Winning Space Race with Data Science

TATA SAI PRASANNA

10-02-2024

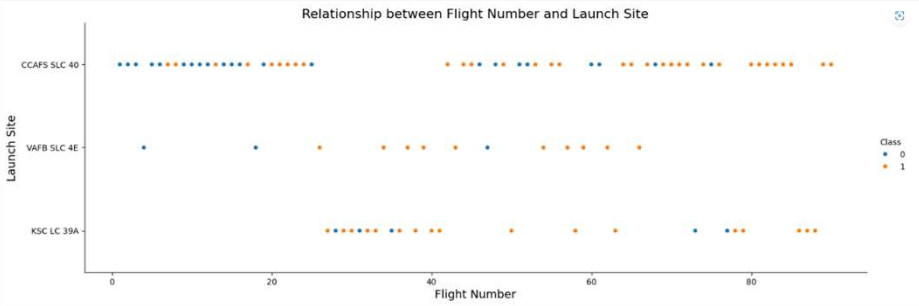
Outline

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

Executive Summary

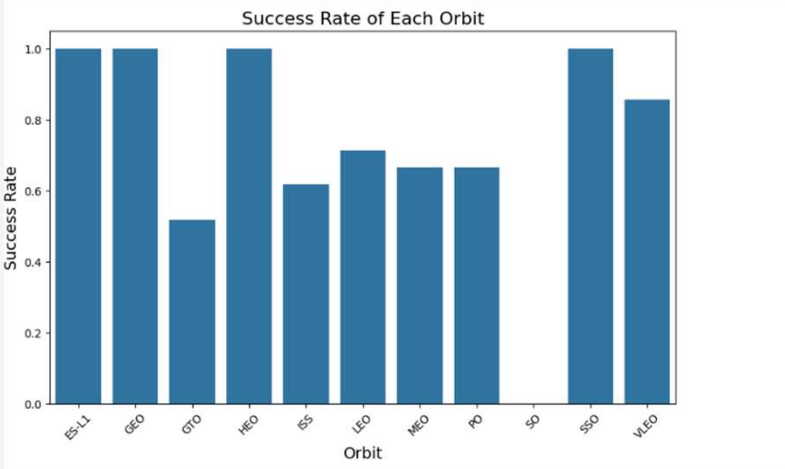
Summary of methodologies -

- Data Collection via API, SQL and Web Scraping Data Wrangling and Analysis
- Interactive Maps with Folium
- Predictive Analysis for each classification model



Summary of all results -

- Data Analysis along with Interactive Vi! Best model for Predictive Analysis



Introduction

- **Project background and context**:

Here we will predict if the Falcon 9 first stage will land successfully. Spacex adverting 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 fand successfully. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problems we want to find answers:

1)With what factors, the rocket will land successfully?

2)The effect of each relationship of rocket variables on outcome.

3)Conditions which will aid Spacex have to achieve the best results.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Via SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One hot encoding data fields for machine learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL

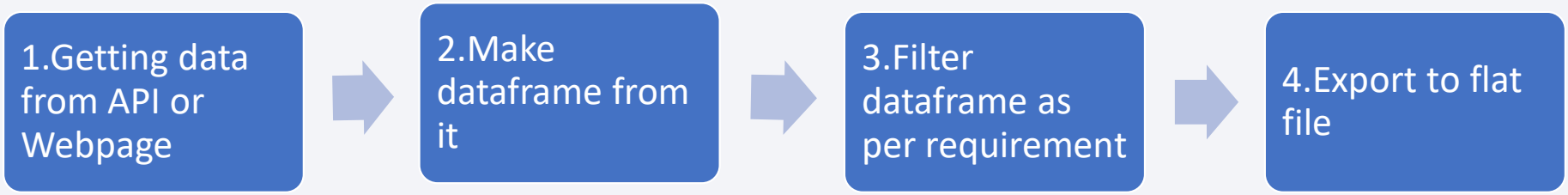
  scatter and bar graphs to show pattern between data
- Perform interactive visual analytics using Folium and Plotly Dash

  Using folium and Plotly Dash Visualizations
- Perform predictive analysis using classification models:

  Build and evaluate classification models

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes



Data Collection – Via SpaceX API

```
%sql SELECT SUM(Payload_Mass__kg_) AS Total_Payload_Mass_CRS FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

```
%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

1. Getting response from API
2. Converting Response to a .json file
3. Apply custom functions to clean data
4. Assign list to dictinary then create dataframe

