IBM Developer SKILLS NETWORK

Winning Space Race with Data Science

Sarbani Gupta

15th May 2023

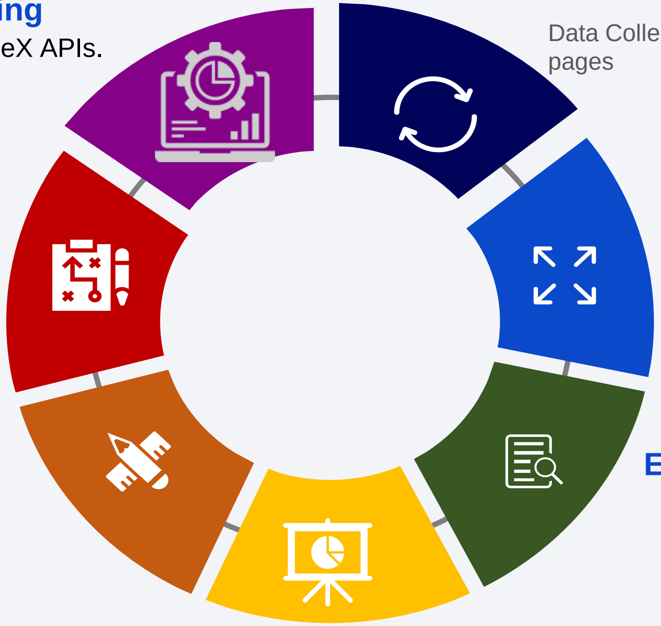
SPACE Y

Outline

SPACE Y

Executive Summary

Summary of methodologies



Data Collection and Preprocessing
Data is collected from SpaceX APIs.

Web Scraping
Data Collection with Web Scrapping the Wiki pages

Data Wrangling
Transforming and Mapping the Data

Exploratory Data Analysis
Data Analysis with SQL

Data Visualization
The scatterplots and bar graphs are extrapolated to represent the relationships of different features.

Interactive Visual Analytics
Interactive maps with Folium
Dashboard creation

Machine Learning
Predictive Analysis for each classification Model.

Summary of all results:

Exploratory Data Analysis result

Interactive Analytics – Screenshots

Predictive Analytics Result

SPACE Y

Introduction

Project background and context:

The aim of the project is to predict that if the Falcon 9 first stage will land successfully.

SpaceX advertises Falcon 9 rocket launch stating that it is about 62% cost effective than other providers, much of the savings of SpaceX is because they can reuse the first stage of their rocket launches. Therefore, our company, SpaceY, wants to analyze and predict if the first stage can land successfully, and use the information to compete with SpaceX for rocket launch.

Problems you want to find answers:

- What are the factors for the rocket's first stage will land successfully?
- The relationship and effect thereof, of each rocket launch variables on the outcome?
- Conditions that will help SpaceY to achieve the best results.

Aim

To predict the landing outcome of first stage of SpaceX's Falcon 9

Reason

If the first stage of the rocket can be reused, it will reduce the cost of launch.

Method

Analysing SpaceX's launching data and creating dashboards and train a machine learning classification model to predict the outcome of a launch.

Section 1

Methodology

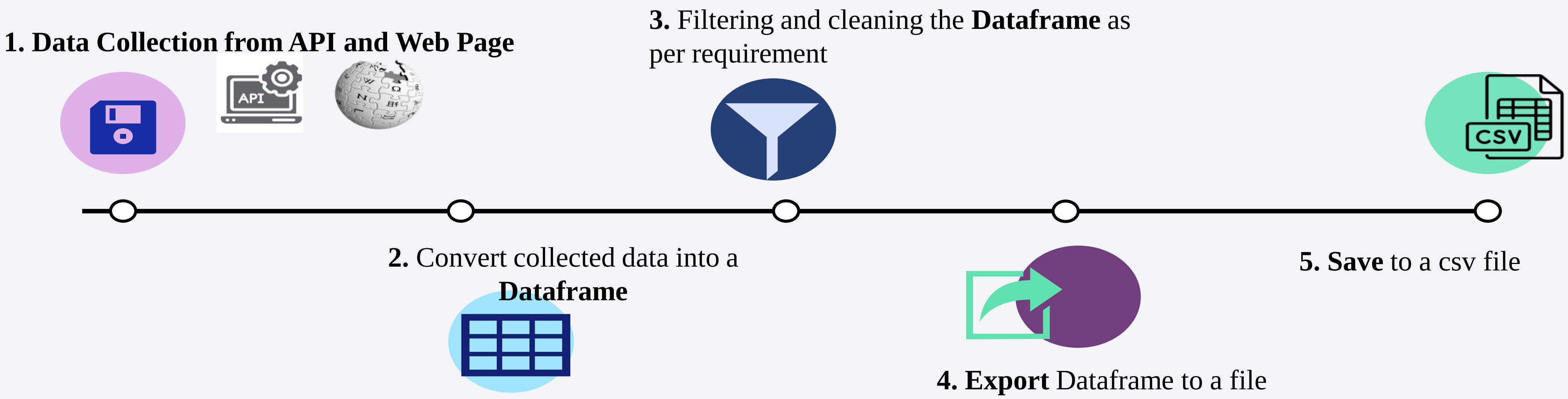
Methodology

- Data collection methodology:
  - The data is collected from open source SpaceX REST API;
  - Web Scrapping the Wikipedia Page of SpaceX
- Perform data wrangling:
  - Transforming data: by one-hot encoding to the categorical features
  - Determining the labels for training the supervised model
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter plot and bar graphs to show relation between the data
- Perform interactive visual analytics
  - Using Folium and Plotly Dash Visualization
- Perform predictive analysis using classification models
  - Build, tune and evaluate the Classification Model by Selecting the best fit model by comparing the following classification algorithms: logistic regression model, support vector machine, Decision Tree classifier, KNN

Data Collection

SPACE Y

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



Data Collection Sources:

1. API

- Gathered data from SpaceX's REST API
- Clean the data

2. Wikipedia web page

- Extracted Falcon 9 launch records from HTML table from Wikipedia page of SpaceX
- Parse the table and convert to Dataframe

Data Collection – SpaceX API

SPACE Y

Getting Response from API

Converting Response to a .json file

Clean the data by applying custom functions

Assign list to dictionary then create dataframe

Filter dataframe and export to a flat-file

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

```
data = pd.json_normalize(response.json())
```

```
data.head()
```

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
launch_data = pd.DataFrame(launch_dict)
```

```
data_falcon9 =launch_data[launch_data ['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

[GitHub link]

Data Collection – Web Scraping

- Getting Response from HTML
- Creating BeautifulSoup Object
- Finding tables
- Getting column names
- Parsing the data, creating dictionary and appending data to keys
- Converting dictionary to dataframe
- Dataframe to .csv file

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
print(response.status code)
```

