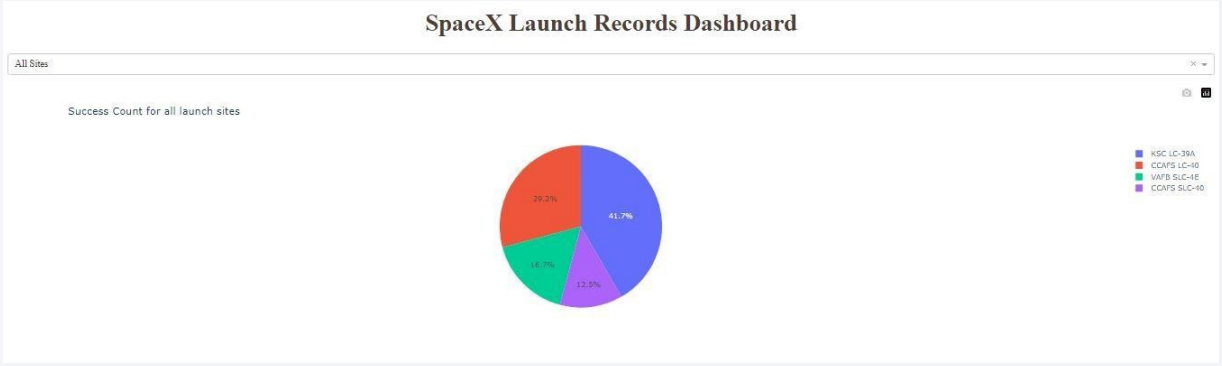
Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly dash is builded.

- Pie charts were plotted, showing the total launches by a certain sites.

- Scatter graph is plotted, showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

The GitHub URL of the completed Plotly Dash lab is:
https://github.com/RaulGarciaCent/Module-9/blob/main/M3_2%20Space%20X%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py



Predictive Analysis (Classification)

- Data was loaded using numpy and pandas, transformed the data, split our data into training and testing.

- Different machine learning models were builded and tune different hyperparameters using GridSearchCV.

- Accuracy was used as the metric for our model, improved the model using feature engineering and algorithm tuning.

- Finally, the best performing classification model was founded.

The GitHub URL of the completed predictive analysis lab is:
https://github.com/RaulGarciaCent/Module-9/blob/main/M4%20SpaceX%20Machine%20Learning%20Prediction.ipynb

