

# Data Science Capstone Project

Sireesha Vemula

26<sup>th</sup> September 2024



# Outline



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

Debdatta Sarkar



# Executive Summary

---

Flight Number vs. Payload Mass

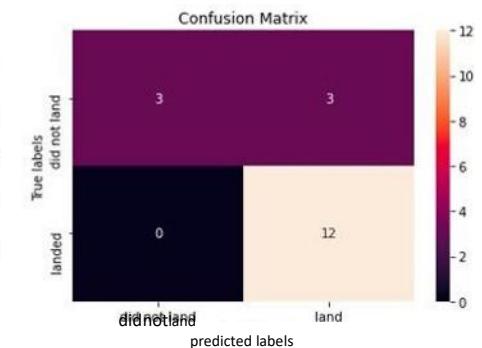
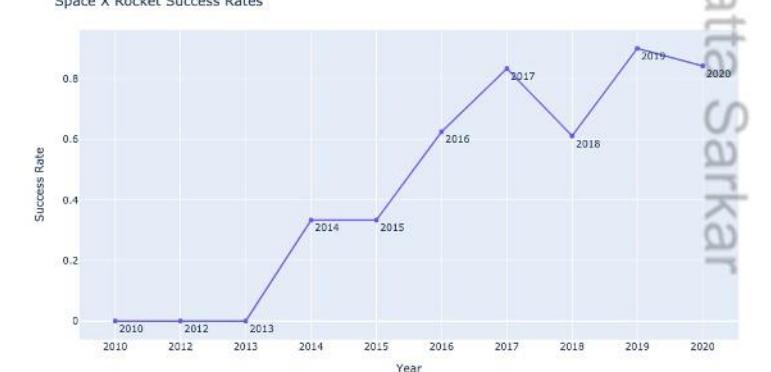
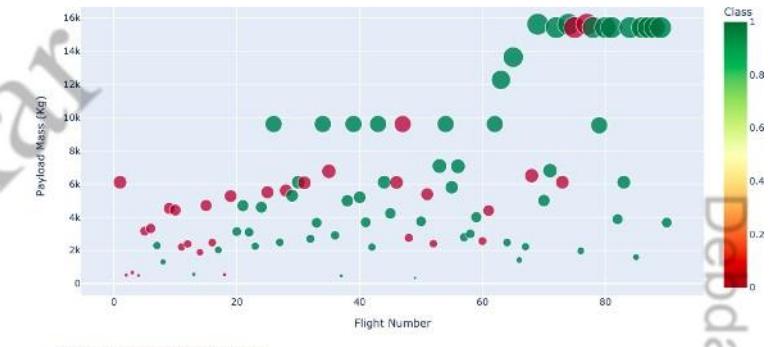
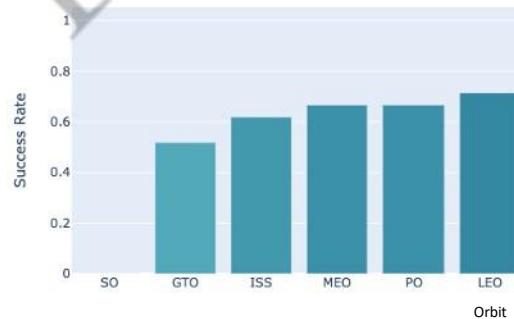
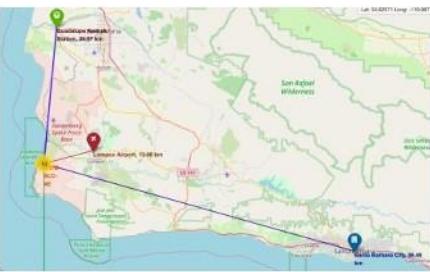
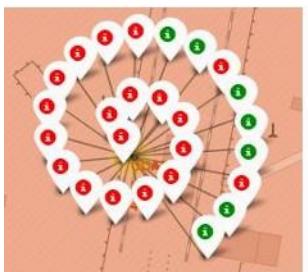
## ■ Summary of methodologies -

- 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 -

Debdc...

- Data Analysis along with Interactive Visualizations
- Best model for Predictive Analysis





## ■ Project background and context:

Here 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 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:

- With what factors, the rocket will land successfully?
- The effect of each relationship of rocket variables on outcome.
- Conditions which will aid SpaceX have to achieve the best results.

# Methodology



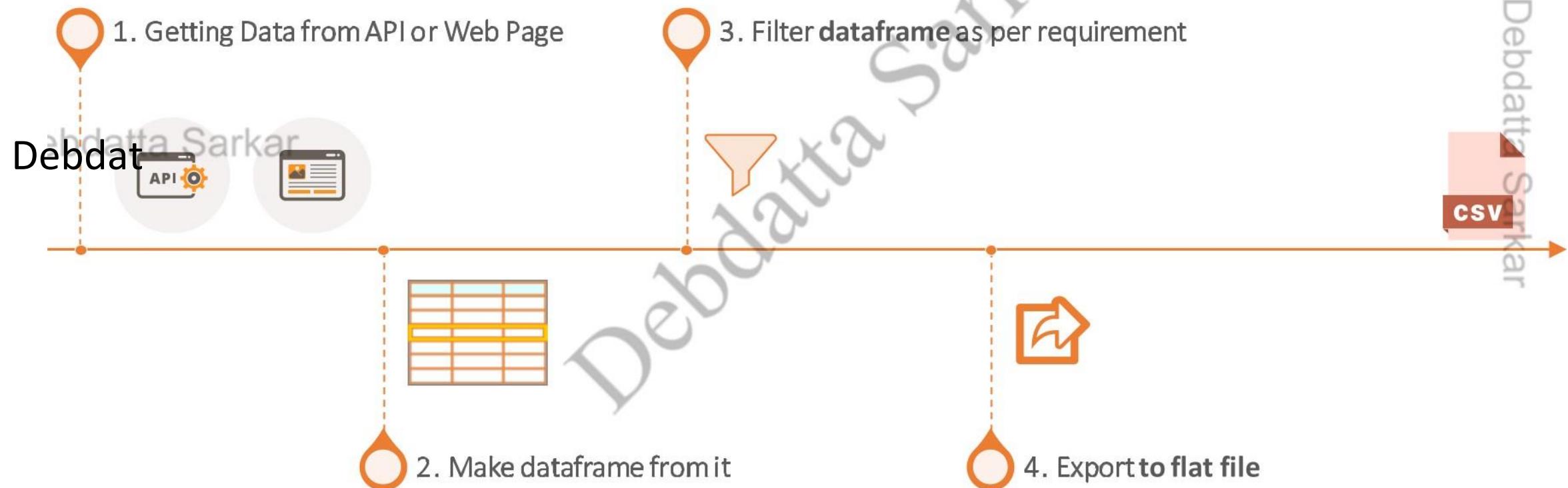
- **Data collection methodology:**
  - Via SpaceX Rest API
  - Web Scrapping from [Wikipedia](#)
- **Perform data wrangling:**
  - One hot encoding data fields for machine learning and dropping irrelevant columns (Transforming data for Machine Learning)
- **Perform exploratory data analysis (EDA) using visualization and SQL:**
  - Scatter and bar graphs to show patterns between data
- **Perform interactive visual analytics:**
  - Using Folium and Plotly Dash Visualizations
- **Perform predictive analysis using classification models:**
  - Build and evaluate classification models

Debdatta Sarkar

# Methodology



# Data Collection



# 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

---

# Data Collection

```
spacex_url="https://api.spacexdata.com/v4/launches/past"  
response = requests.get(spacex_url)
```

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

```
data_falcon9.drop(data_falcon9[data_falcon9['BoosterVersion']!='Falcon 9'].index, inplace=True)  
data_falcon9.loc[:, 'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))  
data_falcon9  
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Getting Response from API

Converting Response to a .json file

Apply custom functions to clean data

Assign list to dictionary then create dataframe

Filter dataframe and export to flat file

Debdatta Sarkar

```
jlist = requests.get(static_json_url).json()  
df = pd.json_normalize(jlist)  
df.head()
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),  
'Date': list(data['date']),  
'BoosterVersion':BoosterVersion,
```

Debdatta Sarkar

# Data Collection

# Data Collection

<u>URL</u>	2013-	Falcon 9	3170.000000	GTO	CCSFS	1	None	False	False	None	1.0	o B1004	<a href="#">GitHub</a>
8	5	12-03		SLC	40	None							

---

# Data Collection

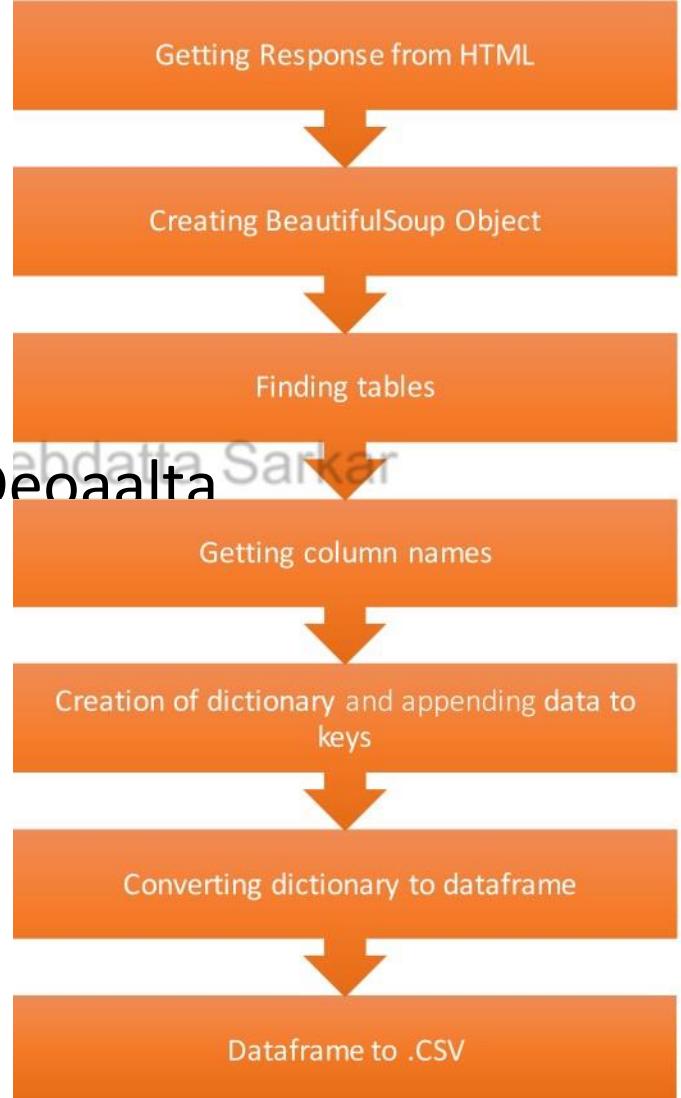
- Via Web Scraping

# Data Collection

Debdatta Sarkar

Debdatta Sarkar

[GitHub URL](#)



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

```
soup = BeautifulSoup(data, 'html5lib')
```

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

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

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

	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	Success	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.0B0007.1	No attempt	1 March 2013	15:10

# Data Collection

15:43

Payload	Orbit Customer	Version	Booster	<u>GitHub URL</u>	
				Booster	Date Time
				landing	
CCAFS Dragon Spacecraft Qualification Unit	o LEO	SpaceX	Success F9 VI .OB0003.1	Failure	4 June 2010 18:45
CCAFS	Dragon	o LEO	success F9 VI .OB0004.1	Failure	8 December 2010
CCAFS	Dragon	525 kg	LEO	Success F9 VI .080005.1	No attempt 22 May 2012 07:44
CCAFS	Spacex CRS-I	4,700 kg	LEO	Success F9 VI .OB0006.1	No attempt 8 October 2012 00:35

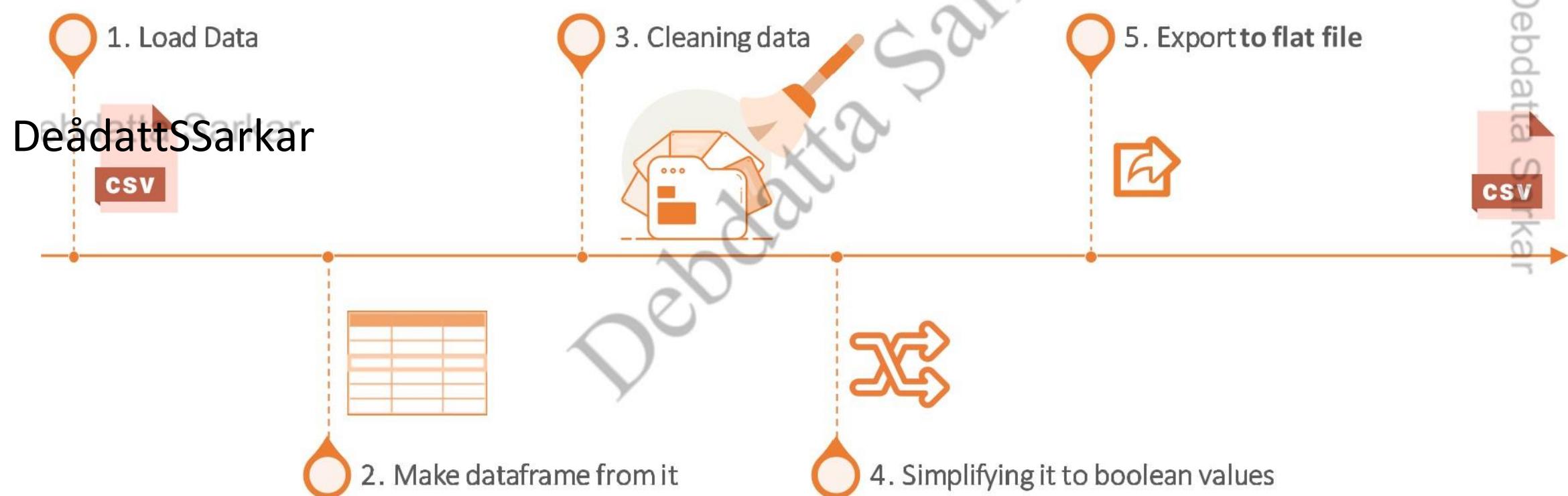
# Data Wrangling - Meaning & Basic Steps

---

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

Here we mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
```



