Data

Science Capstone

T Mohana
1 september 2024

Project

Outline

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

Executive Summary

Flight Number vs. Payload Mass

Summary of methodologies -

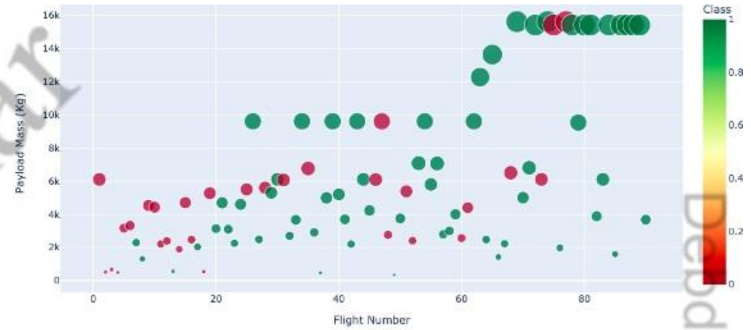
Summary of methodologies -

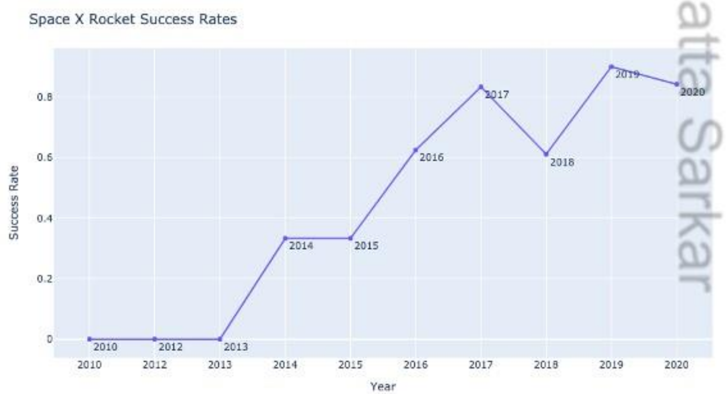
- Data Collection via API, SQL and Web Scraping
- Data Wrangling and Analysis
- Interactive Maps with Folium
- Predictive Analysis for each classification model

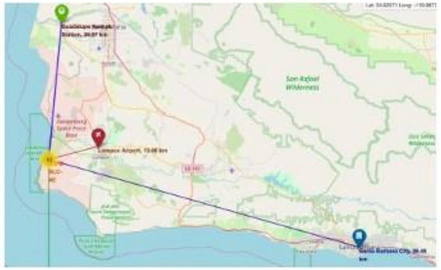
Summary of all results -

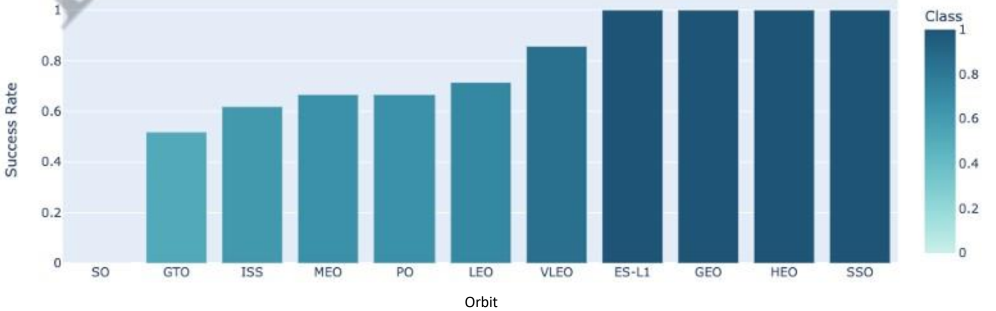
Debdc..

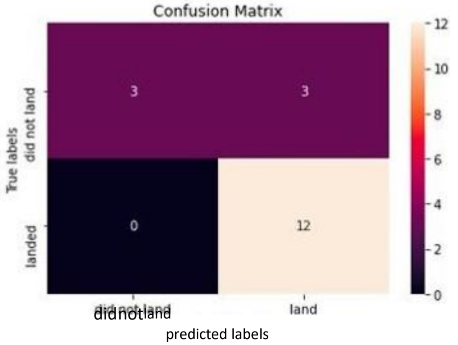
- Data Analysis along with Interactive Visualizations
- Best model for Predictive Analysis











- **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.

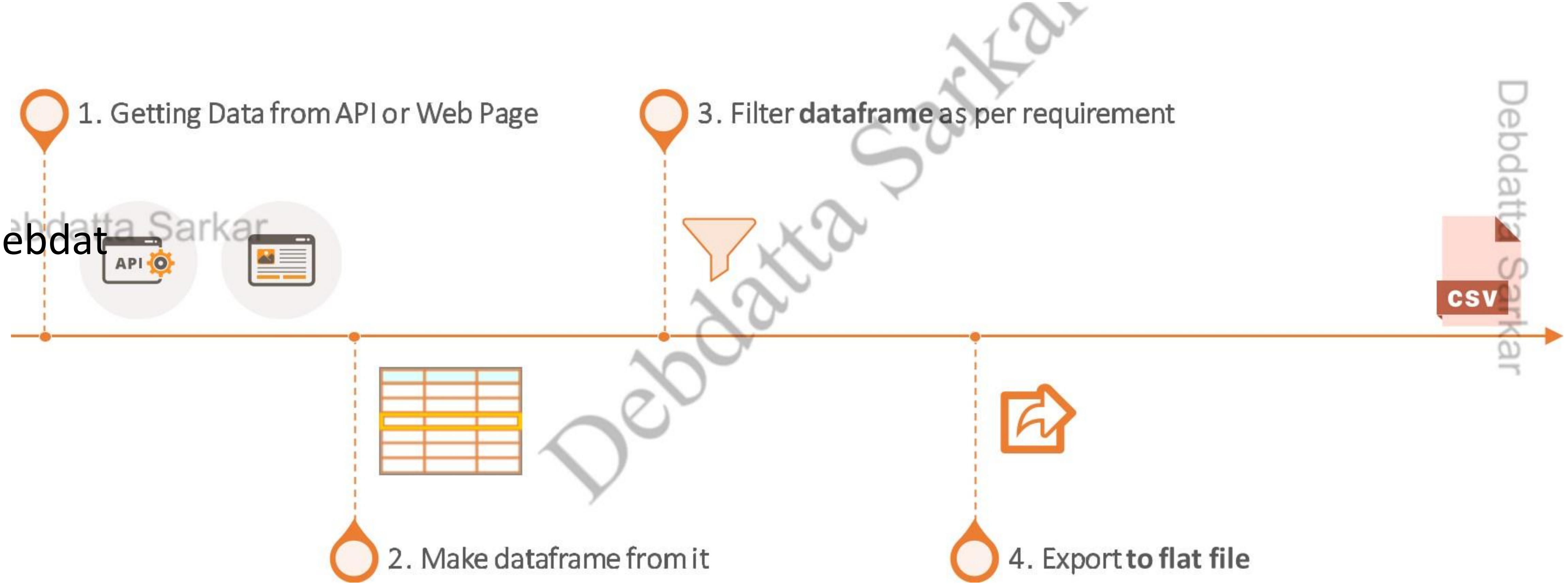
Introduction

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

Methodology

Data Collection



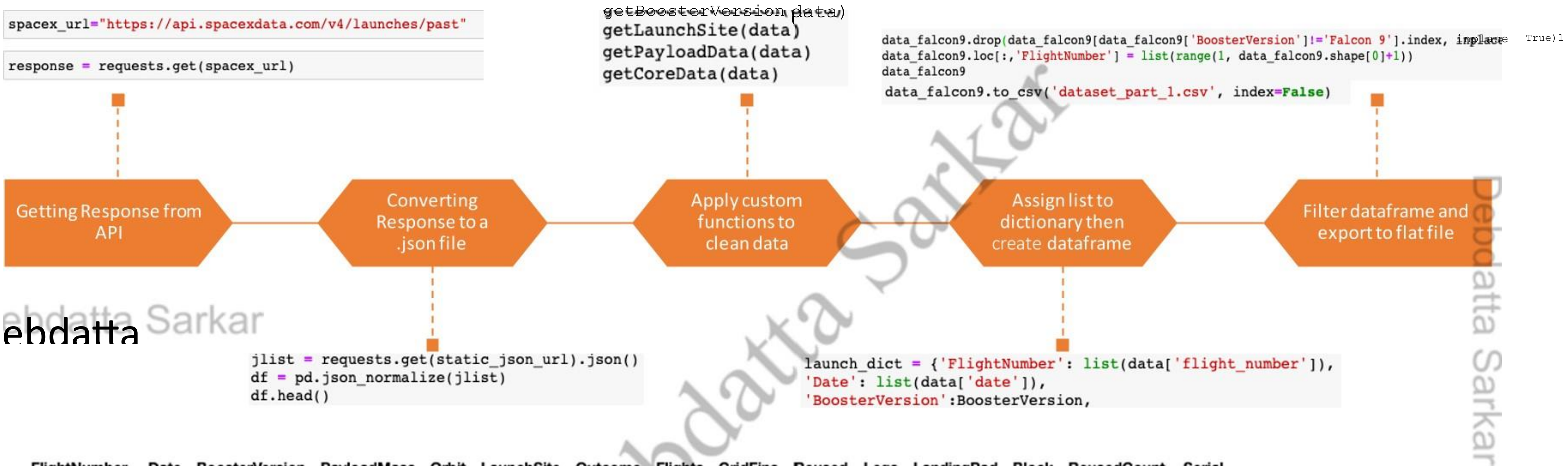
Data Collection

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 ou comes.

