Winning Space Race with Data Science

Click to add text

Radhika Radhakrishnan Nair

11/15/2021

Outline

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

Executive Summary

In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.

In this capstone, the role of a data scientist working for a new rocket company Space Y that would like to compete with SpaceX is to determine the price of each launch. This can be done by gathering information about Space X and creating dashboards using the information. The data scientist will also determine if SpaceX will reuse the first stage.

The data scientist will be collecting data from various sources. After your raw data has been collected, data quality is improved by performing data wrangling.

Then we start exploring the processed data. We use SQL to query the data and gather insights. Further insights into the data is obtained by applying some basic statistical analysis and data visualization, We will be able to see directly how variables might be related to each other.

Finally we drill down into finer levels of detail by splitting the data into groups defined by categorical variables or factors in your data.

Introduction

In this capstone, the role of a data scientist working for a new rocket company Space Y that would like to compete with SpaceX is to determine the price of each launch. This can be done by gathering information about Space X and creating dashboards using the information. The data scientist will also determine if SpaceX will reuse the first stage.

The data scientist will be collecting data from various sources. After your raw data has been collected, data quality is improved by performing data wrangling.

Then we start exploring the processed data. We use SQL to query the data and gather insights. Further insights into the data is obtained by applying some basic statistical analysis and data visualization, We will be able to see directly how variables might be related to each other. Finally we drill down into finer levels of detail by splitting the data into groups defined by categorical variables or factors in your data.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
- The data required for analysis is collected by using the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages.
- **Perform data wrangling**
- We use the collected data and perform data wrangling operations.For example we use the column Outcome(indicates if the first stage successfully landed) and convert it to converted to Classes y(either 0 or 1). 0 is a bad outcome, that is, the booster did not land. 1 is a good outcome, that is, the booster did land. The variable Y will represent the classification variable that represents the outcome of each launch.
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**

- **Perform predictive analysis using classification models**
- we will build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully. This will include: Preprocessing, allowing us to standardize our data, and Train_test_split, allowing us to split our data into training and testing data, We will train the model and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best. Using the best hyperparameter values, we will determine the model with the best accuracy using the training data.You will test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Finally, we will output the confusion matrix

Data Collection

- The data required for analysis is collected by using the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- The end point used api.spacexdata.com/v4/launches/past.We will use this URL to target a specific endpoint of the API to get past launch data. We will perform a get request using the requests library to obtain the launch data, which we will use to get the data from the API.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages. You will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Then you need to parse the data from those tables and convert them into a Pandas data frame for further visualization and analysis. We want to transform this raw data into a clean dataset which provides meaningful data on the situation we are trying to address: Wrangling Data using an API, Sampling Data, and Dealing with Nulls.

Data Collection – SpaceX API

- End Point used :- api.spacexdata.com/v4/launches/past

GitHub URL:

- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Data%20Collection.ipynb

```
url = "api.spacexdata.com/v4/launches/past"
response = requests.geturl(url)
response.json()
data = pd.json_normalize(response.json())
```

Data Collection - Scraping

GitHub URL:

https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Data%20Collection%20with%20webscraping.ipynb

Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages. You will be using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Then you need to parse the data from those tables and convert them into a Pandas data frame for further visualization and analysis. We want to transform this raw data into a clean dataset which provides meaningful data on the situation we are trying to address: Wrangling Data using an API, Sampling Data, and Dealing with Nulls.

Data Wrangling

- The raw data collected using the data collection process has to be transformed into a clean dataset which provides meaningful data on the situation we are trying to address: Wrangling Data using an API, Sampling Data, and Dealing with Nulls.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

Click to add text

- We use data wrangling techniques to convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- **GitHub URL:-**
- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

- EDA was done by plotting various charts to find relevant relation between various features in the dataset. The different charts plotted were :-

1) **FlightNumber vs. PayloadMass** :-From the scatter plot we can see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return

2) **FlightNumber vs LaunchSite** :-From the scatter plot we see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%. VAFB-SLC launch site fewer rockets are launched (less than 80)

3) **LaunchSite vs PayloadMass**(scatter chart) :-VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

4) **Success rate vs Orbit type**(Bar Plot):-From the bar plot we see that the orbits GEO,HEO,SSO&ES-L1 have higher success rates

5) **FlightNumber vs Orbit** (Scatter Plot):-LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

6) **Payload vs Orbit** (Scatter Plot) :- 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.

7)**launch success yearly trend**:-sucess rate since 2013 kept increasing till 2020

GitHub URL:-

https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/EDA%20with%20VisualizationLab.ipynb

EDA with SQL

- The various SQL queries performed :-
- **Names of the unique launch sites in the space mission** :- select DISTINCT(launch_site) from SPACEXDATASET
- **5 records where launch sites begin with the string 'CCA'** :- select * from SPACEXDATASET where launch_site like '%CCA%' limit 5
- **total payload mass carried by boosters launched by NASA (CRS)** :- select sum(payload_mass__kg_) from SPACEXDATASET where customer = 'NASA (CRS)'
- **average payload mass carried by booster version F9 v1.1 :-** select avg(payload_mass__kg_) from SPACEXDATASET where booster_version ='F9 v1.1'
- **date when the first successful landing outcome in ground pad was achieved** :- select min(date) from SPACEXDATASET where LANDING__OUTCOME = 'Success (ground pad)'

- **names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 :-** select BOOSTER_VERSION from SPACEXDATASET where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_<6000
- **total number of successful and failure mission outcomes**:-select count(MISSION_OUTCOME) as SUCCESSFUL_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Success%' and select count(MISSION_OUTCOME) as FAILED_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Failure%'
- **names of the booster_versions which have carried the maximum payload mass** :- select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ in (SELECT max(PAYLOAD_MASS__KG_) from SPACEXDATASET)
- **failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 :**-select LANDING__OUTCOME ,BOOSTER_VERSION ,LAUNCH_SITE ,DATE from SPACEXDATASET where LANDING__OUTCOME ='Failure (drone ship)' and year(DATE) = 2015
- **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** :- select LANDING__OUTCOME ,COUNT(LANDING__OUTCOME) as frequency from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY FREQUENCY DESC

- **GitHub URL:-**
- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Exploratory%20data%20Analysis.ipynb

Build an Interactive Map with Folium

- Create a folium **`Map`** object and use `folium` **`Circle`** and **`Marker`** to add a highlighted circle area with a text label on a specific coordinate. For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label using the marker object. Marker clusters can be a good way to simplify a map containing many markers having the same coordinate.
- To explore and analyze the proximities of launch sites :- add a **`MousePosition`** on the map to get coordinate for a mouse over a point on the map. explore the proximity to find any railway, highway, coastline, etc. Move your mouse to these points and mark down their coordinates (shown on the top-left) in order to the distance to the launch site. calculate the distance between two points on the map based on their `Lat` and `Long` values. After the coordinates and distance is obtained, create a **`Marker`** to show the distance. Draw a **`PolyLine`** between a launch site to the selected point