```
soup = BeautifulSoup(response.text)
soup.title
```

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

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name is not None and len(name)>0:
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)
```

```
df = pd.DataFrame(launch_dict)
df.head()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	[[SpaceX], \n]	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	[[.mw-parser-output .plainlist ol,.mw-parser-o...	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	[[NASA], (, [COTS],)\n]	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	[[NASA], (, [CRS],)\n]	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	[[NASA], (, [CRS],)\n]	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

GitHub link

Data Wrangling

SPACE Y

Data wrangling is the process of cleaning and unifying complex/ incomplete data sets for easy access and analysis.

1. Data loading

2. Create useful **Dataframe** determine the training labels

3. Cleaning the **Data,** handle the missing value

4. Simplifying it to Boolean values

5. Export to a flat-file

6. Save to a csv file

Calculate number of launches at each site

Calculate number of occurrence of each orbit

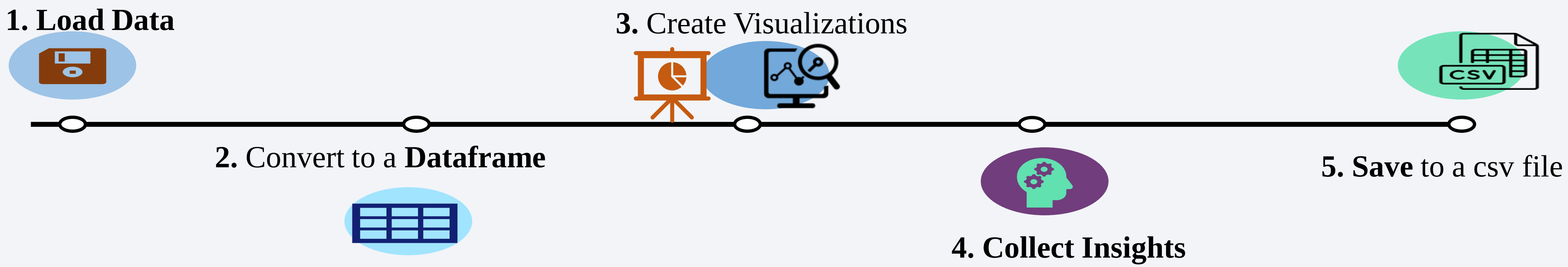
Calculate occurrence of mission outcome per orbit type

Create landing outcome label from Outcome column

GitHub link

EDA with Data Visualization

SPACE Y



Scatter Graphs used in EDA:

- Payload vs Flight Number
- Flight Number vs Launch Site
- Payload vs Launch Site
- Flight Number vs Orbit Type
- Payload vs Orbit Type

Scatterplots show dependency of any two attributes (categorical data) on each other. Once a pattern is determined from the graph, it is easier to predict which factors will lead to maximum probability of success landing and outcome.

Bar Graphs used in EDA:

- Success Rate vs Orbit Type

Bar graph helps to easily interpret the relationship between the attributes and compare several categorical data. Here, bar graph helps to determine which orbits have highest probability of success.

Line Graphs used in EDA:

- Launch Success Yearly Trend

Line graph is best to depict time series data and trends clearly, hence can be used in future predictions.

[GitHub link]

EDA with SQL

SQL is a crucial tool for storing, manipulating and retrieving data from RDBMS which is used in analyzing and extrapolating useful insights from data. Here, IBM’s Db2 for Cloud is used on SpaceX Dataset for Data Analysis.

SQL queries performed to gather information from SpaceX Dataset:

- Displaying the names of the unique launch sites in the space mission.
- Displaying 5 records where launch sites begin with the string ‘CCA’
- Displaying the total payload mass carried by booster version F9v1.1
- Listing the date where the successful landing outcome in ground pad was achieved
- Listing the names of the booster which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes
- Listing the names of booster versions which have carried the maximum payload mass
- Listing the records which displays the month names & failure landing outcomes in drone ships, their booster versions and launch site names for the year 2015
- Ranking the count of landing outcomes, such as Failure (drone ship) or Success (ground pad, between the date 2010-06-04 and 2017-03-20, are in descending order.

GitHub link

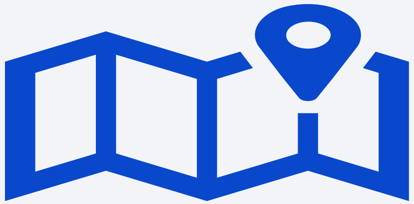
Build an Interactive Map with Folium

SPACE Y