Data Collection

-Via SpaceX API

Data Collection



Data Collection

4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003

FlightNumber Date BoosterVersion PayloadMass Orbit LaunchSite Outcome • Fli ghts GridFins Reused Legs LandingPad Block ReusedCount Serial

Data Collection

2013- Falcon 9 3170.000000 GTO CCSFS 1 NoneFalse False False None 1.0 0 B1004 GitHub
URL
8 5
12-03 SLC 40 None

Data Collection

- Via Web Scraping

Data Collection

Getting Response from HTML

Creating BeautifulSoup Object

Finding tables

Getting column names

Creation of dictionary and appending data to keys

Converting dictionary to dataframe

Dataframe to .CSV

```
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)
```

GitHub URL

	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

[GitHub URL]

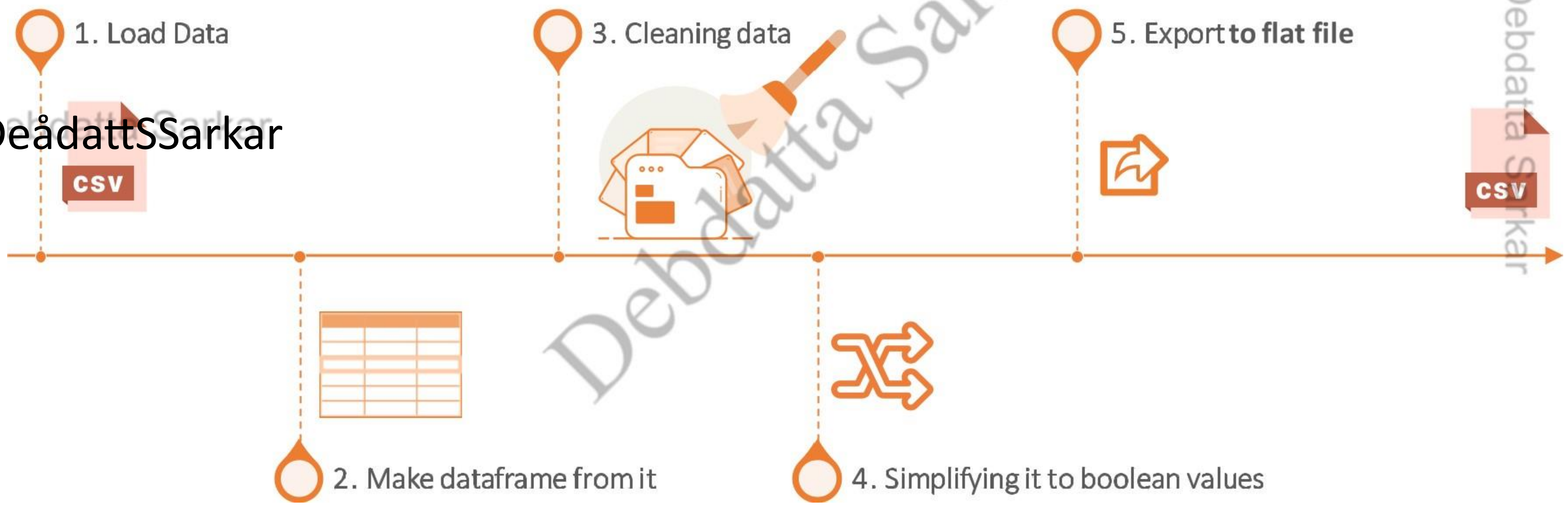
		Payload	Orbit	Customer		Version Booster	Booster landing	Date	Time
CCAFS	Dragon Spacecraft Qualification Unit	o	LEO	SpaceX	Success	F9 VI .OB0003.1	Failure	4 June 2010	18•.45
CCAFS	Dragon	o	LEO		success	F9 VI .0B0004.1	Failure	8 December2010	15:43
CCAFS	Dragon	525 kg	LEO		Success	F9 VI .080005.1	No attempt	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

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

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
```

```
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

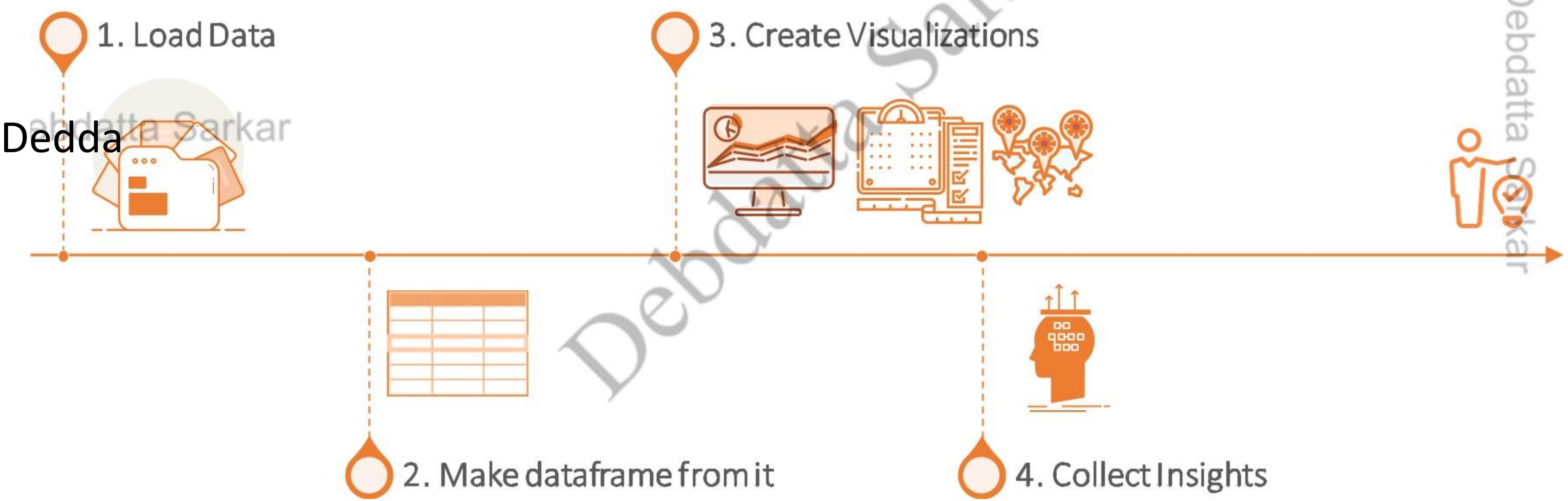
CCAFS None
LEO False False False NaN 1.0 O B0003
SLC 40 None

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
None ASDS	2
False Ocean	2
False RTLS	1

	2	201205-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	O B0005
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	O B0007
	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	ase	NaN	1.0	O B1003
4	5	201312-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None 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.



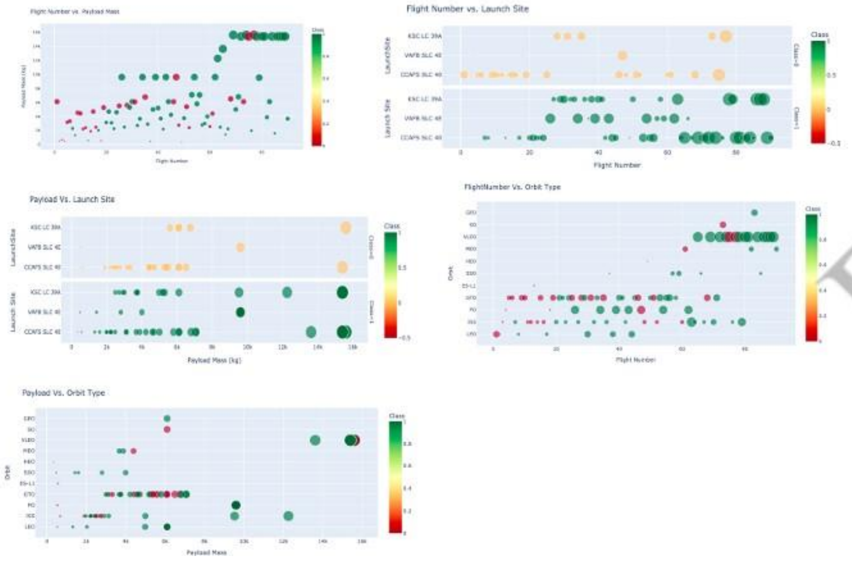
EDA with Data Visualization

Scatter Graphs Drawn:

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.

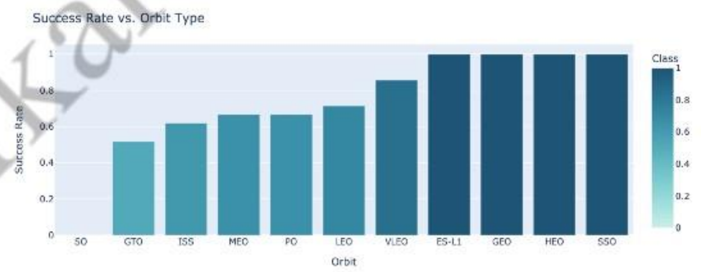


Bar Graph Drawn:

Bar Graph Drawn:

Success Rate VS. Orbit Type

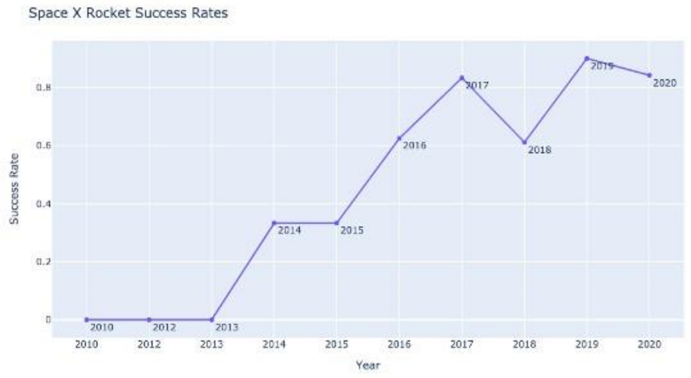
Bar graphs are easiest to interpret a relationship between attributes. Via this bar graph we can easily determine which orbits have the highest probability of success.



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.