- **GitHub URL:-**
- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb
- https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/f1f8189-8b05-4b32-8328-df446826bac0/view?access_token=b4fdfaf0578b4df86d21bd751976fc7c9049989ea4db82f873d41d0f7a048e89

Build a Dashboard with Plotly Dash

- This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.A Launch Site Drop-down Input Component is added to select a particular launch site or all launch sites.
- Add a callback function to render `success-pie-chart` based on selected site dropdown. The general idea of this callback function is to get the selected launch site from `site-dropdown` and render a pie chart visualizing launch success counts. A pie chart graph showing the success (`class=1`) count and failed (`class=0`) count for the selected site is displayed .
- Add a Range Slider to Select Payload to find if variable payload is correlated to mission outcome
- Add a callback function to render the `success-payload-scatter-chart` scatter plot. Plot a scatter plot with the x axis to be the payload and the y axis to be the launch outcome (i.e., `class` column) to observe how payload may be correlated with mission outcomes for selected site(s).color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters

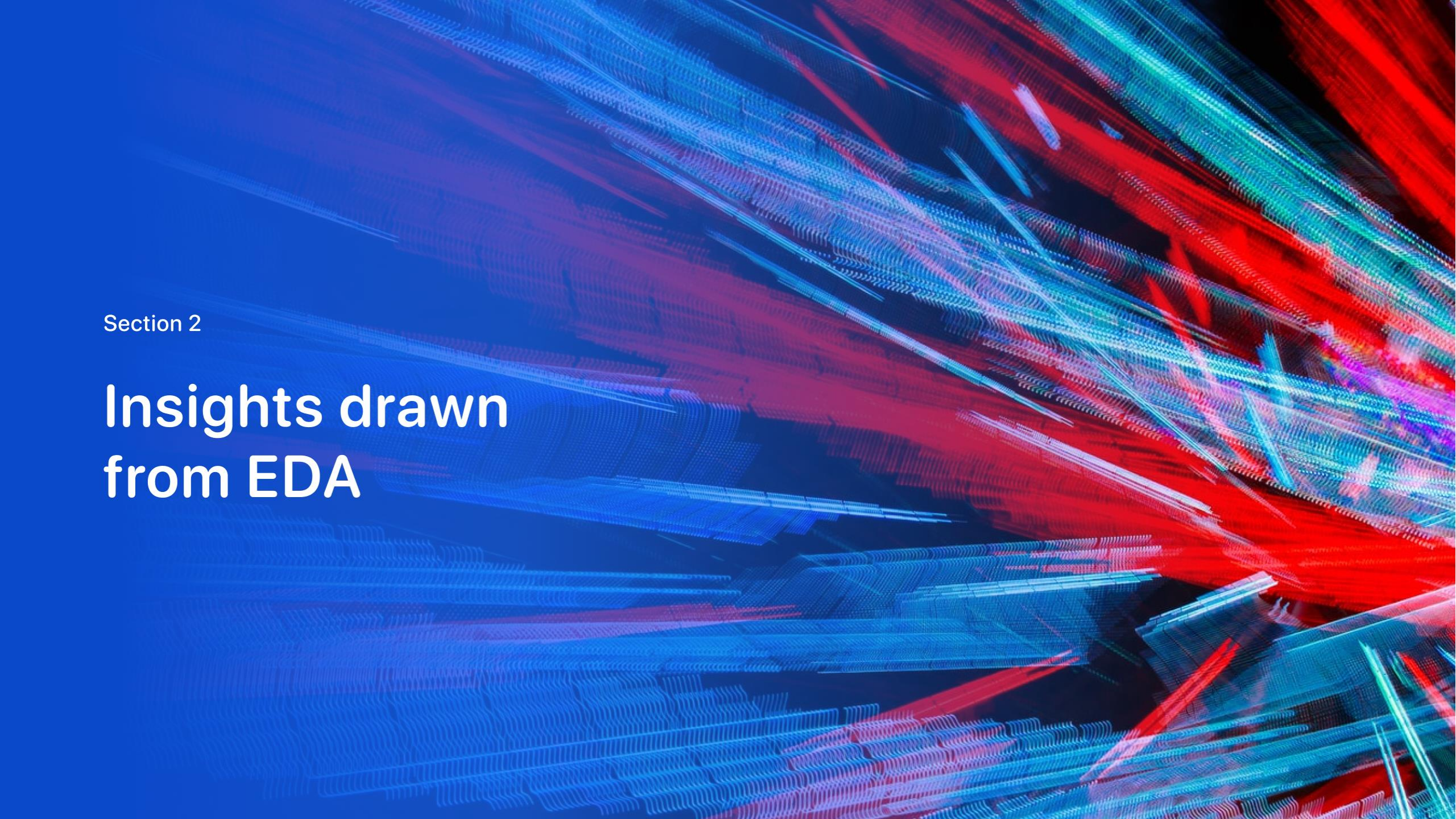
- **GitHub URL:-**
- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- We will build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully. This will include:
- Preprocessing, allowing us to standardize our data,
- Train_test_split, allowing us to split our data into training and testing data.
- We will train the model and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best. Using the best hyperparameter values, we will determine the model with the best accuracy using the training data. We will test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Finally, we will output the confusion matrix.
- **GitHub URL:-**
- https://github.com/RadhikaR91/Applied-DataScience-Capstone/blob/master/Predictive%20Analysis%20(Classification).ipynb