Data Collection – Via Web Scraping

```
%sql SELECT SUM(Payload_Mass__kg_) AS Total_Payload_Mass_CRS FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

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

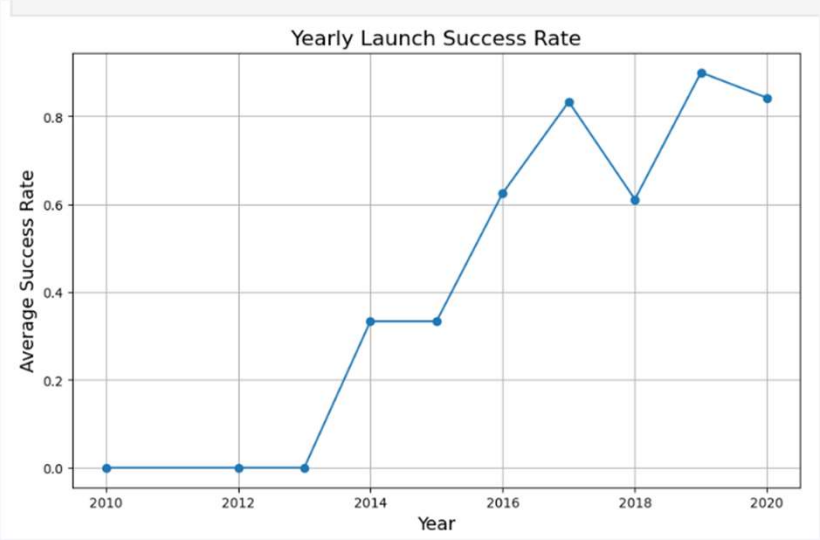
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

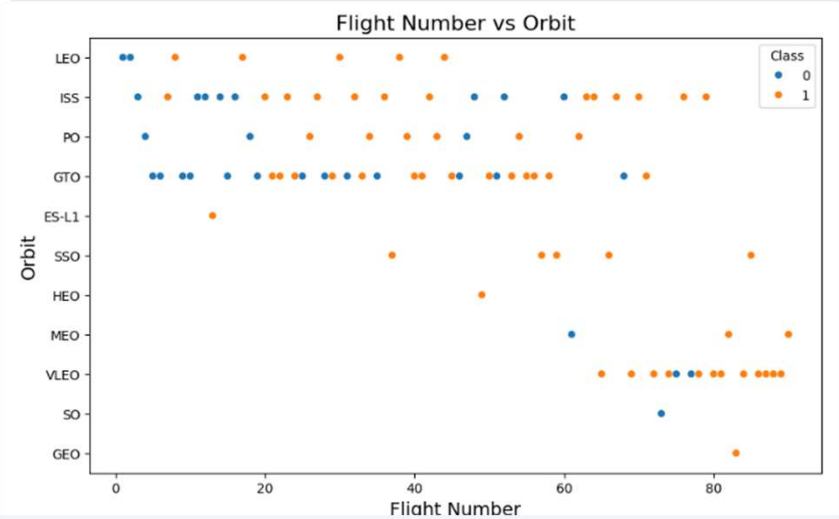
1. Getting Response from html
2. Creating BeautifulSoup object
3. Finding tables
4. Getting column names
5. Creation of dictionary and appending data to keys
6. Converting dictionary to dataframe
7. Dataframe to .CSV

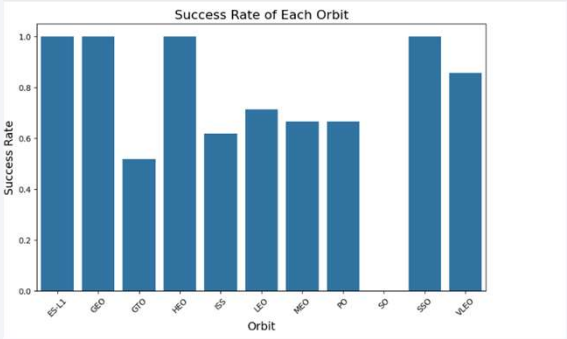
Data Wrangling

1. Load data
2. Make dataframe from it
3. Cleaning data
4. Simplifying it to Boolean values
5. Export to flat file

EDA with Data Visualization







EDA with SQL

SQL is an indispensable tool for Data Scientists and analysts. as most of the real-world data is stored in databases. it's not only the standard language for