# Data Wrangling

Calculate number of launches at each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55  
KSC LC 39A     22  
VAFB SLC 4E     13
```

Calculate number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27  
ISS      21  
VLEO     14  
PO       9  
LEO      7  
SSO      5  
MEO      3  
ES-L1    1  
HEO      1  
SO       1  
GEO      1
```

Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()  
landing_outcomes
```

```
'Class') = 'Outcome'.Ja  
'Class' I l .head(8)  
a landing_class :
```

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)  
df[['Class']].head(8)
```

Create landing outcome label from Outcome column

Export dataset as .CSV

```
df.to_csv("csvs/dataset_part_2.csv", index=False)
```

[GitHub URL](#)

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1 2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003

index=False )

[GitHub URL](#)

CCAFS	None													
LEO		False									NaN	1.0	O	B0003

Debdatta Sarkar

D

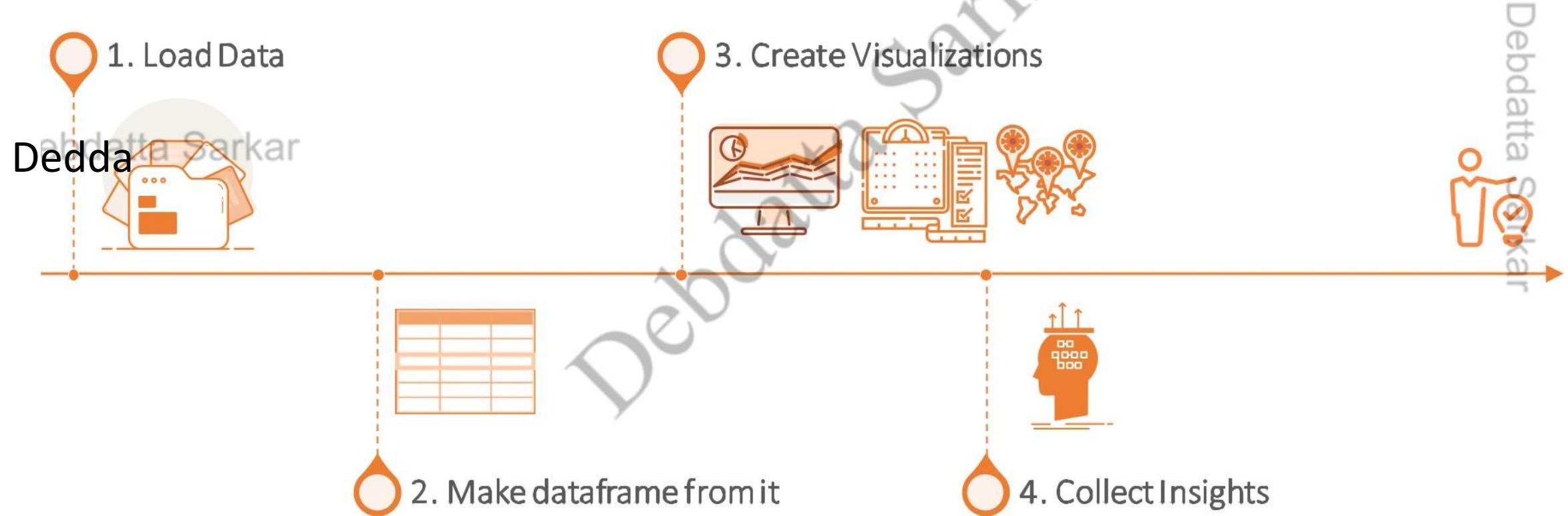
True	ASDS	41		2	201205-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None	1	False	False	False	NaN	O B0005	
None	None	19			2013-				CCAFS	None					1.0		
True	RTLS	14							SLC 40	None	1	False	False	False	NaN	O B0007	
False	ASDS	6	2	3	03-01	Falcon 9	677.000000	ISS	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	O B1003
True	Ocean	5			2013-											O B1004	
None	ASDS	2		4	09-29	Falcon 9	500.000000	PO									
False	Ocean	2															
False	RTLS	1	4	5	201312-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None	1	False	False	False	NaN	1.0	O B1004

# EDA - Meaning & Basic Steps

---

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, using statistical graphics

and other data visualization methods.



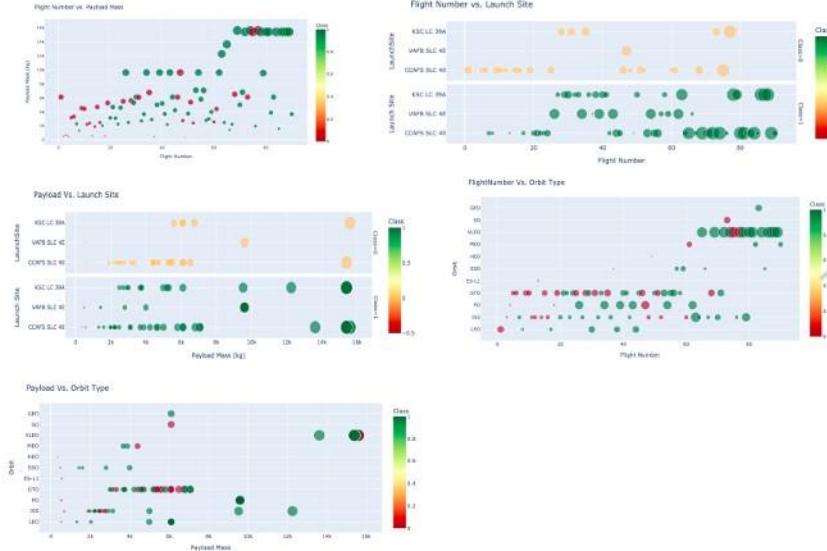
# EDA with Data Visualization

## Scatter Graphs Drawn:

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

Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs it's very easy to predict which factors will lead to maximum probability of success in both outcome and landing.

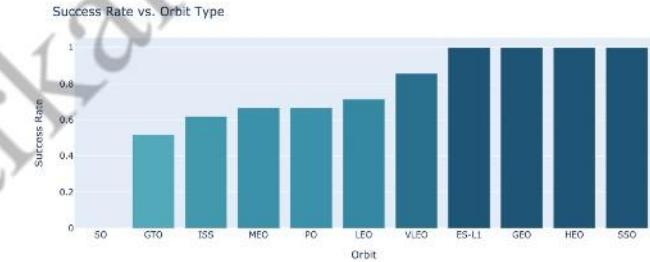
Debdatta Sarkar



## Bar Graph Drawn:

### Success Rate VS. Orbit Type

Bar graphs are easiest to interpret a relationship between attributes. Via this bar graph we can easily determine which orbits have the highest probability of success.

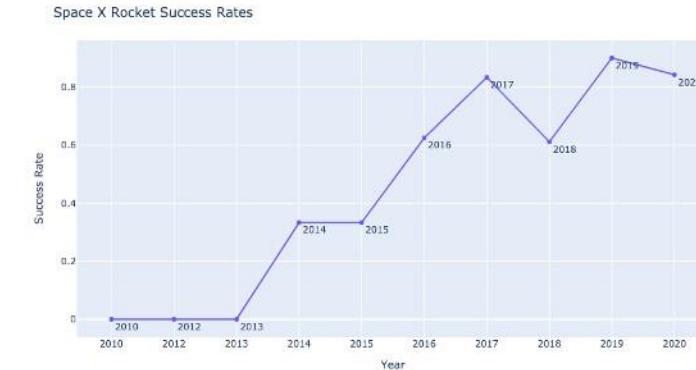


Debdatta Sarkar

## Line Graph Drawn:

### Launch Success Yearly Trend

Line graphs are useful in that they show trends clearly and can aid in predictions for the future.

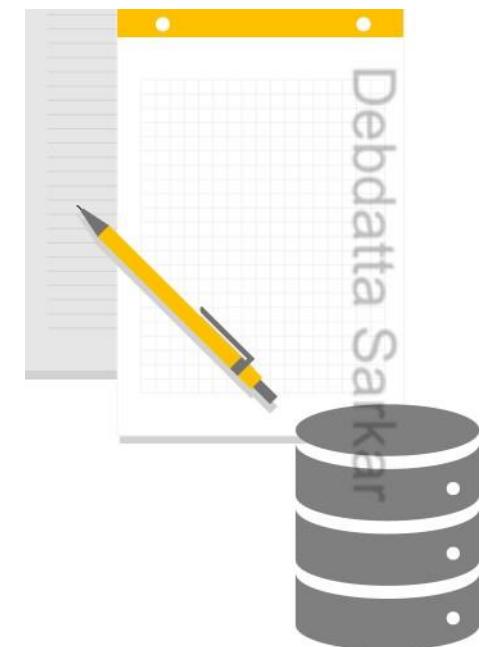


Seaborn Graphs GitHub URI

# 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.

```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
%load_ext sql

%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name
%sql <your query>
```

## Debdatta Sarkar

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 F9 v1.1
- 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 names of the booster\_versions which have carried the maximum payload mass

Cloud, which is a fully managed SQL Database

provided as a service.

- Listing the failed landing\_outcomes in drone ship, their boosterversions, 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, in descending order

[GitHub URL](#)

# Build an Interactive Map with Folium

---

Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. We use the latitude and longitude coordinates for each launch site and added a Circle Marker around each launch site with a label of the name of the launch site. It is also easy to

visualize the number of success and failure for each launch site with Green and Red markers on the map.

Map Objects	Code	Result
Map Marker	folium.Marker()	Map object 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 same coordinate.
AntPath	folium.plugins.AntPath()	Create an animated line between points.

[Original Project - GitHub URL](#)

[Clean Distance Markers GitHub URL](#)

visualize the number of success and failure for each launch site with Green and Red markers on the



# Build a Dashboard with Plotly Dash

Pie Chart showing the total success for all sites or by certain launch site

- Percentage Success rate in certain launch site

Scatter Graph showing the correlation between Payload and Success for all sites or by certain launch site

- It shows the relationship between Success rate and Booster Version Category.

Map Objects	Code	Result	De
Dash and its components With Dash Open Source, dash_html_components as Component library components import more.	import dash from dash.dependencies import Input, user-friendly Python classes.  Pandas Plotly Dropdown	import dash from dash.dependencies import Input, user-friendly Python classes.  import pandas as pd from plotly import express as px  dcc.Dropdown(  as html as dcc t Input, Output  Plot the graphs with interactiveplotlylibrary Create a dropdown for launchsites Create a rangeslider for PayloadMass range selection Creating the Pie graph for Success percentage display Creating the Scatter graph for correlation display	Pyton's.leauRdataviz and UI libraries. Dash apps run onWOur import html local laptop or serve . he sh Core contains a set of higher-level dash_core_components as dcc , tables, and  Output t e available HTML tags as  CSV and creating a dataframe  co
			Deb datta arkar

Rangeslider	dcc.RangeSlider()
Pie Chart	px.pie()
Scatter Chart	px.scatter()

[GitHub Code URL](#)

Used "Python Anywhere" to host a live website. The live site dashboard is built with Flask and Dash.

[Live Site URL](#)



# Predictive Analysis (Classification)

Load our feature engineered data into dataframe  
Transform it into NumPy arrays

```
y = data['Class'].to_numpy() train =  
preprocessing.StandardScaler() X = train  
form .fit(X).transform (X)  
X_train, X_test, Y_train, Y_test = train_test_split(X, y, te  
ra ndo tal  
_test.shape
```

```
algorithms = KNN ':knn_cv.best_score_,  
Decision Tree':tree_cv.best_score
```

## Building Model

Standardize and transform data

Split data into training and test data sets  
best\_algorithm = max(algorithms, key=lambda x:

Check how many test samples has been created

```
Logistic Regression' best_score_}  
algorithms
```

List down machine learning algorithms we want to use

Set our parameters and algorithms to GridSearchCV

Fit our datasets in to the GridSearchCV objects and train our model

Find the Best Performing Classification Model}

The model with best accuracy score wins the best performing model

Debda+, La.Sar. a

CO

Best Model

## Evaluating Model

```
for algorithm in algorithms:  
    algorithm.fit(X_train, y_train)  
    y_pred = algorithm.predict(X_test)
```

Check accuracy for each model

- Get best hyperparameters for each type of algorithms
- Plot Confusion Matrix

Code

[GitHub URL](#)



# Results

Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

Debdatta Sarkar





Debdatta Sarkar

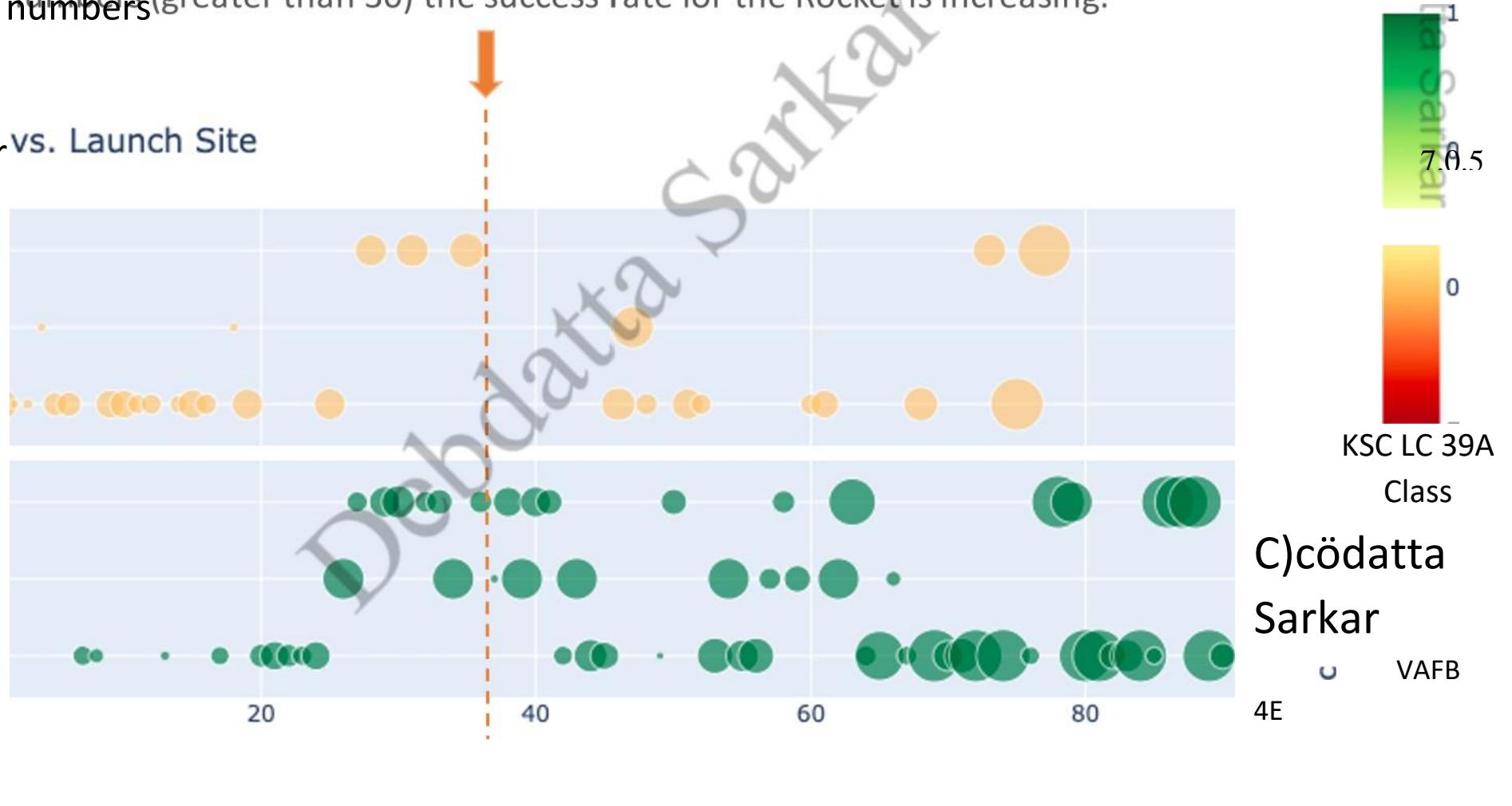
# EDA with Visualization

# Flight Number vs. Launch Site

---

- With higher flight numbers (greater than 30) the success rate for the Rocket is increasing.