Results

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



Section 2

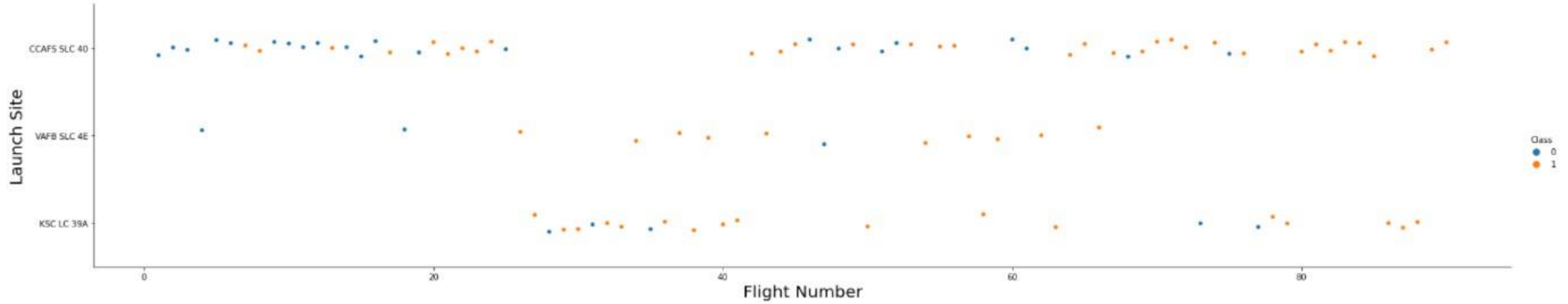
Insights drawn from EDA

Flight Number vs. Launch Site

parameter `hue` to `'class'`

15]:

```
# 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(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

From the scatter plot we see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%. VAFB-SLC launch site fewer rockets are launched (less than 80)

Payload vs. Launch Site

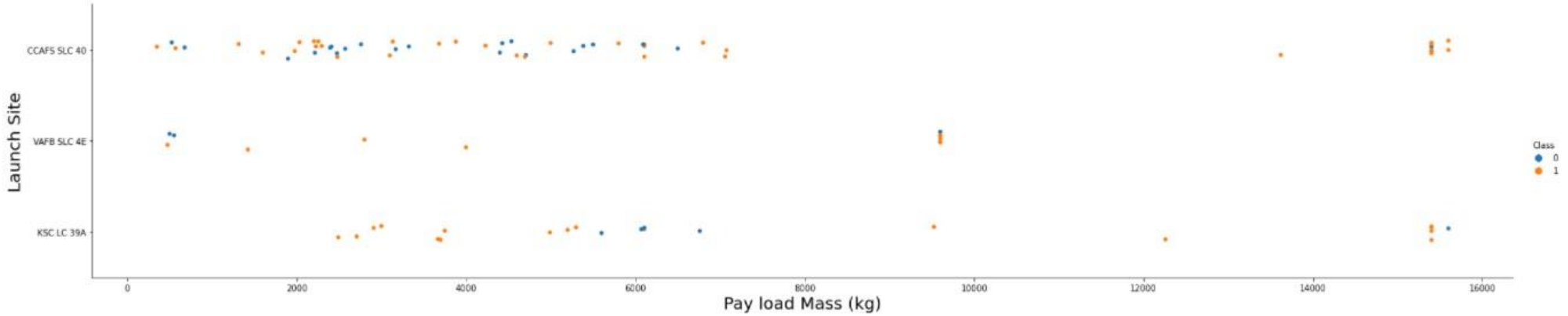
- From the plot we can see that VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

In [16]:

```
# 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(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

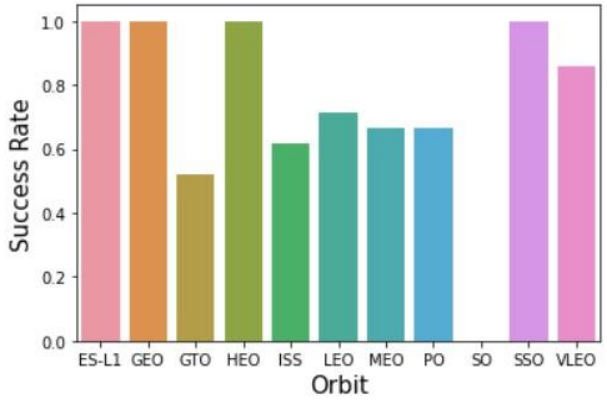
- From the bar plot we see that the orbits GEO,HEO,SSO&ES-L1 have higher success rates

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

17]:

```
# HINT use groupby method on Orbit column and get the mean of Class column
df_orbit = df.groupby('Orbit' , axis = 0).mean()
df_orbit.reset_index(inplace=True)
df_orbit.head()
sns.barplot(x='Orbit', y='Class', data=df_orbit)
plt.xlabel("Orbit",fontsize=15)
plt.ylabel("Success Rate",fontsize=15)
plt.show()
```



Flight Number vs. Orbit Type

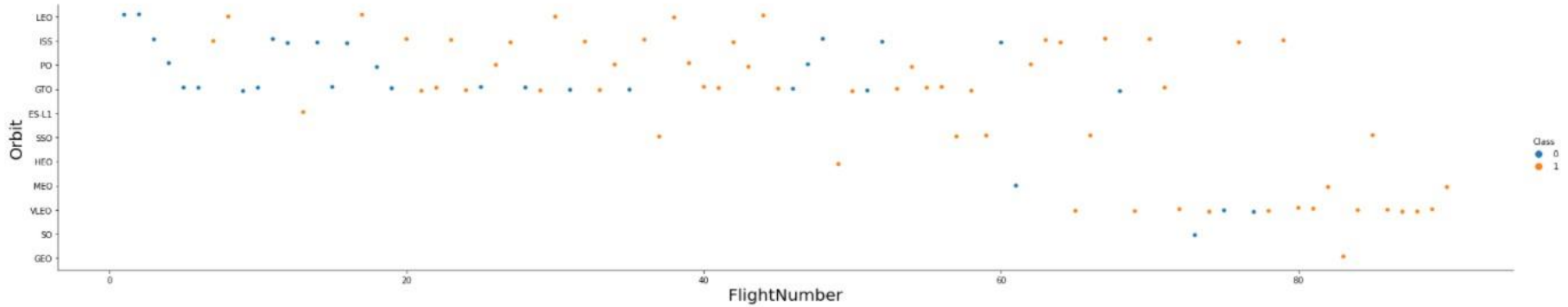
- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

In [18]:

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



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

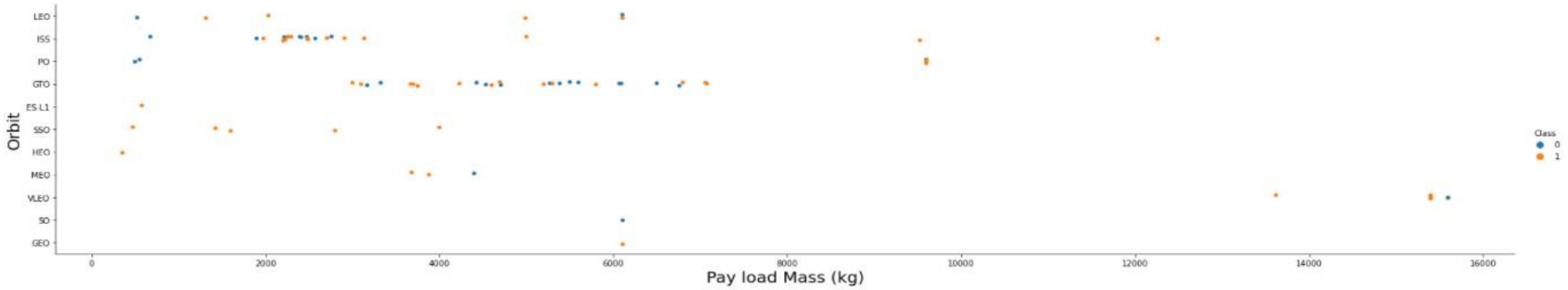
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there.

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

[19]:

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



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.