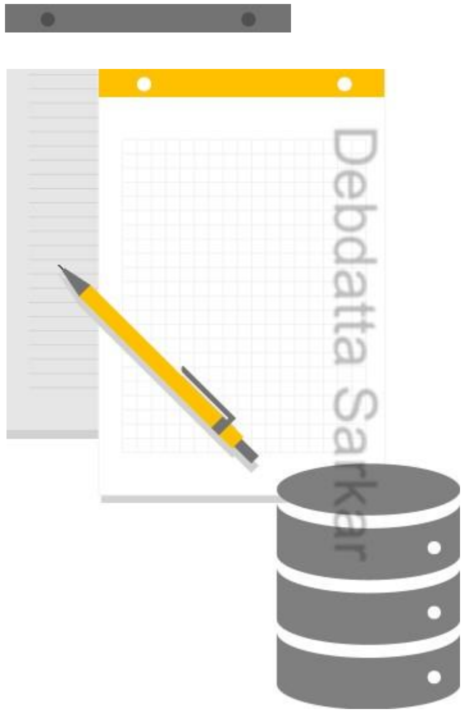


[Seaborn Graphs GitHub URl]

Plotly Graphs GitHuh URL

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>
```


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 of success in relation to launch site

Percentag of succes in relatio to launchsi

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
Dash and its components	import dash Plotly stewards With Dash Open Source, dash_html_components as Component library contains import more. as html as dcc Input, Output	local laptop or server. The Da sliders, graphs, dropdowns, t Dash provides all of the avail Pyt on's.leaåuRdatavizand UI libraries. Dash apps run onWÖur import html local laptop or serve . he sh Core a set of higher-level components dash_core_components as dcc , tables, and like
Debdatta Sarkar from dash.dependencies import Input, user-friendly Python classes.		Output t e available HTML tags as
		Fetching values from CSV and
		Plot the graphs with interactive plotly library
		Create a dropdown for launch sites
Pandas	import pandas as pd rom	CSV and creating a dataframe co
		Create a rangeslider for Payload Mass range selection
Plotly	import plotly.express as PX	Creating the Pie graph for Success percentage display
Dropdown	dcc.Dropdown(	Creating the Scatter graph for correlation display

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

GitHub Code URL

Used "Python Anywhere" to hosta 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() tra n s form = prep rocessin g.Sta ndardScaler() X = tra ns form .fit(X). transform (X)

X_tra i n, X_test, Y_train, Y_test = tra in_test_split(X, y, te ra ndo ta

_test.shape

a l gorith ms = KNN ' :knn_cv.best_score_,

Decision Tree':tree cv.best score

Building Model

Standardize and transform data

Logi stic Regression' best_score_}

Split data into training and testdata setsbest_algorithm = ma x(algorith ms, key= lambda x:

Check how many test samples has been created

a l gorith ms

List down machine learning algorithms we want to use

Set our parameters and algorithms to GridSearchCV

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

lh Ing Best Performing Classification Modéj}

The model with best accuracy score wins the best performing model

Debda+,La.Sar. a

CO

Best Model

oge

Evaluating Model

yh a t=a lgorithm .pred
pl ot_co nfusion_m yh at)

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

GitHub URL

Results

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

EDA with Visualization

Flight Number vs. Launch Site

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

KSC LC 39A

Launch Site

VAFB SLC 4E

CCAFS SLC 40

-0.5

Sea horn Graphs GitHuh URL

Flight Number

PI Qtly Gra GitHuh URL

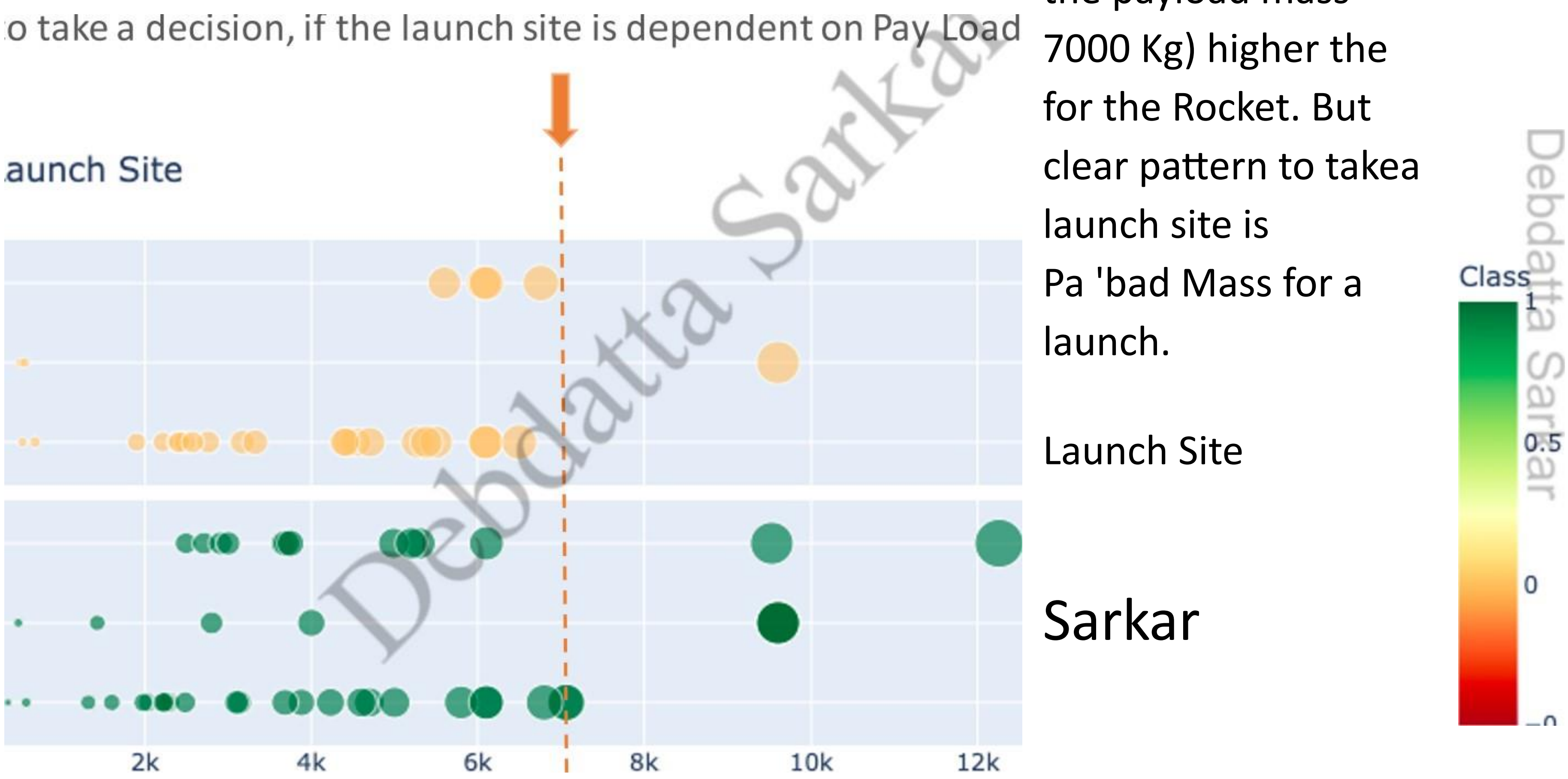
Payload vs. Launch Site

GitHuh

GitHuh

Payload Vs. Launch Site

- The greater the payload mass (greater than 7000 Kg) higher the success rate for the Rocket. But there's no clear pattern to takea decision, if the launch site is dependent on Pa 'bad Mass for a success launch.



KSC LC 39A

De%atta Sarkar

VAFB SLC 4E

CCAFS SLC 40

KSC LC 39A

Graphs GitHuh

PI Qtly Gra GitHuh

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

- ES-LI, GEO, HEO, SSO has

Success Rate

sso highest Sucess rates.

Graphs GitHuh

PI Qtly Gra GitHuh

Debcytta Sarkar

0.6

0.4

0.2

SO

Orbit

Sea horn Graphs IJR[

Plotly Graphs IJRI

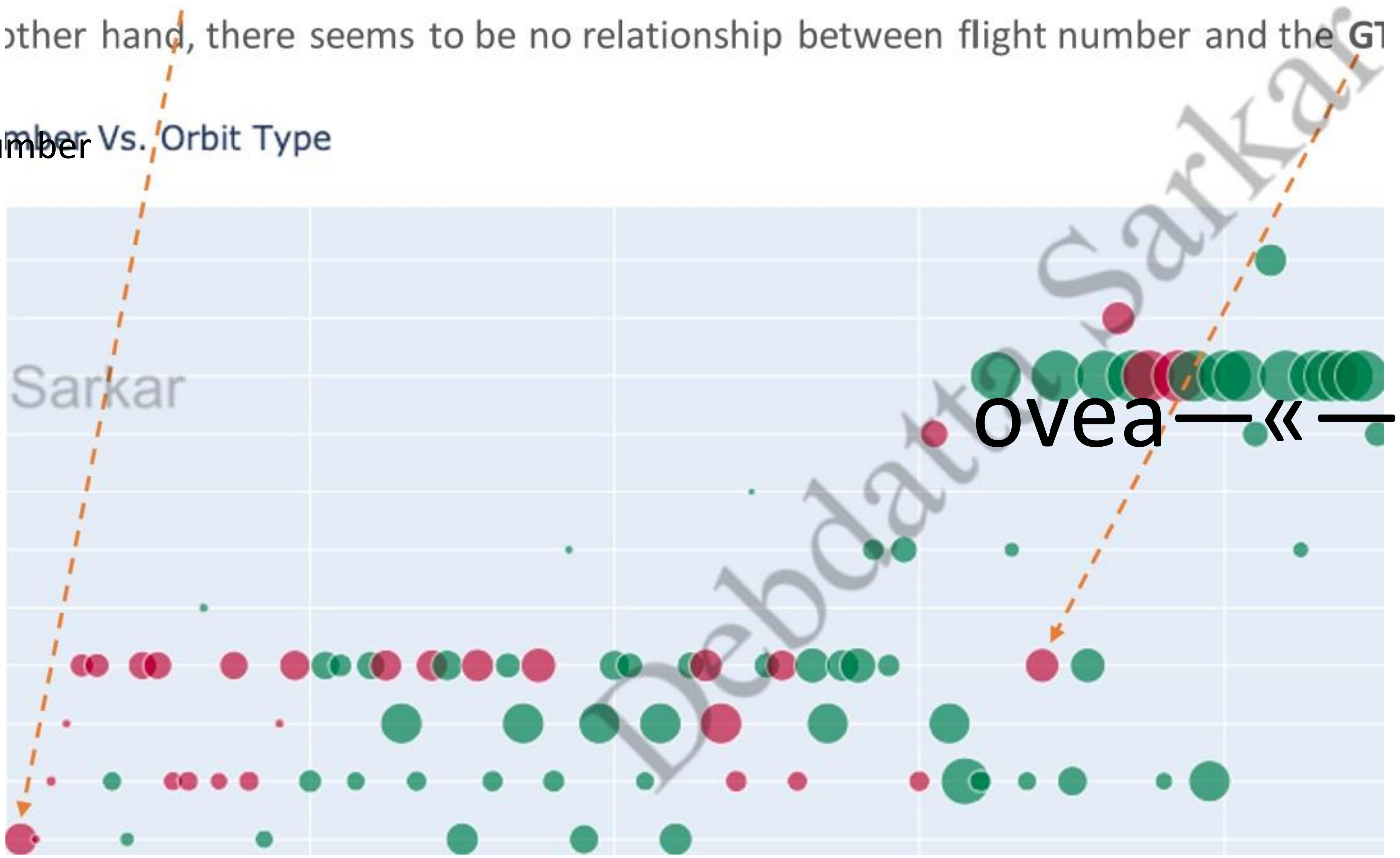
Flight Number vs. Orbit Type

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

GitHuh

GitHuh

FlightNumber



GTO

PO
0.2
ISS

ovea—«—

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

so

MEO

ES-LI
0.4

GEO Class

DebYE?tta
Sarkar0.8

HEO0.6 sso

Graphs GitHuh

PI Qtly Gra GitHuh

LEO

20 40 60 80

Flight Number

Sea horn URL

URL

GitHuh

GitHuh

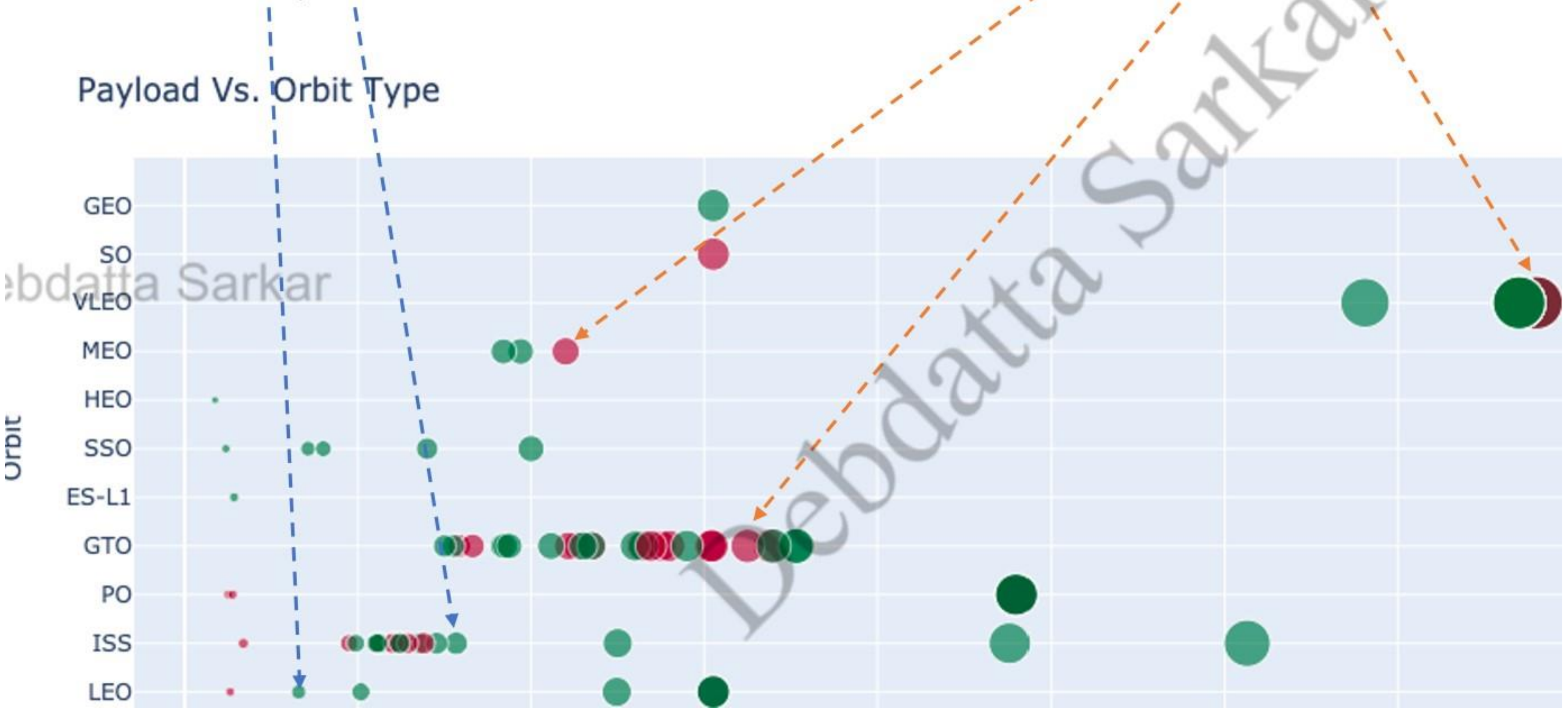
Payload vs. Orbit Type

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

GitHub

PI Qtly GitHub

- Positive on LEO, ISS orbits



GEO

SO

Debde&ga SarÆar

M
E
O

H
E
O
s
s
o

ES-LI

GTO

ISS

LEO

2k

4k

6k 8k 10k 12k 14k 16k

Payload Mass

Class

0.8

0.6

Debdatta Sarkar

0.4

0.2

Seaborn Graphs URL

Graphs URL

Launch Success Yearly Trend

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

2020

co

2010 2012 2013
2014 2015 2016
2017 2018 2019
2020

Sea horn IJRI

URI

With SQL

All Launch Site Names

SQL Query

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

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

Description

Desqription

Using the word DISTINCT in the query we pull uniqu VGi'åes for Launch_Site

column from table SPACEX.

Launch Site Names begin with 'CCA

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 withzWild 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
201210-08	00:35:00	F9 1.0 B0006	CCAFS LC40	SpaceX CRS-I		LEO OSS)	NASA (CRS)		No attempt
201303-01	15:10:00	F9 VI .0 B0007	CCAFS LC40	SpaceX CRS-2	677	OSS)	NASA (CRS)		No attempt

co

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

CO

YLOAD MASS KG and WHERE clause filters the datå 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

SQL Query