Flight Number vs. Launch Site



KSC LC 39A

Launch Site

VAFB SLC 4E  
CCAFS SLC 40

-0.5

[Sea horn Graphs GitHub URL](#)

Flight Number

[PI Qtly Gra GitHub URL](#)

# Payload vs. Launch Site

---

GitHub

GitHub

- The greater (greater than success rate there's no decision, if the dependent on success

o take a decision, if the launch site is dependent on Pay Load

Launch Site

Payload

KSC LC 39A

De\*atta

VAFB SLC 4E

CCAFS SLC 40

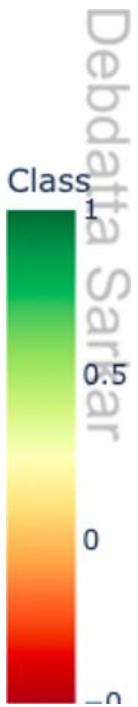
KSC LC 39A



the payload mass (7000 Kg) higher the for the Rocket. But clear pattern to take a launch site is Pa 'bad Mass for a launch.

Vs. Launch Site

Sarkar



Graphs GitHub

PI Qtly Gra GitHub

Launch Site

VAFB SLC 4E

CCAFS SLC 40

-0.5

14k

16k

Payload Mass (kg)

Sea horn URL

URL

# Success Rate vs. Orbit Type

---

GitHuh

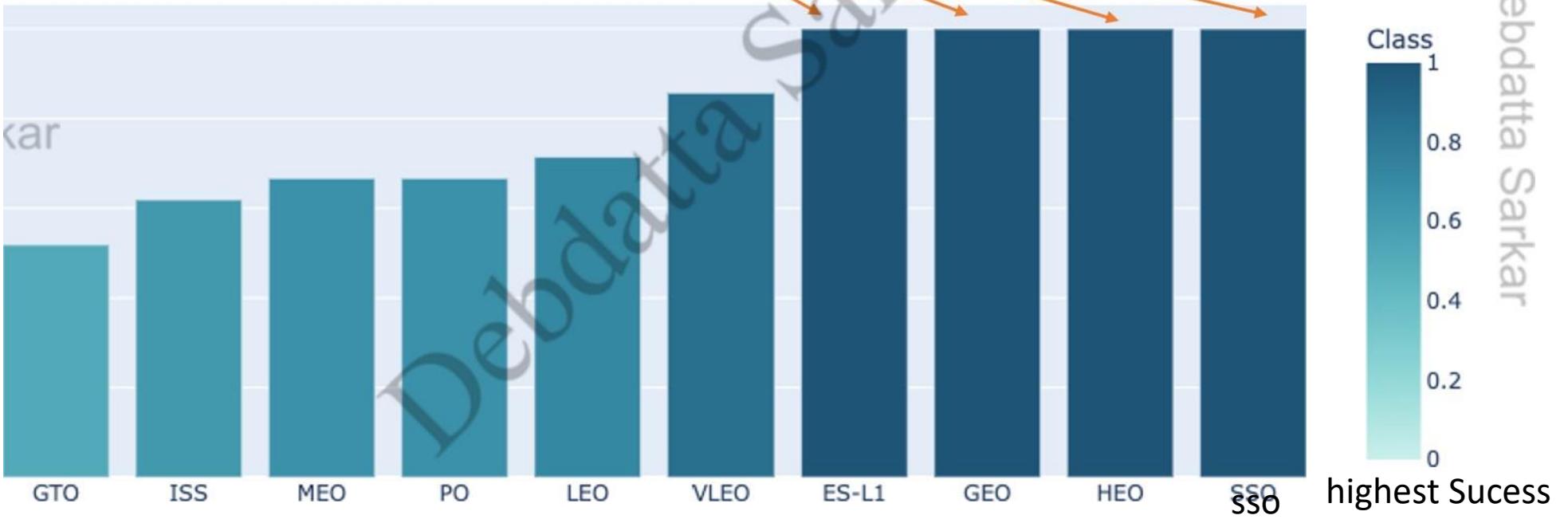
GitHuh

- 

MEO, GEO has highest success rates.

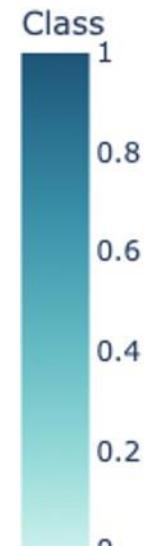
### Success Rate vs. Orbit Type

Success Rate



ES-L1, GEO,  
HEO, SSO has

Debattar Sarker



rates.

Graphs GitHub

PI Qtly Gra GitHub

Debcytta Sarkar

0.6

0.4

0.2

so

Orbit

[Sea horn Graphs](#) [IJR\[](#)

[Plotly Graphs](#) [IJR\]](#)

# Flight Number vs. Orbit Type

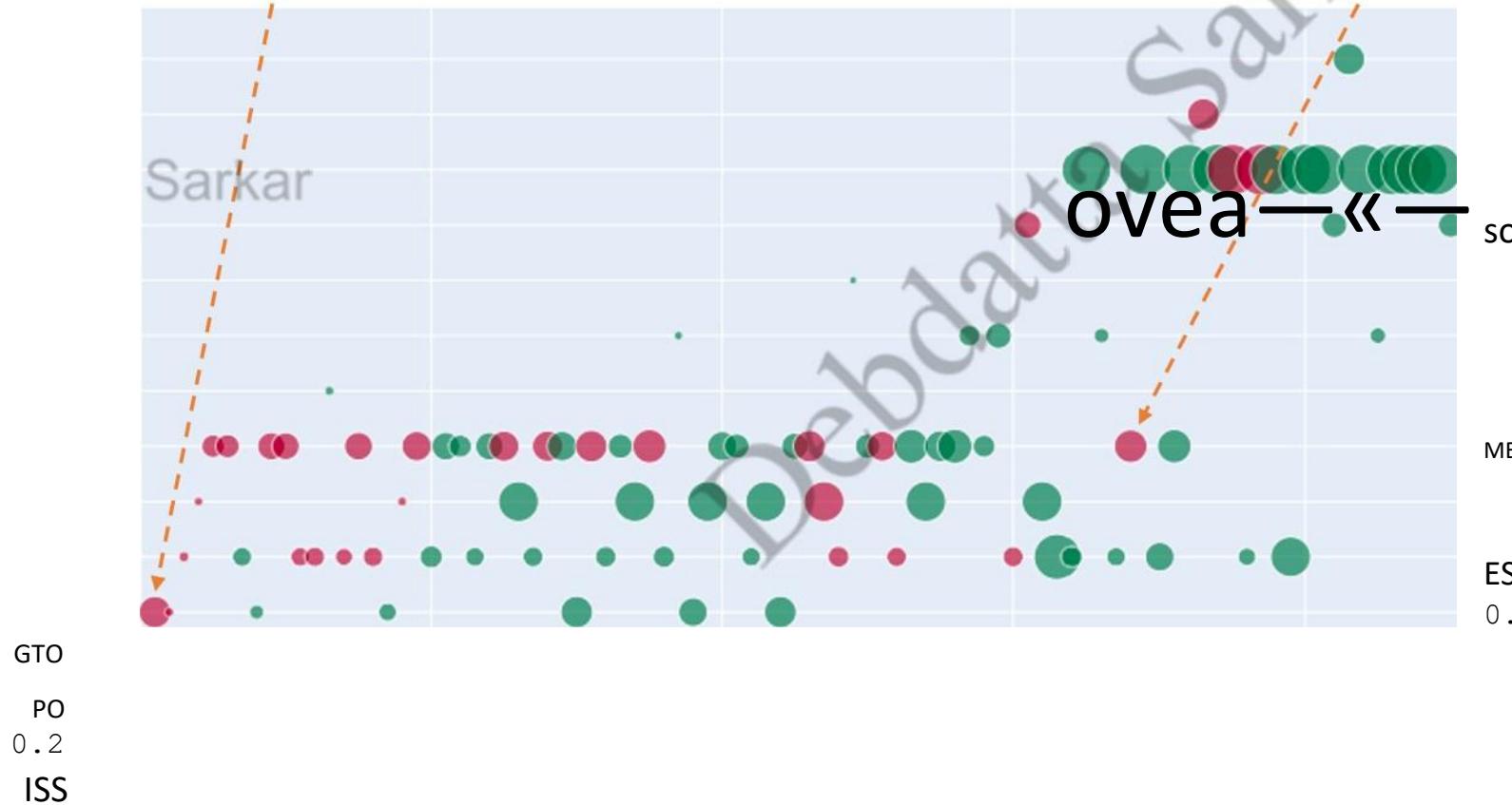
---

- We see that for LEO orbit the success increases with the number of flights

GitHub

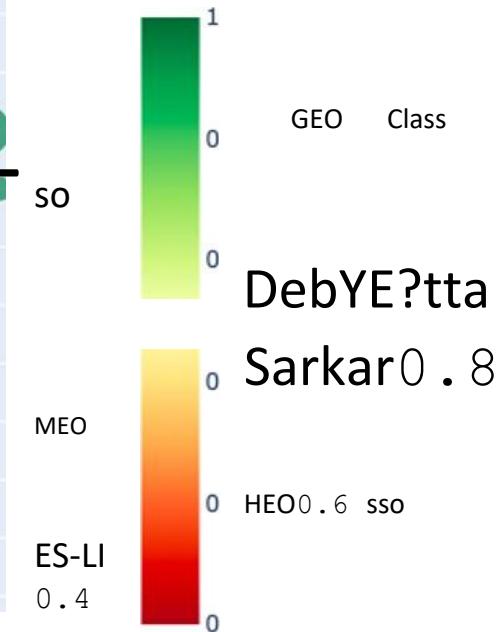
GitHub

FlightNumber Vs. Orbit Type



other hand, there seems to be no relationship between flight number and the GTO orbit.

- On the other hand, there seems to be no relationship between flight number and the GTO orbit.



Debdatta Sarkar

Graphs GitHuh

PI Qtly Gra GitHuh

LEO

20

40

60

80

Flight Number

Sea horn URL

 URL

GitHuh

GitHuh

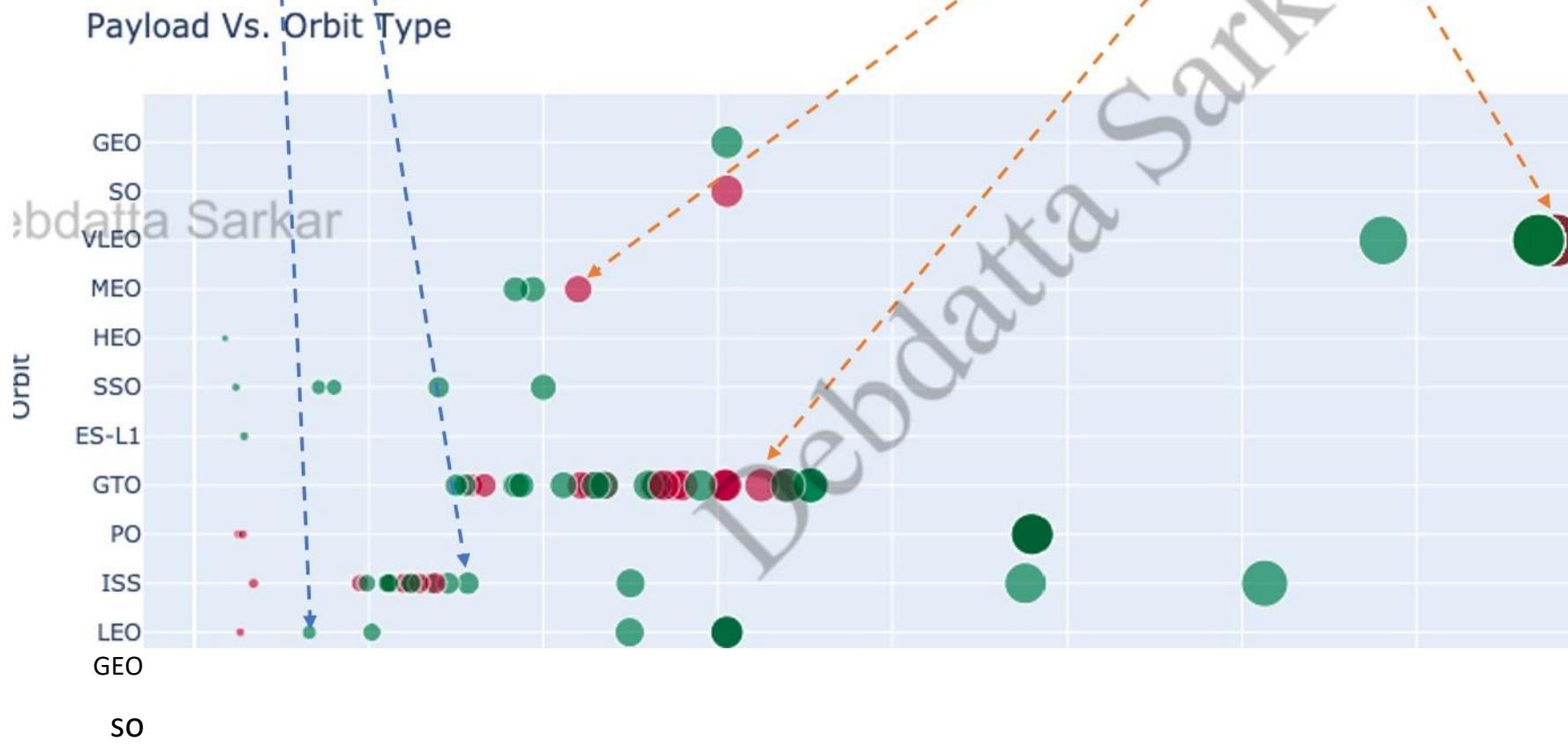
# Payload vs. Orbit

## Type

---

- We observe that heavy payloads have a negative influence on MEO, GTO, VLEO orbits

- Positive on LEO, ISS orbits



Debde & ga  
Sar Æar

M  
E  
O  
H  
E  
O  
s  
s  
o  
ES-LI  
GTO  
  
ISS  
LEO

2k

4k



# Launch Success Yearly Trend

---

We can observe that the sucess rate since 2013 kept increasing relatively though there is slight dip after 2019.