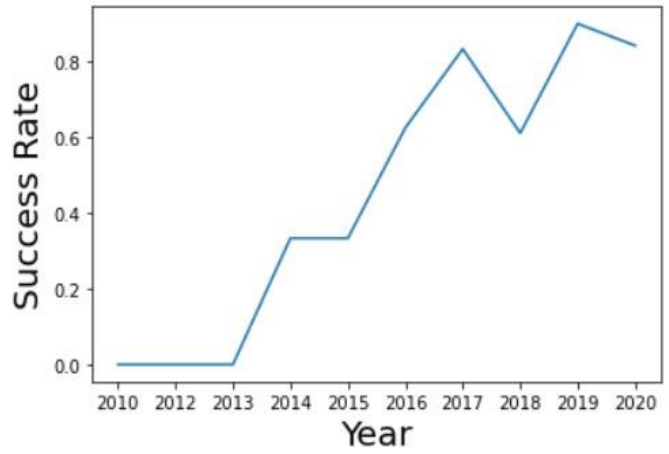
Launch Success Yearly Trend

- sucess rate since 2013 kept increasing till 2020

n [21]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
if(len(year)==0 or len(year)<90):
    df['Year'] = Extract_year(df['Date'])

df_YearlySuccess = df.groupby('Year' ,axis=0).mean()
sns.lineplot(x='Year',y='Class',data=df_YearlySuccess)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites
- **Query :- select DISTINCT(launch_site) from SPACEXDATASET.**

Task 1

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

```
4]: %sql select DISTINCT(launch_site) from SPACEXDATASET
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

4]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`.
- **c select * from SPACEXDATASET where launch_site like '%CCA%' limit 5**

Task 2

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

[5]:
```
%sql select * from SPACEXDATASET where launch_site like '%CCA%' limit 5
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

[5]:

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

Task 3

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- **Query:-select sum(payload_mass__kg_) from SPACEXDATASET where customer = 'NASA (CRS)'**

Task 3

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

```
[6]: %sql select sum(payload_mass__kg_) from SPACEXDATASET where customer = 'NASA (CRS)'

 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
ut[6]:  1
```

45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- **Query :- select avg(payload_mass__kg_) from SPACEXDATASET where booster_version ='F9 v1.1'**

Task 4

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

In [7]:
```
%sql select avg(payload_mass__kg_) from SPACEXDATASET where booster_version ='F9 v1.1'
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[7]:

1
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- **Query :- select min(date) from SPACEXDATASET where LANDING__OUTCOME = 'Success (ground pad)'**

List the date when the first successful landing outcome in ground pad was achieved.

Hint:Use min function

[8]:
```
%sql select min(date) from SPACEXDATASET where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

[8]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Query:- **`select BOOSTER_VERSION from SPACEXDATASET where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_<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

In [9]:
```
%sql select BOOSTER_VERSION from SPACEXDATASET where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_<6000
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[9]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- **Query :-**

1. select count(MISSION_OUTCOME) as SUCCESSFUL_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Success%'
2. select count(MISSION_OUTCOME) as FAILED_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Failure%'

List the total number of successful and failure mission outcomes

```
In [10]: %sql  select count(MISSION_OUTCOME) as SUCCESSFUL_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Success%'
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
Out[10]: successful_outcomes
                        100
```

```
In [11]: %sql  select count(MISSION_OUTCOME) as FAILED_OUTCOMES from SPACEXDATASET where MISSION_OUTCOME like '%Failure%'
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
Out[11]: failed_outcomes
                        1
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- **Query :- select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ in (SELECT max(PAYLOAD_MASS__KG_) from SPACEXDATASET)**

Task 8

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

In [12]:
```
%sql select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ in (SELECT max(PAYLOAD_MASS__KG_) from SPACEXDATASET)
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[12]:

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- **Query :- select LANDING__OUTCOME ,BOOSTER_VERSION ,LAUNCH_SITE ,DATE from SPACEXDATASET where LANDING__OUTCOME ='Failure (drone ship)' and year(DATE) = 2015**

Task 9

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

In [13]:
```
%sql select LANDING__OUTCOME ,BOOSTER_VERSION ,LAUNCH_SITE ,DATE from SPACEXDATASET where LANDING__OUTCOME ='Failure (drone ship)' and year(DATE) = 201
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[13]:

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- 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
- **Query :- select LANDING__OUTCOME ,COUNT(LANDING__OUTCOME) as frequency from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY frequency DESC**