Relational Database operations, but also an incredibly powerful tool for analyzing data and drawing useful insights from it .Here we use IBM's Db2 for cloud, which is a fully managed SQL Database provided as a service.

We performed SQL queries to gather information from given dataset:

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

Display 5 records where launch sites begin with the string CCA

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

Displaying average payload mass carried by booster version F9v11

Listing the date where the successful landing outcome in drone ship was achieved

Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

Listing the total number of successful and failure mission outcomes

Listing the failed landingoutcomes in drone ship, their boosters versions, and launch site names for the year 2015

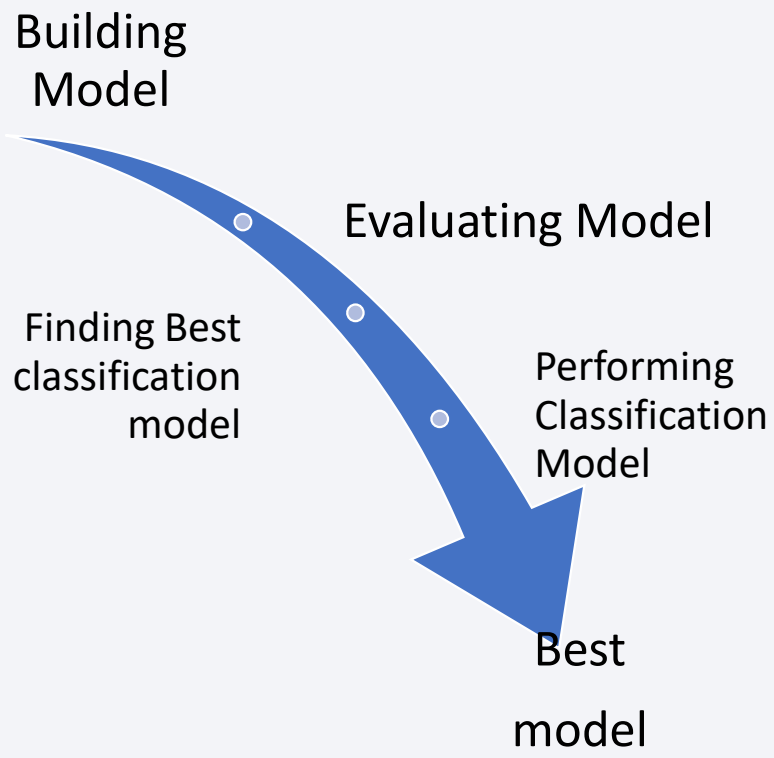
Build an Interactive Map with Folium

Map Object	Code	Result