## Space X Rocket Success Rates



# With SQL

# All Launch Site Names

---

SQL Query

```
*sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

## Description

## Description

Using the word DISTINCT in the query we pull unique values for Launch\_Site

column from table SPACEX.

# Launch Site Names begin with 'CCA'

---

## Launch\_Sites

CCAFS LC-40

CCAFS SLC-40

KSC LC-39A

VAFB SLC-4E

# SQL Query

```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

## Description

Using keyword 'LIMIT 5' in the query we fetch 5 records from table spacex and with condition LIKE keyword with Wild card - 'CCA%'. The percentage in the end suggests that the Launch\_Site name must start with CCA.

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 (OSS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 1.0 B0006	CCAFS LC40	SpaceX CRS-I		LEO (OSS)	NASA (CRS)		No attempt
2013-03-01	15:10:00	F9 VI .0 B0007	CCAFS LC40	SpaceX CRS-2	677	LEO (OSS)	NASA (CRS)		No attempt

# Total Payload Mass

---

# SQL Query

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

## Description

Using the function SUM calculates the total in the column PAYLOAD\_MASS\_KG\_ and WHERE clause filters the data to fetch Customer's by name "NASA (CRS)".

**Total Payload Mass by NASA (CRS)**

---

45596

PAYLOAD MASS KG and WHERE clause filters the data to

# Average Payload Mass by F9 VI-I

---

## SQL Query

```
%sql SELECT AVG(PAYLOAD_MASS_KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

## Description

Using the function AVG works out the average in the column PAYLOAD\_MASS\_KG\_.  
The WHERE clause filters the dataset to only perform calculations on Booster\_version "F9 v1.1".

### Average Payload Mass by Booster Version F9 v1.1

2928

column PAYLOAD MASS KG \_p rform  
calculations on Booster version "F9 v 1.1".

# First Successful Ground Landing

Date

## De SQL Query

```
%sql SELECT MIN(DATE) AS "First Successful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING_OUTCOME = 'Success (ground pad)' ;
```

### Description

Using the function MIN works out the minimum date in the column Date and WHERE clause filters the data to only perform calculations on Landing\_Outcome with values "Success (ground pad)".

#### First Successful Landing Outcome in Ground Pad

2015-12-22

gave. In the column Date and WHERE clause filters the outcome with values "Success (ground pad)".

# Successful Drone Ship Landing with Payload between 4000 and 6000

---

## SQL Query

```
*sql SELECT BOOSTER VERSION FROM SPACEX WHERE LANDING_.OUTCO      ' Success (drone ship)'  
AND PAYLOAD MASS KG > 4000 AND PAYLOAD MASS KG < 600          \
```

D hdatta Sukar

bescnpt•onco

Selecting only Booster\_Version,

WHERE clause filters the dataset to  
Landing\_Outcome: Success (drone)  
AND clause specifies additional filter  
Payload\_MASS\_KG >4000 AND  
Payload\_Mass\_KG < 6000

# Total number of Successful and Failure Mission

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

# Outcomes

## SQL Query

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission",  
       sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \  
FROM SPACEX;
```

Debdatta Sarkar

## Description

Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.

Case when MISSION\_OUTCOME like "%Success%" then 1 else 0 end"

Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.

Successful Mission	Failure Mission
100	1

Case when MISSION OUTCOME LIKE '0/0Success%' then 1 else 0 end"  
returns a Boolean value which we sum to get the result needed.

## Boosters carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX \
WHERE PAYLOAD_MASS_KG_ =(SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEX);
```

## Description

Using MAX function works out the maximum payload in the column PAYLOAD\_MASS\_KG\_ in sub query.

WHERE clause filters Booster Version which had that maximum payload.

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

## SQL Query

# 2015 Launch Records

---

## SQL Query

```
*sql SELECT {fn MONTHNAME(DATE)} as "Month" , BOOSTER VERSION , LAUNCH SITE FROM SPACEX WHERE year (DATE ) = '2015' AND \ LANDING OUTCOME = 'Failure (drone ship) ;
```

## Description

## Debdatta Sarkar

We need to list the records which will display month names, failure landing\_outcomes in booster versions, launch site for the months in a 015.

Via year function we extract the year and future where clause 'Failure (drone ship)' fetches our

Also, am using {fn MONTHNAME(DATE)} to get the Month name.

# Rank Landing Outcomes between 2010-06-04 and 2017-03-20

which will display the outcomes in drone ship, months in year 2015.

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

---

## SQL Query

Nar

```
%sql SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

Debdatta Sarkar

## Description

Selecting only LANDING\_OUTCOME,

WHERE clause filters the data with DATE BETWEEN '2010-06-04' AND '2017-03-20'

Grouping by LANDING\_OUTCOME

Order by COUNT(LANDING\_OUTCOME) in Descending Order.

'2010-06-04' AND '2017-03-20'

Order by

Landing Outcome	Total Count
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