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

[14]:
```
%sql select LANDING__OUTCOME ,COUNT(LANDING__OUTCOME) as frequency from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20' GROUP BY LANDING
```

```
 * ibm_db_sa://vvt78396:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

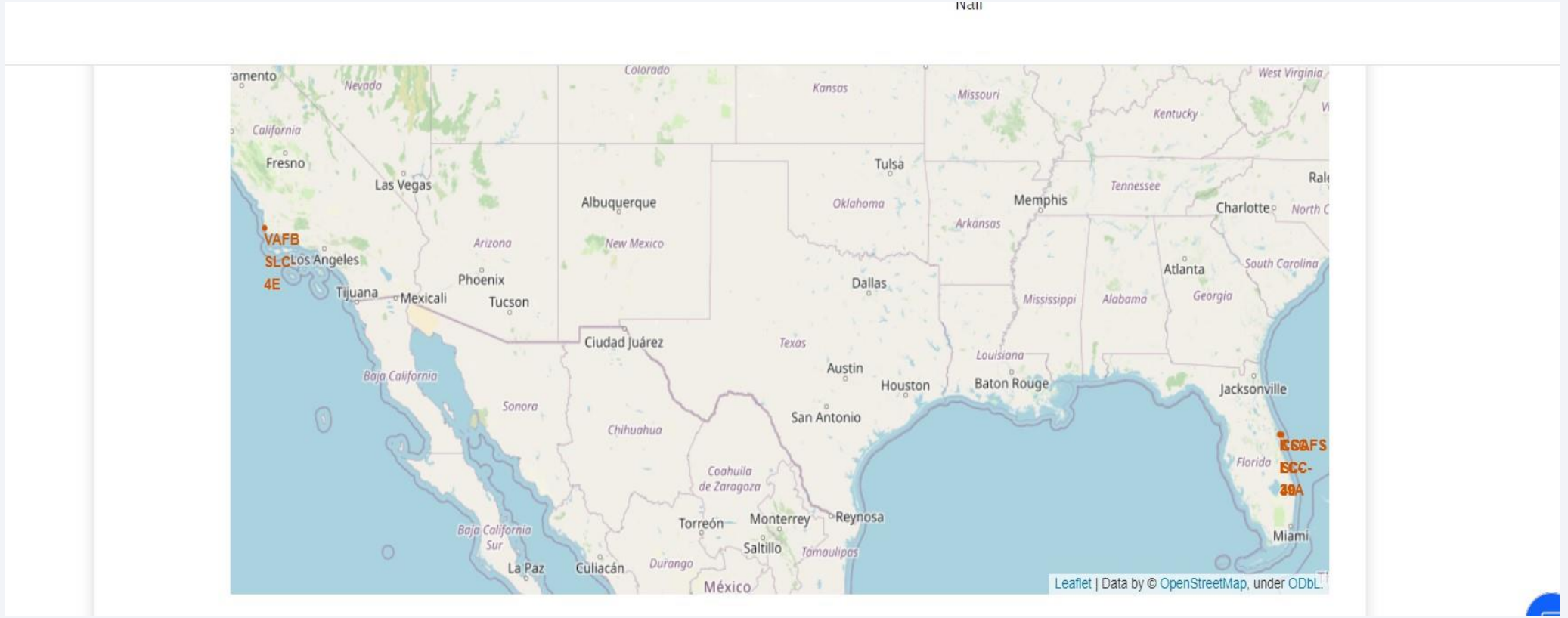
t[14]:

landing__outcome	frequency
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

Section 4

Launch Sites Proximities Analysis

All launch sites on a map



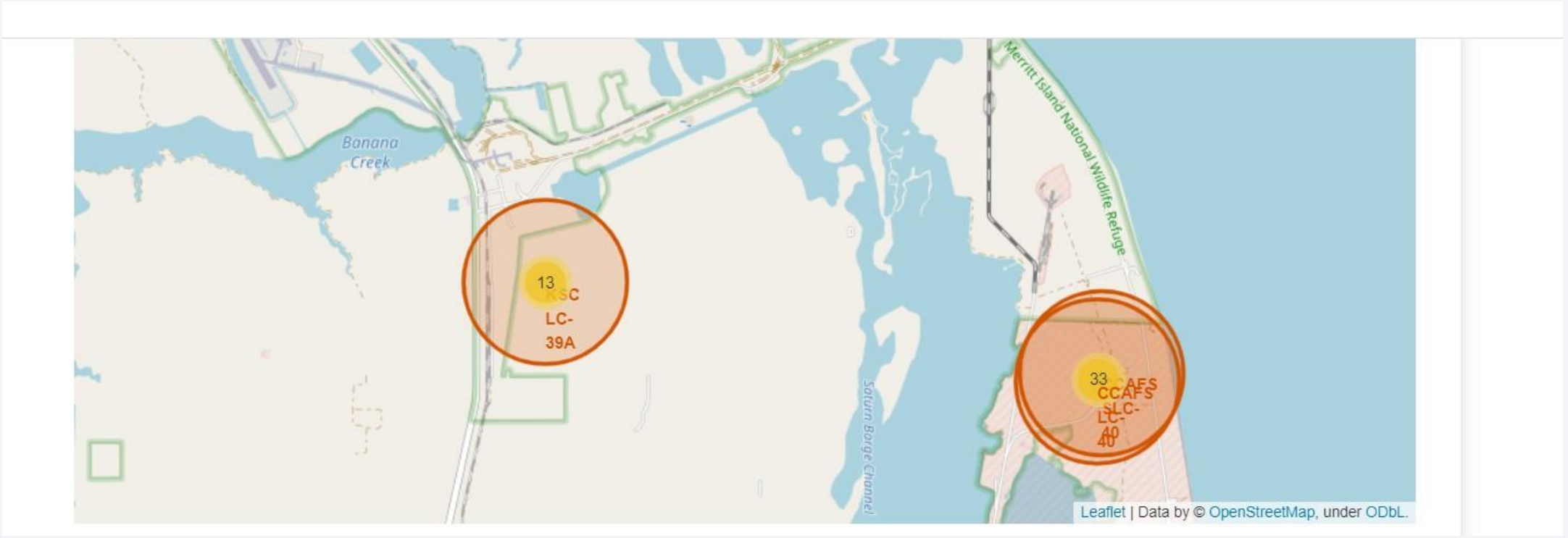
- Are all launch sites in proximity to the Equator line?

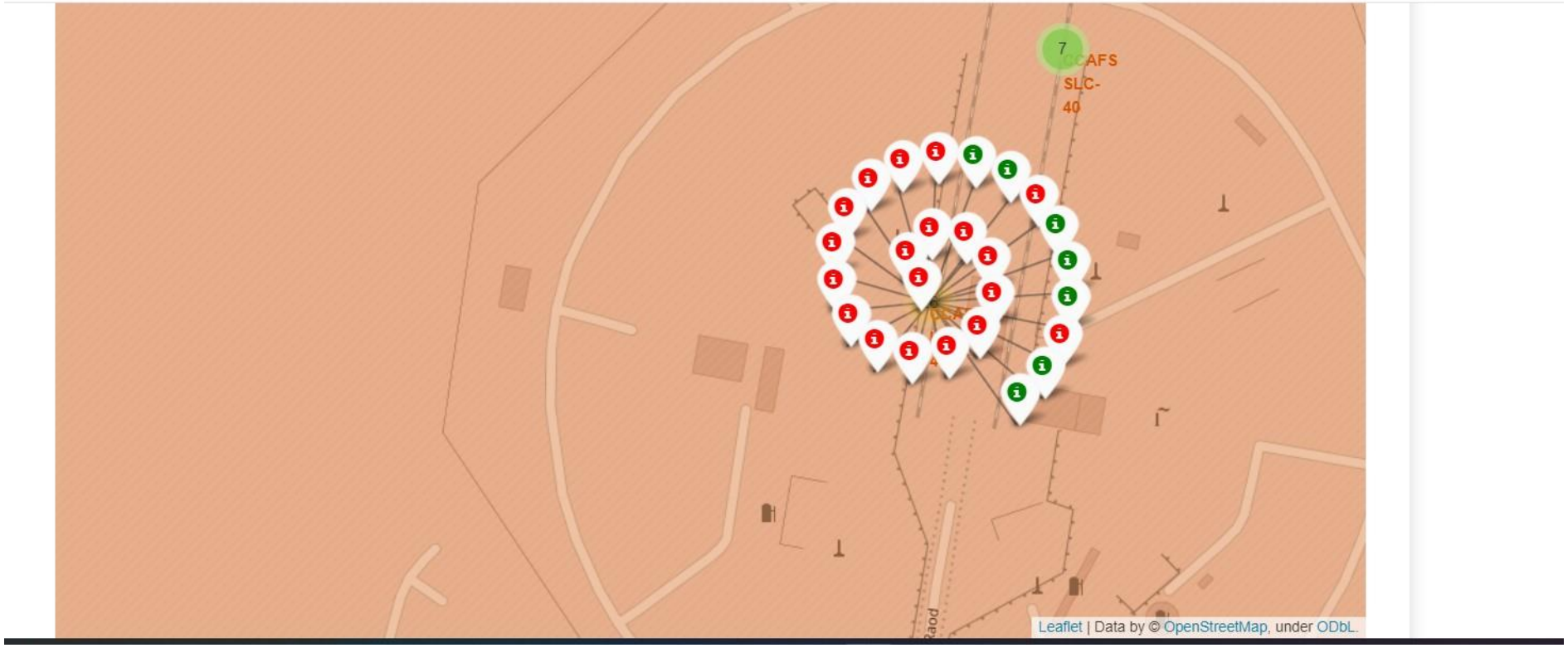
  Yes the launch sites are in proximity to the equator line.

- Are all launch sites in very close proximity to the coast?
- Yes all the launch sites are in close proximity to the coast .Such a location for the launch site is chosen in an effort to minimize (if not completely eliminate) the risk to human life in the case of a failure during the initial ascent of a rocket.