Map Marker	folium.Marker(	Map objects to make a mark on map.
Icon Marker	folium.Icon(	Create an icon on map.
Circle Marker	folium.Circle(	Create a circle where marker is being placed.
PolyLine	folium.PolyLine(	Create a line between points.
Marker Cluster Object	MarkerCluster()	This is a good way to simplify a map containing many markers having the some coordinates.
Antpath	folium.plugins.Antpath(	Create an animated line between points.

Build a Dashboard with Plotly Dash

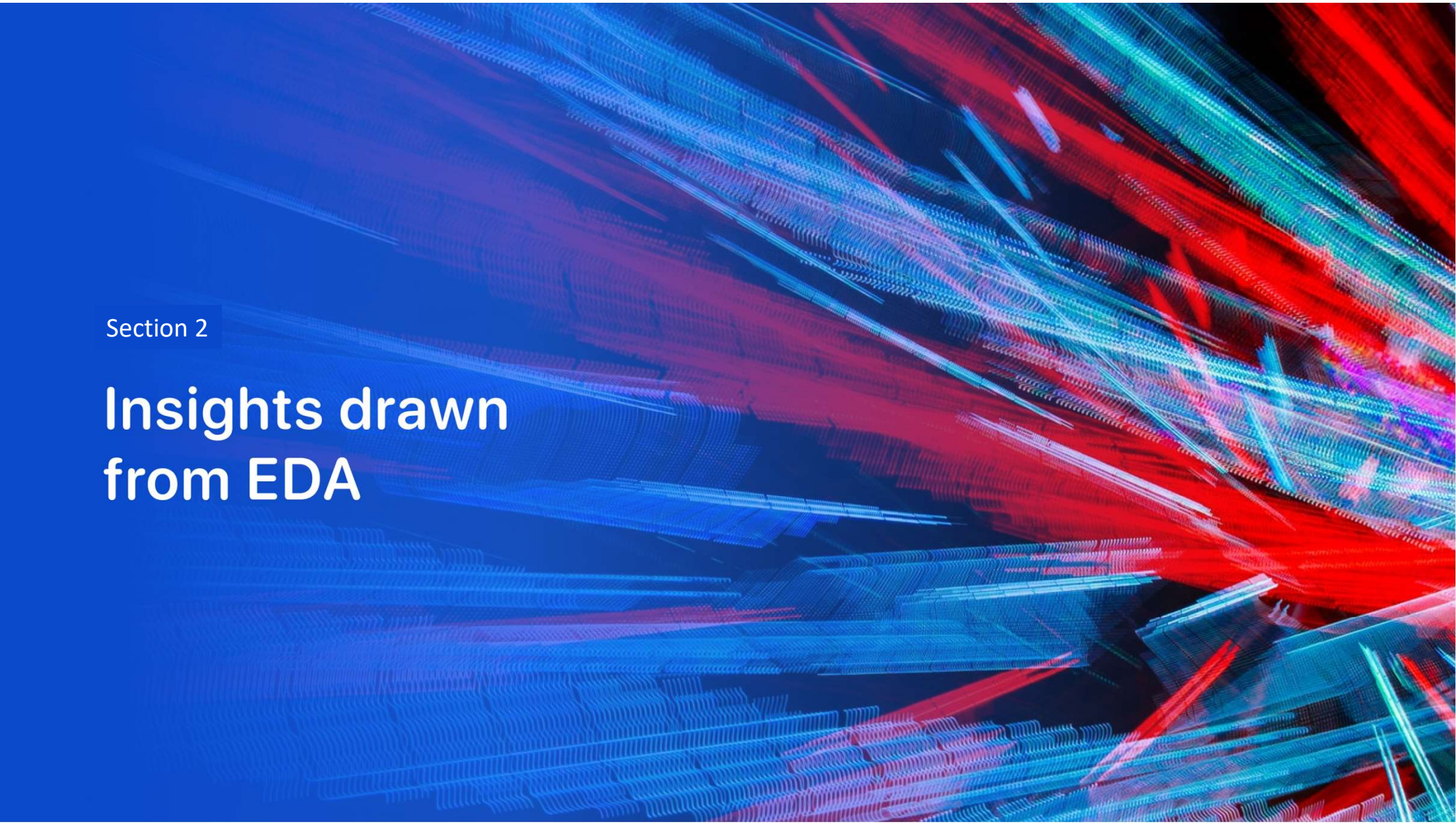
Map Object	Code	Result
Dash and its components	import dash import dash_html_components as html import dash_core_components as doc arkar from dash.dependencies import Input, Output	Plotly stewards python's leading data vizard and ui libraries.With Dash open source,Dash apps run on your local laptop server.
Pandas	import pandas as pd	Fetching values from CSV and creating a dataframe
Plotly	Import plotly.express as px	Plot the graphs with interactive plotly library