```
%sql SELECT MIN(DATE) AS "First Succesful 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 Succesful Landing Outcome in Ground Pad
2015-12-22

gave.ln the column Date and WHERE clause filters th€)
éöme with values "Success (ground pad)".

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

kar

e

```
*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_Ou co e: Success (drone ship)

AND clause specifies additional filter co Ltio s
Payload_MASS KG >4000 AND Payload_M
ss_KG < 6000

ing_Outcome = Success (dr

ditions

Total number of Successful and Failure Mission

booster_version

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;
```


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"

Successful Mission	Failure Mission
100	1

Selecting multiple count is a complex query. Ih
v within sub query for getting both success
query.

Case when MISSION OUTCOME LIKE 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);
```

Desqription

Description

Using the function MAX works out the maximum payload in the column PAYLOAD_MASS__KG_ in sub query.

Using
PAYLOAD_MASS

WHERE clause filters Booster Version which had that maximum payload.

WHERE

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 displä month names, failure landing_outcomes in booster versions, launch site for the months in a 015.

Via year function we extract the ye clause 'Failure (drone ship)' fetches our

ich will display the
omes in drone ship,
nonths in year 2015.

ear and future where
r required values.

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

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

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

SQL Query

```
%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 ;
```


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

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

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 [1]
AND DATE < '2017-03-20'

Interactive map with Folium

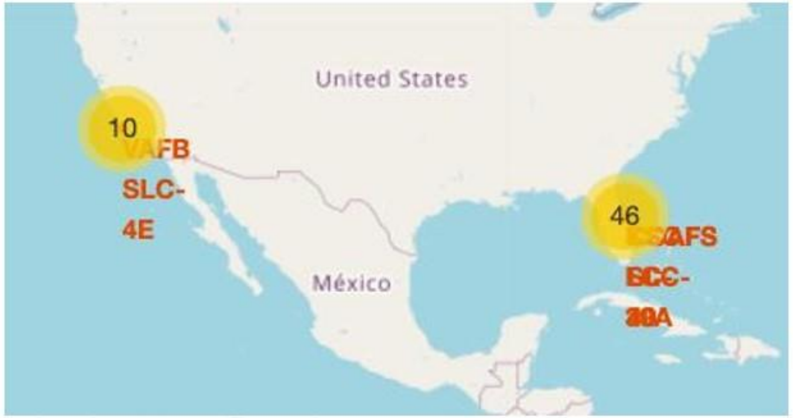