success/failed launches for each site on the map

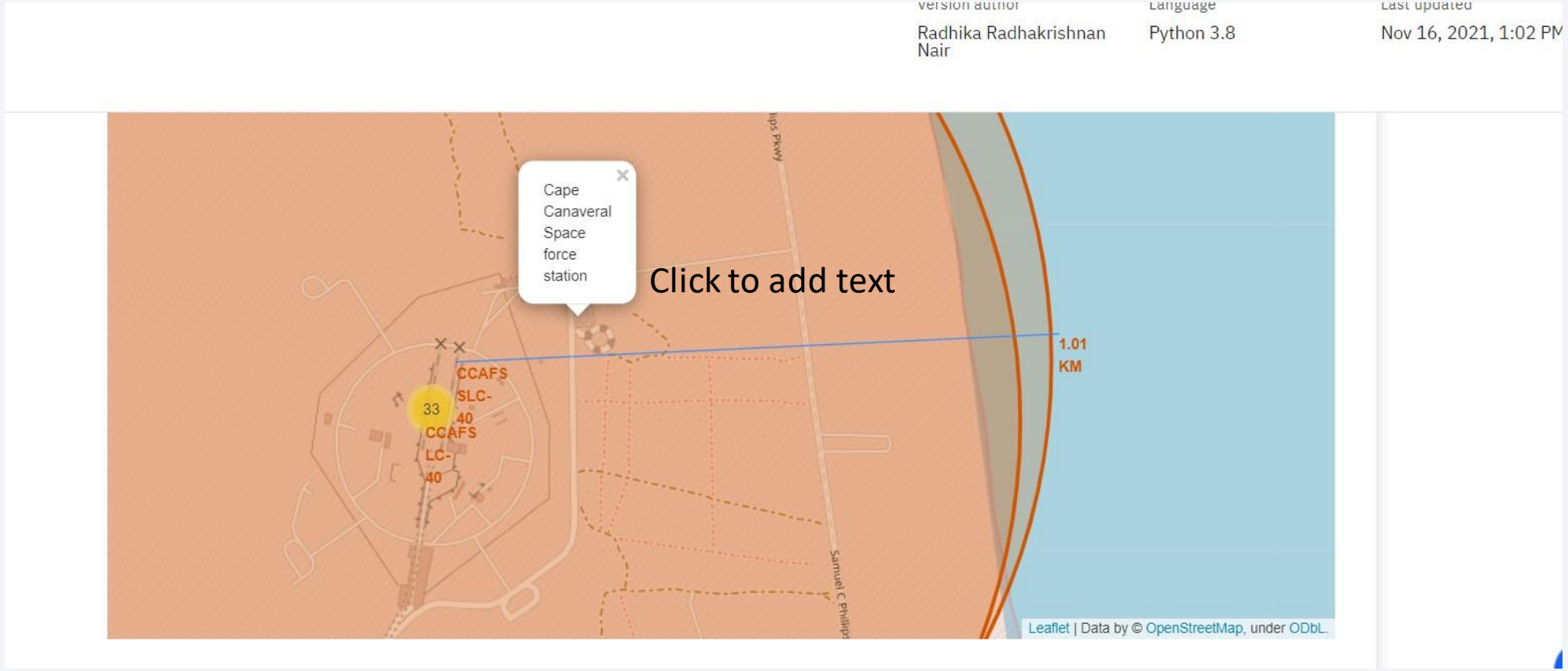
- enhance the map by adding the launch outcomes for each site, and see which sites have high success rates.



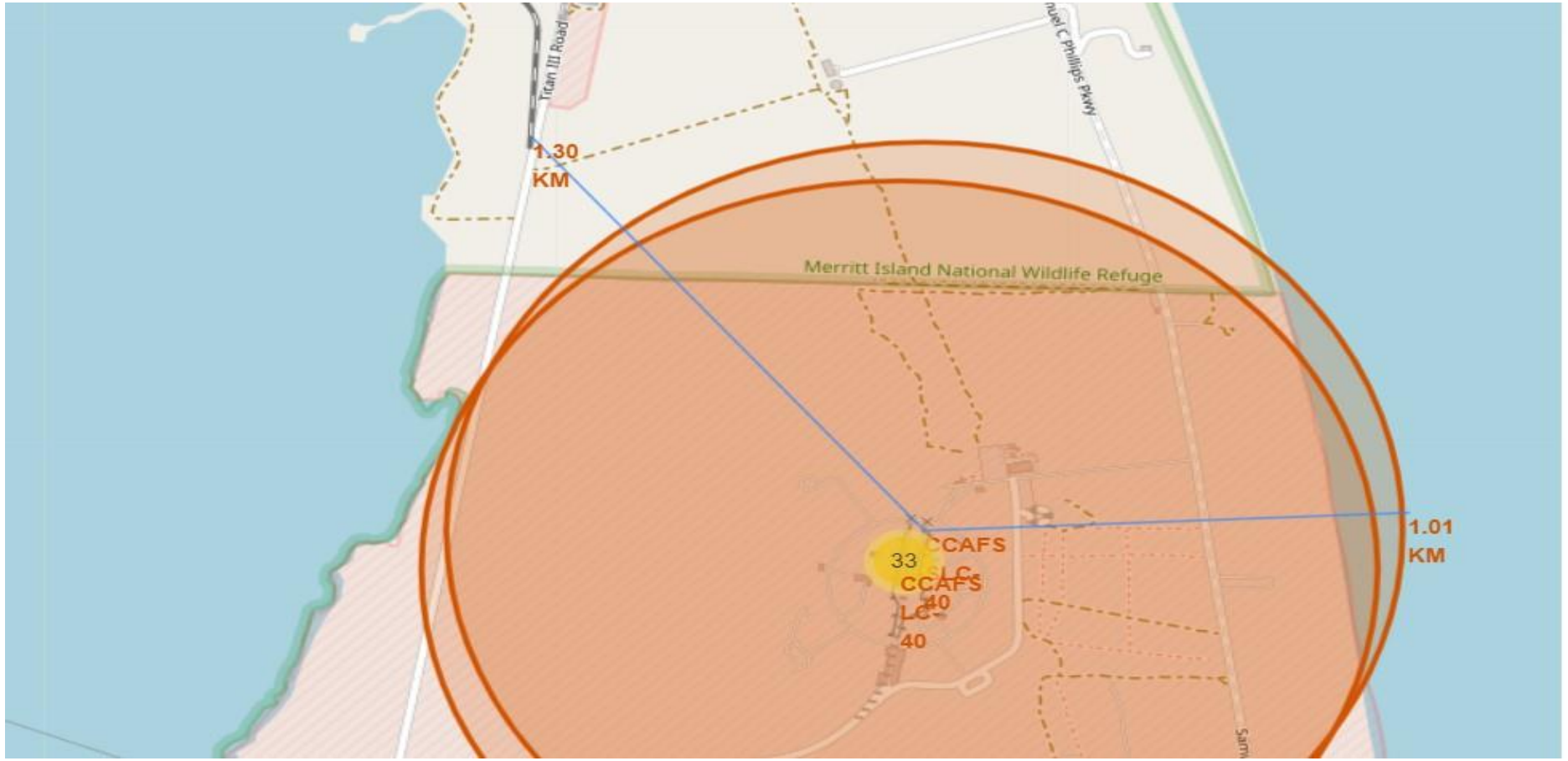


KSC LC-39A has more successful launches almost 77%.From the map we can see that the launch site KSC LC 39A has more color-labeled markers .

Distances between a launch site to its proximity to coastline

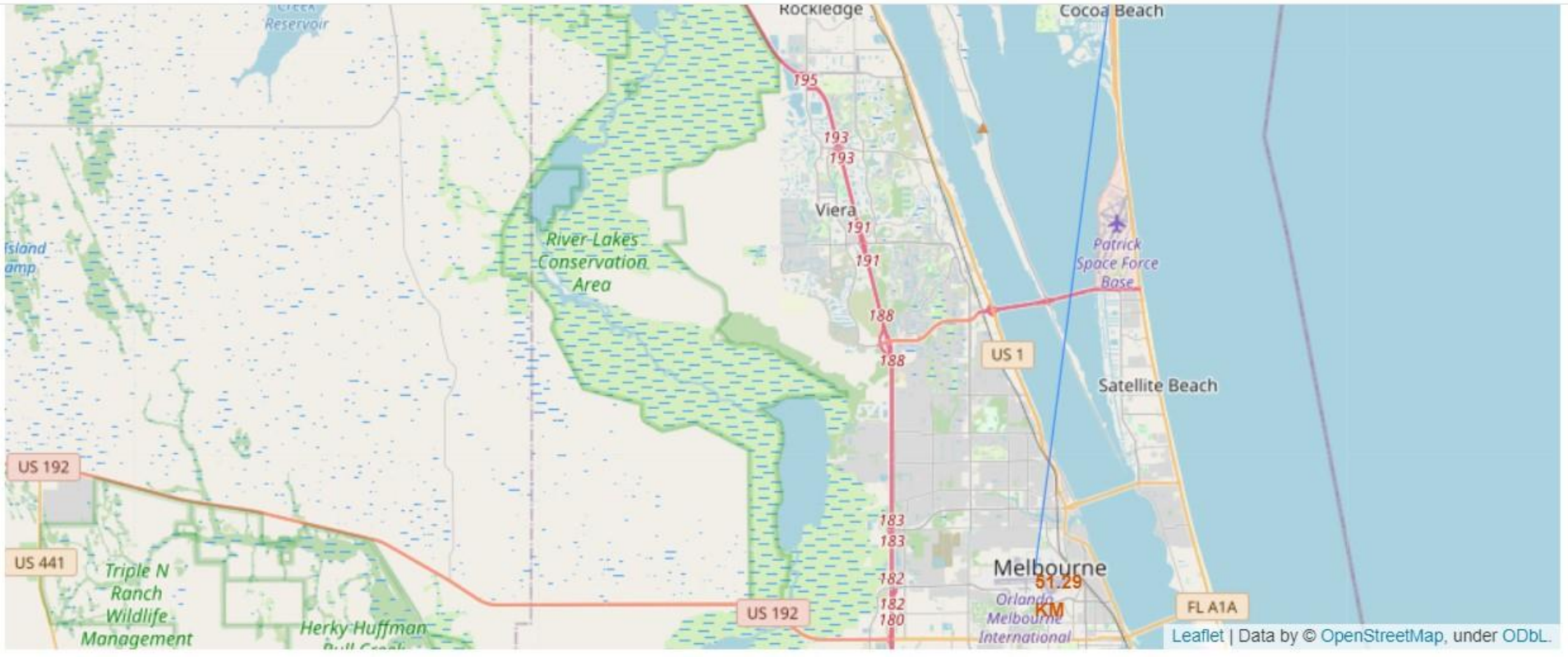


Distances between a launch site to its proximity to railway



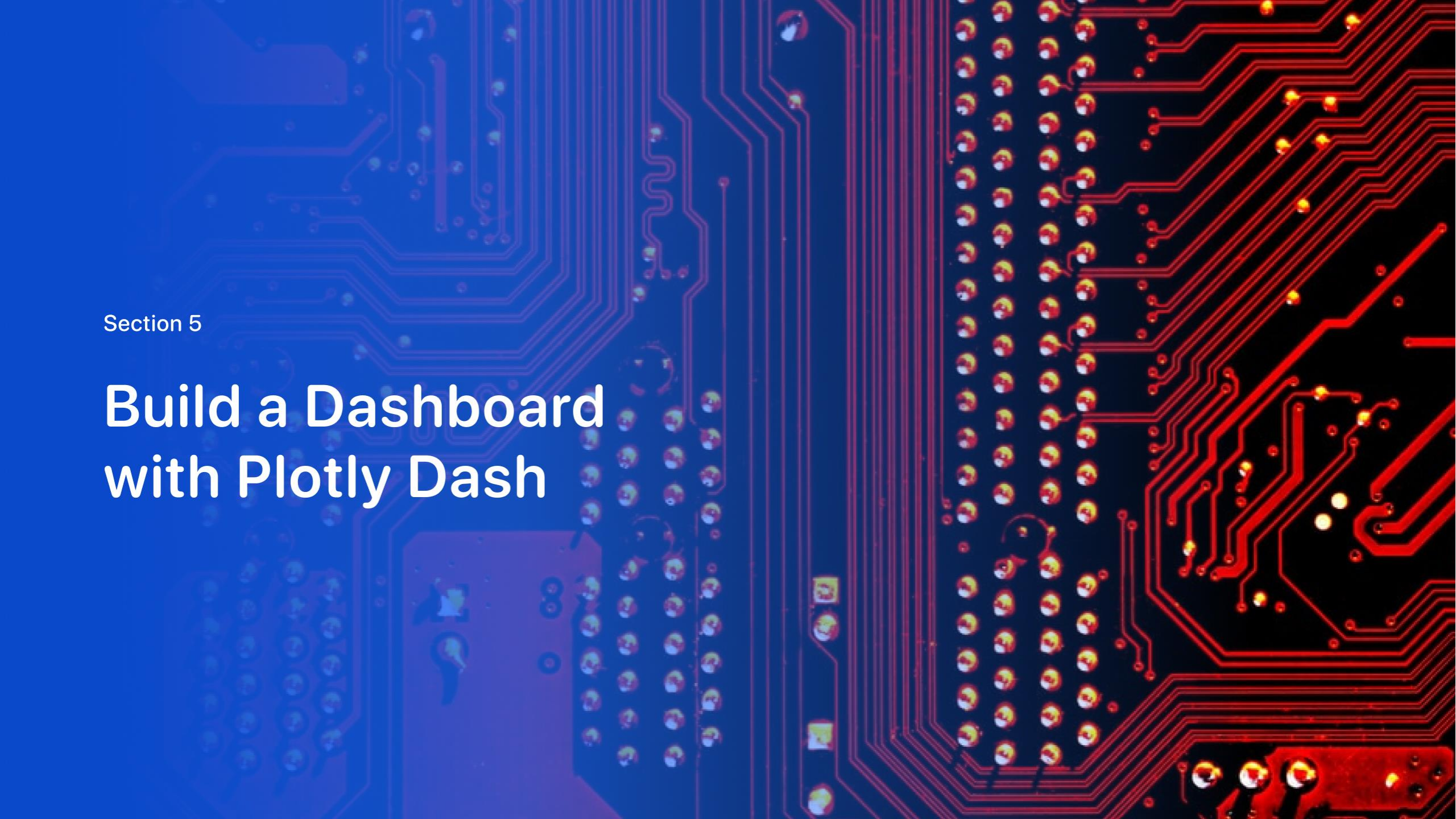
Distances between a launch site to its proximity to highway.

Distances between a launch site to its proximity to city.



Findings from the map

- The launch sites are in close proximity to coastline(1km) ,highway(0.59km) and railway(1.30km).
- The launch site is far away from city(59km).



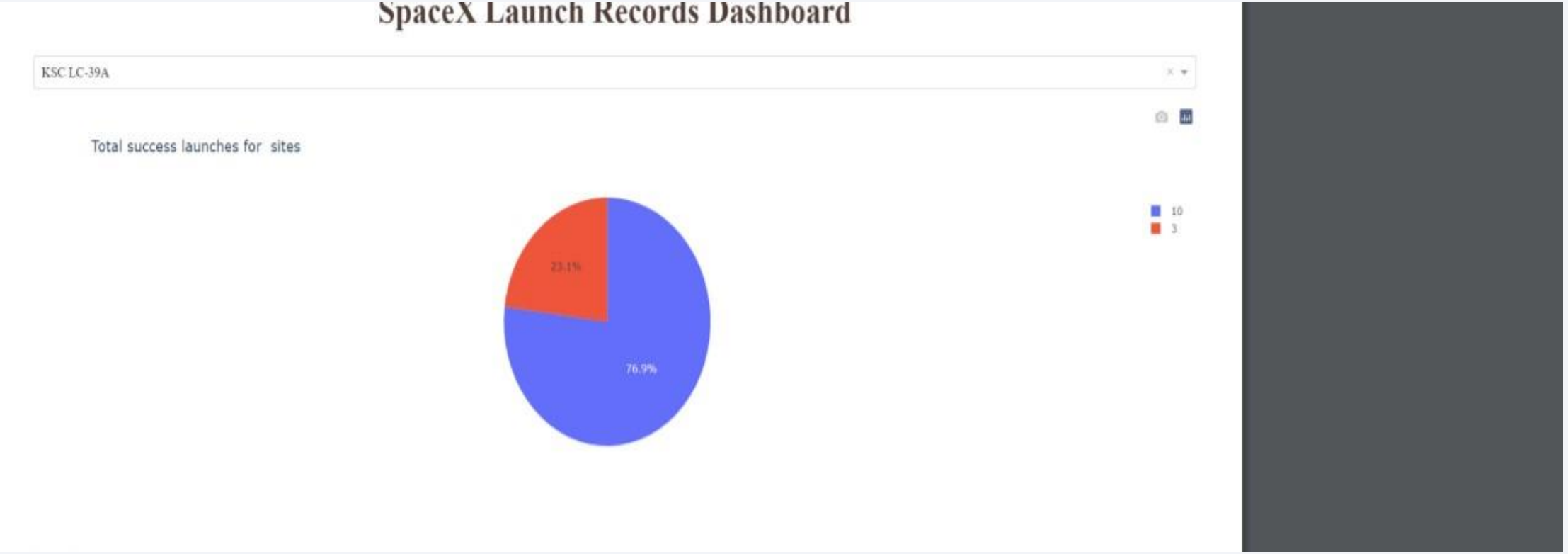
Section 5

Build a Dashboard with Plotly Dash

Launch Success for ALL sites

From the pie chart it is clear that the launch site KSC LC 39A has the most successful launches (41.7%),followed by the launch site CCAFS LC 40 (29%) while the launch site VAFB SLC 4E has success rate of 16% and the launch site CCAFS SLC 40 has the least success with 12.5%

Highest Launch Success Ratio



Payload vs. Launch Outcome scatter plot

Section 6

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

[104…]
```
from sklearn.metrics import f1_score
print("F1 Score for LogisticRegression :", f1_score(Y_test, logreg_cv.predict(X_test), average='weighted'))
print("F1 Score for SVM :", f1_score(Y_test, svm_cv.predict(X_test), average='weighted'))
print("F1 Score for Decision Tree Classifier :", f1_score(Y_test, tree_cv.predict(X_test), average='weighted'))
print("F1 Score for K Nearest Neighbors Classifier :", f1_score(Y_test, knn_cv.predict(X_test), average='weighted'))
```