Folium helps to visualize the data on an interactive leaflet map. The latitude and longitude for each launch site is used and a Circle Marker is added around each launch site with a label comprising its name.

Map objects	Code	**Use**
Map Marker	folium.Marker(…)	Map object to make a mark on the map
Icon Marker	folium.Icon(…)	Create an icon on the map
Circle Marker	folium.Circle(…)	Create a circle where the Marker is being placed
PolyLine	folium.PolyLine(..)	Create a line between two points on the map
Marker Cluster Object	MarkerCluster()	A method to simplify a map containing many markers having the same coordinates

- Mark all the launch sites on a map
- Mark each launch site on map with Green and Red for easy visualization of the number of Success and Failure launches.
- Calculate the distances between a launch site to its proximities to-
    - The coastal line
    - The railway
    - To the highway
    - To the city



GitHub link

Build a Dashboard with Plotly Dash

SPACE Y

- **Pie-Chart** shows the total success for all launch site or a distinct selected launch site from the dropdown table.

  The pie-chart helps to visualize the percentage of successful launches in relation to launch sites

- **Scatter Graph** is used to show the correlation between Payload and Success for all sites or a particular selected launch site

  We can use the range slider on the payload attribute to select the payload mass

  It shows the relationship between Success rate and Booster Version category

Dashboard objects	Code	Function
Dashboard and some of its components	*import* dash *import* dash_html_components as html *import* dash_core_components as dcc *from* dash.dependencies import Input, Output	With the Dash Open Source, dash apps run on the local laptop or server. The Dash Core component library contains a set of higher-level components like sliders, dropdown, graphs etc.
Dropdown	dcc.Dropdown(…)	Create a dropdown list for launch sites
RangeSlider	dcc.RangeSlider(…)	Create a range slider for payload mass for selection
Pei-Chart	px.pie(…)	Create a pie-chart for displaying Success percentage of lunches
Scatter Chart	px.scatter(…)	Create a scatter plot for displaying the correlation between payload mass and

GitHub link

Predictive Analysis (Classification)

SPACE Y

Building Machine Learning Model

Data Loading & Preprocessing

- Load our feature engineered data into dataframe
- Transform it into NumPy arrays
- Standardize and transform data

Machine Learning and Training model

- List the machine learning algorithms we want to use
- Set parameters and algorithms to GridSearchCV
- Fit our dataset and train out model

Split data to training/test sets

- Split the dataset into training and test sets
- Check how many test samples are created

Classification Algorithms

- Logistic Regression
- Support Vector Machine
- Decision Tree Classifier
- K-Nearest Neighbor

Evaluating Model

- Check accuracy of each model
- Get best hyperparameters for each algorithm
- Plot confusion matrix

Find Best performing Classification model

- The model with best accuracy score is the best performing model

Evaluating the Model

Best Model

- The Decision tree is the best performing model

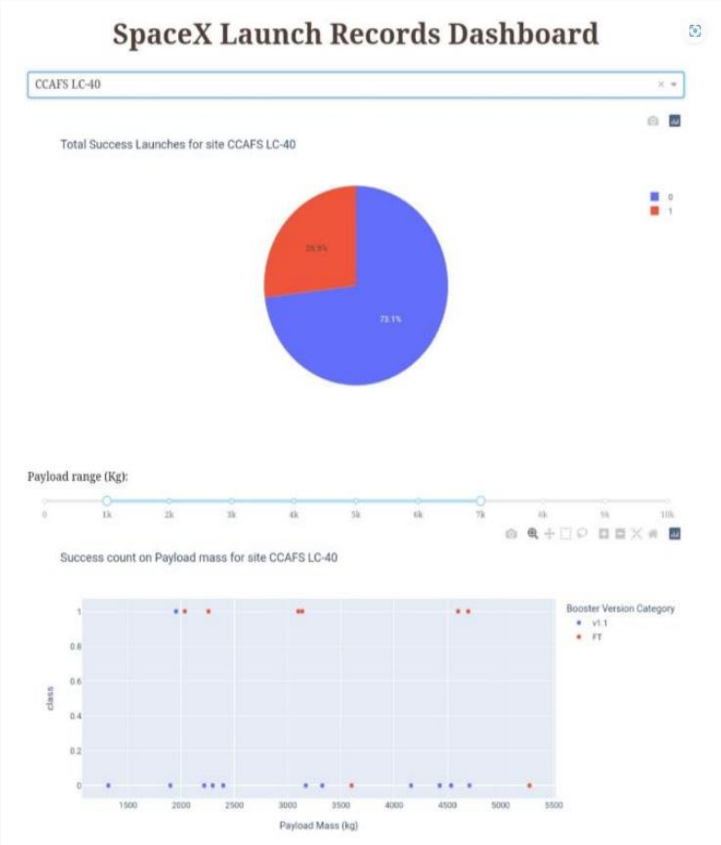
GitHub link

Results

Exploratory Data Analysis

- KSC LC-39A and VBFB SLC 4E has a success rate of 77%
- VAFB SLC 4E has no payload above 10000 kg
- In LEO orbit the Success rate is directly related to the Number of flights
- The successful/positive landing rate for Polar, LEO & ISS are more with heavy payloads