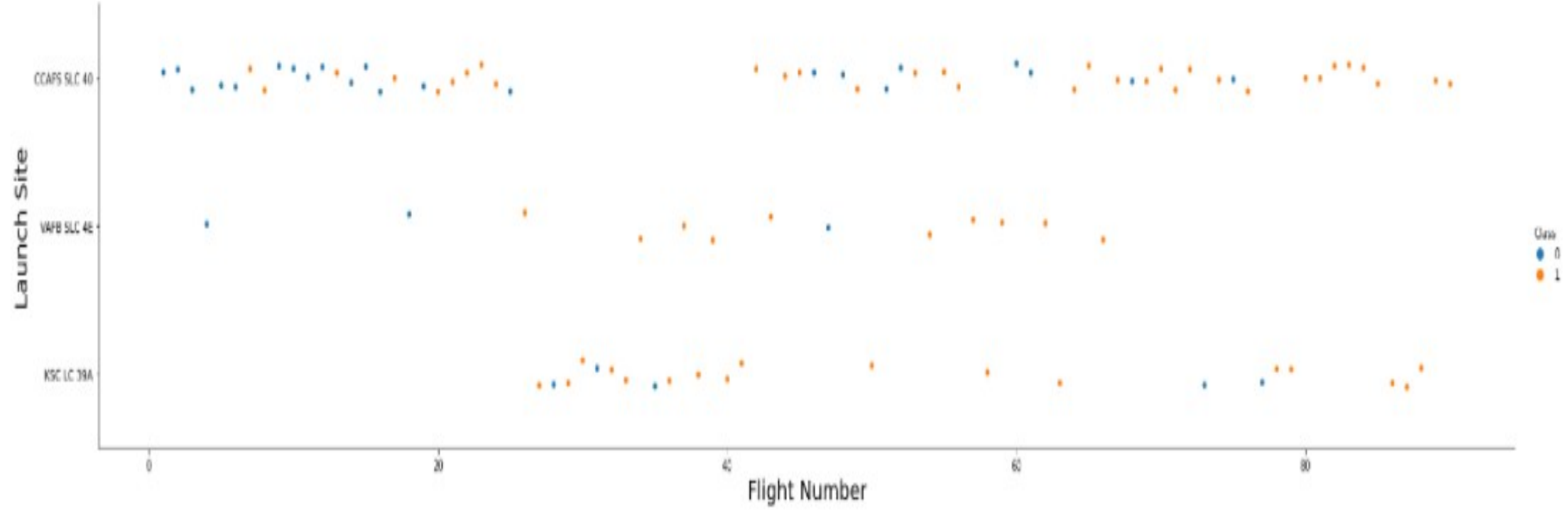
Results

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

Section 2

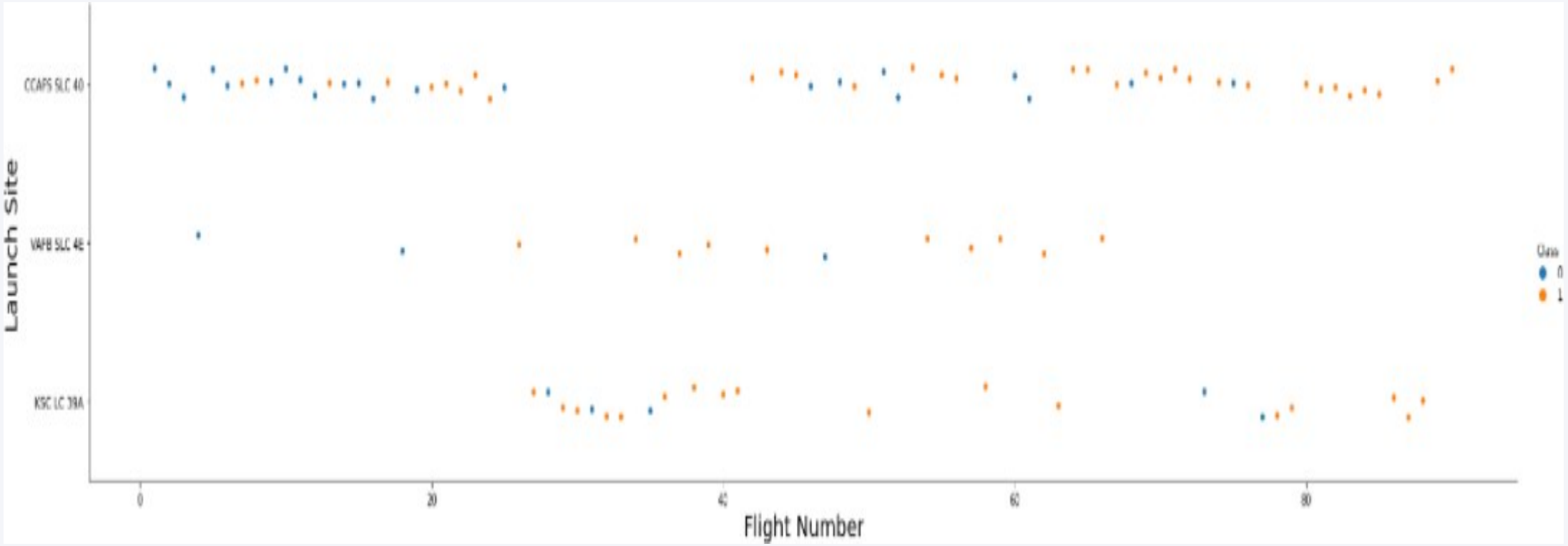
Insights drawn from EDA

Flight Number vs. Launch Site



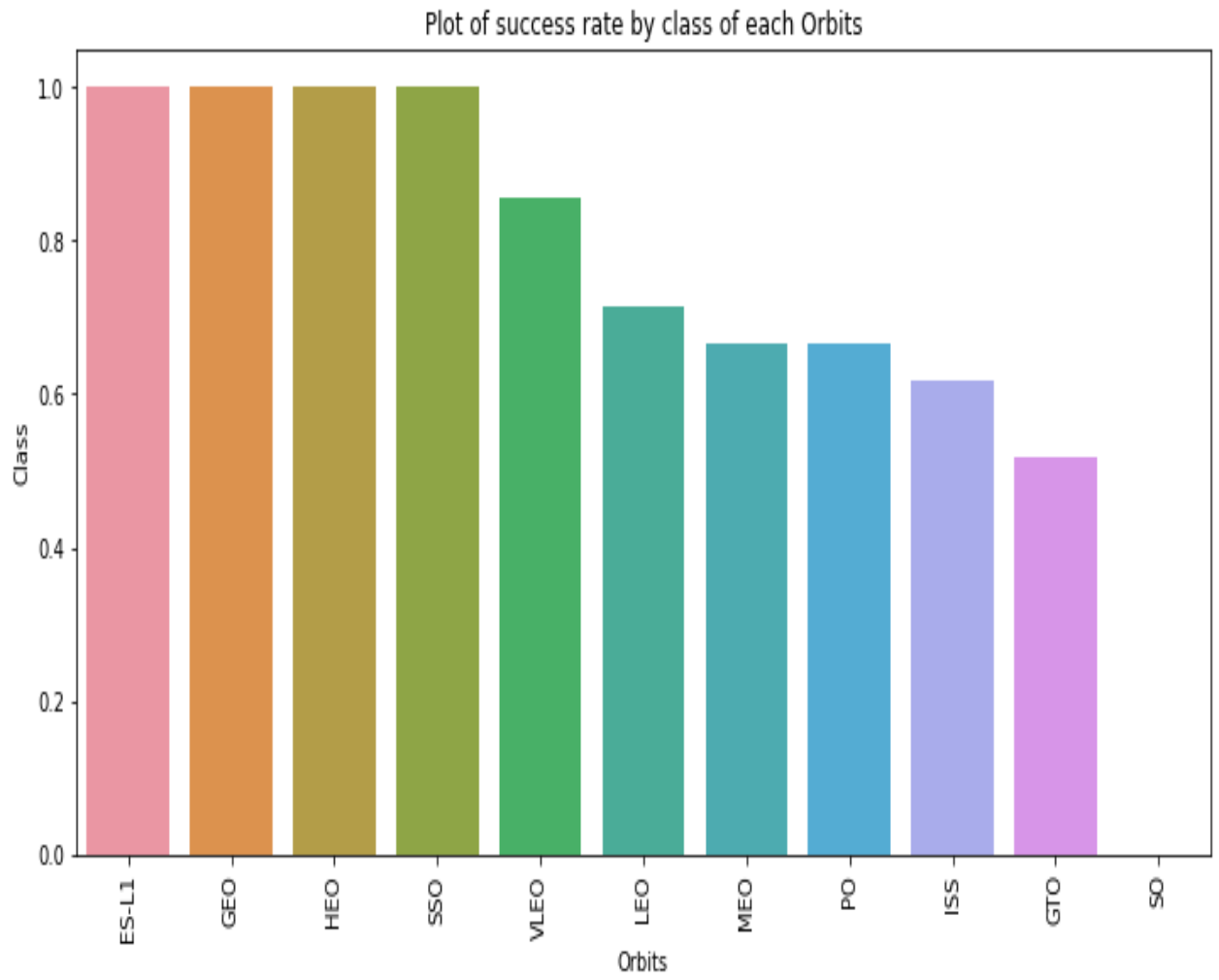
- As we can see from this plot, the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site