```
F1 Score for LogisticRegression : 0.8148148148148149
F1 Score for SVM : 0.8148148148148149
F1 Score for Decision Tree Classifier : 0.8148148148148149
F1 Score for K Nearest Neighbors Classifier : 0.8148148148148149
```

[107…]
```
print("accuracy :",svm_cv.best_score_)
print("accuracy :",logreg_cv.best_score_)
print("accuracy :",knn_cv.best_score_)
print("accuracy :",tree_cv.best_score_)
```

```
accuracy : 0.8482142857142856
accuracy : 0.8464285714285713
accuracy : 0.8482142857142858
accuracy : 0.8767857142857143
```

[108…]
```
print("The best accuracy is {} for Decision Tree classifier".format(tree_cv.best_score_))
```

```
The best accuracy is 0.8767857142857143 for Decision Tree classifier
```

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

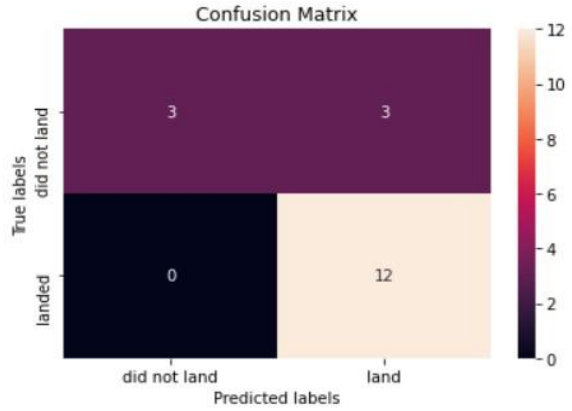
Calculate the accuracy of tree_cv on the test data using the method `score`:

```
BestTree = tree_cv.best_estimator_
print("accuracy of test data:", BestTree.score(X_test, Y_test))
```

```
accuracy of test data: 0.8333333333333334
```

We can plot the confusion matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

By analysing the data for Spacex launches we found the following :-

- Different launch sites has different success rate of which KSC LC 39A has the highest success rate .
- The Decision Tree classifier has the highest accuracy for predicting whether a launch would be successful or not.

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!