- The overall success rate of launch has kept increasing since 2013 till 2020

Interactive Analytics- Screenshot



Predictive Analysis

The Decision Tree Classifier model is the best performing classification Model.
The accuracy of the model is 88.75% with the best hyperparameters:

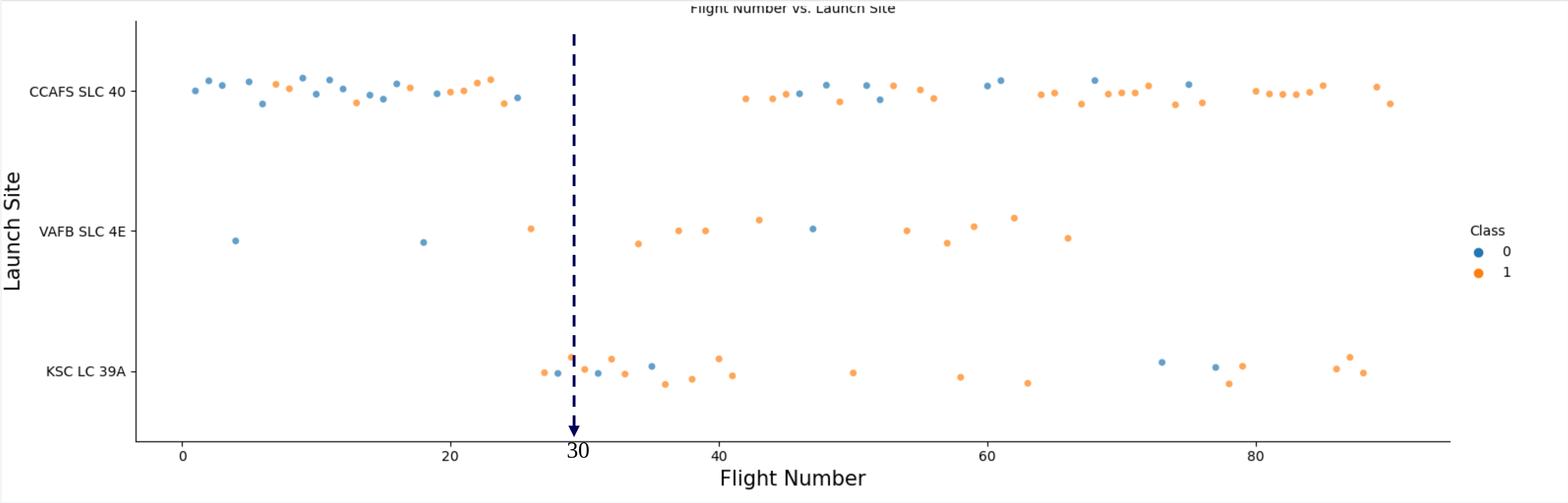
{'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- The Flight Number vs Launch Site plot depicts that with more number of flights (greater than 30) the success rate for Booster landing increases.
- KSC LC 39A has highest success rate, while CCAFS SLC 40 has the lowest

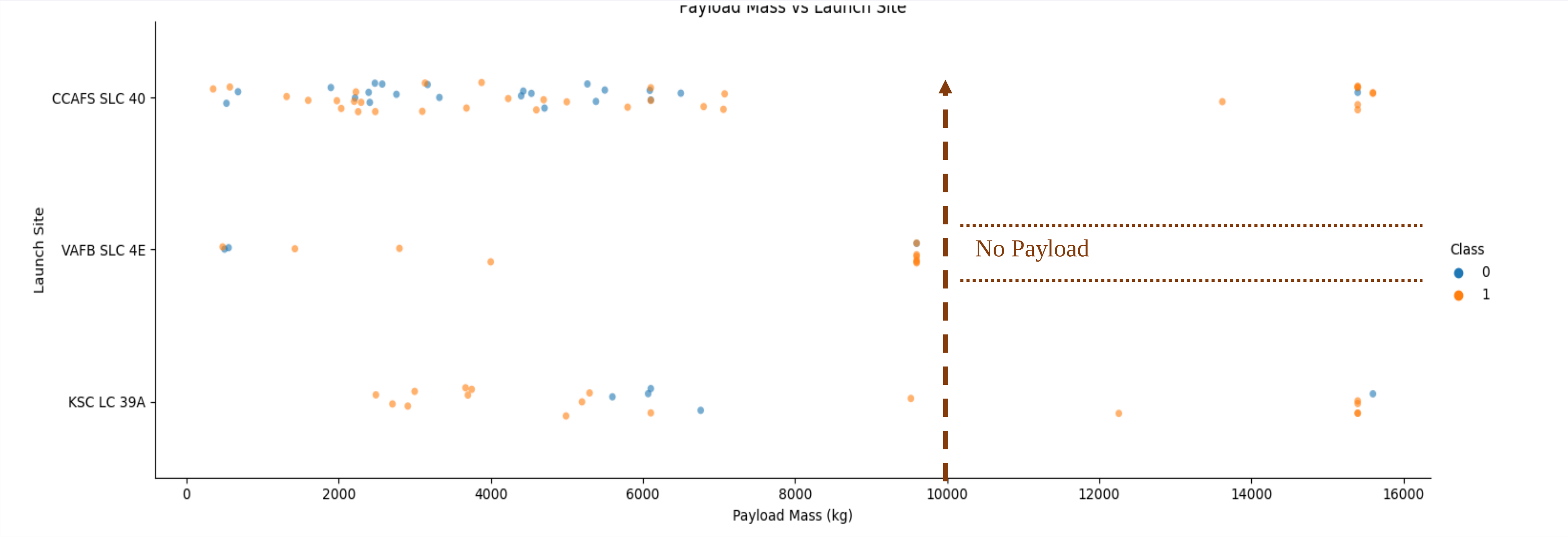


Class 1= Booster landing successful
Class 0 = Unsuccessful

Payload vs. Launch Site

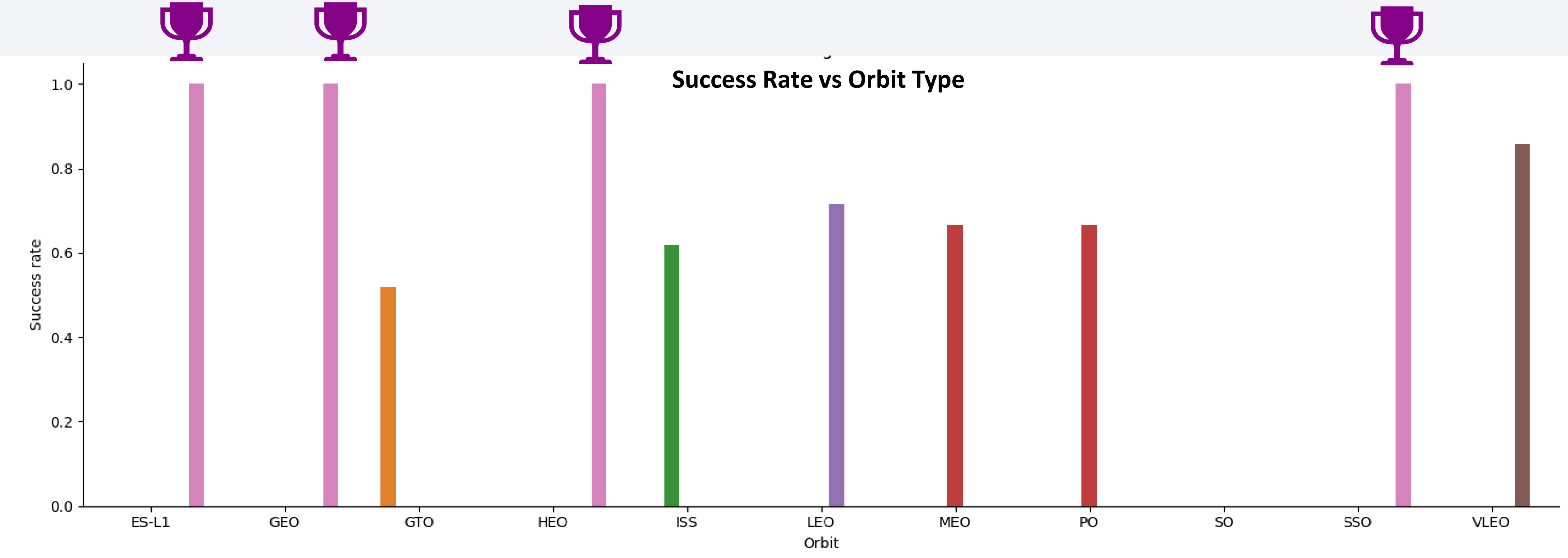
The Payload vs Launch Site graph shows that VAFB SLC 40 has no payload above 10000 kg.

But, there is no clear pattern to infer that whether the choice launch site is dependent on the payload for a successful launch.



Success Rate vs. Orbit Type

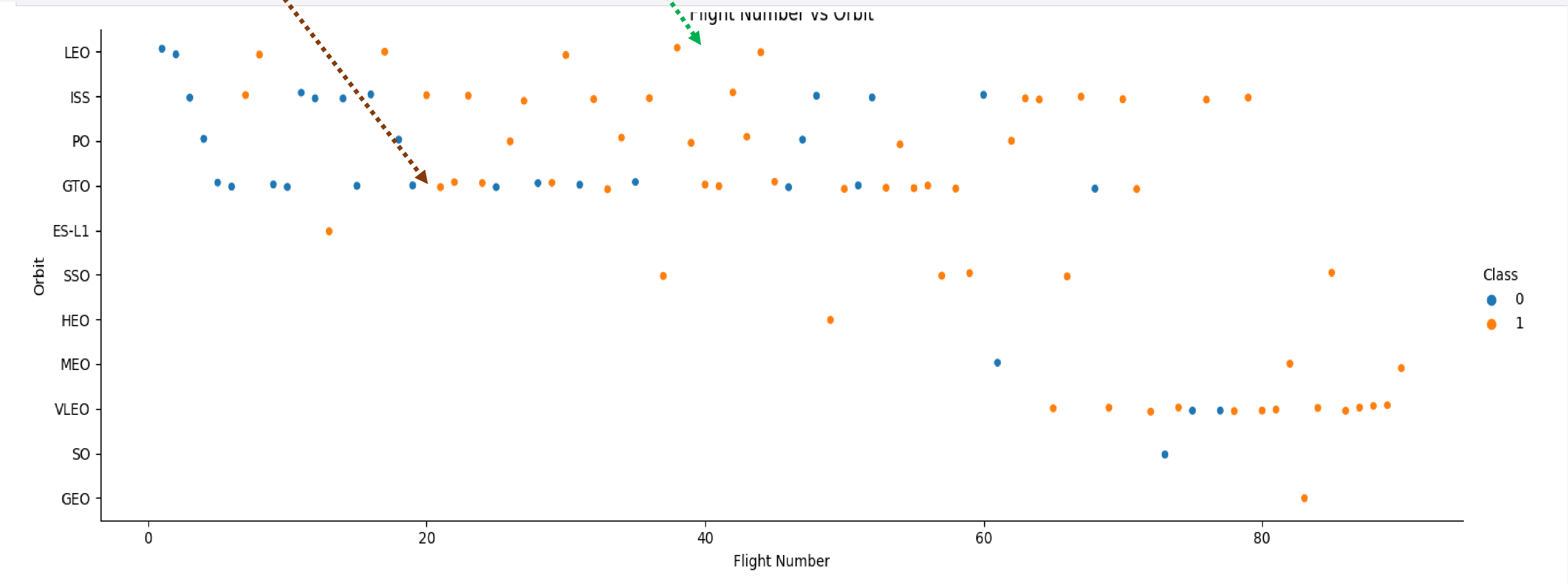
The Success Rate vs Orbit Type graph represents that the orbit type:- **ES-L1, GEO, HEO & SSO** have the highest success rate



= Highest Success rate

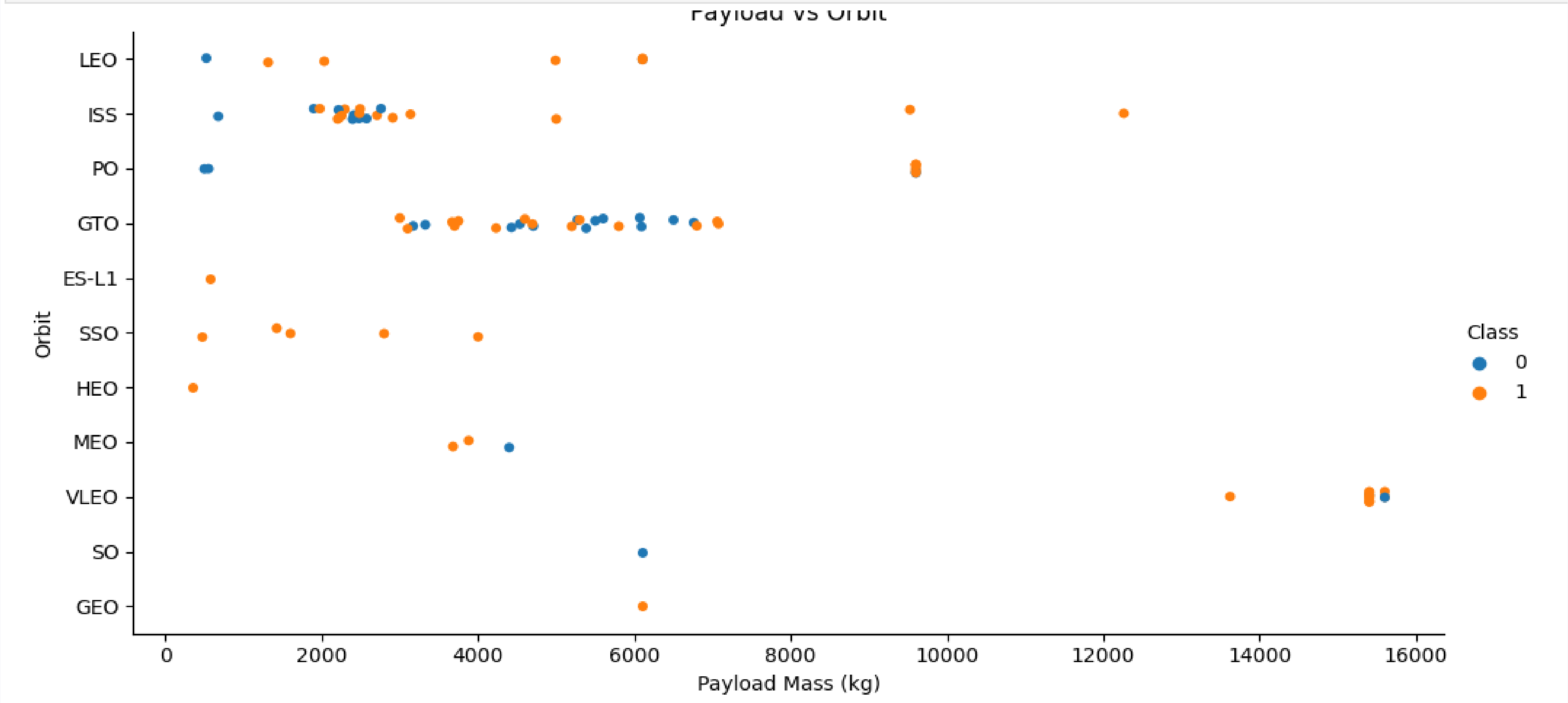
Flight Number vs. Orbit Type

- The scatter plot shows that in the **LEO orbit** the Success increases with the number of flights
- But, in the **GTO orbit,** there seems to be no relation between the flight number and success



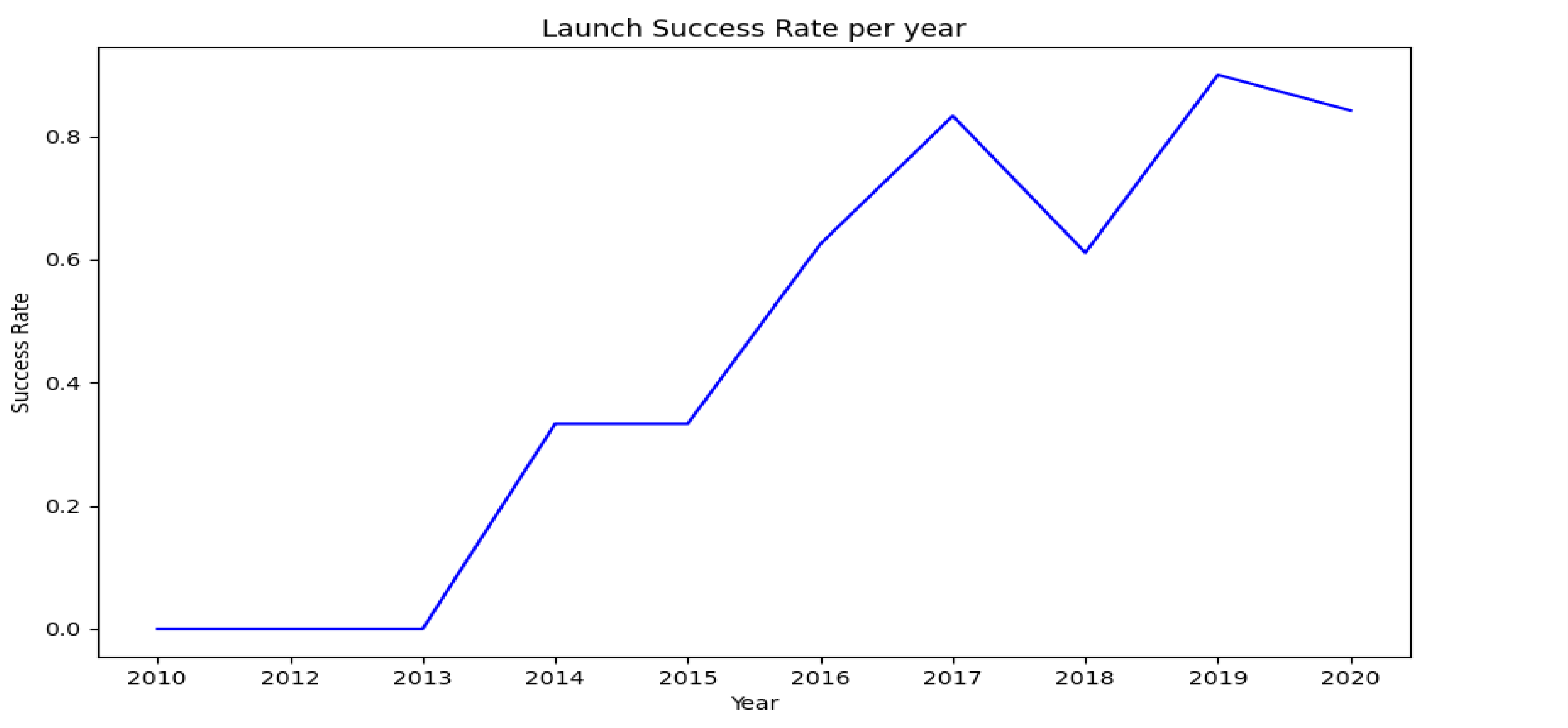
Payload vs. Orbit Type

- The payload vs. orbit type graph depicts that the heavy payloads have a positive landing rate (successful landing) with orbit type **LEO & ISS, PO**
- On the other hand, the heavy payloads have a negative influence on MEO, GTO, VLEO orbits



Launch Success Yearly Trend

The line graph shows the launch success rate drastically increases in 2013, and keeps on increasing till 2020, with a slight dip in year 2018.



All Launch Site Names

Find the names of the unique launch sites

SQL Query

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

Description

Using 'DISTINCT' in the query we pull unique values for Launch_Site column from the table SPACEX

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

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

<u>SQL Query</u>

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

<u>Description</u>

Using the keyword ‘LIMIT 5’ in the query we can fetch 5 records from the table SPACEX with a condition LIKE keyword with wildcard-’ CCA%’. The ‘%’ sign represents that the Launch_Site name must start with CCA

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

SPACE Y

Calculate the total payload carried by boosters from NASA (CRS)

SQL Query

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

Description

Using the function SUM, we can calculate the total in the column PAYLOAD_MASS_KG__ and the WHERE clause filters the data to fetch the Customer by name ‘NASA (CRS)’

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

- **SQL Query**

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

Description
Using the function AVG calculates the average in the column PAYLOAD_MASS_KG__
The WHERE clause filters the dataset to only perform the calculations on Booster_Version is equal to ‘F9 v1.1’

Avg_Payload
2928.4

First Successful Ground Landing Date

SPACE Y

Find the dates of the first successful landing outcome on ground pad

<u>SQL Query</u>