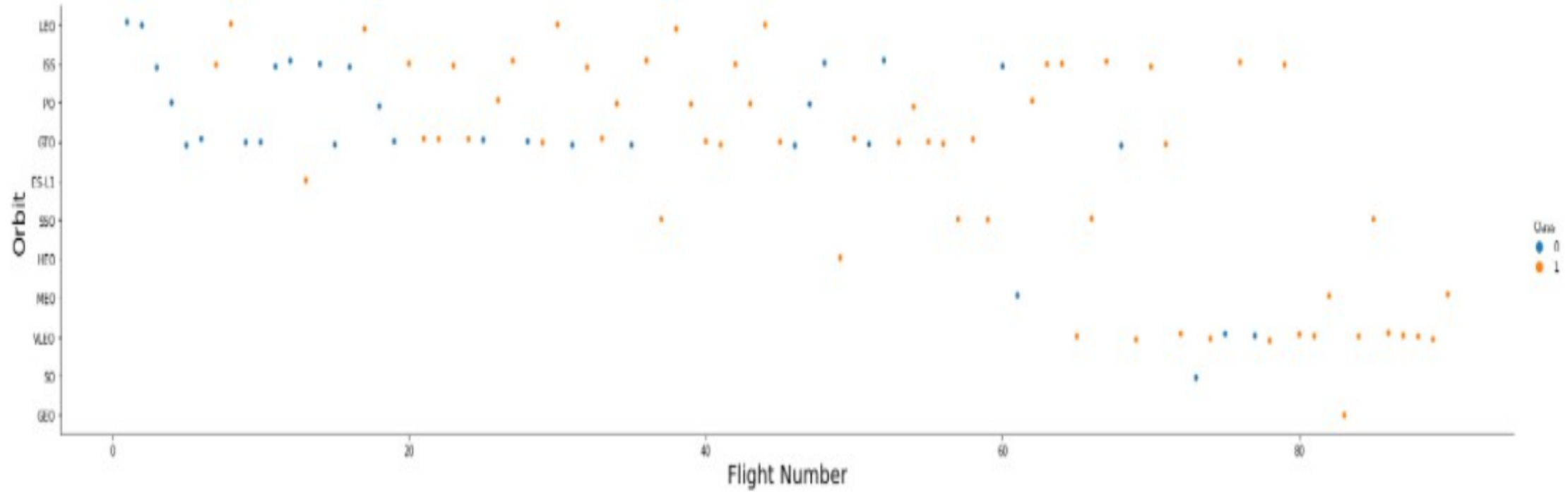
- From this chart, we find the information that the grater the payload mass for launch site from the CCAFS SLC 40 the higher the succes rate for the rocket.

Success Rate vs. Orbit Type



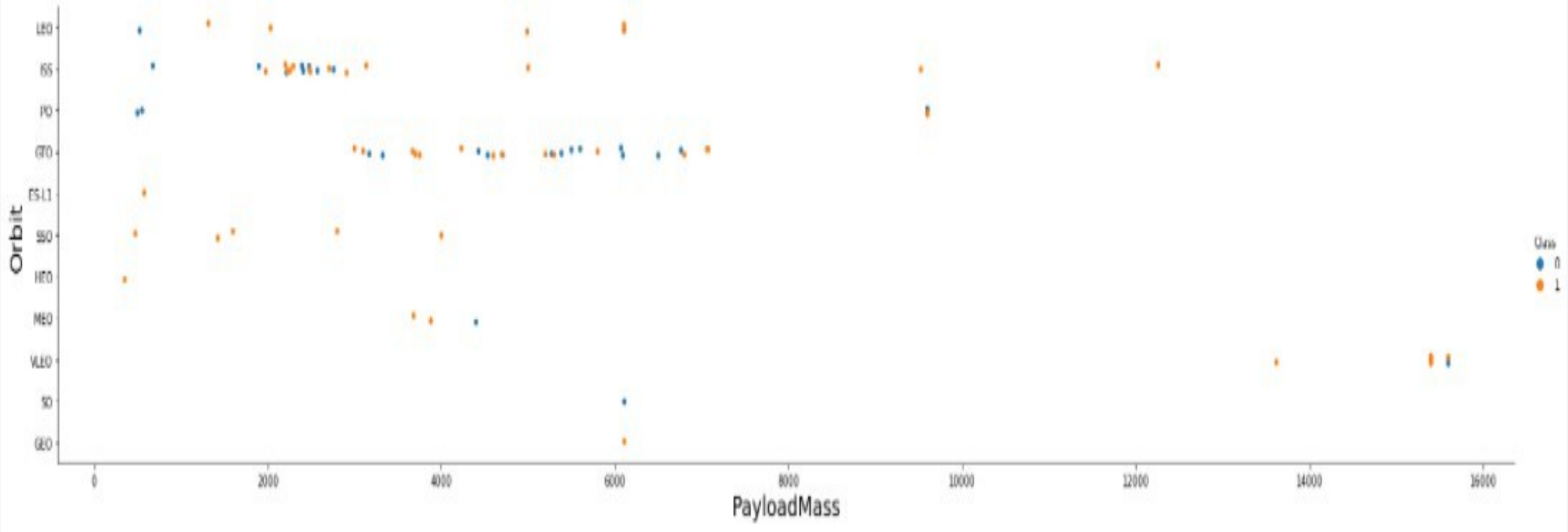
- The following barchart shows that ES-L1, GEO, HEO, SSO,VLEO had the most success rate.