Dropdown	dcc.Dropdown(	Create a dropdown for launch sites
Rangeslider	dcc.RangeSlider(	Create a rangeslide for payload mass range selection
piechart	Px.pie(	Creating the pie graph for success percentage display
Scatter Chart	Px.scatter(	Creating the scatter graph for correlation display

Predictive Analysis (Classification)



Results

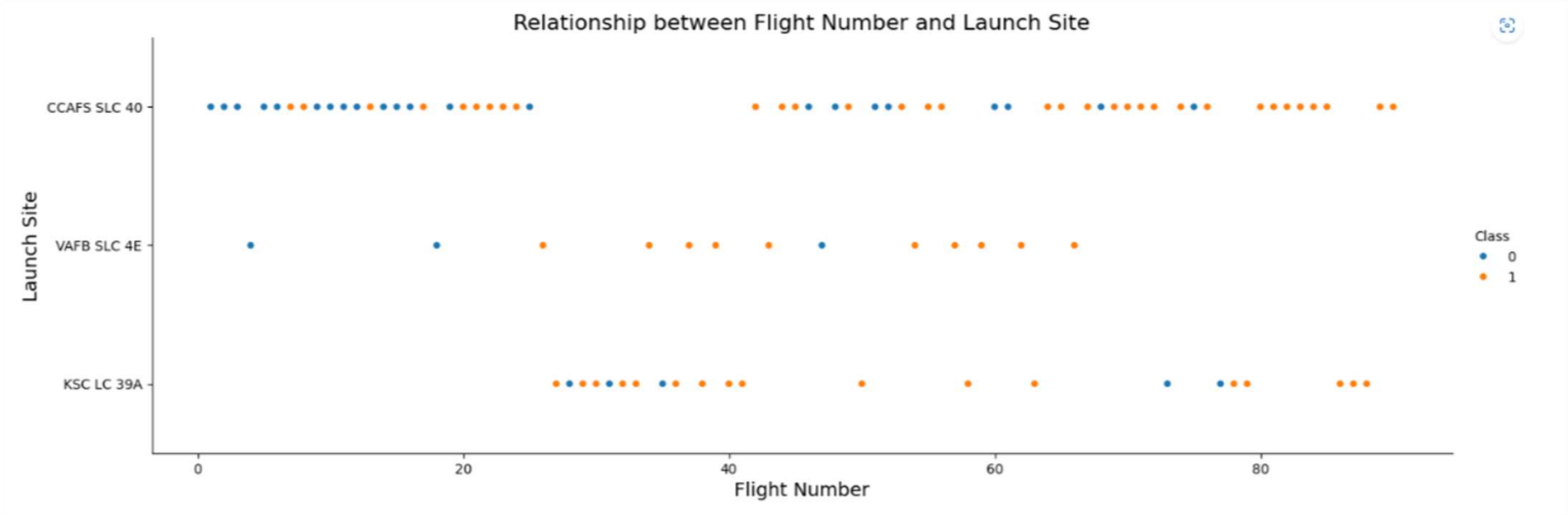
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



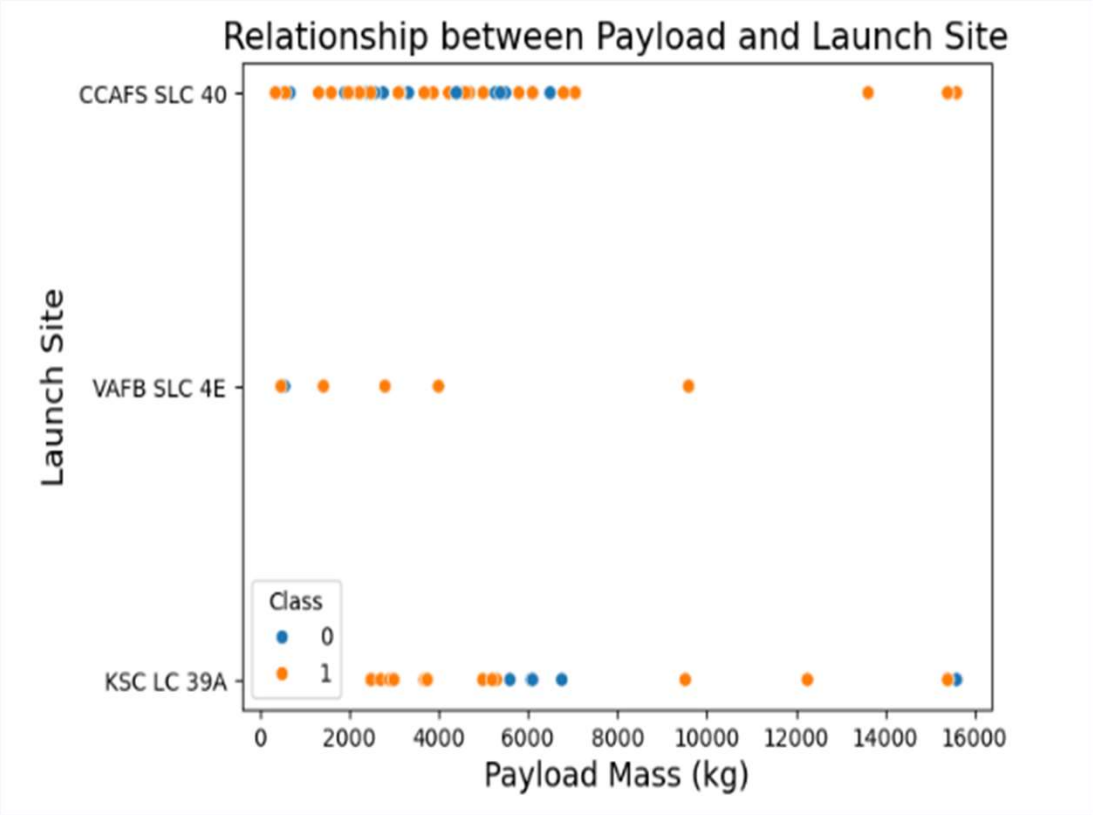
Section 2

Insights drawn from EDA

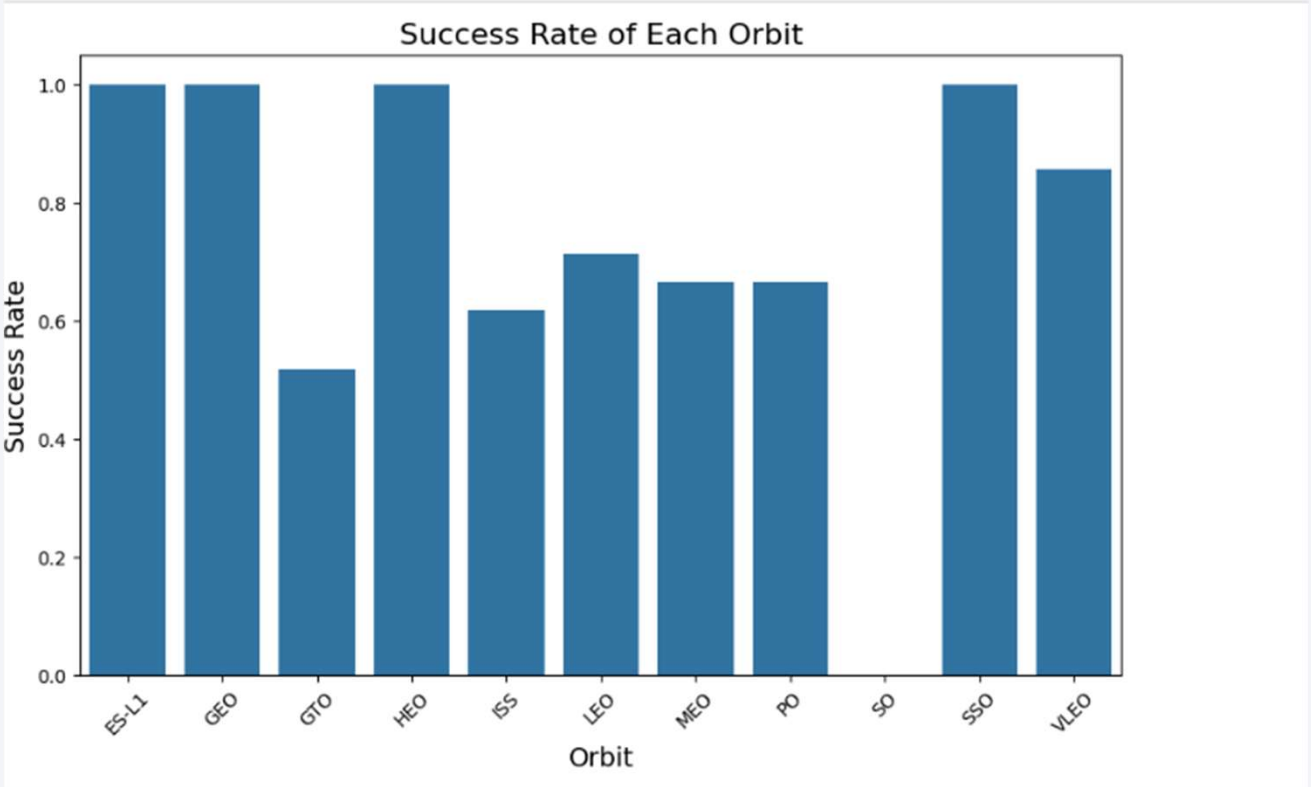
Flight Number vs. Launch Site



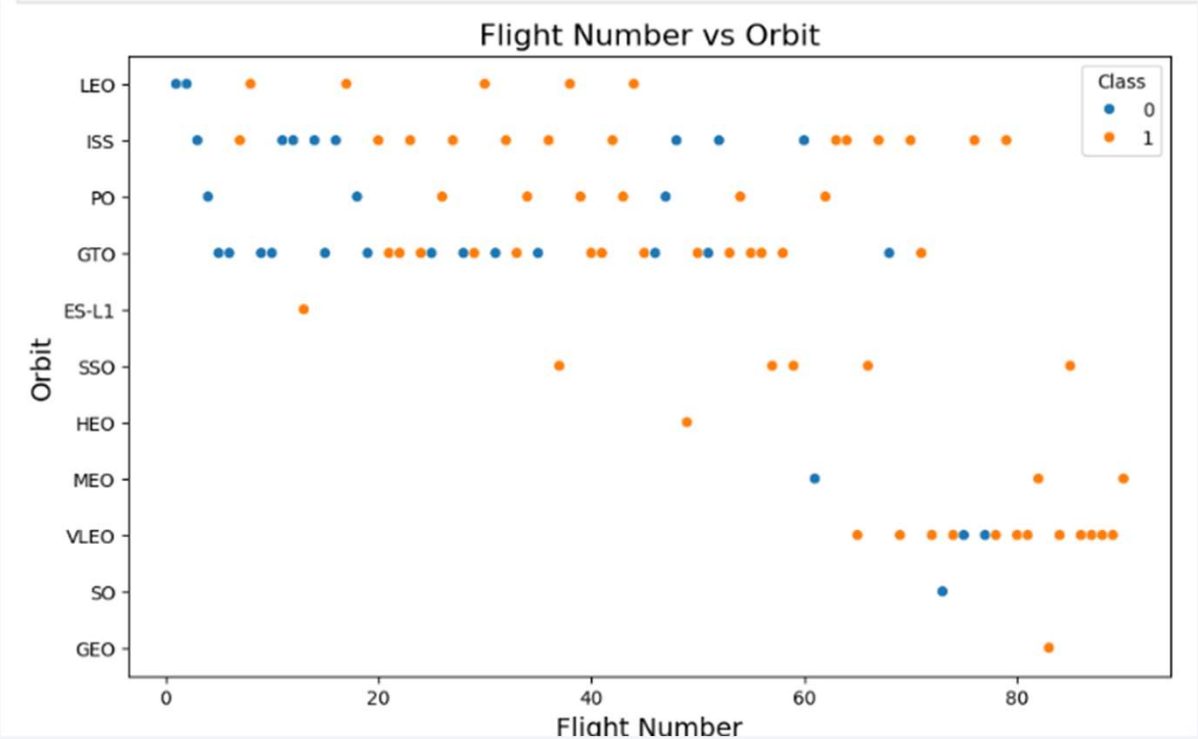
Payload vs. Launch Site



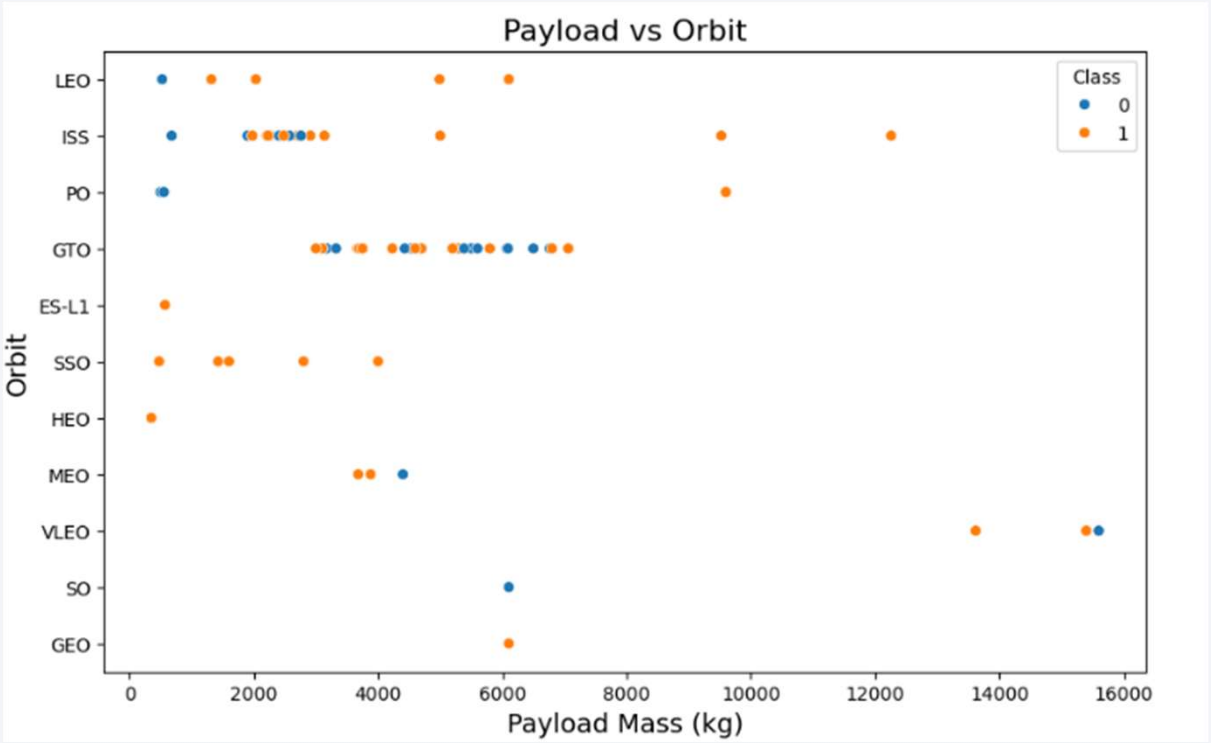
Success Rate vs. Orbit Type



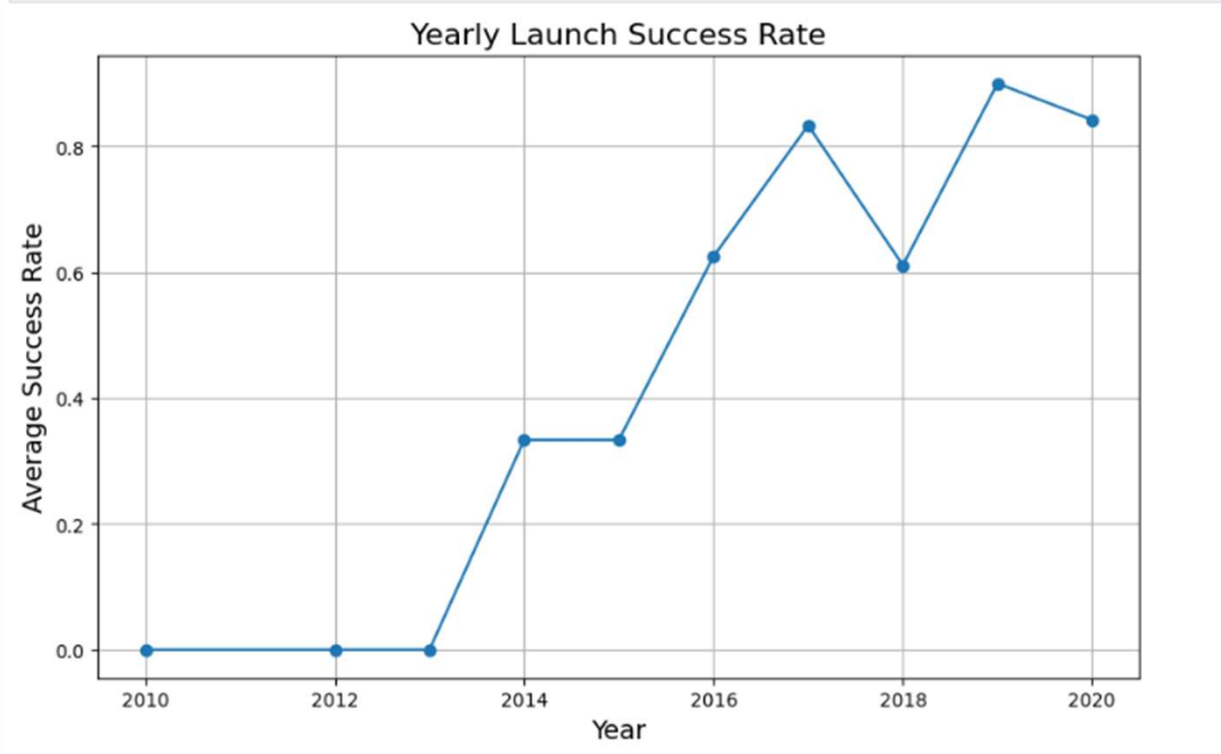
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Task 1

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

[10]:
```
%load_ext sql