Rank Success Count between  
2010-06-04 and 2017-03-20

## SQL Query

```
%sql SELECT COUNT(LANDING_OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-20" FROM SPACEX \
WHERE LANDING_OUTCOME LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

## Description

COUNT counts records in column LANDING\_OUTCOME  
WHERE filters data with '%Success%'  
AND DATE > '2010-06-04'  
AND DATE < '2017-03-20'

**Rank success count between 2010-06-04 and 2017-03-20**

8

COUNT counts records in column LANDING  
WHERE filters data with '%Success%'  
AND DATE > '2010-06-04'  
AND DATE < '2017-03-20'



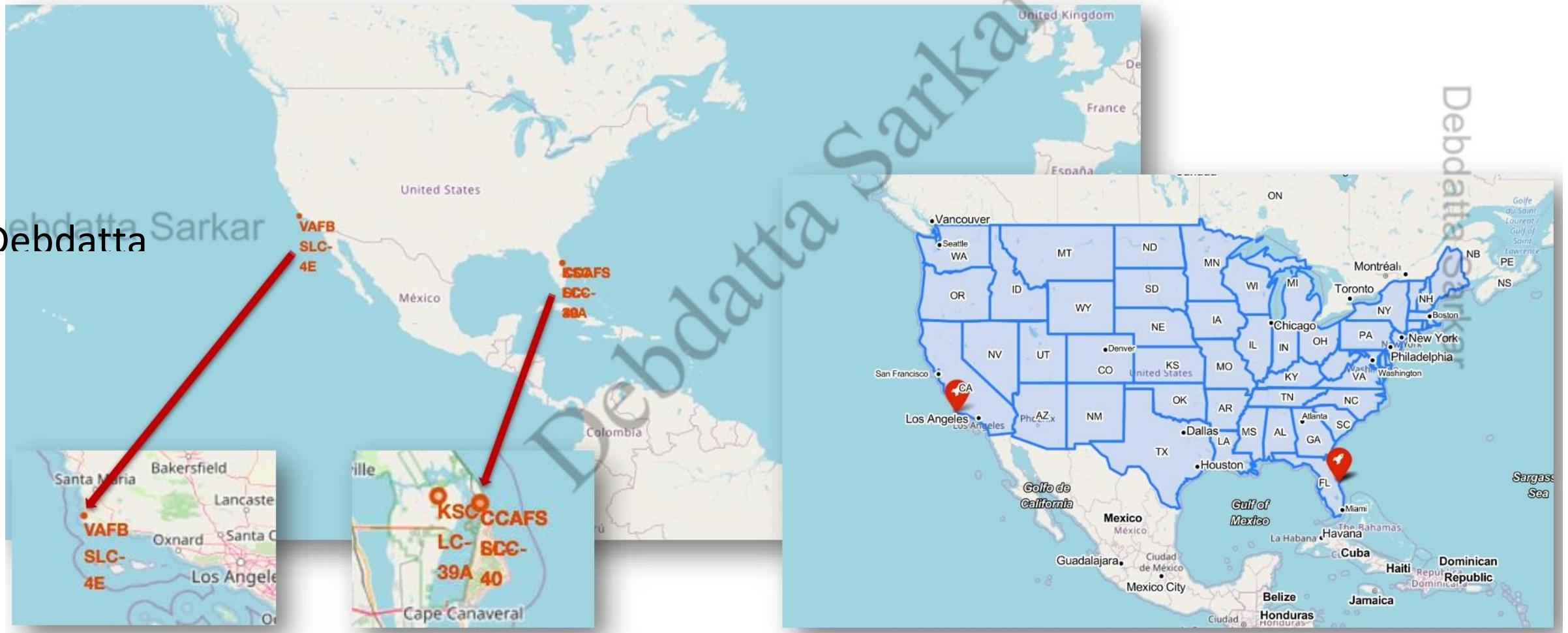
# Interactive map with Folium

# All Launch Sites on Folium Map

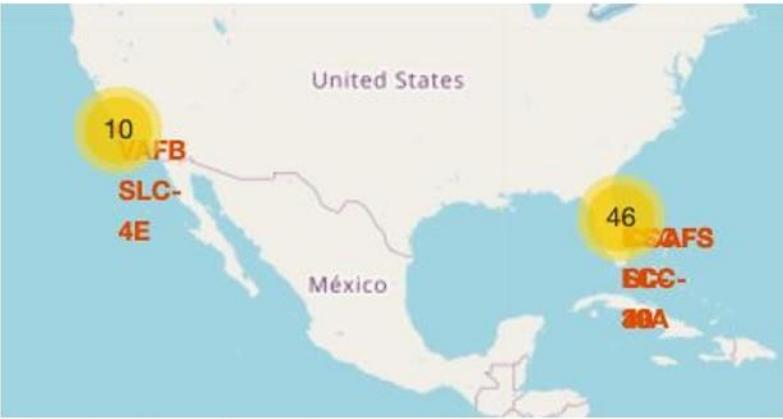
---

We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions.

Debdatta

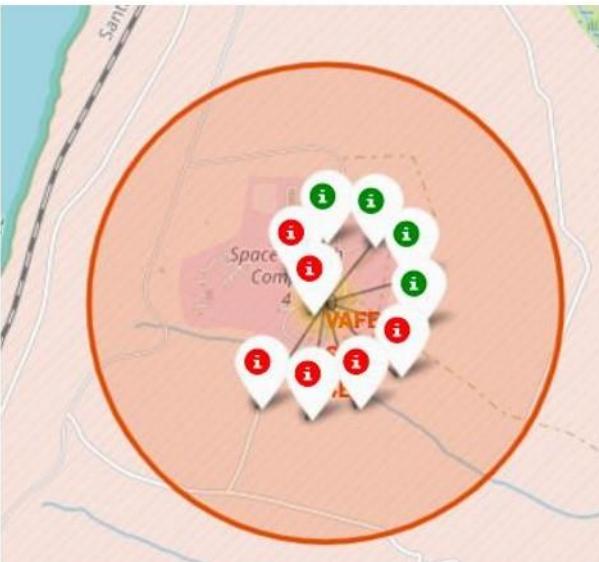


# Color Labeled Launch Records



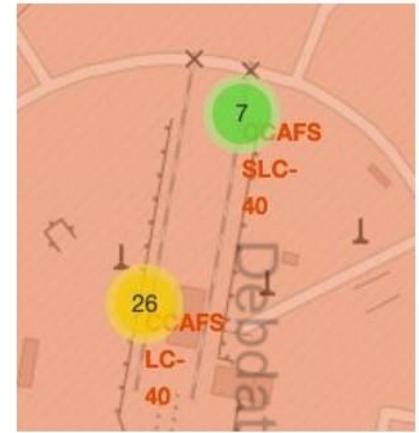
Debdatta Sarkar

VAFB SLC-4E



Green Marker shows successful launches and Red Marker shows failures.

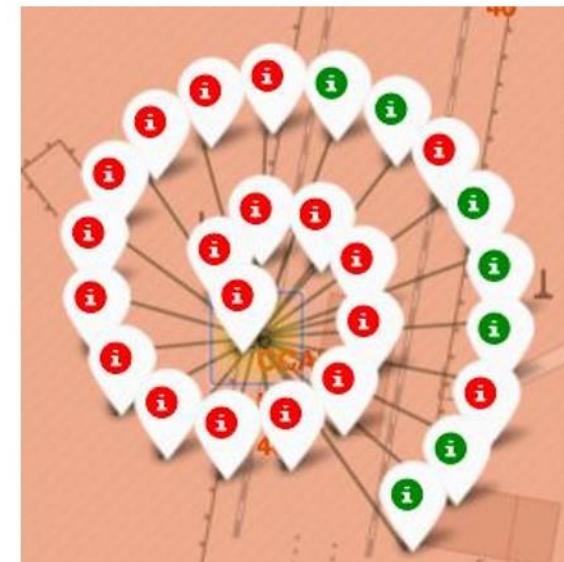
From these screenshots its easily understandable that KSC LC-39A has the maximum probability of success.



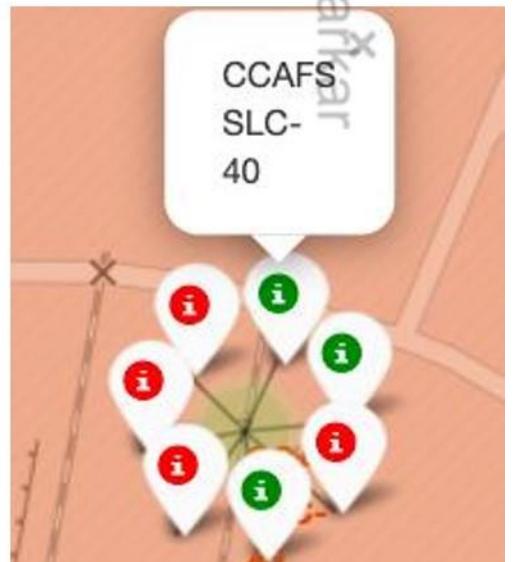
KSC LC-39A



CCAFS LC-40



CCAFS SLC-40

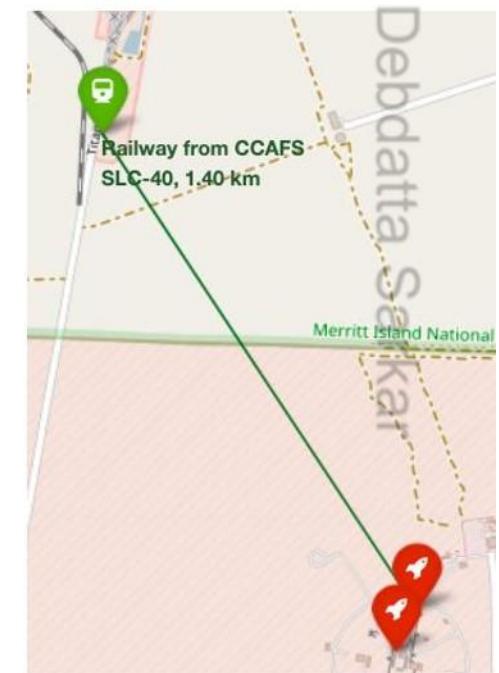
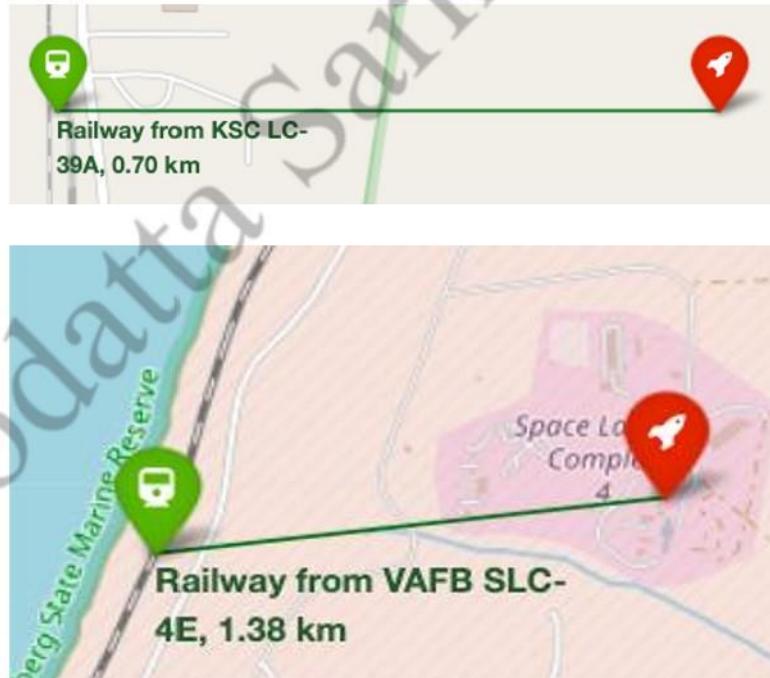


# Launch Site Distances from Equator & Railways

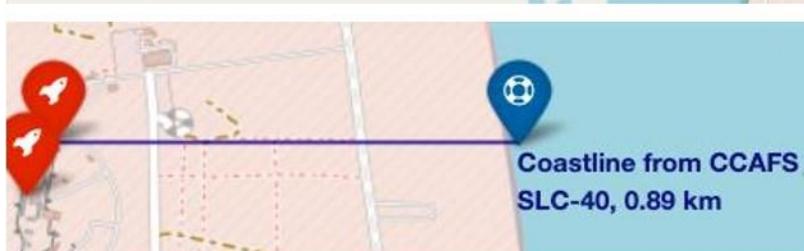
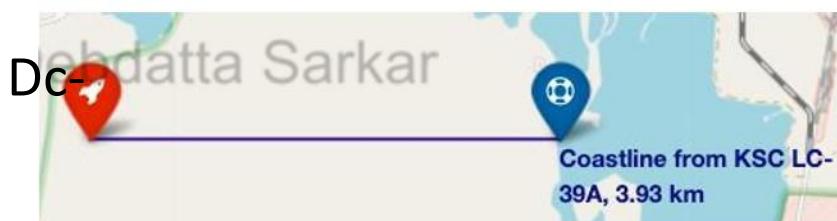
Distance from Equator is greater than 3000 Km for all sites.



Distance for all launch sites from railway tracks are greater than .7 Km for all sites. So, launch sites are not so far away from railway tracks.

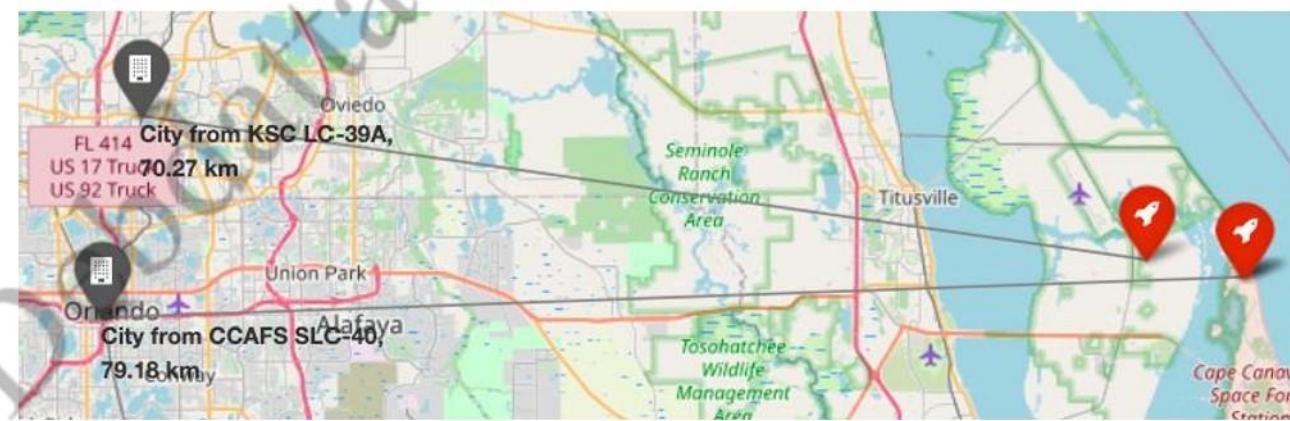


# Launch Site Distances from Coastlines & Cities



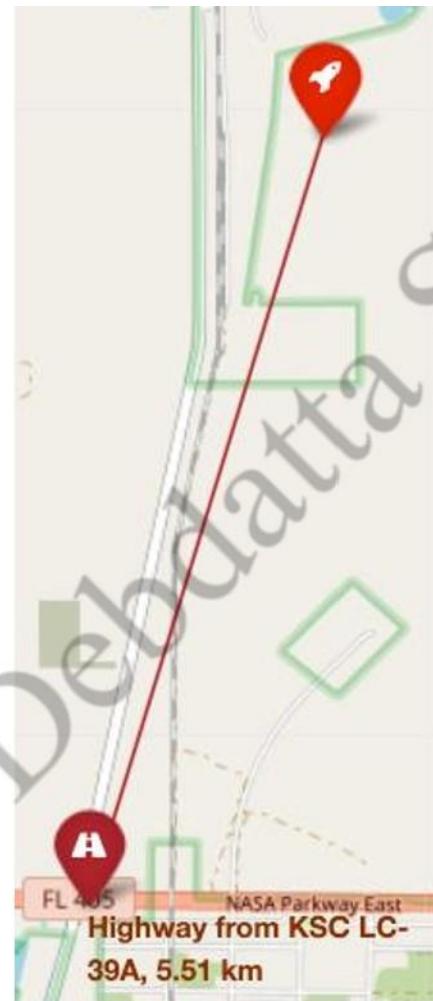
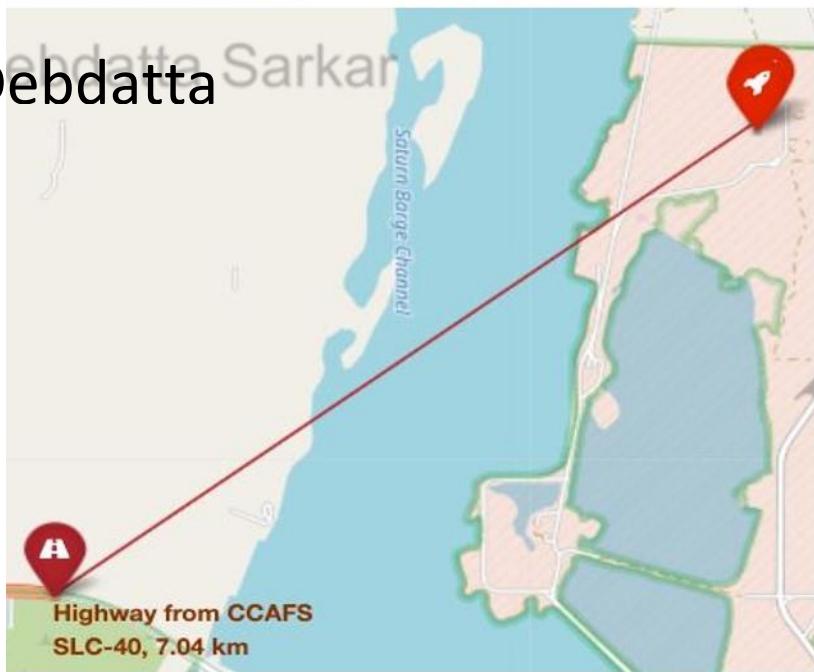
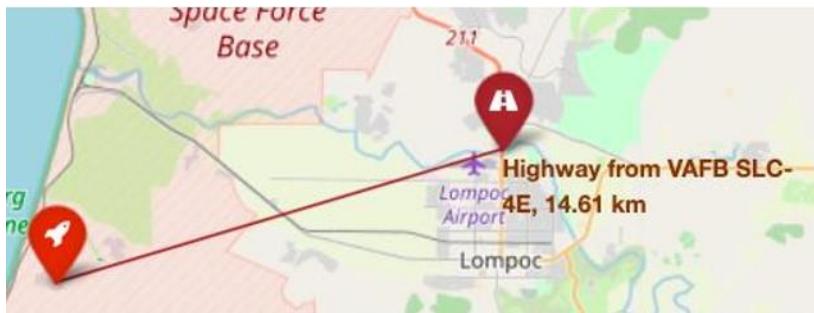
Distance for all launch sites from coastline is less than 4 Km.

Distance for all launch sites from cities is greater than 14 Km for all sites. So, launch sites are far away from cities.



# Launch Site Distances from Highways

Debdatta Sarkar



Distance for all launch sites from highways is greater than 5 Km for all sites. So, launch sites are relatively far away from highways.

### Conclusion:

- Are all launch sites in proximity to the Equator line?  
No ( $4000 \text{ Km} > \text{distance} > 3000 \text{ Km}$ )
- Are launch sites in close proximity to railways?  
Yes ( $2 \text{ Km} > \text{distance} > .5 \text{ Km}$ )
- Are launch sites in close proximity to highways?  
No ( $15 \text{ Km} > \text{distance} > 5 \text{ Km}$ )
- Are launch sites in close proximity to coastline?  
Yes ( $5 \text{ Km} > \text{distance} > .5 \text{ Km}$ )