Flight Number vs. Orbit Type



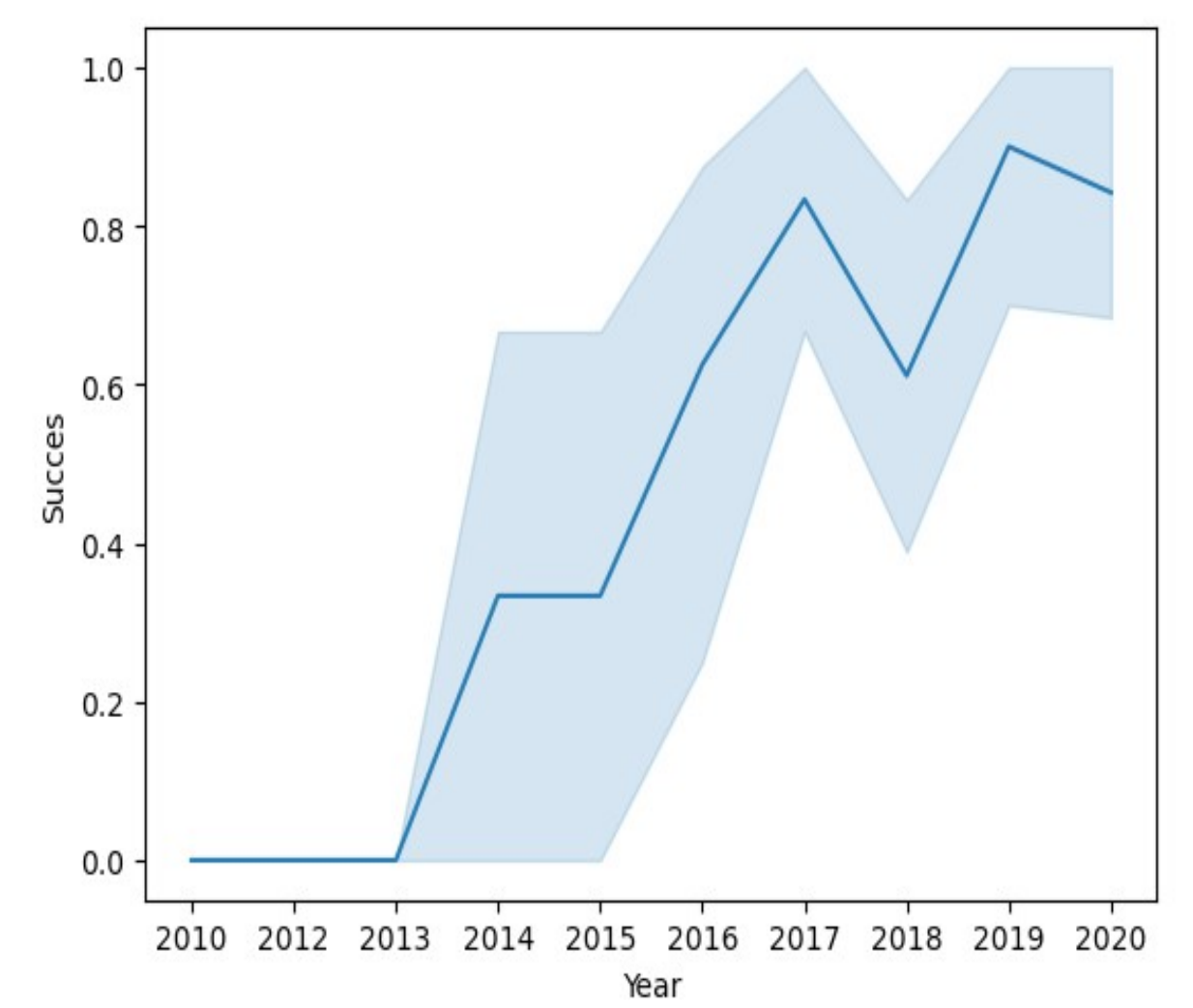
- Now, the plot shows the Flight Number vs. Orbit type. Here, is observed that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type



- From this From this plot is observed that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

Launch Success Yearly Trend



- From the following line chart, is observed that success rate since 2013 kept on increasing till 2020.

All Launch Site Names

- Using the ‘SELECT DISTINCT’ statement to return only the unique launch sites, all Launch Site Names are obteined from the ‘LAUNCH_SITE’ column from the SPACEX table.

Task 1

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

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

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	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Using ‘LIKE’ command with ‘%’ wildcard in ‘WHERE’ clause to select and dispay a table of 5 records where **l**aunch sites begin with the string **'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 Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

- The total payload carried by boosters from NASA is calculated using the ‘SUM()’ function to return and dispaly the total sum of ‘PAYLOAD_MASS_KG’ column 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 payloadmass from SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

payloadmass
2928.4

- The average payload mass carried by booster version F9 v1.1 is calculated using the ‘AVG()’ function to return and dispaly the average payload mass carried by booster version F9 v1.1.

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) from SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

- The date of the first successful landing outcome on ground pad is obtained Using the ‘MIN()’ function to return and dispaly the first date when first successful landing outcome on ground pad ‘Success (ground pad)’ happened.

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 Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" \
    AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

- The list of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 were obtained using the ‘SELECT DISTINCT’ statement to return the names of boosters with operators >4000 and <6000 to only list booster with payloads between 4000-6000 with landing outcome of ‘Success (drone ship)’.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

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

- The total number of successful and failure mission outcomes is calculated using the ‘COUNT()’and the ‘GROUP BY’ statement together to return total number of missions 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, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

- The list of the names of the booster which have carried the maximum payload mass is obtained using a Subquerry to return and pass the Max payload and used it list all the boosters that have carried the Max payload.