# Establish connection to the database
%sql sqlite:///my_data1.db

# Execute SQL query
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
 * sqlite:///my_data1.db
Done.
```

[10]:

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

Launch Site Names Begin with 'CCA'

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

Total Payload Mass

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

```
%sql SELECT SUM(Payload_Mass__kg_) AS Total_Payload_Mass_CRS FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

Total_Payload_Mass_CRS
45596

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(Payload_Mass__kg_) AS Average_Payload_Mass_F9_v1_1 FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

Average_Payload_Mass_F9_v1_1
2928.4

First Successful Ground Landing Date

Hint:Use min function

```
%sql SELECT MIN(Date) AS Date_First_Successful_Landing_Ground_Pad FROM SPACEXTBL WHERE Landing_Outcome = 'Success (g
```

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

Date_First_Successful_Landing_Ground_Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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 FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND Payload_Mass__kg_ > 4000 AND Payload_Mass__kg_ < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT Mission_Outcome, COUNT(*) AS Total_Outcomes
FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

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

[19]:
```
%%sql
SELECT DISTINCT Booster_Version
FROM SPACEXTBL
WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTBL);
```

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

[19]:

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

[20]:
```
%%sql
SELECT SUBSTR(Date, 6, 2) AS Month,
       Booster_Version,
       Launch_Site,
       Landing_Outcome
FROM SPACEXTBL
WHERE SUBSTR(Date, 0, 5) = '2015'
AND Landing_Outcome LIKE '%drone ship%';
```

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

[20]:

Month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)
06	F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)

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

1]:
```
%%sql
SELECT Landing_Outcome,
       COUNT(*) AS Count_of_Outcomes
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
AND (Landing_Outcome LIKE '%Failure (drone ship)%' OR Landing_Outcome LIKE '%Success (ground pad)%')
GROUP BY Landing_Outcome
ORDER BY Count_of_Outcomes DESC;
```

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

1]:

Landing_Outcome	Count_of_Outcomes
Failure (drone ship)	5
Success (ground pad)	3

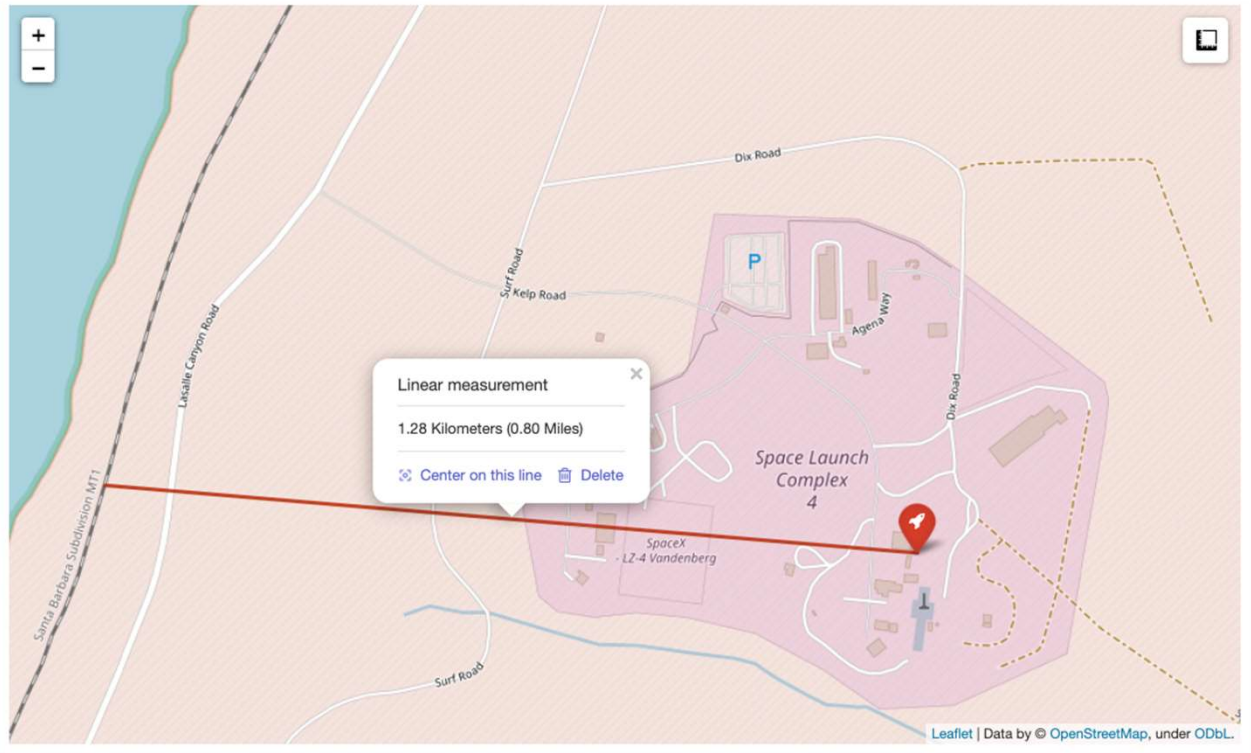
Section 3

Launch Sites Proximities Analysis

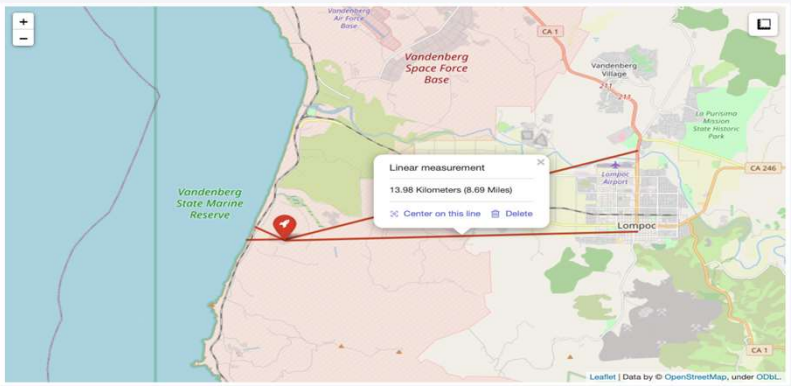
<Folium Map Screenshot 1>

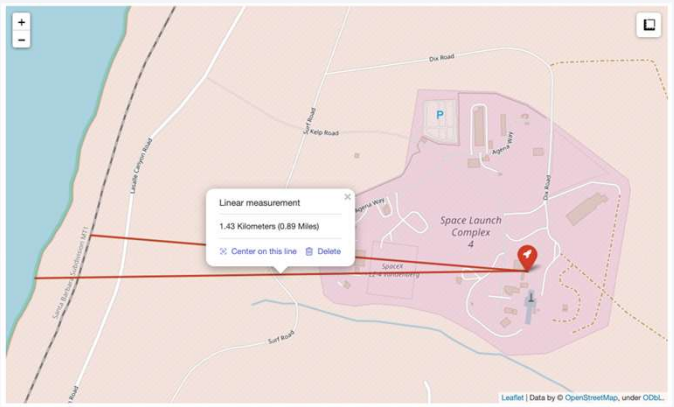
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