- Do launch sites keep certain distance away from cities?  
Yes ( $15 \text{ Km} > \text{distance} > 80 \text{ Km}$ )

Debdatta Sarkar

# Build a Da hboard with Plotly Dash

## Launch Success Count for All Sites

---

# SpaceX Launch Records Dashboard

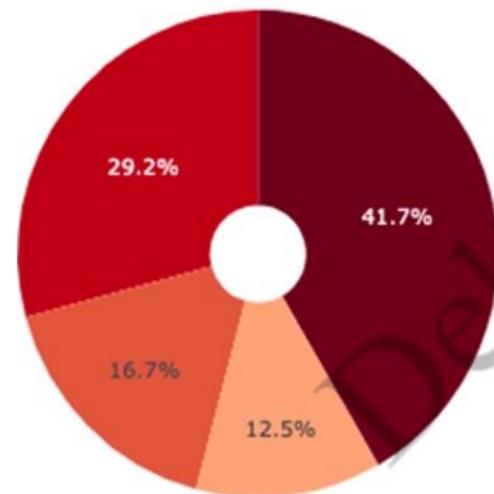
We can see that KSC LC-39A had the most successful launches from all the sites.

Total Success  
Sites  
by All Sites

KSC LC.39A

Debdatta Sarkar

CCAFS LC.40



All Sites

Launches by All

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

Debdatta Sarkar  
CO

[Live Site URL Code](#)

# Payload vs. Launch Outcome Scatter Plot for All Sites

---

We can see the success rates for payloads is higher than the heavy

Low Weighted Payload Okg — 4000kg

0.8

0.4

Débdatta Sarkar

1000

1500

Low Weighted Payload 4000kg-10000kg

0.8

0.6

0.4

0.2



low weighted  
weighted payloads



Booster Version Category

VI.o

VI.I

35

Booster Version Category

• VI.I

Débdatta Sarkar

4k

5k

6k

9k

Payload Mass (kg)

[Live Site URL Code](#)

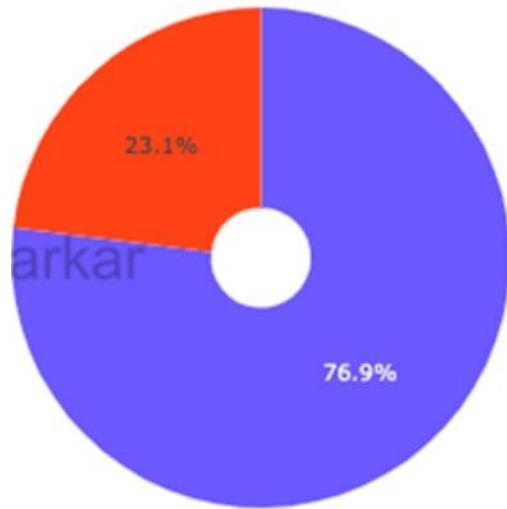
# Launch Site with Highest Launch Success Ratio

---

Total Success Launches for Site KSC LC-39A

KSC LC-39A achieved a 76.9% success rate while getting a

ysis using the  
able to obtain



Debdatta S  
ch  
launch success

Which payload  
highest launch

Which payload  
the lowest

23.1% failure rate.

After visual analysis usin  
some insights to answer these questions: site has the highest  
rate?

- Which site has the highe  
**KSC LC – 39A**
- Which payload range(s) |  
**2000 Kg – 10000 Kg**
- Which payload range(s) |  
**0 Kg – 1000 Kg**
- Which F9 Booster versio

Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the  
highest launch success rate?

dashboard, we are

co

range(s) has the  
success rate? ↗

range(s) has  
launch  
success rate?

[Live Site URL](#) [Code](#)

# Predictive analysis (Classification)

## Confusion Matrix

---

Out here for all models unfortunately, we have same confusion matrix.

Deb@  
Debdatta Sarkar

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

**Accuracy:**  $(TP+TN)/Total = (12+3)/18 = 0.83333$

**Misclassification Rate:**  $(FP+FN)/Total = (3+0)/18 = 0.1667$

**True Positive Rate:**  $TP/Actual\ Yes = 12/12 = 1$

**False Positive Rate:**  $FP/Actual\ No = 3/6 = 0.5$

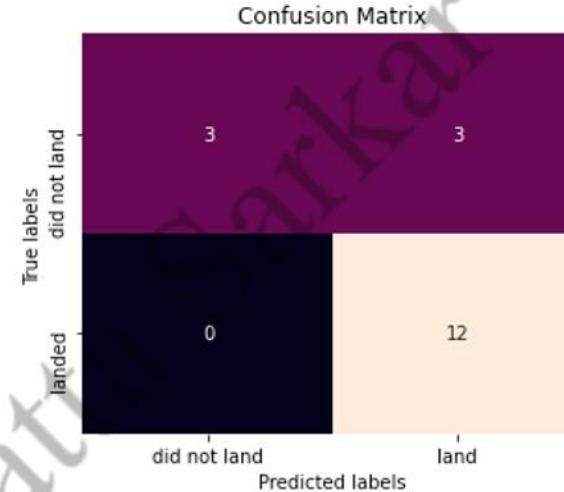
**True Negative Rate:**  $TN/Actual\ No = 3/6 = 0.5$

**Precision:**  $TP/Predicted\ Yes = 12/15 = 0.8$

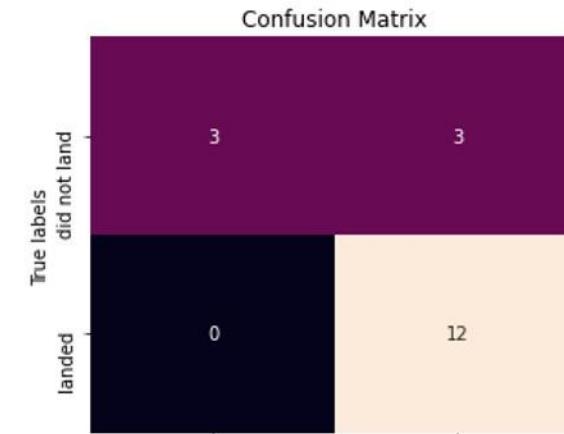
**Prevalence:**  $Actual\ yes/Total = 12/18 = 0.6667$

Predicted

### Logistic Regression



### Decision Tree



		Yes
True Positive	TP = 12	
3	15	Total Cases
		= 18

Accuracy:  $(TP+TN)/Total = (12+3)/18 = 0.83333$

Misclassification Rate:  $(FP+FN)/Total = (3+0)/18 = 0.16667$