```
%sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success (ground pad)';
```

<u>Description</u>

Using the function MIN calculates the minimum date in the column Date and the WHERE clause filters the data to only perform calculations on Landing _Outcome with values ‘Success (ground pad)’

First_Successful_Landing_Date
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

SPACE Y

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

SQL Query

```
%sql SELECT DISTINCT "Booster_Version", "Landing _Outcome", PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "Landing _Outcome" ='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Description

Selecting only the Booster_Version,
WHERE clause filters the dataset to Landing _Outcome = Success (drone ship)

AND clause specifies additional filter conditions and BETWEEN gives the range value
PAYLOAD_MASS_KG__ BETWEEN 4000 AND 6000

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

SPACE Y

Calculate the total number of successful and failure mission outcomes

SQL Query

```
%sql SELECT Mission_Outcome, Count(*) AS Number_of_Missions FROM SPACEXTBL GROUP BY Mission_Outcome;
```

Description

Selecting the Mission_Outcome column and Count(*) clause gives the counts of all the Mission_Outcome , the GROUPBY statement groups the rows together which have same values from table SPACEX

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

Boosters Carried Maximum Payload

SPACE Y

List the names of the booster which have carried the maximum payload mass

SQL Query

```
%sql SELECT Booster_Version, Max_Payload FROM (SELECT Booster_version, MAX(PAYLOAD_MASS__KG_) AS Max_Payload FROM SPACEXTBL GROUP BY Booster_Version) AS Sub;
```

Booster_version	Max_Payload
F9 B4 B1039.2	2647
F9 B4 B1040.2	5384
F9 B4 B1041.2	9600
F9 B4 B1043.2	6460
F9 B4 B1039.1	3310
F9 B4 B1040.1	4990
F9 B4 B1041.1	9600
F9 B4 B1042.1	3500
F9 B4 B1043.1	5000
F9 B4 B1044	6092
F9 B4 B1045.1	362
F9 B4 B1045.2	2697
F9 B5 B1046.1	3600
F9 B5 B1046.2	5800
F9 B5 B1046.3	4000
F9 B5 B1046.4	12050
F9 B5 B1047.2	5300
F9 B5 B1047.3	6500

Description

Using the function MAX gives the maximum payload in the column PAYLOAD_MASS_KG__ in sub query

2015 Launch Records

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

SQL Query

```
%sql SELECT SUBSTR(Date,4,2) AS Month, Booster_Version, Launch_Site FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Failure%' AND SUBSTR(Date,7,4)='2015';
```

Description

Using substr(Date,4,2) function to get the value of month and substr(Date,7,4)='2015' to get the value of year i.e. 2015, we can list out failed landing_outcomes in drone ship with their specific booster-version

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

SQL Query

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

Description

Selecting only Landing _Outcome,
WHERE clause filters the data with date BETWEEN ’04-06-2010’ AND ’20-03-2017’
Group by Landing _Outcome
Order by Count of (Number_of_successful_landings) in Descending Order

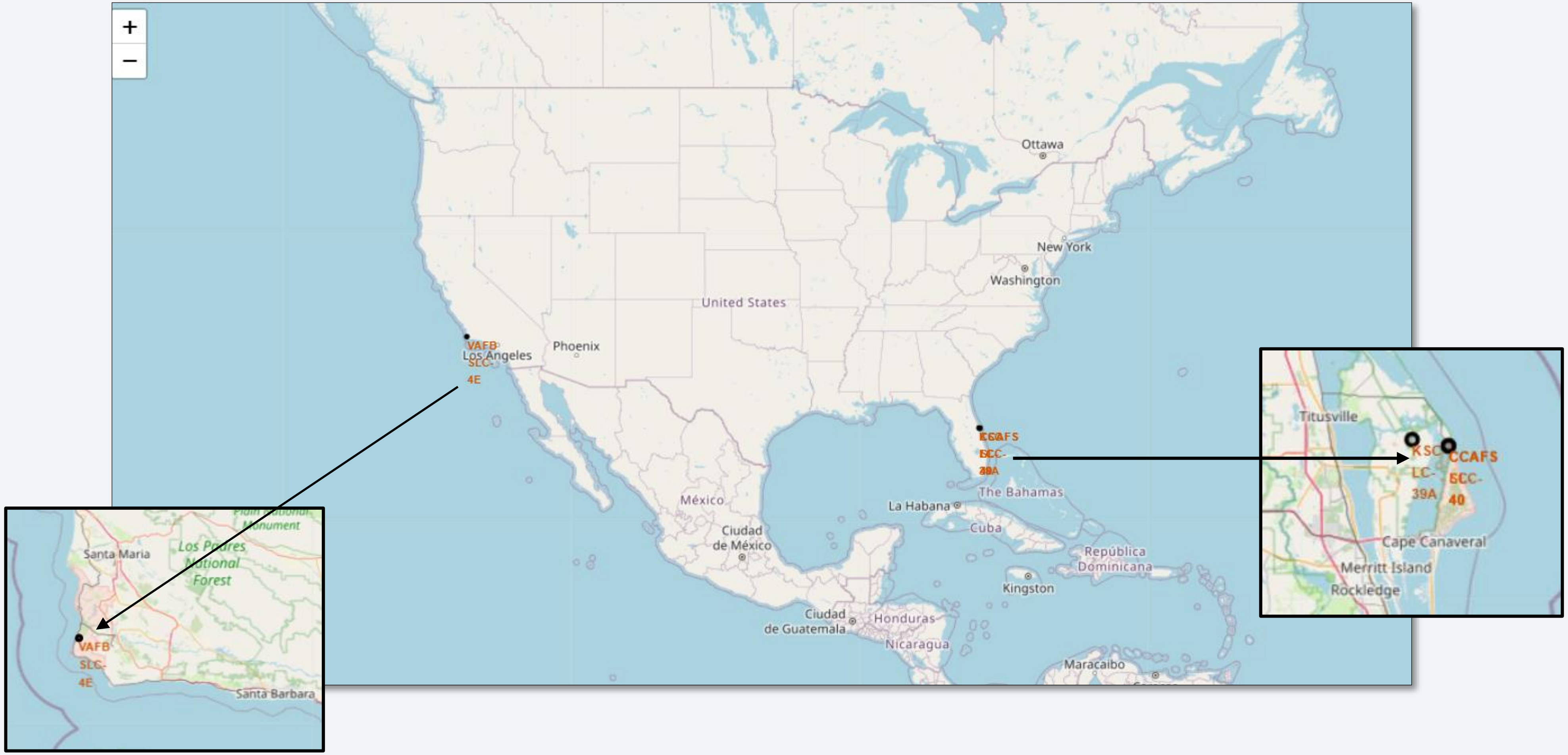
Landing _Outcome	Number_of_successful_landings
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

Launch Sites Proximities Analysis

All Launch Sites on Folium Map

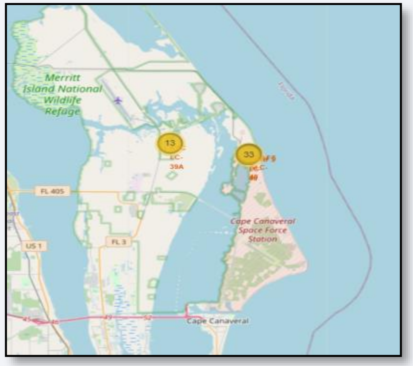
The Folium map shows that the different SpaceX launch sites are in the coastal regions of U.S.A. viz. Florida and California states; and none of the launch site is in the proximity of Equator line.