All Launch Sites on Folium Map

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

Debdatta

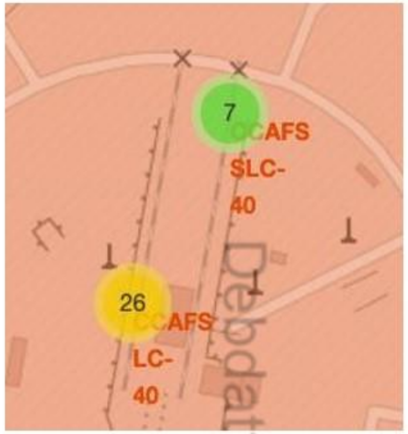
VAFB
SLC-
4E
CCAFS
SLC-
39A
United States
México
Colombia
United Kingdom
France
España
Santa Maria
Bakersfield
Lancaster
Oxnard
Santa C
Los Angele
KSC
LC-
39A
CCAFS
SLC-
40
Cape Canaveral
Debdatta Sarkar

Color Labeled Launch Records



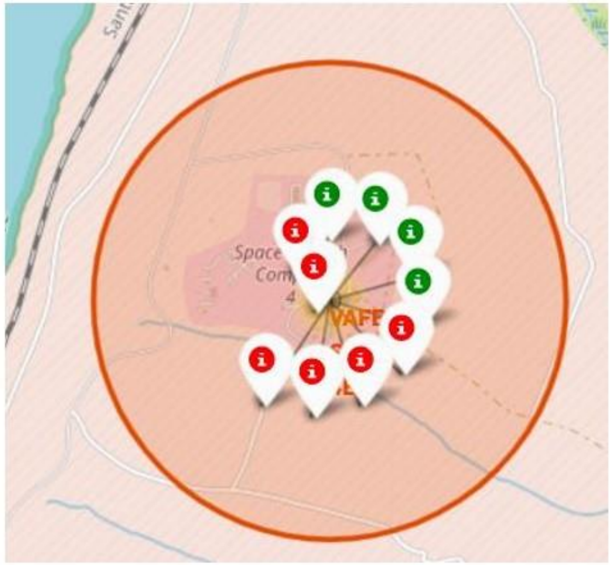
Green Marker shows successful launches and **Red Marker** shows failures.

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

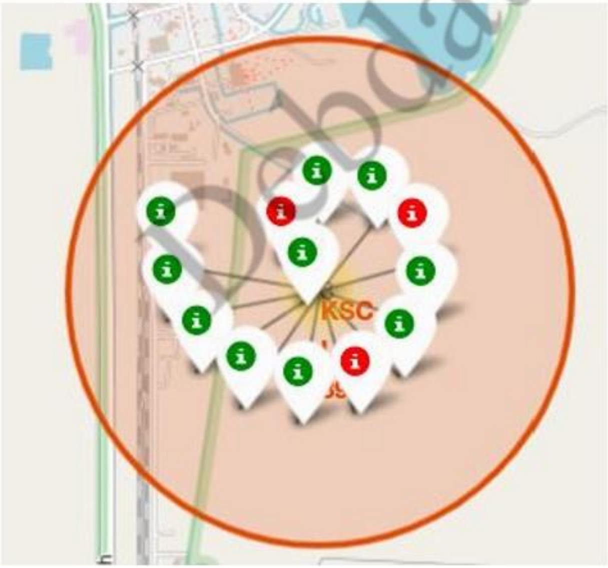


Debdatta

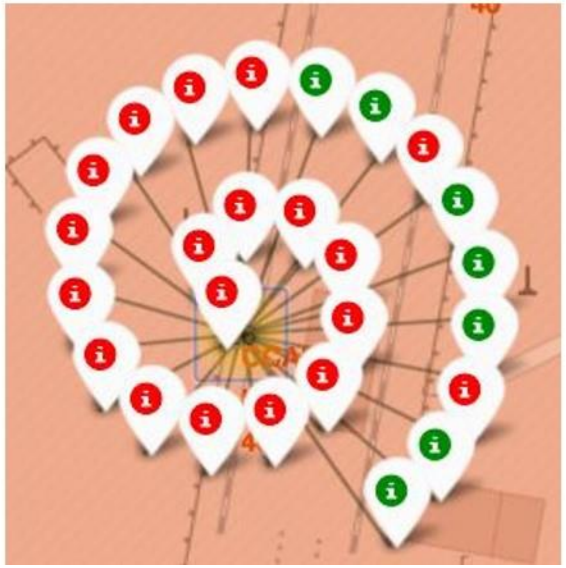
VAFB SLC-4E



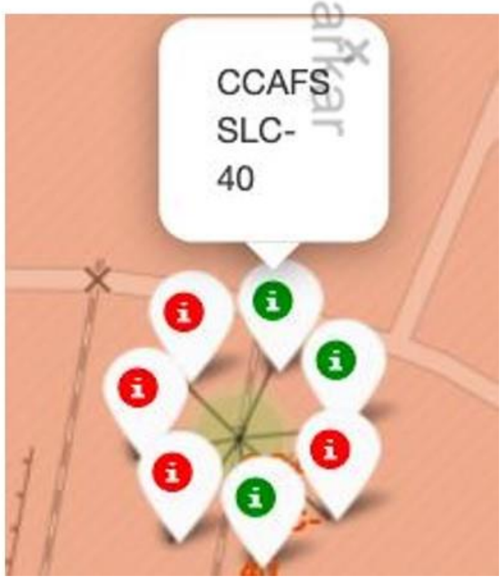
KSC LC-39A



CCAFS LC-40



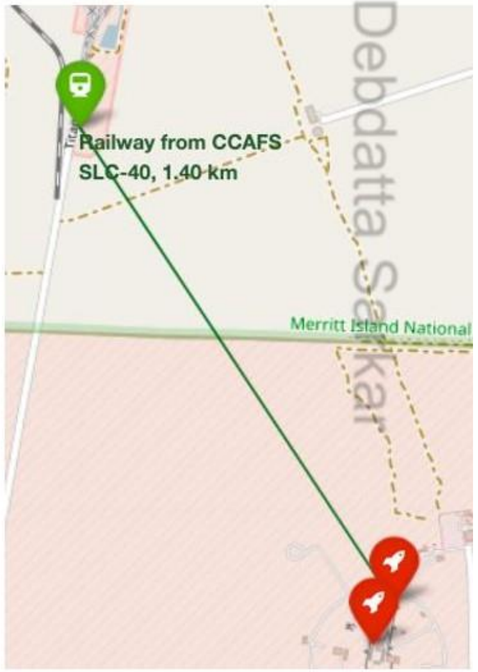
CCAFS SLC-40



Launch Site Distances from Equator & Railways

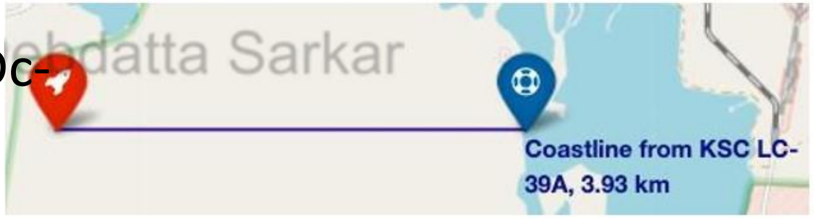
Distance from Equator is greater than 3000 Km for all sites.

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

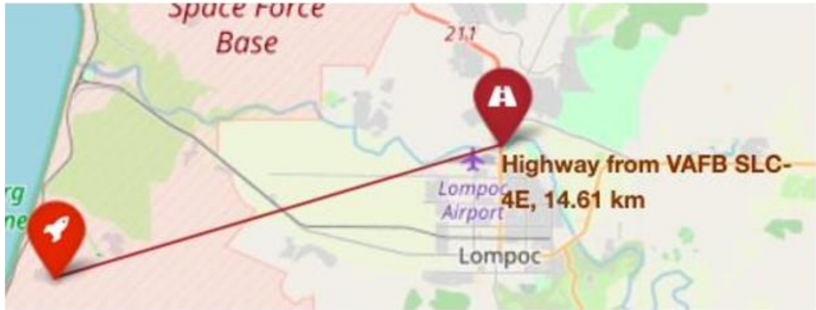
Dc-



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



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 Km > distance > 3000 Km)
- Are launch sites in close proximity to railways?

  Yes (2 Km > distance > .5 Km)
- Are launch sites in close proximity to highways?

  No (15 Km > distance > 5 Km)
- Are launch sites in close proximity to coastline?

  Yes (5 Km > distance > .5 Km)
- Do launch sites keep certain distance away from cities?

  Yes (15 Km > distance > 80 Km)

Build a Da hboard with Plotly Dash

Launch Success Count for All Sites

SpaceX Launch Records Dashboard

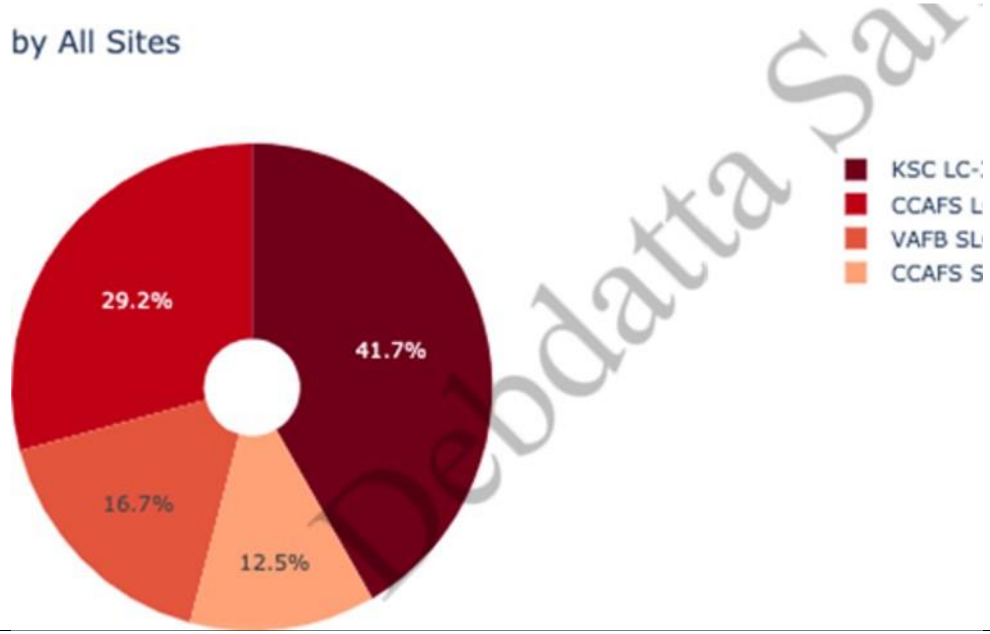
We can see that KSC LC-39A had the most successful launches from all the sites.

Total Success Launches by All Sites

KSC LC.39A

Debdatta Sarkar

CCAFS LC.40



VAFB SLC-4E

CCAFS SLC-40

All Sites

Live Site URL Code

Payload vs. Launch Outcome Scatter Plot for All Sites

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