True Positive Rate:  $TP/Actual\ Yes = 12/12 = 1$

False Positive Rate:  $FP/Actual\ No = 3/6 = 0.5$

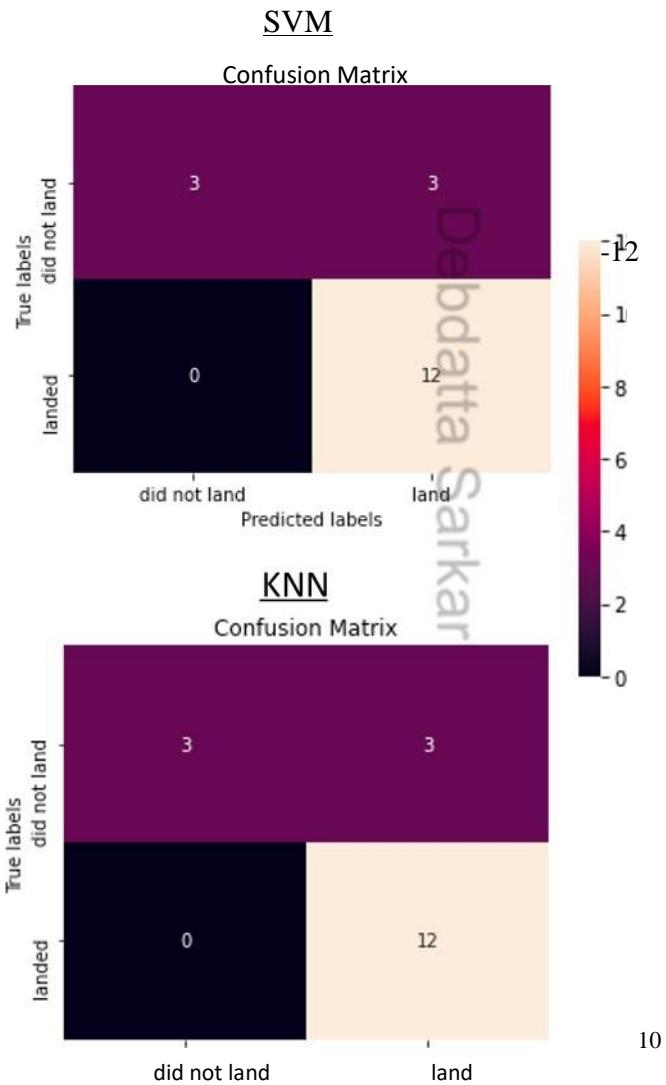
True Negative Rate:  $TN/Actual\ No = 3/6 = 0.5$

Precision:  $TP/Predicted\ Yes = 12/15 = 0.8$

Prevalence:  $Actual\ yes/Total = 12/18 = 0.6667$

	did not land	land
did not land		
land		

Predicted labels



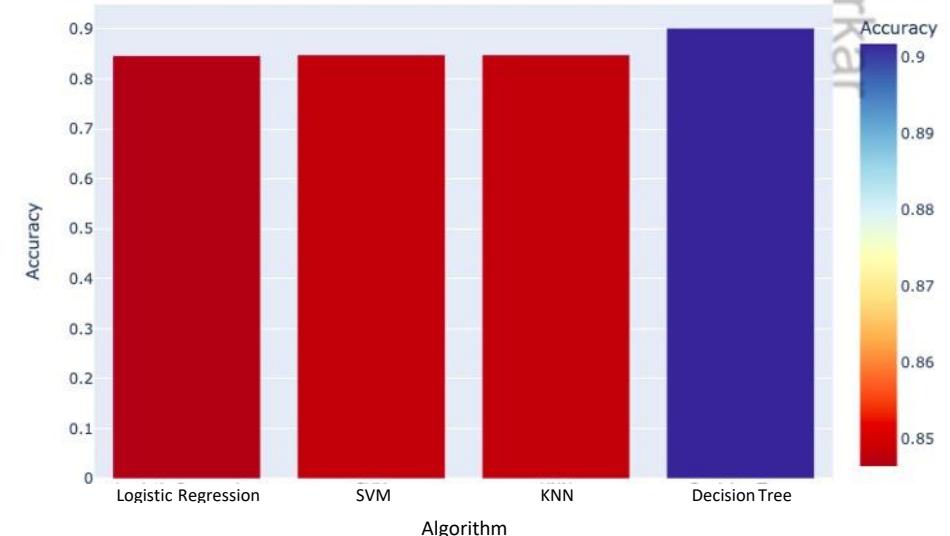
Predicted labels

# Classification Accuracy

with a score of 0.90178.

Algorithm	Accuracy	Accuracy on Test Data	Tuned Hyperparameters
Logistic Regression	0.846429	0.833334	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM	0.848214	0.833334	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
KNN	0.848214	0.833334	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
Decision Tree	0.901786	0.833334	{'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}

We trained four different models which each had an 83% accuracy rate.



---

As you can see our accuracy is extremely close, but we do have a clear winner which performs best - "Decision Tree"

# Conclusion

1

Orbits ES-L1, GEO, HEO, SSO has highest Success rates

2

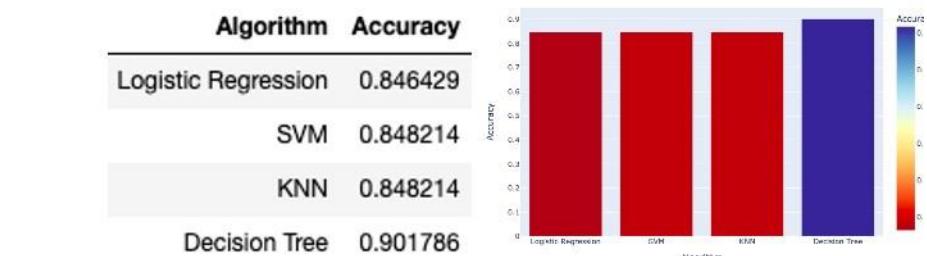
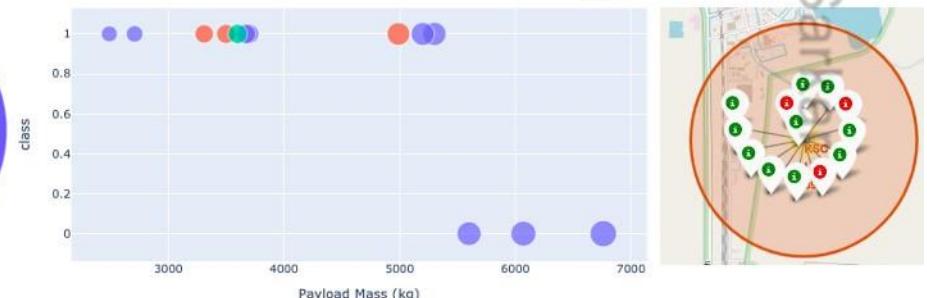
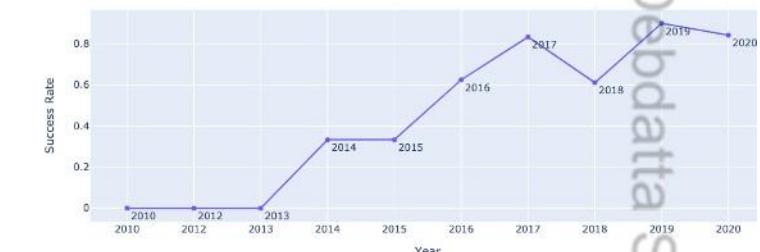
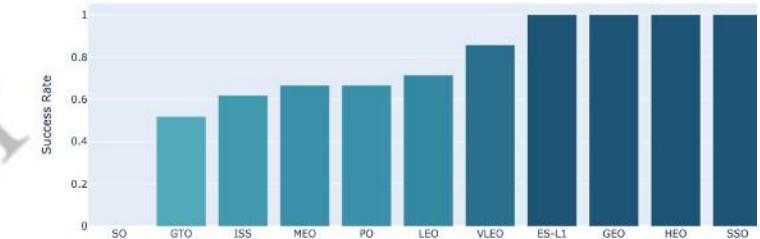
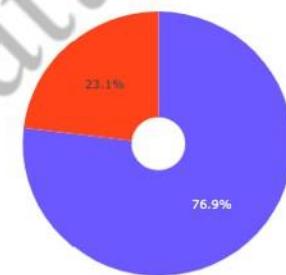
Success rates for SpaceX launches has been increasing relatively with time and it looks like soon they will reach the required target

3

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

4

Decision Tree Classifier Algorithm is the best for Machine Learning Model for provided dataset



Deb



# Appendix

Interactive Plotly

Folium MeasureControl Plugin Tool

Folium Custom Title Layers with Labels

IBM Cognos Visualization Tool

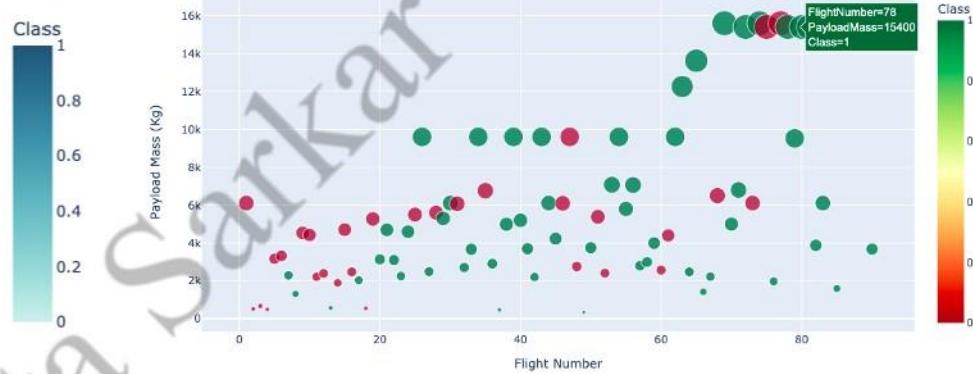
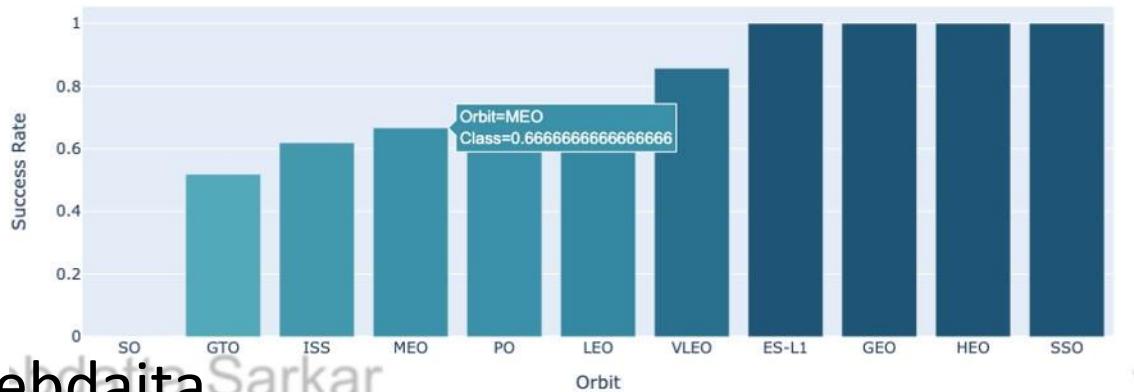
Basic Decision Tree Construction

Debattasarkar

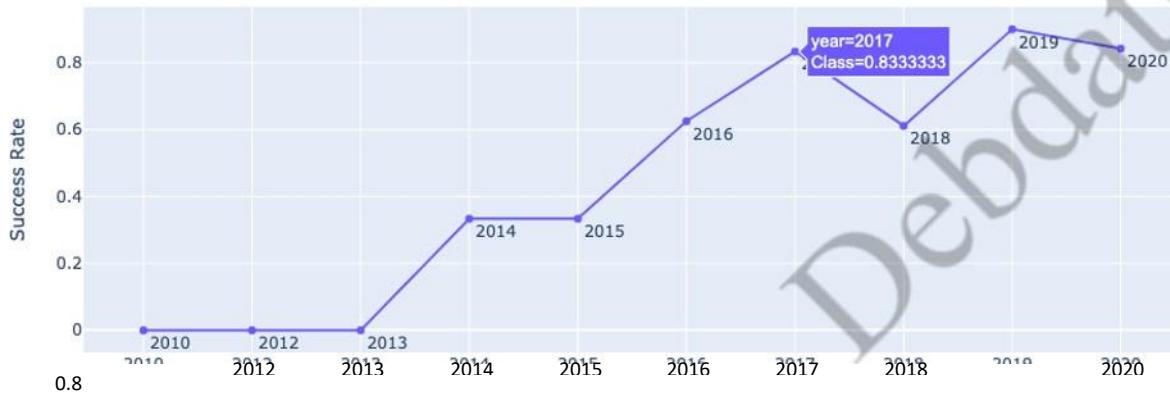


# Interactive Plotly

Used plotly instead of seaborn. They are more interactive and easily customizable as well.



Debdatta Sarkar



Debdatta Sarkar

Code

0.6

0.4

0.8

0.6

0.4

0.2

2010

Year

2019

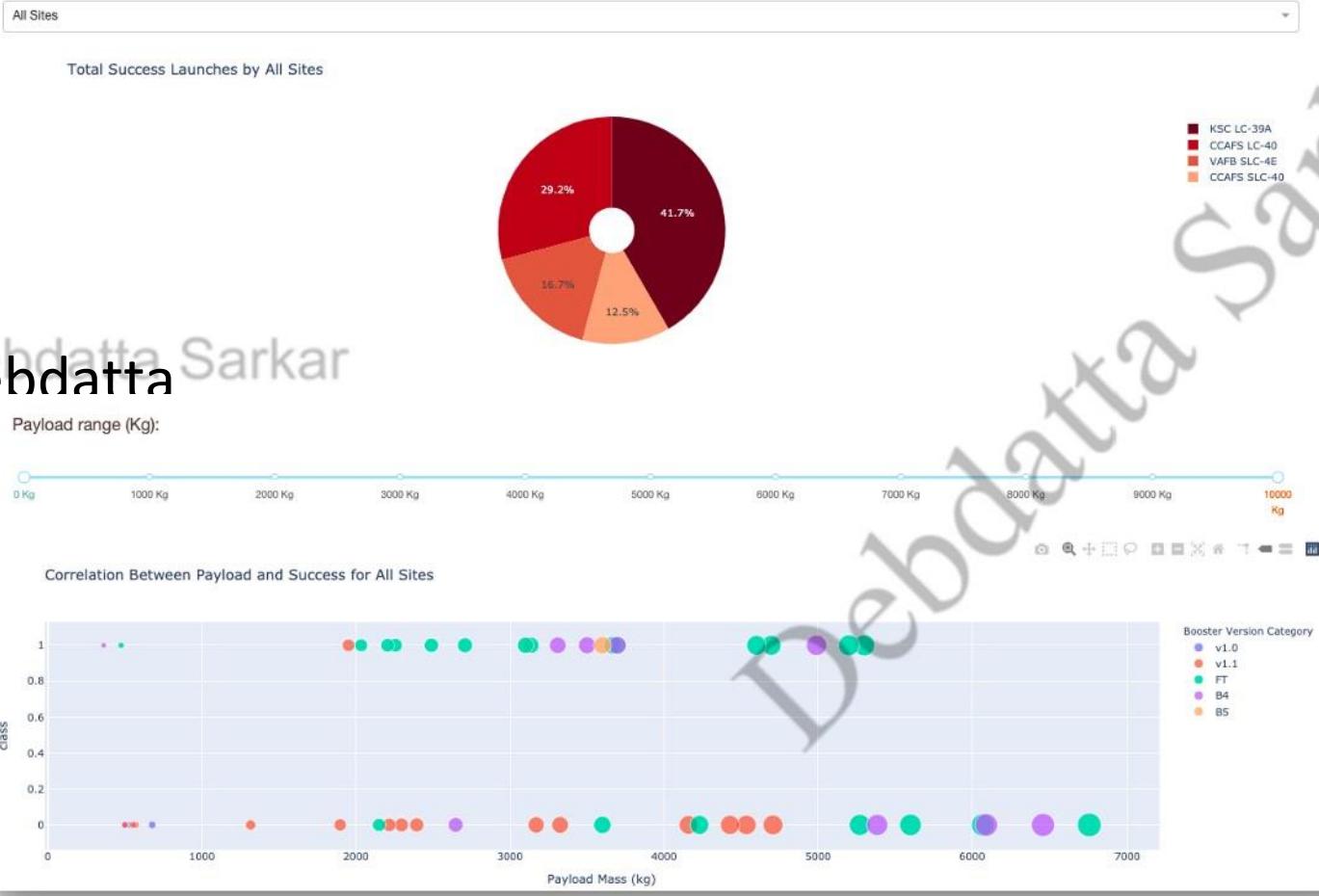
Flight Number

# " Python Anywhere" Live Site for Plotly Dashboard

---

[Code](#)

## SpaceX Launch Records Dashboard



Used "Python Anywhere" to host a live website. The live site dashboard is built with Flask and Dash.

Debdatta Sarkar

[Live Site](#)

Code

# Folium MeasureControl Plugin Tool

---

With Measure Control Plugin Tool, we don't need to write manual distance calculation code and it's very easy to use.

Code

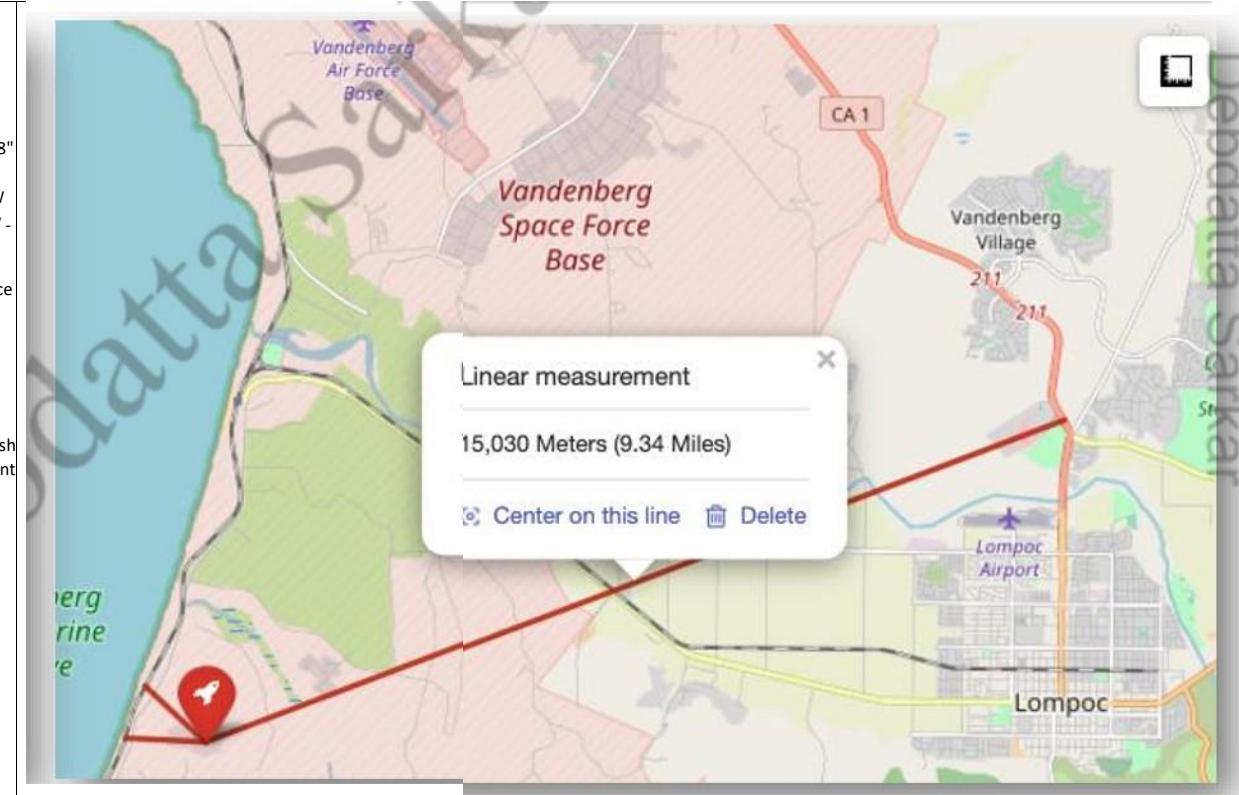
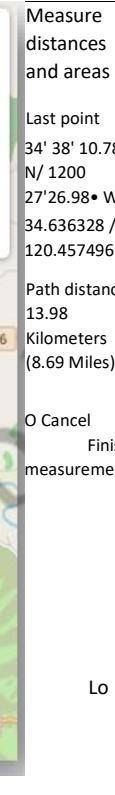
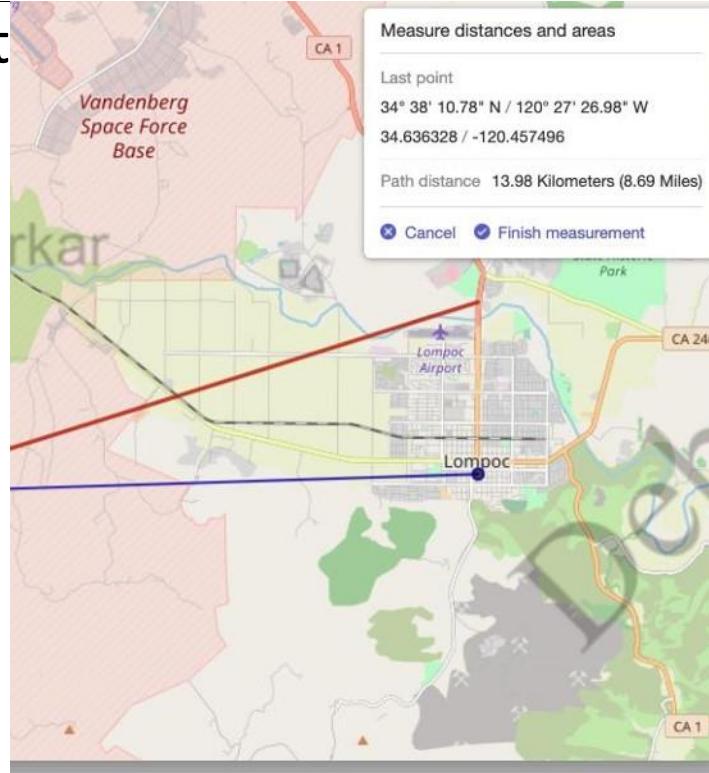
---