Launch Outcomes on Map

SPACE Y

The screenshots of these folium maps depicts the colour labelled outcomes of various launches. Green Marker shows successful launches and the Red Marker shows failure.



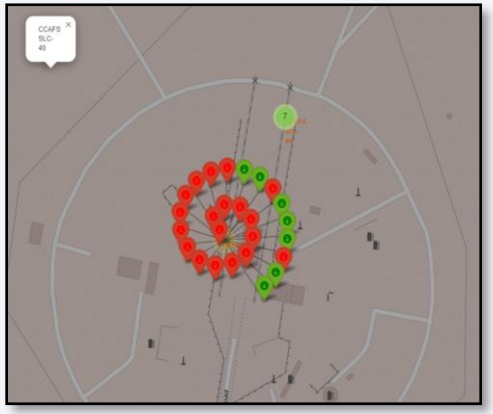
The following colour-labelled marker maps shows in a simplified way that **KSC LC-39A is the launch site which has highest success rate** among the 4 launch sites**.**

- VAFB SLC-4E has a total of 10 launch trails.
- CCAFS LC-40 has 26 launches, while CCAFS SLC-40 has 7 trails
- KSC LC-39A has a total of 13 launch trails

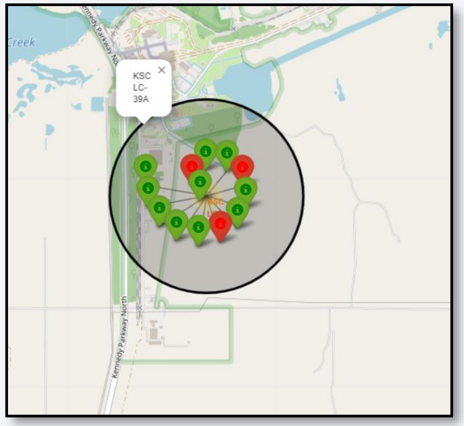
VAFB SLC-4E

CCAFS SLC-40

CCAFS LC-40



KSC LC-39A

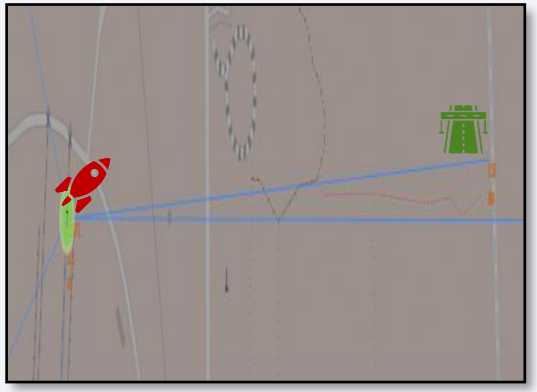


Proximity of the Launch Sites

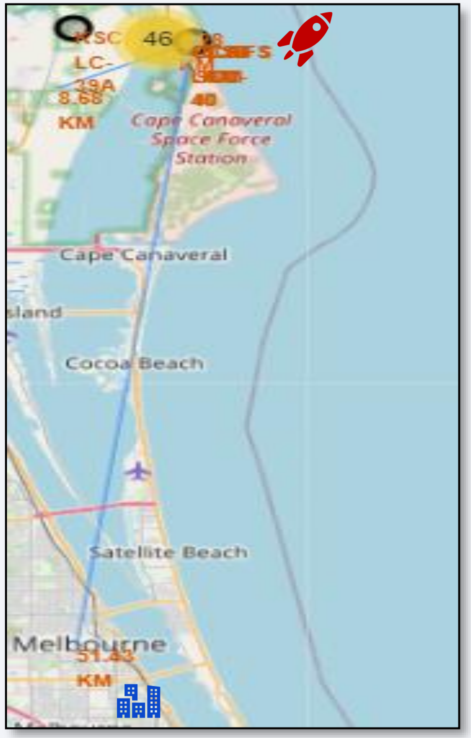
The following maps represents the distance of CCAFS SLC-40 launch site to its proximities- railway, highway, coastline.