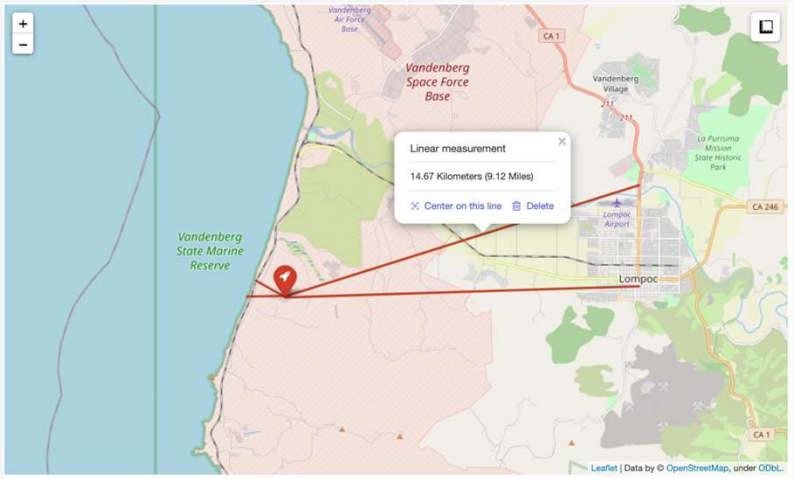
<Folium Map Screenshot 2>

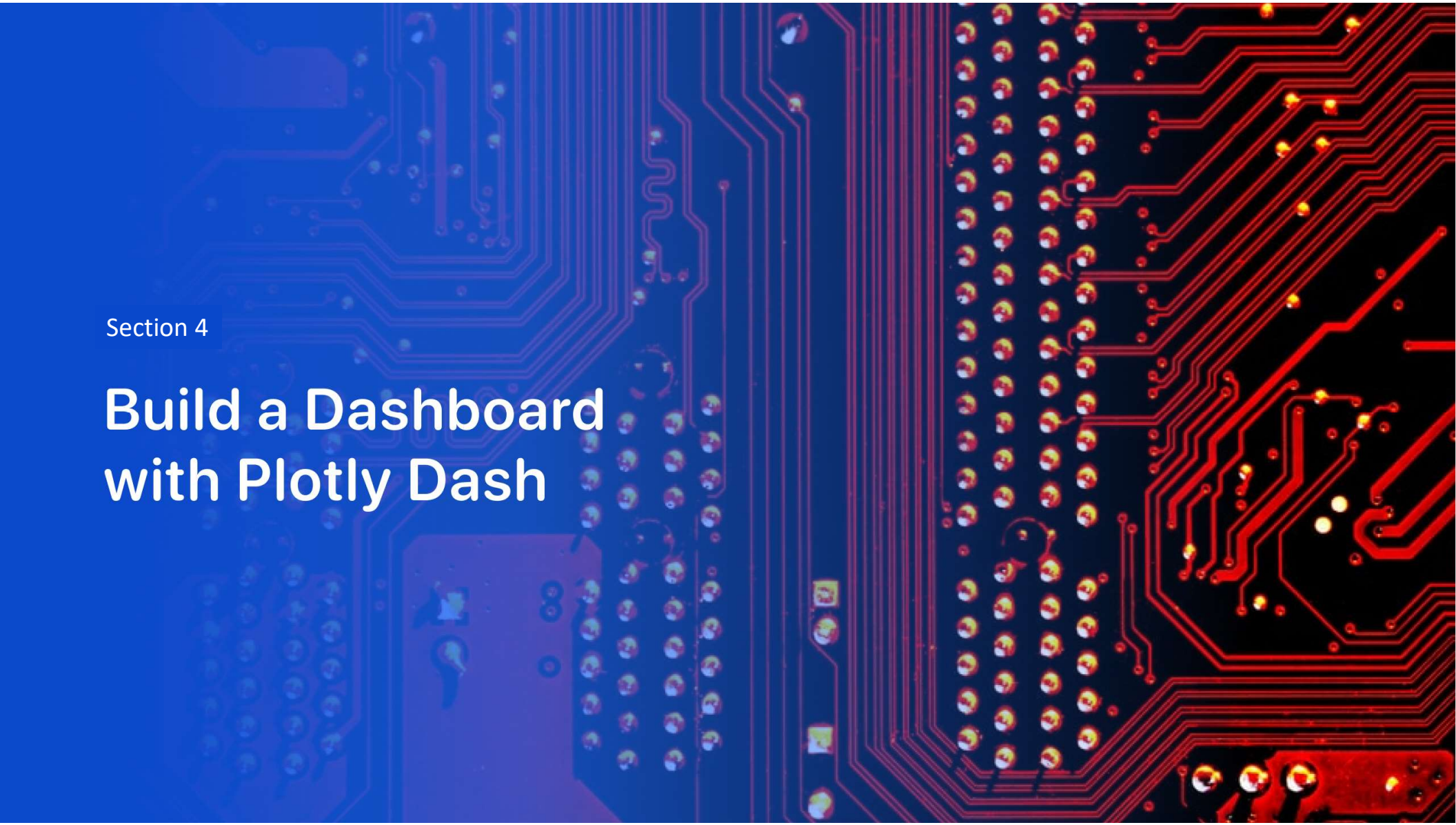


<Folium Map Screenshot 3>





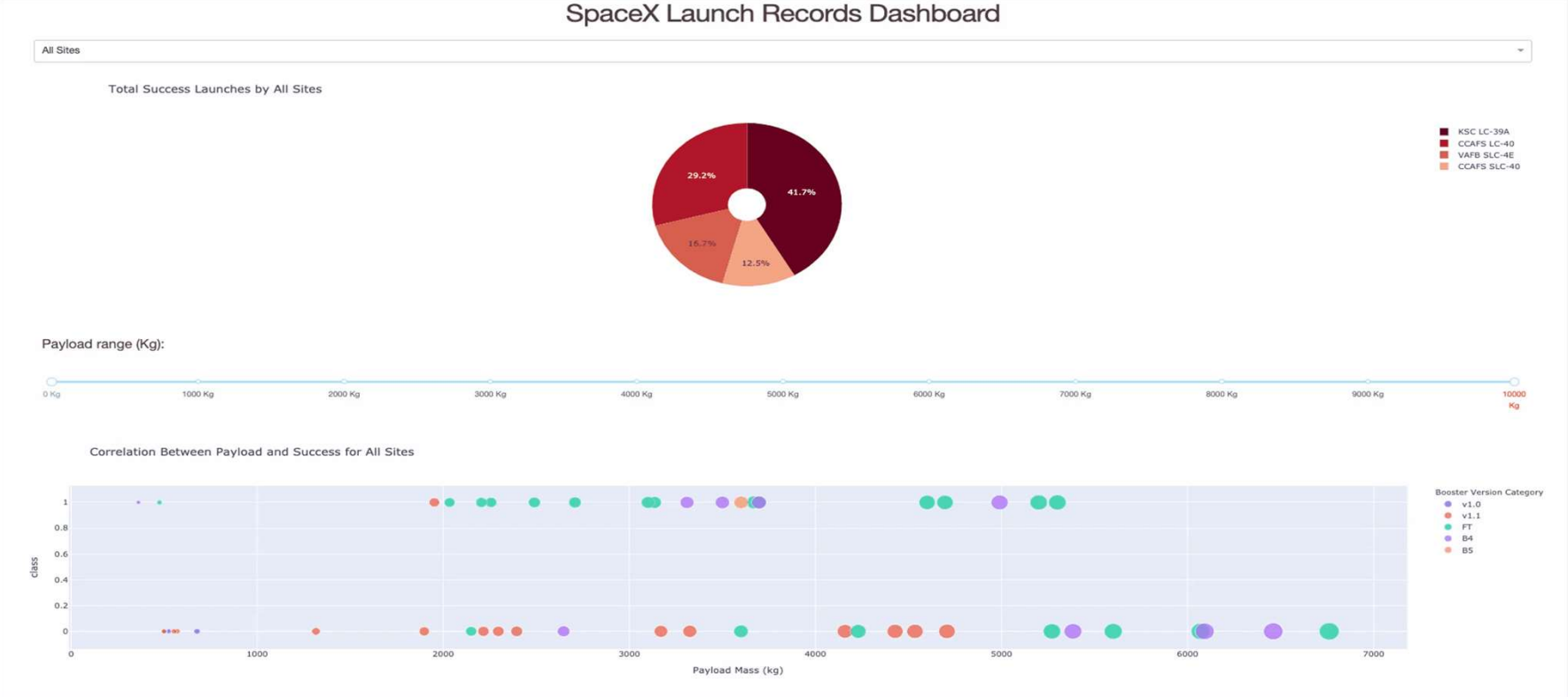




Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



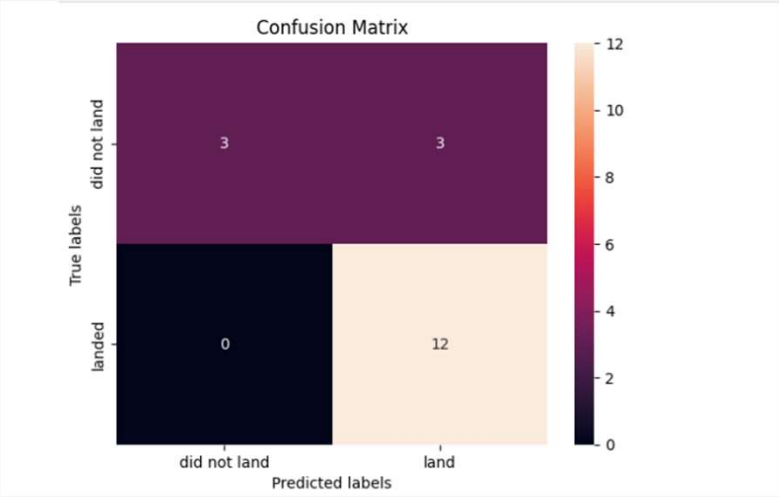
Section 5

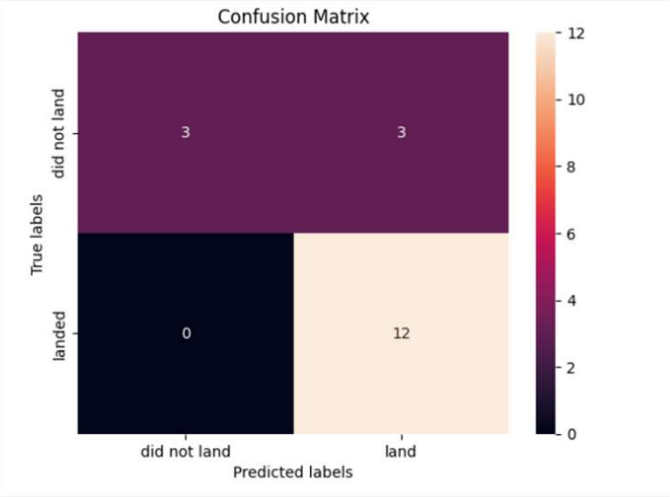
Predictive Analysis (Classification)

Classification Accuracy

Algorithm	Accuracy	Accuracy on Test Data	Tuned Hyperparameters
Logistic Regression	o.846429	0.833334	{‘c’:0.01,’penalty’:’12’,’solver’}
SVM	0.848214	0.833334	{‘c’:1.0,’gamma’:0.0316227766016837 9}
KNN	0.848214	0.833334	{‘algorithm’:’auto’:’ n_neighbors’:10}
Decision Tree	0.901786	0.833334	{‘criterion’:’gini’,’max_depth’:10,’max_features’}

Confusion Matrix





Conclusions

- Orbit ES-L1,GEO,HEO,SSO has highest success rates
- Success rates for Spacex launches has been increasing relatively with time and it looks like soon they will reach the required target
- KSC LC-39A had the most successful launches but increasing payload mass seems to have negative impact on success
- Decision Tree Classifier Algorithm is the best for Machine Learning for provided dataset.

Appendix

- Interactive Plotly
- Folium MeasureControl Plugin Tool
- Folium Custom Title Layers with Lables
- IBM Congo Visualization Tool

Thank you!