```
from folium.plugins import MeasureContr01 site_map. add child (MeasureContr01  
primary_length_unit='kilometers', '#ba2f00')) site_map
```

```
active_color='#0900ba', completed_color=
```

bdat

Vandenberg  
Marine  
Reserve

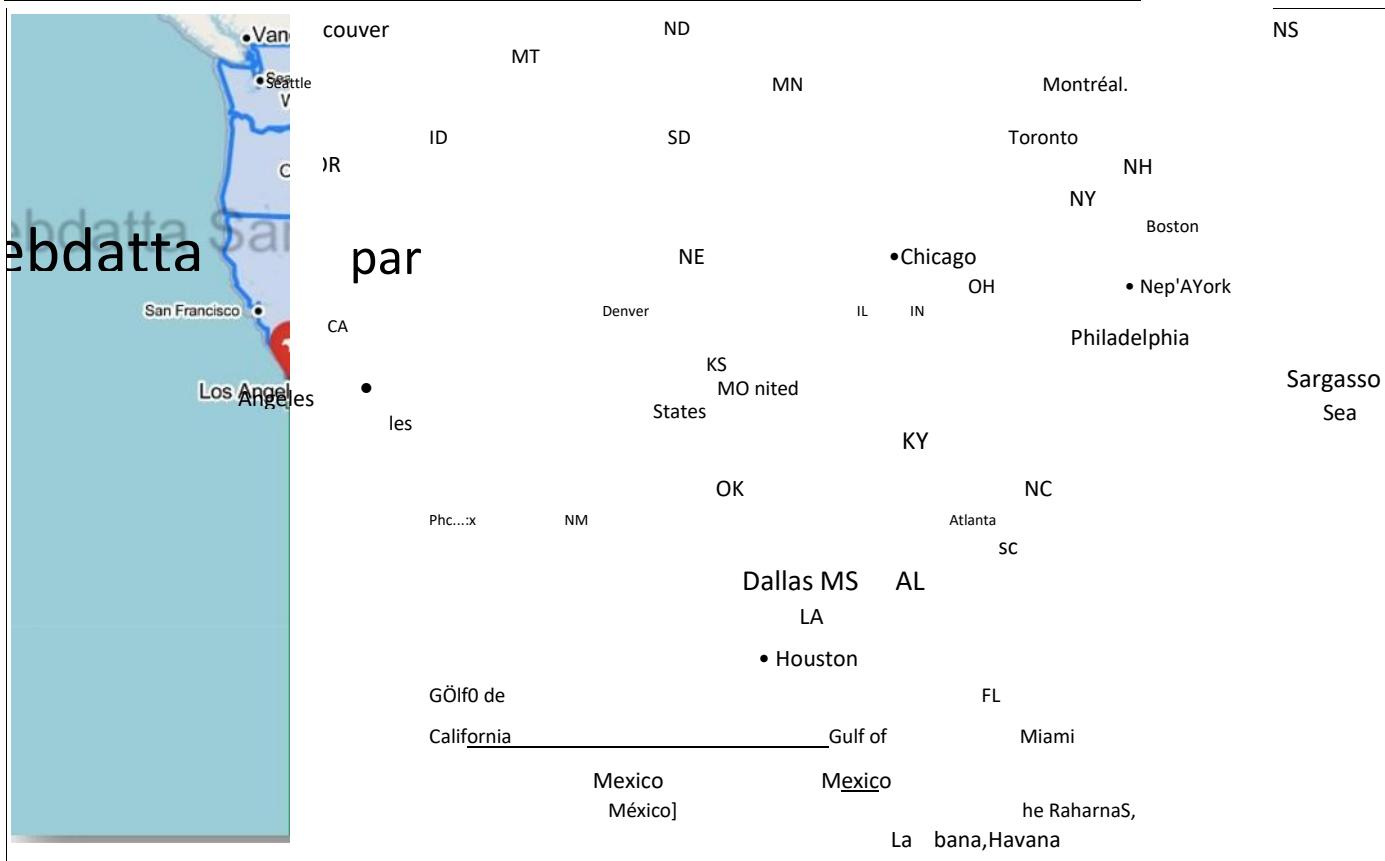


D

## Folium Custom Title Layers with Labels

Code

```
folium.GeoJson(geo_json_data).add_to(site_map).CustomPane("labels").add_to(  
site_map).folium.TileLayer("stamentonerlabels", pane="labels").add_to(site_map).site_map
```



Code

Ciudad  
Guadalajara'  
"de México

Mexico City

Belize

Ciudad de Guatemala

Honduras

El Salvador

It aragua

Cuba

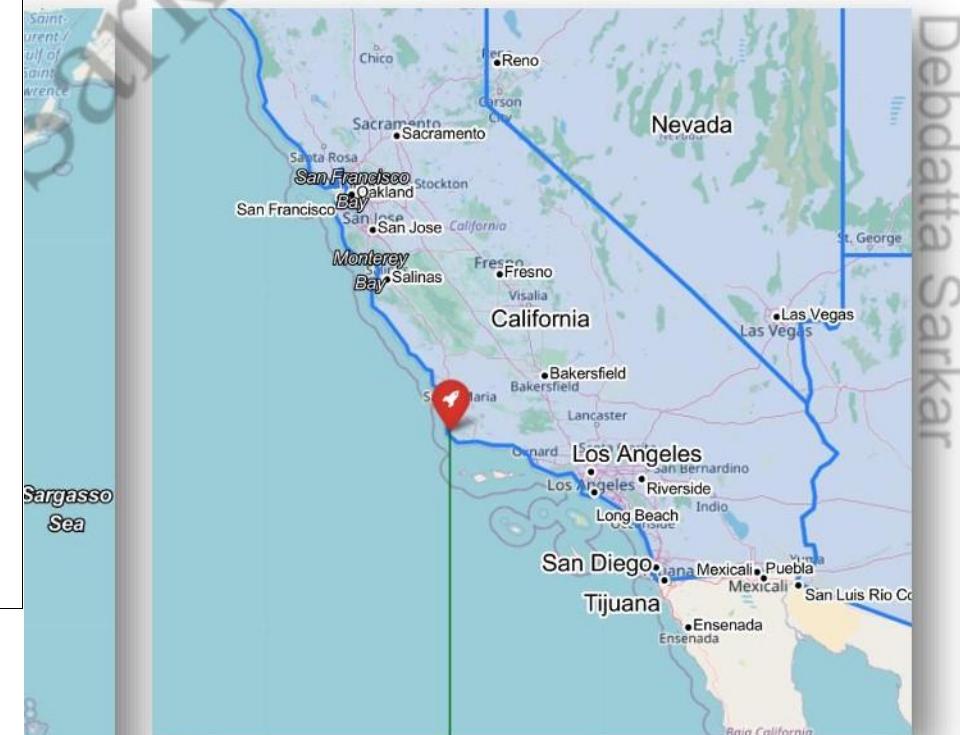
Dominican Republic

Haiti L R

Dominic O' 4---.

locations of launch site in a better way.

Created Custom Title Layer to understand the



[Code](#)

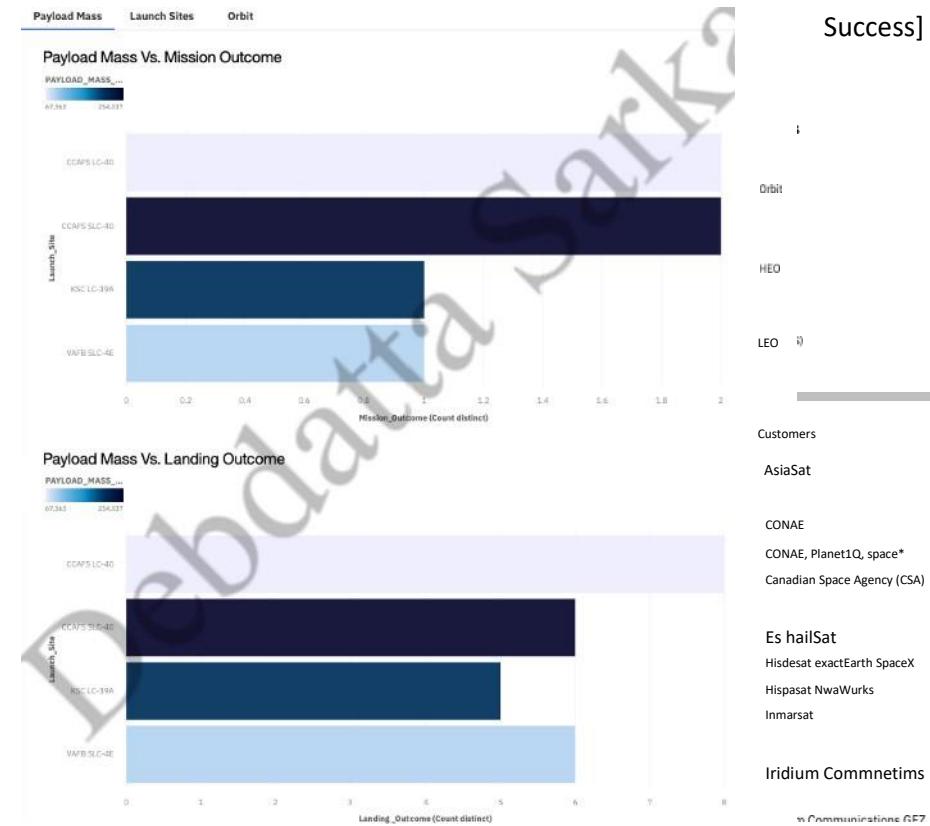
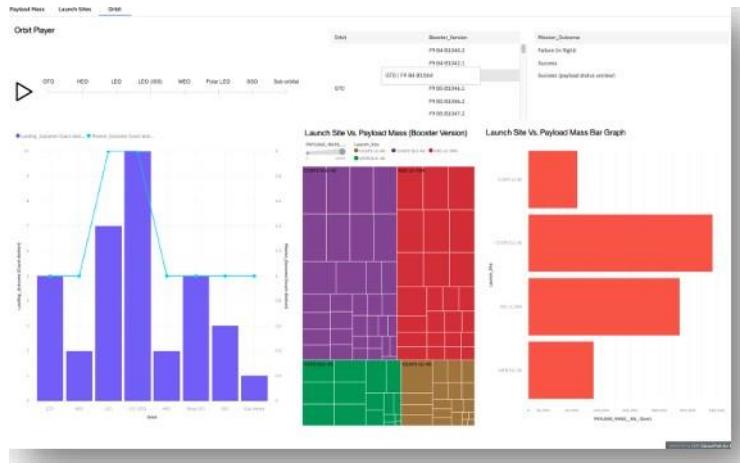
[Code](#)

# IBM Cognos Visualization Tool

IBM Cognos Analytics provides analytic insights that help you to detect and validate important relationships and meaningful differences based on the data that is presented by the visualization.



Debadatta Sarkar



Success] Success

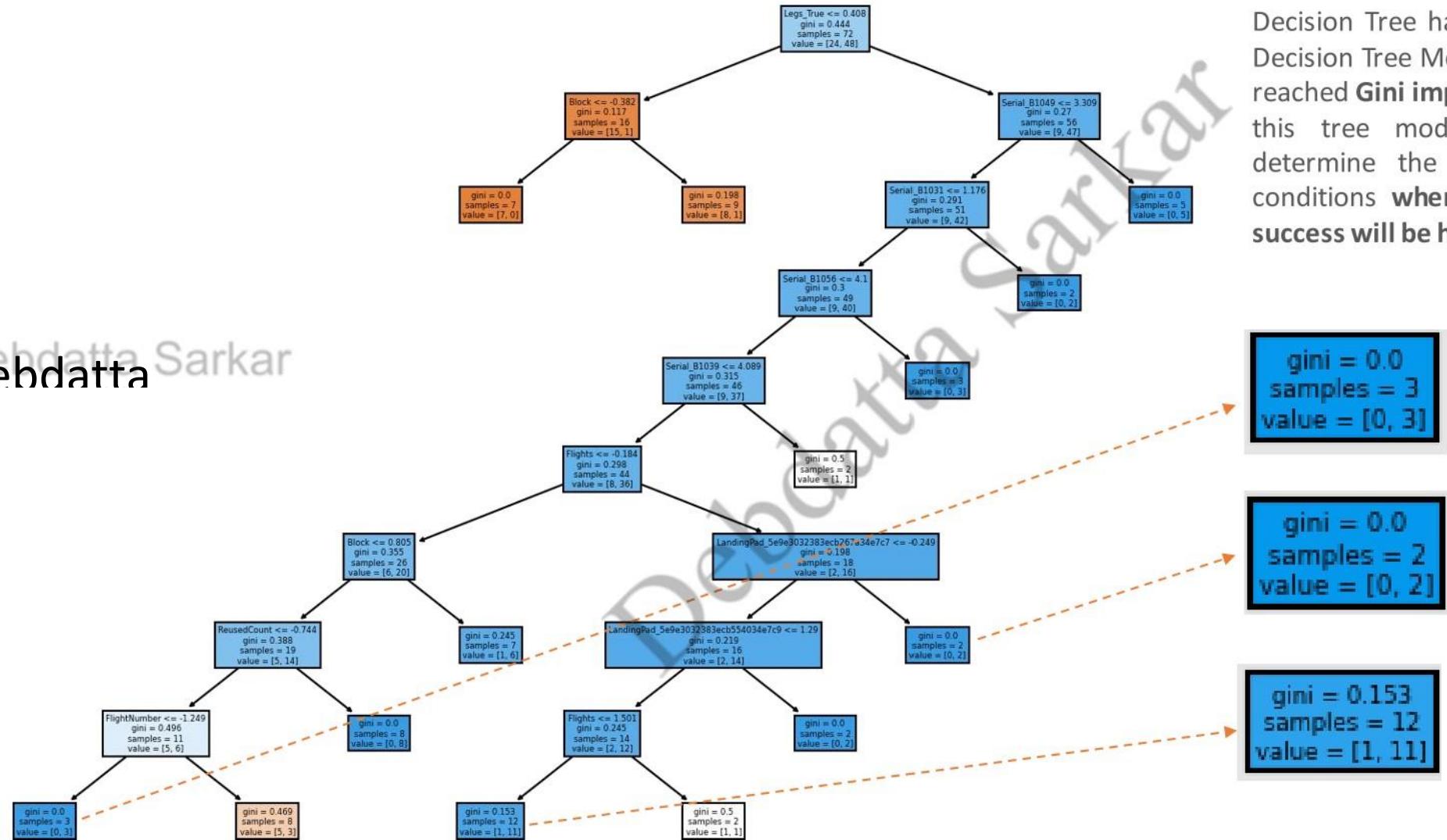
620K

PAYLOAD\_MASS\_KG\_

Catalyst

Dashboard

# Basic Decision Tree Construction



Decision Tree has been constructed, with Decision Tree Model. We see that we have reached **Gini impurity almost near to 0** via this tree model. From this we can determine the correct combination of conditions where the probability of the success will be highest.

Debdatta Sarkar



Thank You! Debdatta Sarkar