Low Weighted Payload Okg — 4000kg

0.8

0.4

Booster Version Category

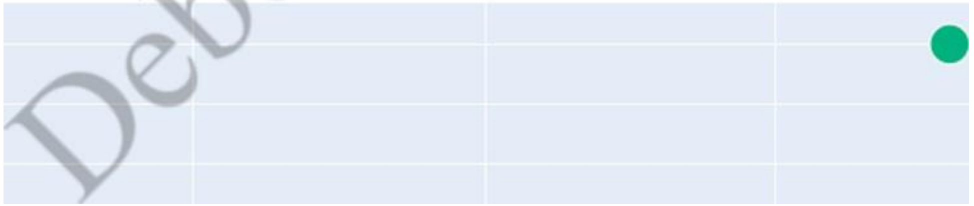
VI.o

VI.I

35

Débdatta Sarkar

1000 1500



Low Weighted Payload 4000kg-10000kg

Booster Version Category

• VI.I

0.8

0.6

0.4

0.2

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

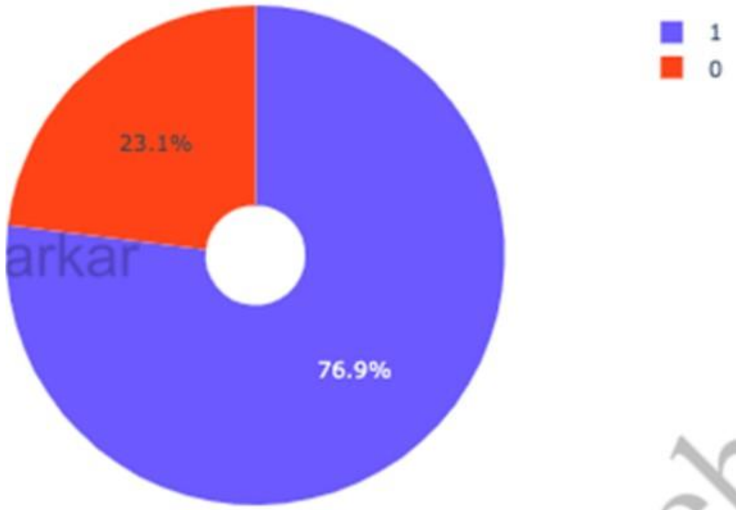
KSC LC-39A achieved a 76.9% success rate while getting a

lysis 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:

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

dashboard, we are

site has the highest
rate?

co

range(s) has the
success rate?

range(s) has
launch
success rate?

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

Live Site URL Code

Predictive analysis (Classification)

Confusion Matrix

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

Predicted Values
Predicted Values

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

Actual Values

Deb@a

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

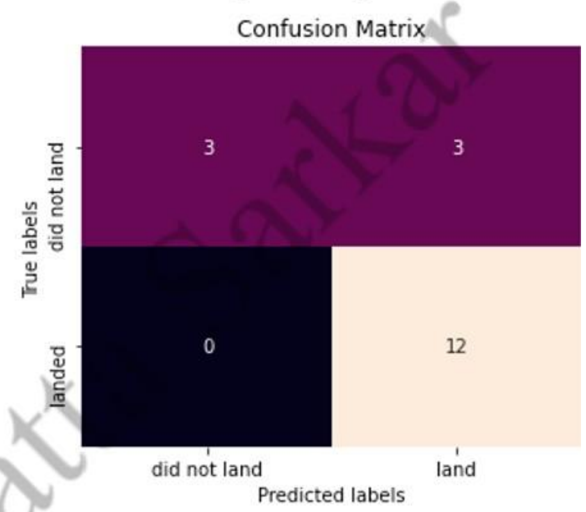
True Negative Rate: TN/Actual No = 3/6 = 0.5

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

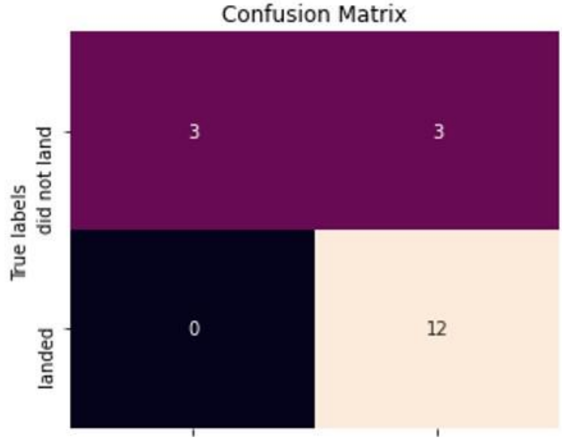
Prevalence: Actual yes/Total = 12/18 = 0.6667

Predicted

Logistic Regression



Decision Tree



Yes

True
Positive
TP = 12

Total Cases

3 15 = 18

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

Misclassification Rate: (FP+FN)/Total = (3+0)/1

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

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

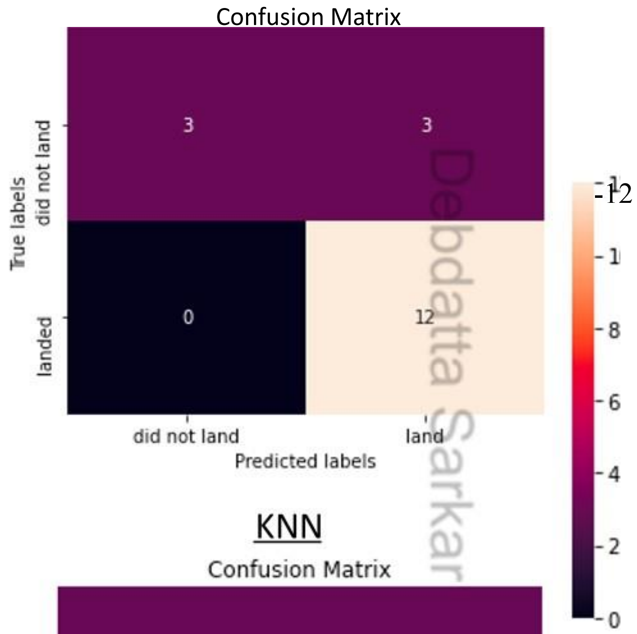
True Negative Rate: TN/Actual No = 3/6 = 0.5

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

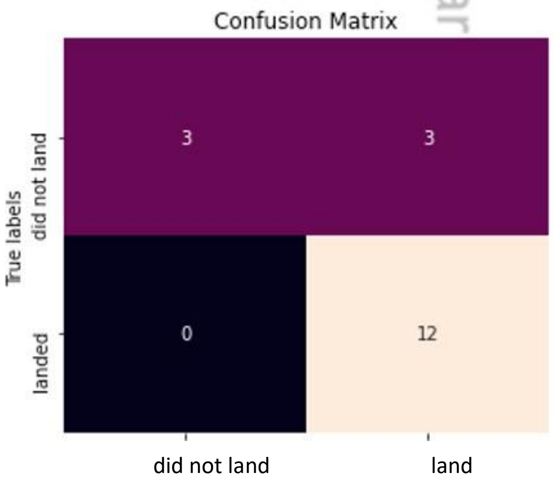
Prevalence: Actual yes/Total = 12/18 = 0.6667

did not land land
Predicted labels

SVM



KNN



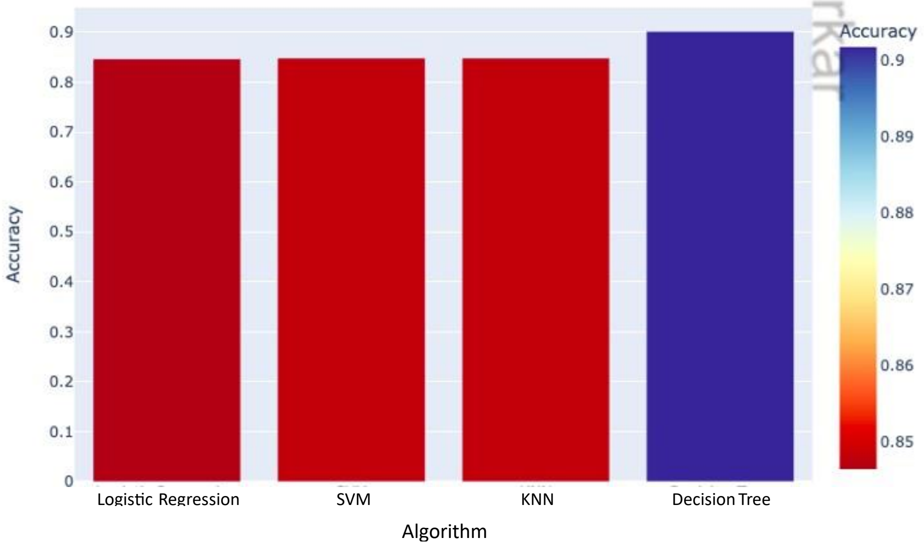
Predicted labels

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'}

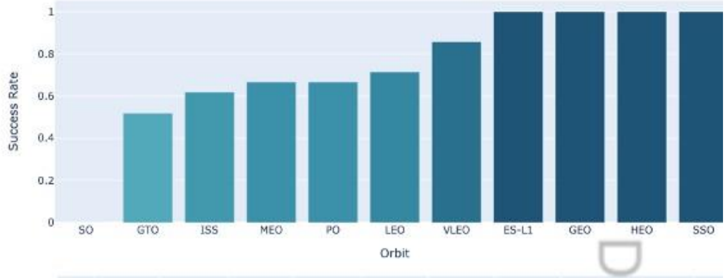
We trained four different models which each had an 83% accuracy rate.