2015 Launch Records

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date,6,2) as month, DATE, BOOSTER_VERSION, LAUNCH_SITE, Landing_Outcome from SPACEXTBL\
WHERE Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- The list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 is obtained using the ‘subsrt()’ in the select statement to get the month and year from the date column where substr(Date,7,4)='2015' for year and Landing_outcome was ‘Failure (drone ship)’ and return the records nmatching the filter.

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(*) as count_outcomes from SPACEXTBL where DATE between '2010-06-04' and '2017-20-03' GROUP BY\
    LANDING_OUTCOME ORDER BY count_outcomes DESC;
```

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

Landing_Outcome	count_outcomes
Success (drone ship)	12
No attempt	12
Success (ground pad)	8
Failure (drone ship)	5
Controlled (ocean)	4
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Finally, 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 is ranked aplying the ‘COUNT(*)’ statement and the ‘ORDER BY’ clause.

Section 3

Launch Sites Proximities Analysis

Global map launch sites markers

- The launch sites are in proximity to the Equator, and are in very close proximity to the coast too as we can see in the map.

Launch outcomes for each site on the map With Color Markers

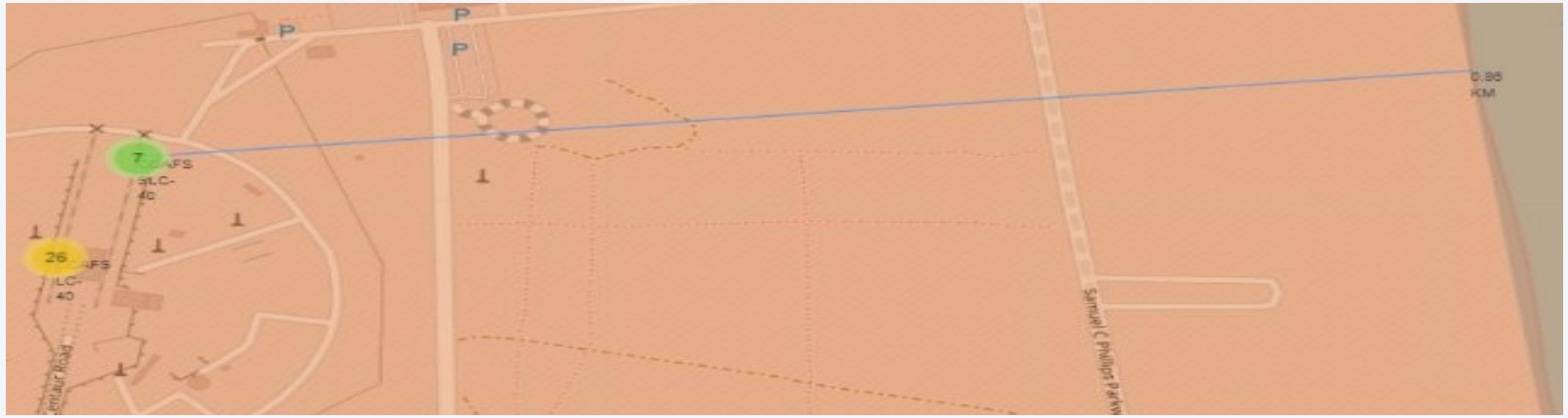
- Florida/East Coast

Launch site KSC LC-39A has relatively high success rates in comparison to CCAFS SLC-40 & CCAFS LC-40.

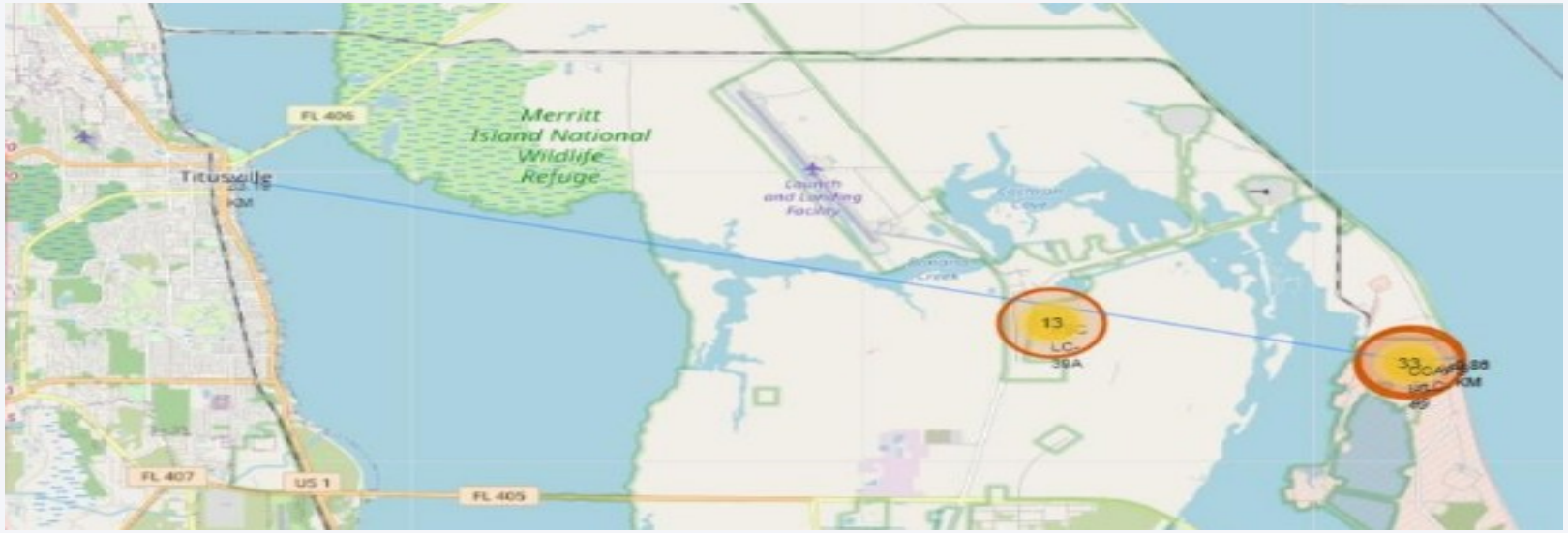
- California/Weast Coast

Here, Launch site VAFB SLC-4E has a relatively lower success rates (4/10) compared to KSC LC- 39A launch site in the Eastern Coast of Florida.

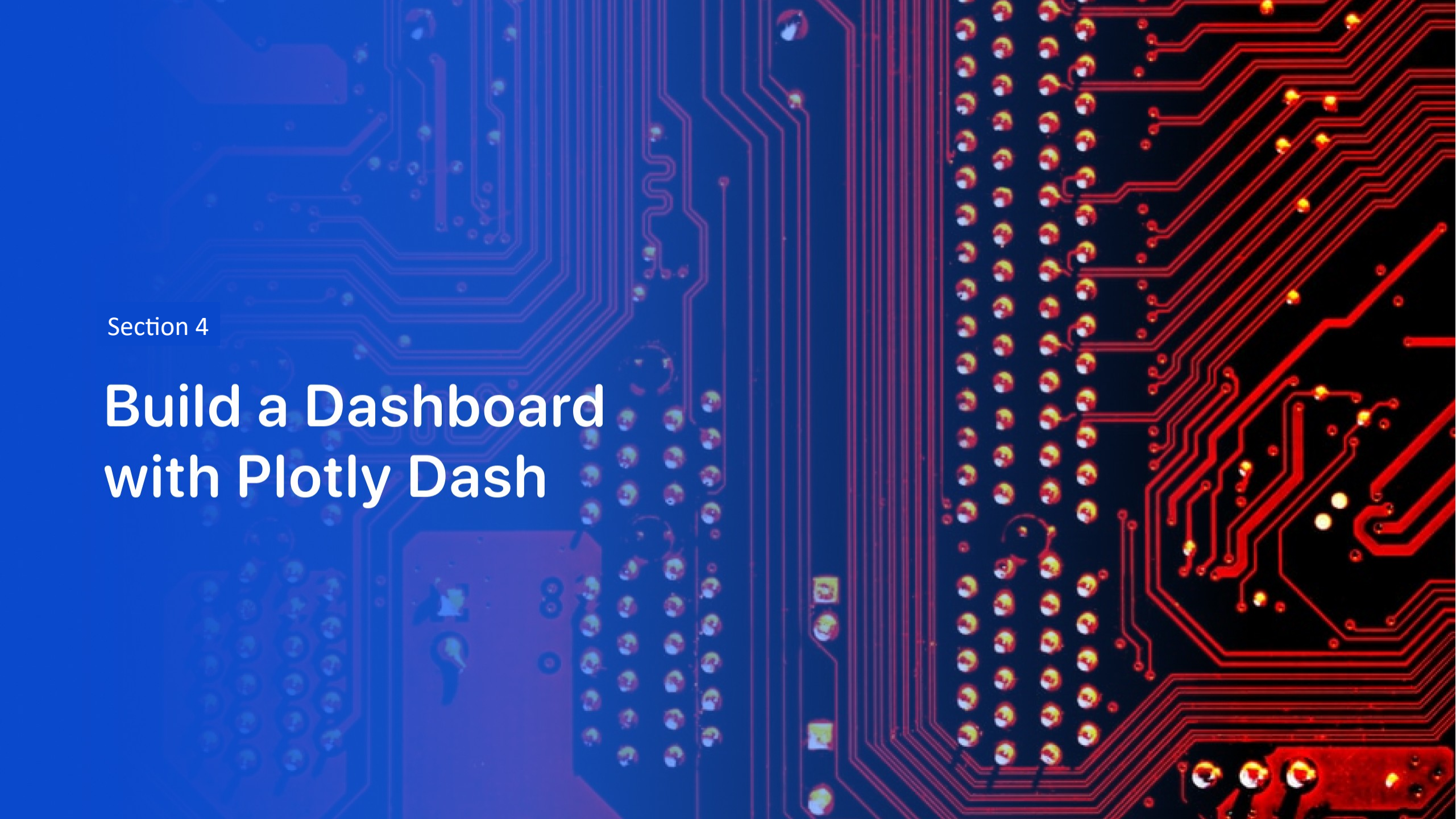