Distance to coastline

Distance to Parking-way

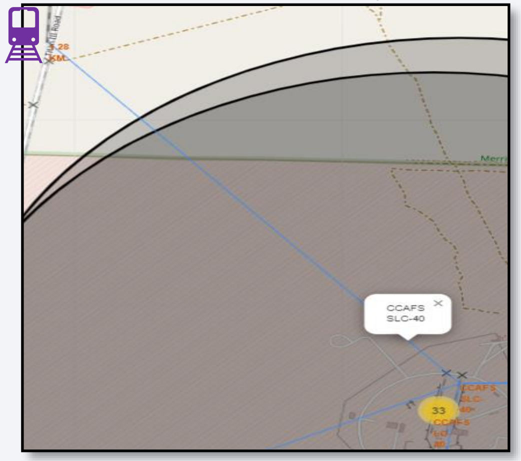


Distance to city



Distance to Highway

Distance to Railway



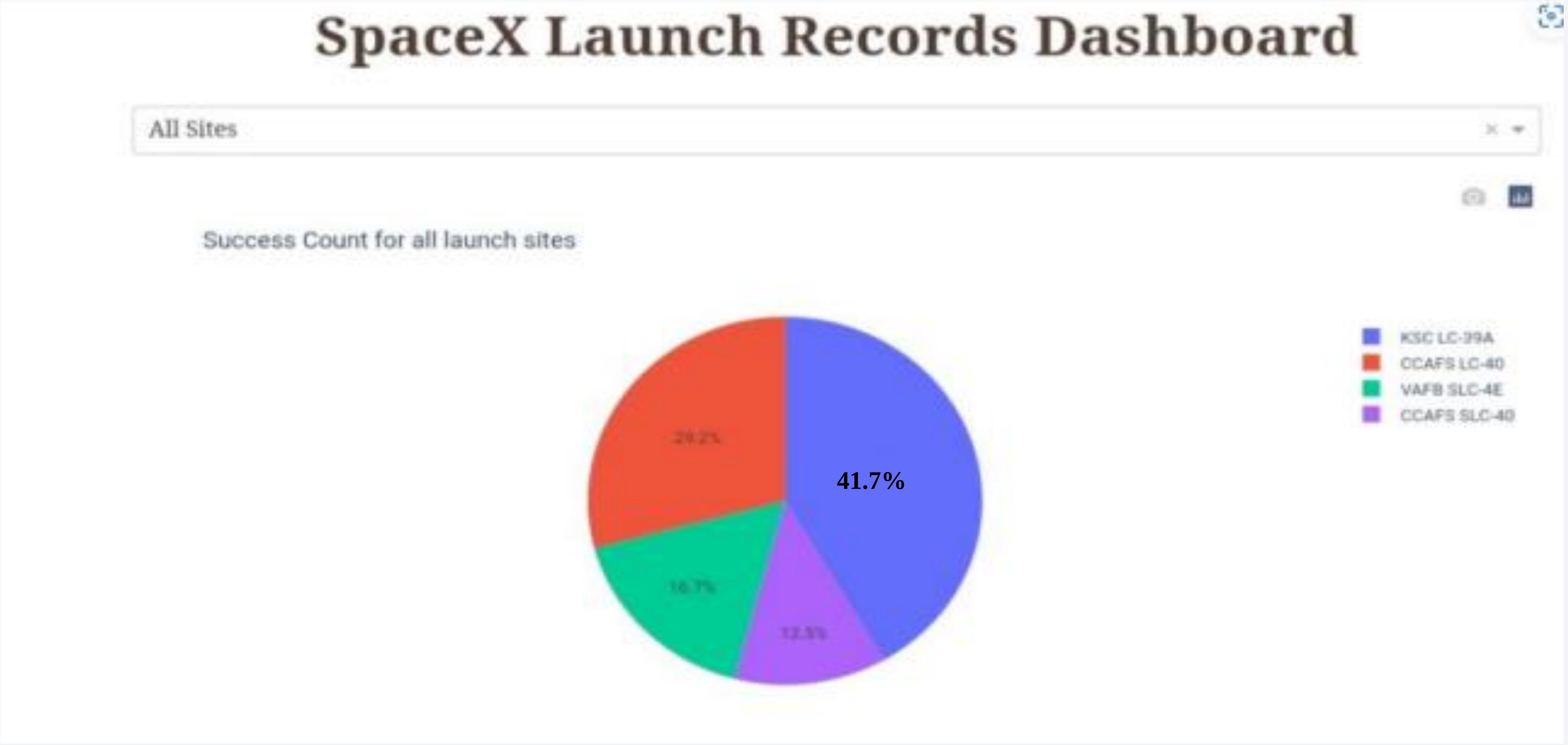
Conclusion

- Are all launch sites in proximity to the equator line?
  No (4000 km > distance > 3000 km)
- Are launch sites in close proximity to coastline?
  Yes (CCAFS SLC-40 is 0.88 km far from coastline. All the launch sites are in close distance from coastline, 5 km > distance 0.5 km)
- Are launch sites in close proximity to highways?
  No, although the closest parkway is 0.58 km away, the highway is 8.04 km away from the launch site. (15 km > distance > 5 km)
- Are launch sites in close proximity to railways?
  Yes, distance of nearest railway is 1.28 km from the launch site. All launch sites are relatively close to railway (1.5 km >distance > 0.5 km)
- Do launch sites keep certain distance away from cities?
  Yes, closest city from CCAFS SLC-40 site is Melbourne 51.4 km away.

Section 4

Build a Dashboard with Plotly Dash

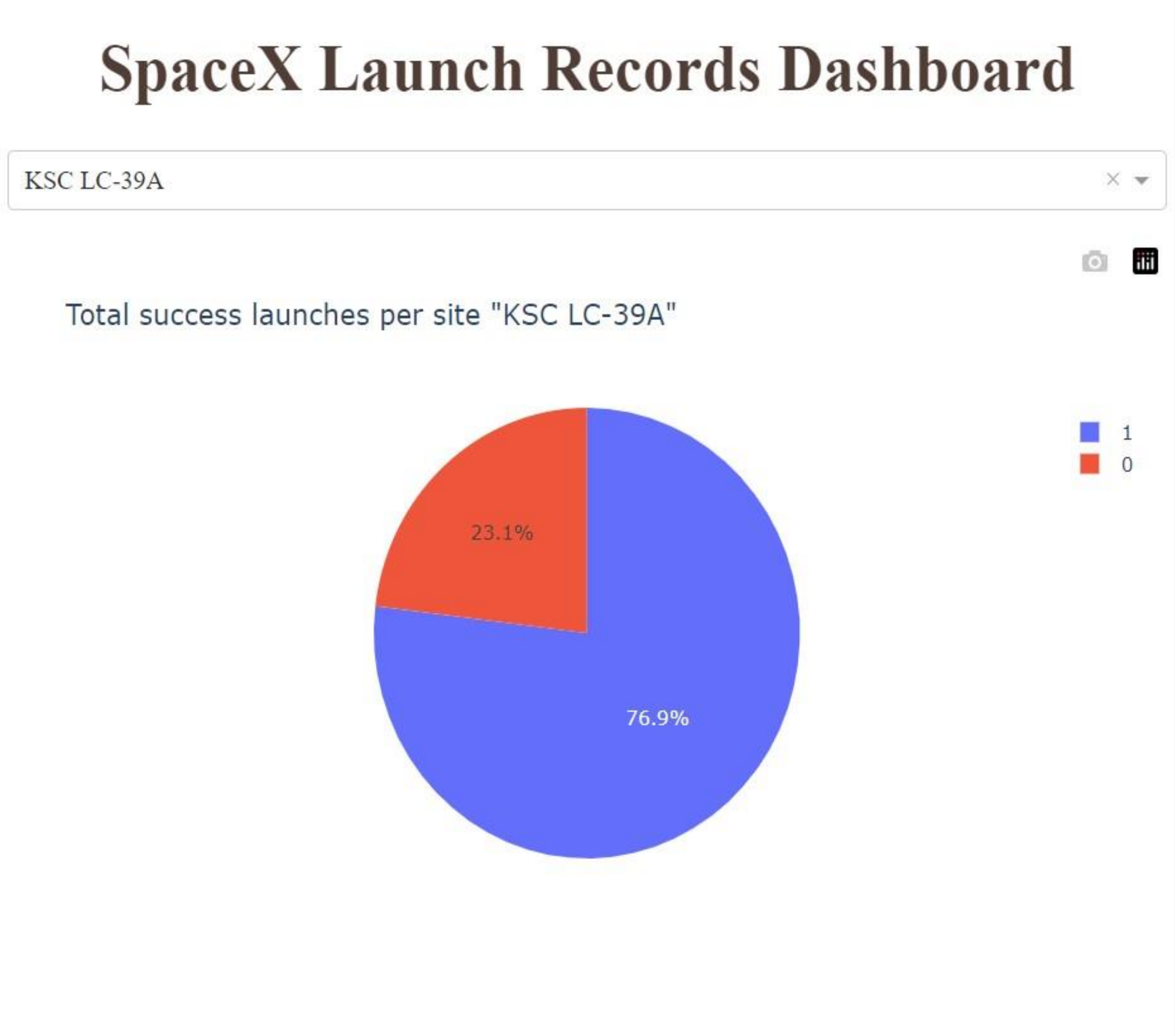
Launch Success Count for All Sites



KSC LC-39A : 41.7%
CCAFS LC-40 :29.2%
VAFB SLC-4E :16.7%
CCAFS SLC-40 :12.5%

The pie-chart shows that KSC LC-39A has the most successful launches from all the sites.

Highest Launch Success Ratio



KSC LC-39A launch site has a 76.9% success rate and only a 23.1% failure rate.

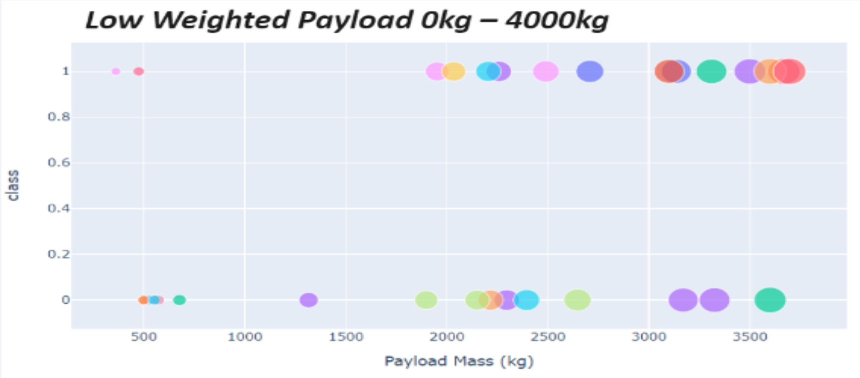
Insights from Visual Analysis using the Dashboard:

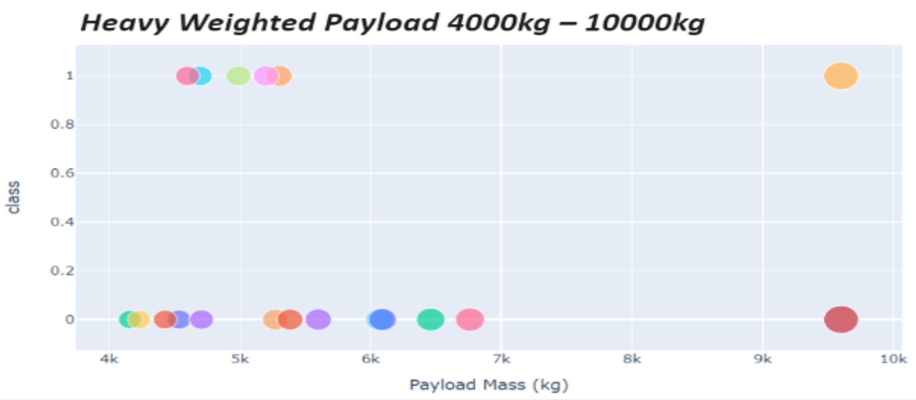
- Which site had the highest launch success rate?
  KSC LC-39A
- Which payload range(s) has the highest launch success rate?
  2000 kg – 10000 kg
- Which payload range(s) has the lowest launch success rate?
  0 kg – 1000 kg
- Which F9 Booster Version has the highest launch success rate?
  FT