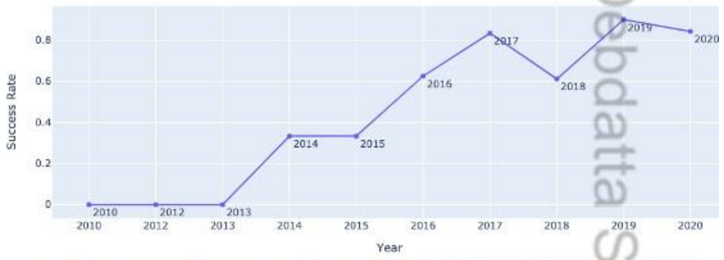
As you can see our accuracy is extremely close, but we do have a clear winner which performs best - "Decision Tree"

Conclusion

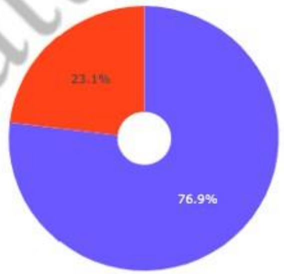
1 Orbits ES-L1, GEO, HEO, SSO has highest Sucess rates

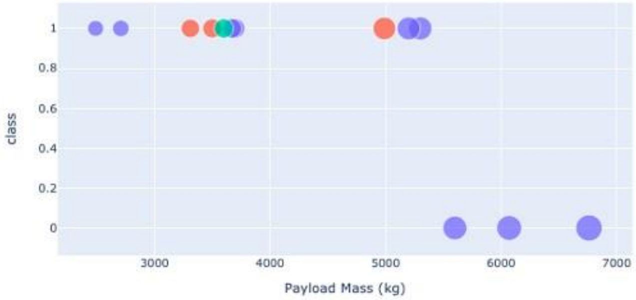


2 Success rates for SpaceX launches has been increasing relatively with time and it looks like soon they will reach the required target



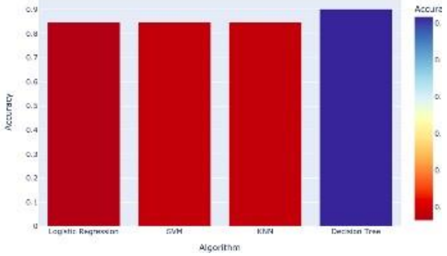
3 KSC LC-39A had the most successful launches but inceasing payload mass seems to have negative impact on success





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

Algorithm	Accuracy
Logistic Regression	0.846429
SVM	0.848214
KNN	0.848214
Decision Tree	0.901786

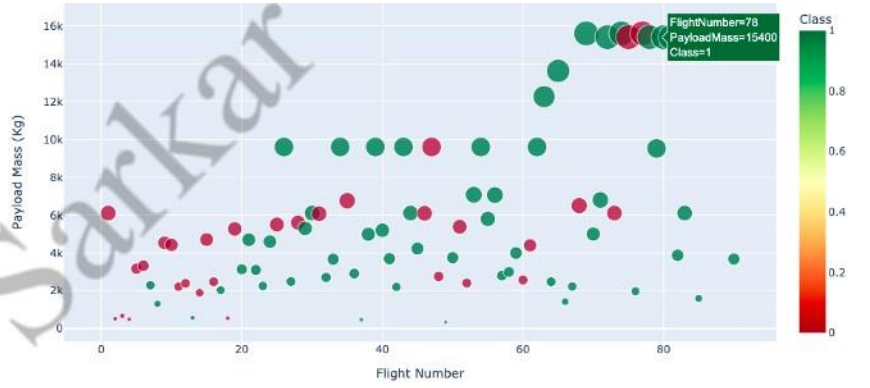


- Interactive Plotly
- Folium MeasureControl Plugin Tool
- Folium Custom Title Layers with Labels
- IBM Cognos Visualization Tool
- Basic Decision Tree Construction

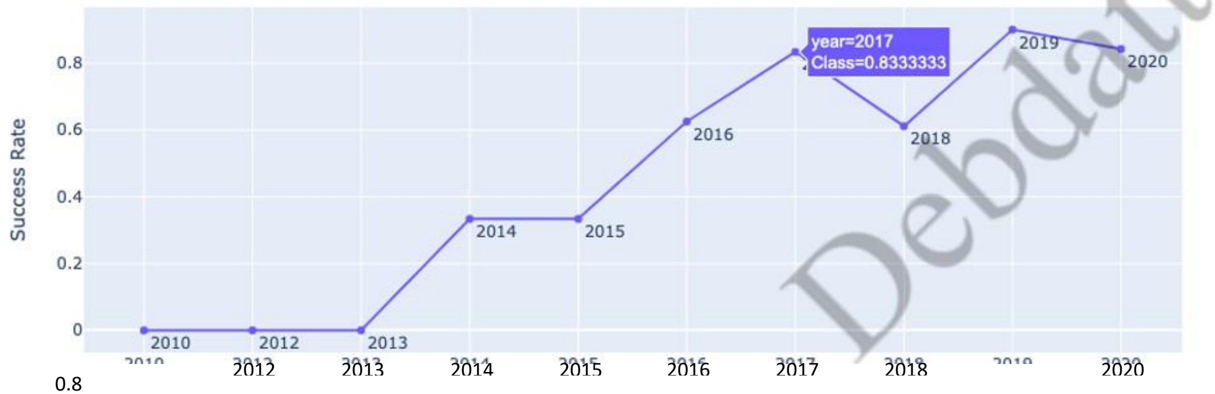
Appendix

Interactive Plotly

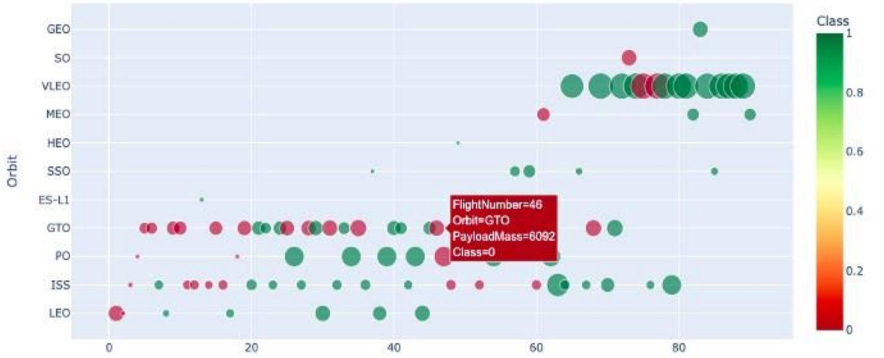
Used plotly instead of seaborn. They are more interactive and easily customizable as well.



Debdaita



0.8



Code

0.6

0.4

0.8

0.6

0.4

0.2

2010

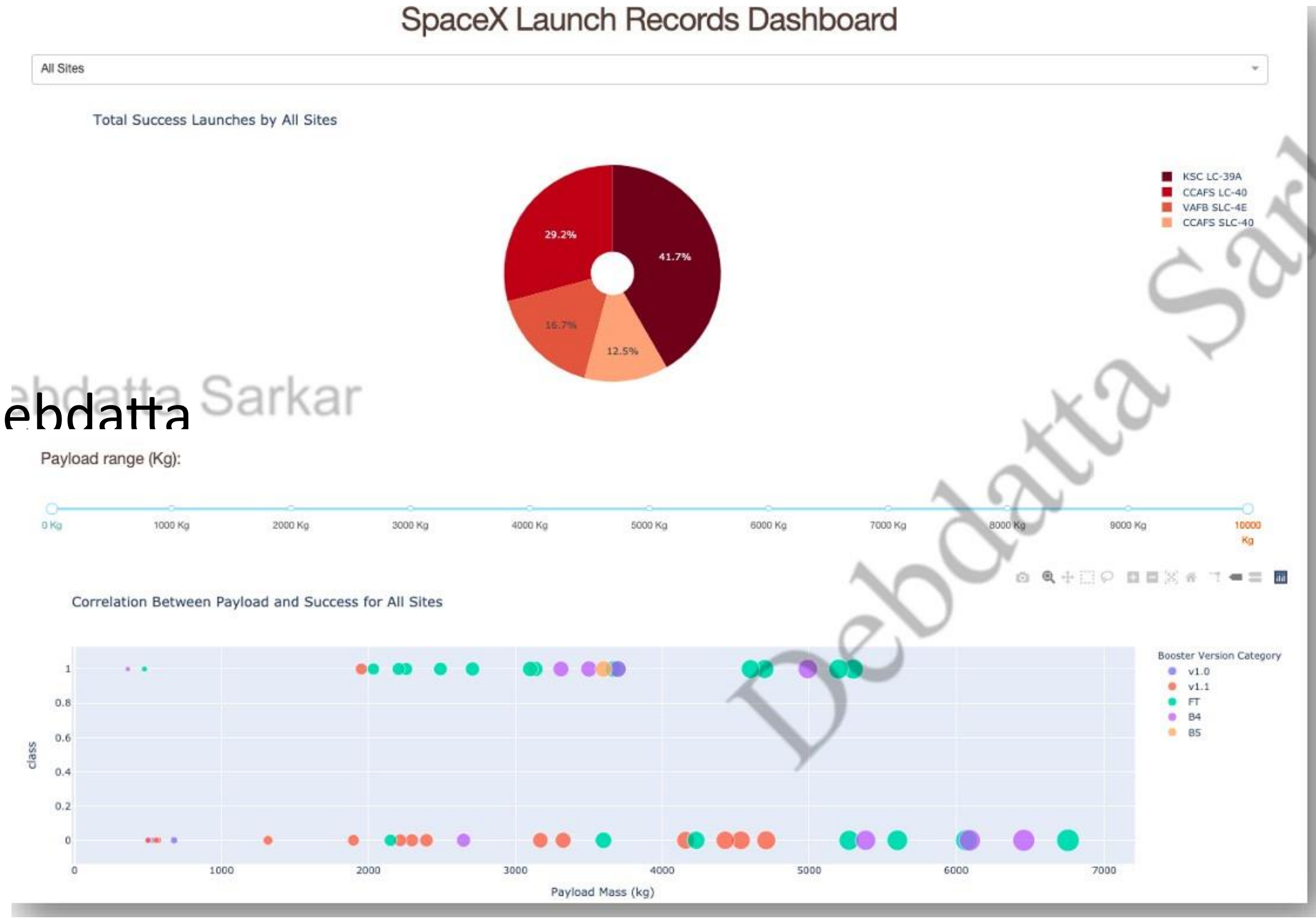
2019

Flight Number

Year

" Python Anywhere" Live Site for Plotly Dashboard

Code



Debdatta

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

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.

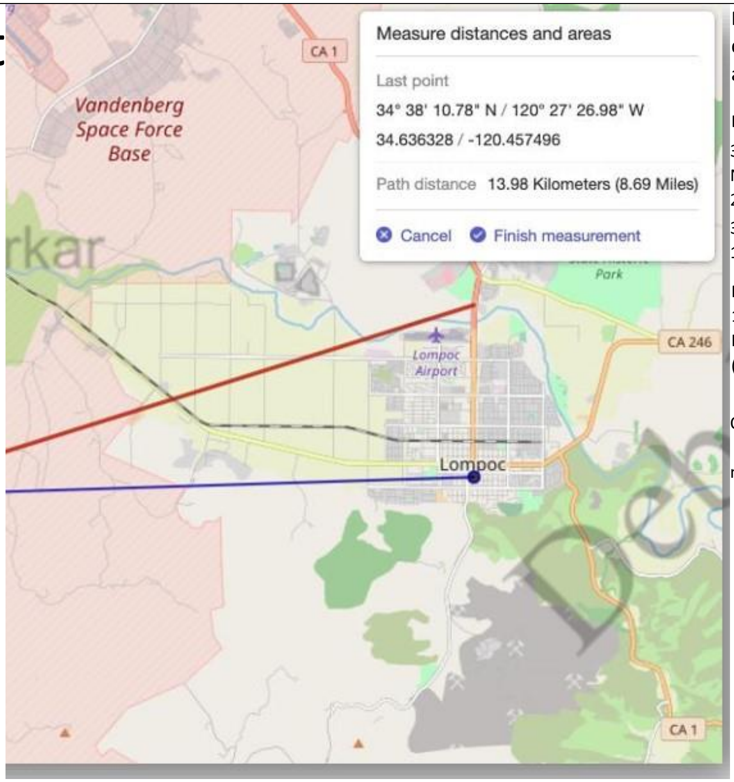
Code

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

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

bdat

Idenberg
Marine
?eserve



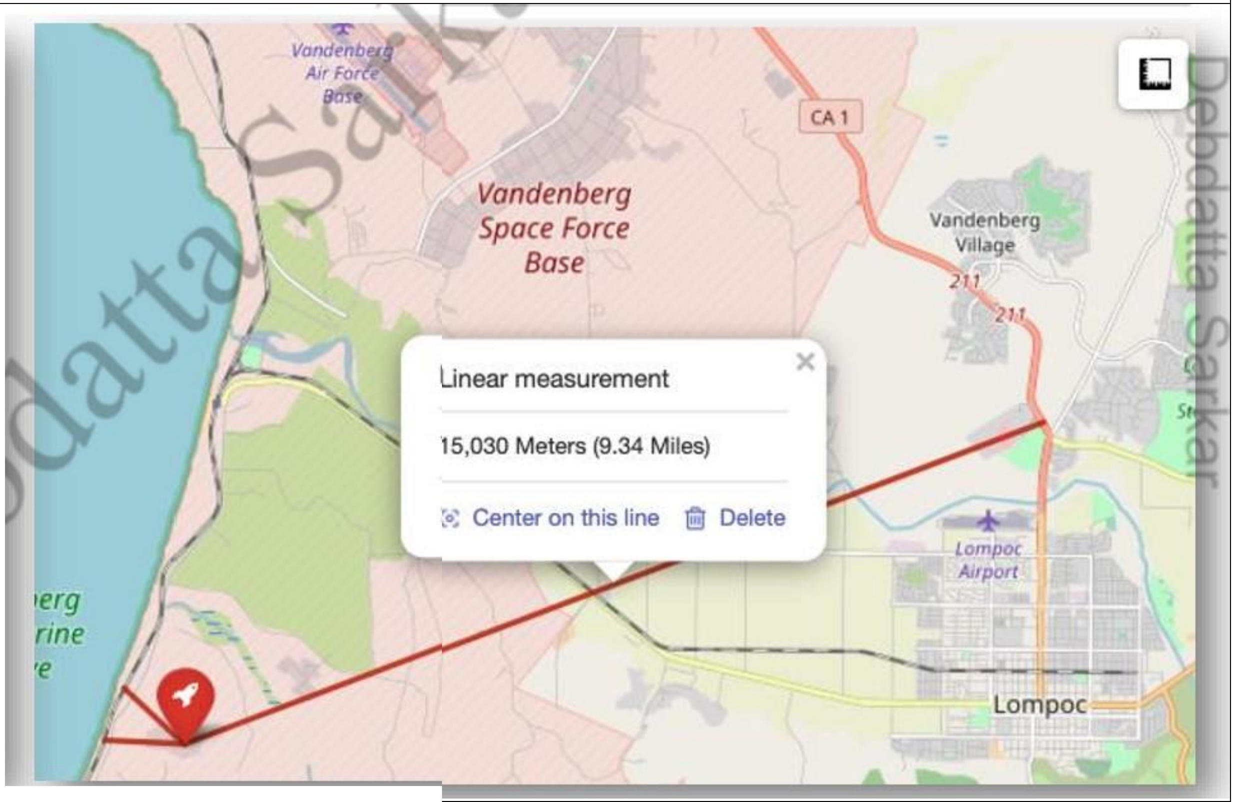
Measure distances and areas

Last point
34' 38' 10.78" N/ 1200 27'26.98• W
34.636328 / - 120.457496

Path distance 13.98 Kilometers (8.69 Miles)

O Cancel
Finish measurement

Lo



D

Folium Custom Title Layers with Labels

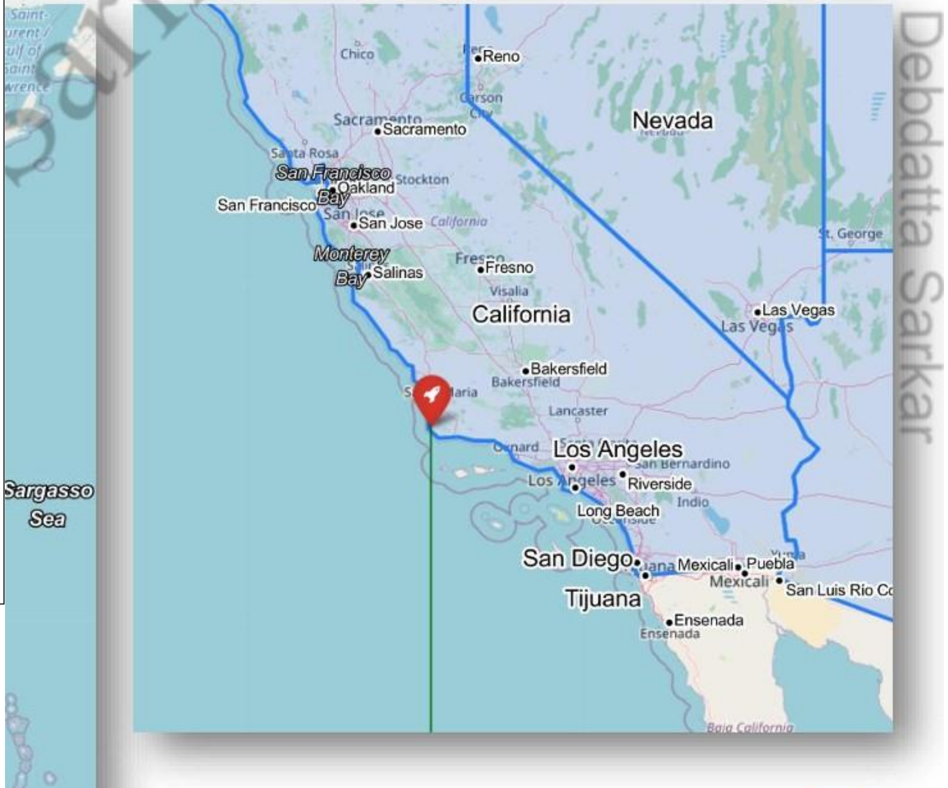
Code