Distances between a launch site to its proximities



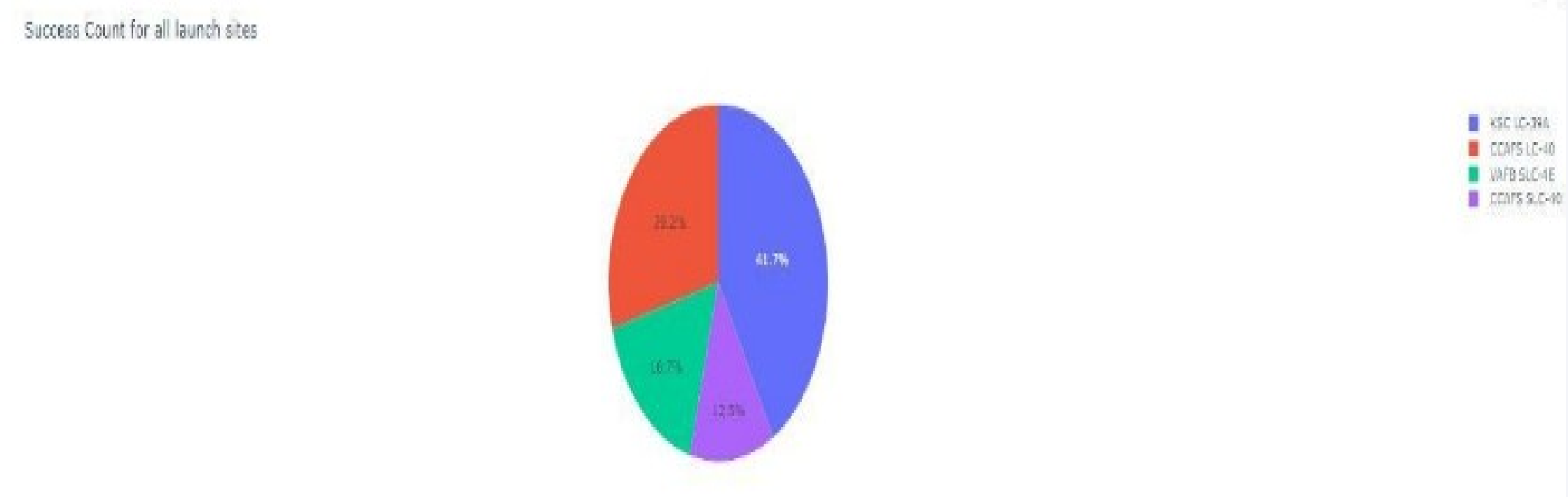
The proximity of the Launch site CCAFS SLC-40 to the coastline is 0.86km



The closest Launch site CCAFS SLC-40 to highway (Washington Avenue) is 23.19km.

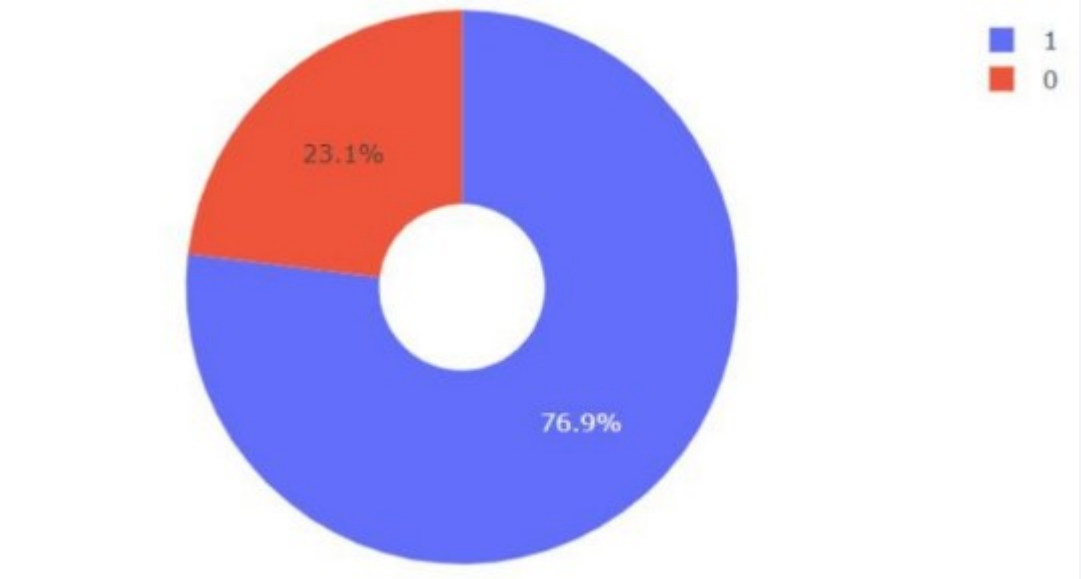

Section 4
Build a Dashboard with Plotly Dash

Pie-Chart for launch success count for all sites



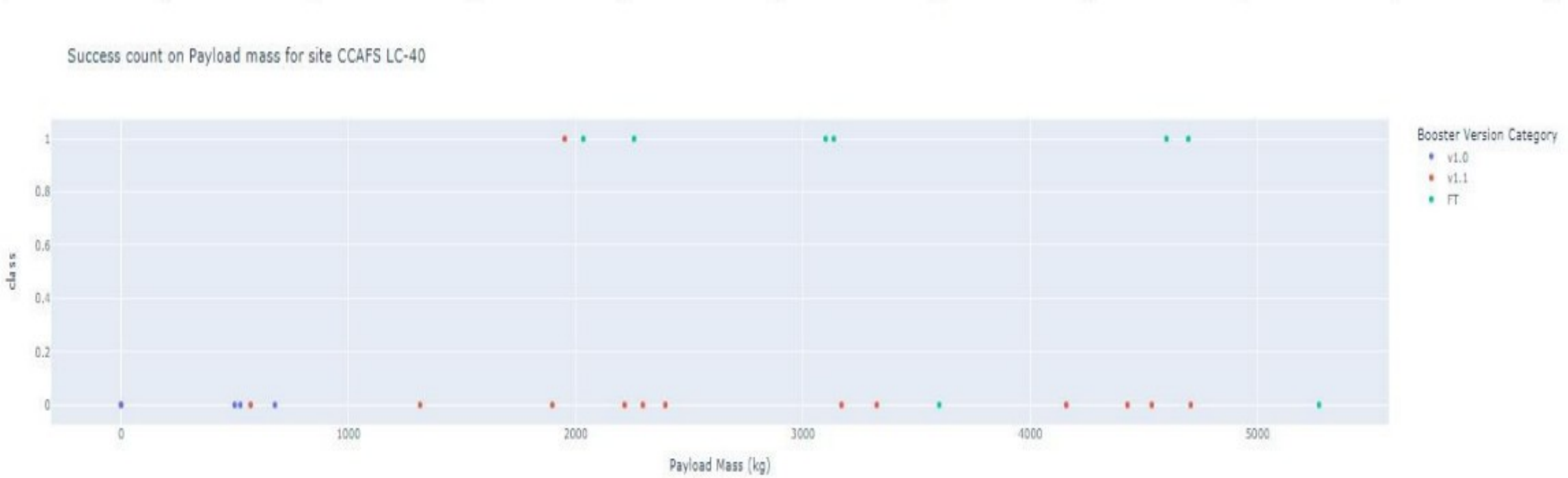
The Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and launch site CCAFS SLC-40 has the worst launch succes with a success rate of 13%.

Pie chart for the launch site with the highest launch success ratio



The KSC LC-39A obtained a 76,9% succes rate and a 23,1% failure rate.

Payload vs. Launch Outcome scatter plot for all sites



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

TASK 12

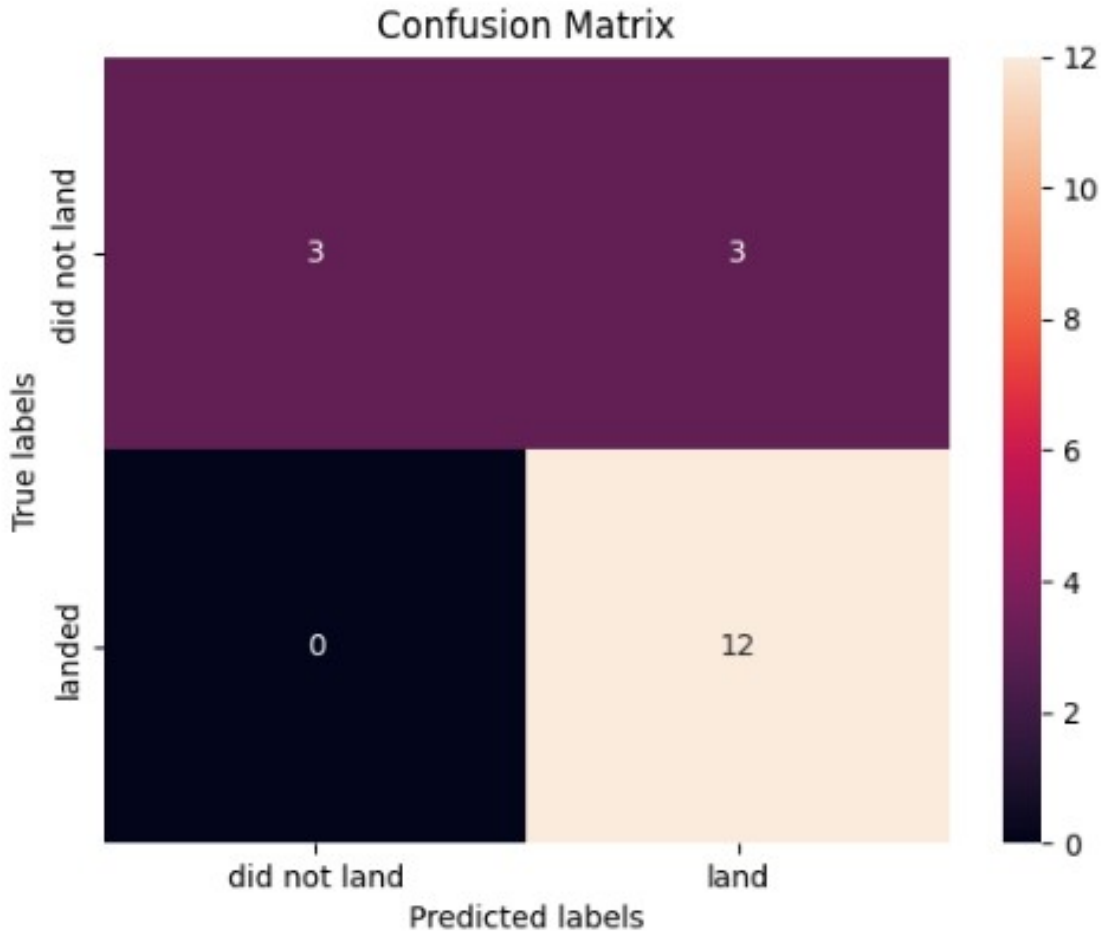
Find the method performs best:

```
Report = pd.DataFrame({'Method' : ['Test Data Accuracy']})

knn_accuracy=knn_cv.score(X_test, Y_test)
Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
SVM_accuracy=svm_cv.score(X_test, Y_test)
Logistic_Regression=logreg_cv.score(X_test, Y_test)

Report['Logistic_Reg'] = [Logistic_Regression]
Report['SVM'] = [SVM_accuracy]
Report['Decision Tree'] = [Decision_tree_accuracy]
Report['KNN'] = [knn_accuracy]

Report.transpose()
```

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

- The four methods developed have the same ‘Test Data Accuracy’ 0.8333333, therefore all methods perform equally on the test data.

Confusion Matrix



-The Confusion Matrix is able to distinguish between the different classes.

-The major problem is false positives for all the models (such as unsuccessful landing marked as successful landing by the classifier).

Conclusions

- Different launch sites have different success rates. CCAFS LC-40, has a 60% success rate, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

- Observing the ‘Payload Vs. Launch Site’ scatter point chart, there are no rockets launched for heavypayload mass(>10000Kg) for the VAFB-SLC launchsite

- 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.

- LEO orbit the Success appears related to the number of flights; on the other hand, it seems to be no relationship between flight number when in GTO orbit.

- The successful landing rate for Polar, LEO and ISS is higher with heavy payloads. On the other hand, is not posible to distinguish this well for GTO beacause both positive and negative landings are both present.

- Finally, the success landing rate has increased since 2013 until 2020.

Thank you!