Payload vs Launch Outcome Scatter Plot

It is observed that the success rates for low weighted payloads is higher than the heavy weighed payloads

Payload vs Success count on all sites with slide ranger





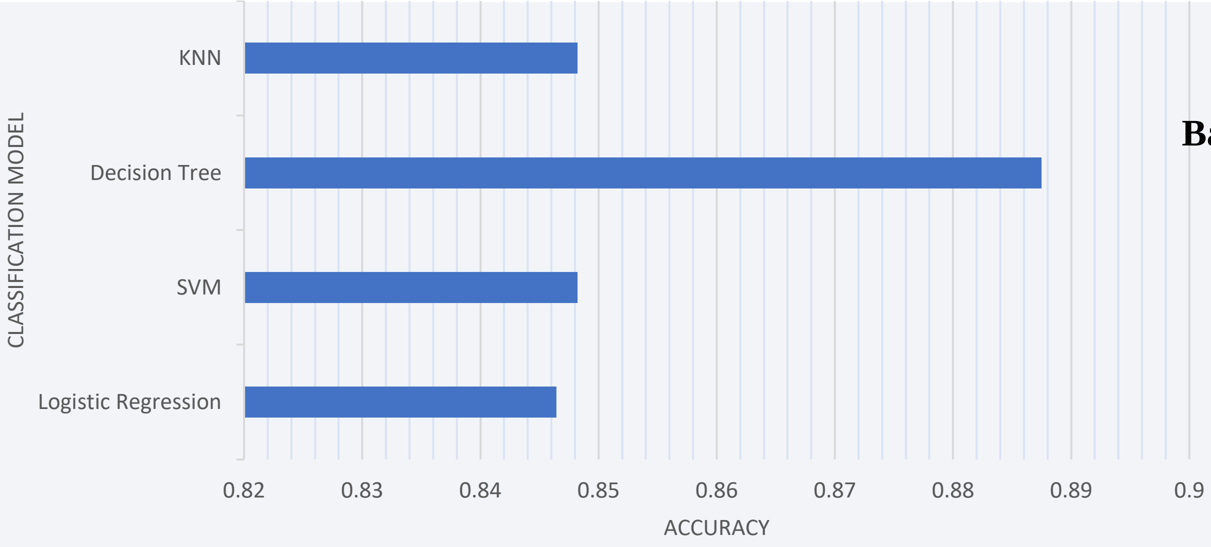
Section 5

Predictive Analysis (Classification)

Classification Accuracy

The best performing classification model is **“Decision Tree”** with a score of **0.8875**

Algorithm	Accuracy	Accuracy of Test Data	Tuned Hyperparameters
Logistic Regression	0.8464285714285713	0.83333333334	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM	0.8482142857142856	0.83333333334	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
Decision Tree	0.8875	0.83333333334	{'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
KNN	0.8482142857142858	0.83333333334	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}



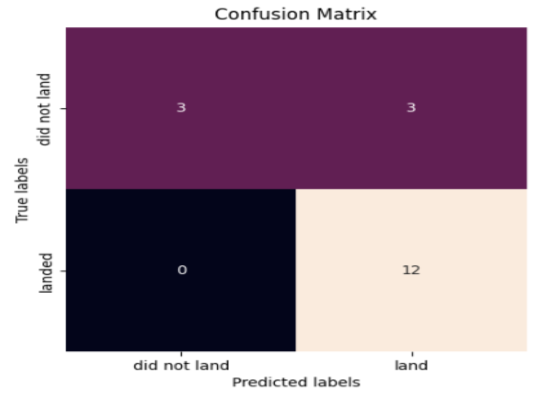
Bar Graph – Classification Accuracy of 4 Different Models

Confusion Matrix

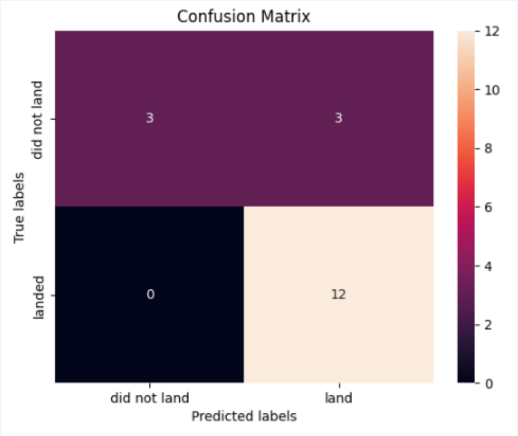
SPACE Y

Determining the best performing model on the basis of Test Dataset Accuracy is unfortunately not possible since all the 4 classification models have the same confusion matrix.

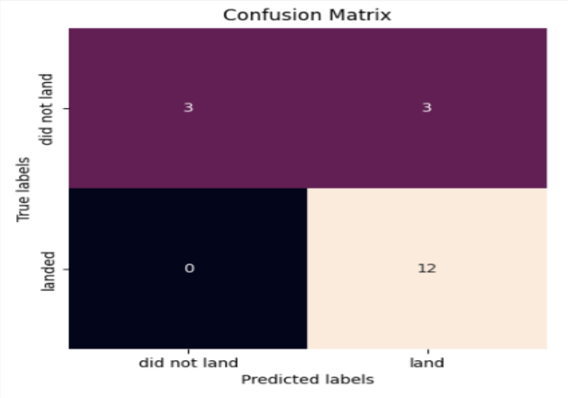
Logistic Regression



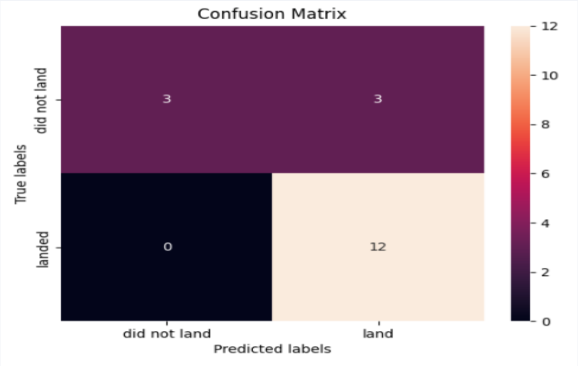
SVM



Decision Tree



Logistic Regression



Predicted Values

Actual Values			
	True Negative TN = 3	**False Positive FP = 3**	Actual No **6**
	False Negative FN = 0	**True Positive TP = 12**	Actual Yes **12**
	Predicted No **3**	Predicted Yes **15**	Total Cases = **18**

Accuracy: (TP+TN)/Total = (12+3)/18 = 0.833334
True Positive Rate: TP/Actual Yes = 12/12= 1
False Positive Rate: FP/Actual No = 3/6 = 0.5
True Negative Rate: TN/Actual No = 3/6 = 0.5
Misclassification Rate: (FP+FN)/Total = (3+0)/18 = 0.1667
Precision: TP/Predicted Yes = 12/15 = 0.8
Prevalence: Actual Yes/Total =12/18 =0.6667

Conclusions

SPACE Y

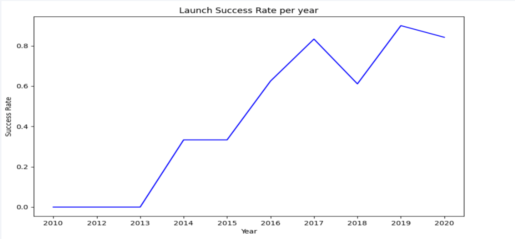
1. The higher the number of flights at a launch site, greater is the success rate.

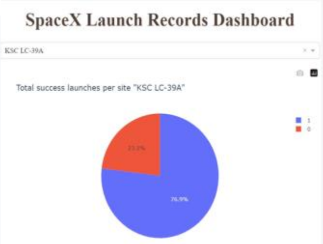
2. Orbits ES-L1, GEO, HEO, SSO have the highest Success rate

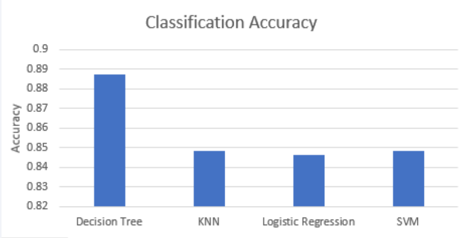
3. Success rate of SpaceX launches have shown a positive trend and has been increasing since 2013.

4. KSC LC-39A had the most successful launches but increasing the payload mass seems to have a negative impact on the success rate

5. Decision Tree Classifier is the best performing Machine Learning Model with a score of 0.8875







Appendix

The GitHub link for the IBM Capstone Final Project SpaceY– [GitHub]

IBM Watson cloud link for the Capstone Final project link- [IBM Watson]

Reference book -Getting started with Data Science- Murtaza Haider

Coursera Skills Laboratory

Thank you!