```
folium.GeoJson(geo_json_data) .add to(site_map) folium.map. CustomPane( " labels " ) . add_to(
site_map) folium.TileLayer( "stamentonerlabels" , pane=" labels" ) .add_to(site_map)s ite_map
```

Code

Ciudåd
Guadalajara'
"de México
Cuba
Dominican
Haiti L R
Republic
Dominic 0'_4—--.
Mexico City
Belize
Jamaica
Ciudad
Honduras
de Guatemala
El SalvadorIt aragua

Created Custom Title Layer to understand the locations of launch site in a better way.

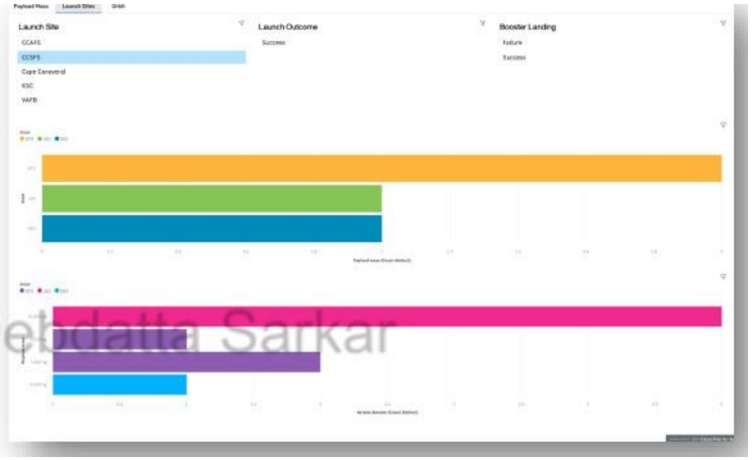


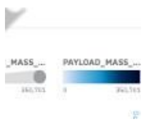
Code

Code

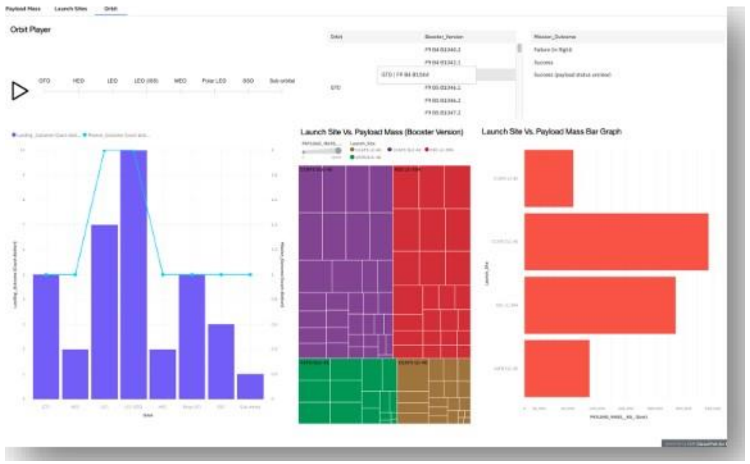
IBM Cognos Visualization Tool

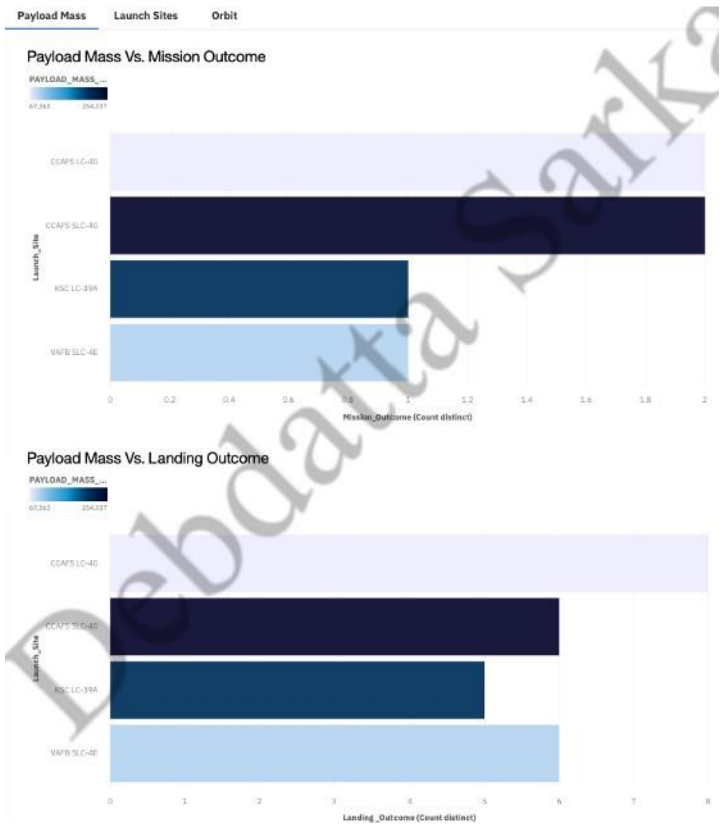
IBM Cognos Analytics provides analytic insights that help you to detect and validate important relationships and meaningful differences based on the data that is presented by the visualization.





payload Mass





Success] Success

620K

PAYLOAD_MASS KG_

Orbit

HEO

LEO

C

Booster Version

F9 8481039.2

F9 B4 B1040.2

F9 B4 B1041.2

F9 a. BIOC3_2

F9 84 B1039.1

Customers

AsiaSat

CONAE

CONAE, Planet1Q, space*

Canadian Space Agency (CSA)

Es hailSat

Hisdesat exactEarth SpaceX

Hispasat NwaWurks

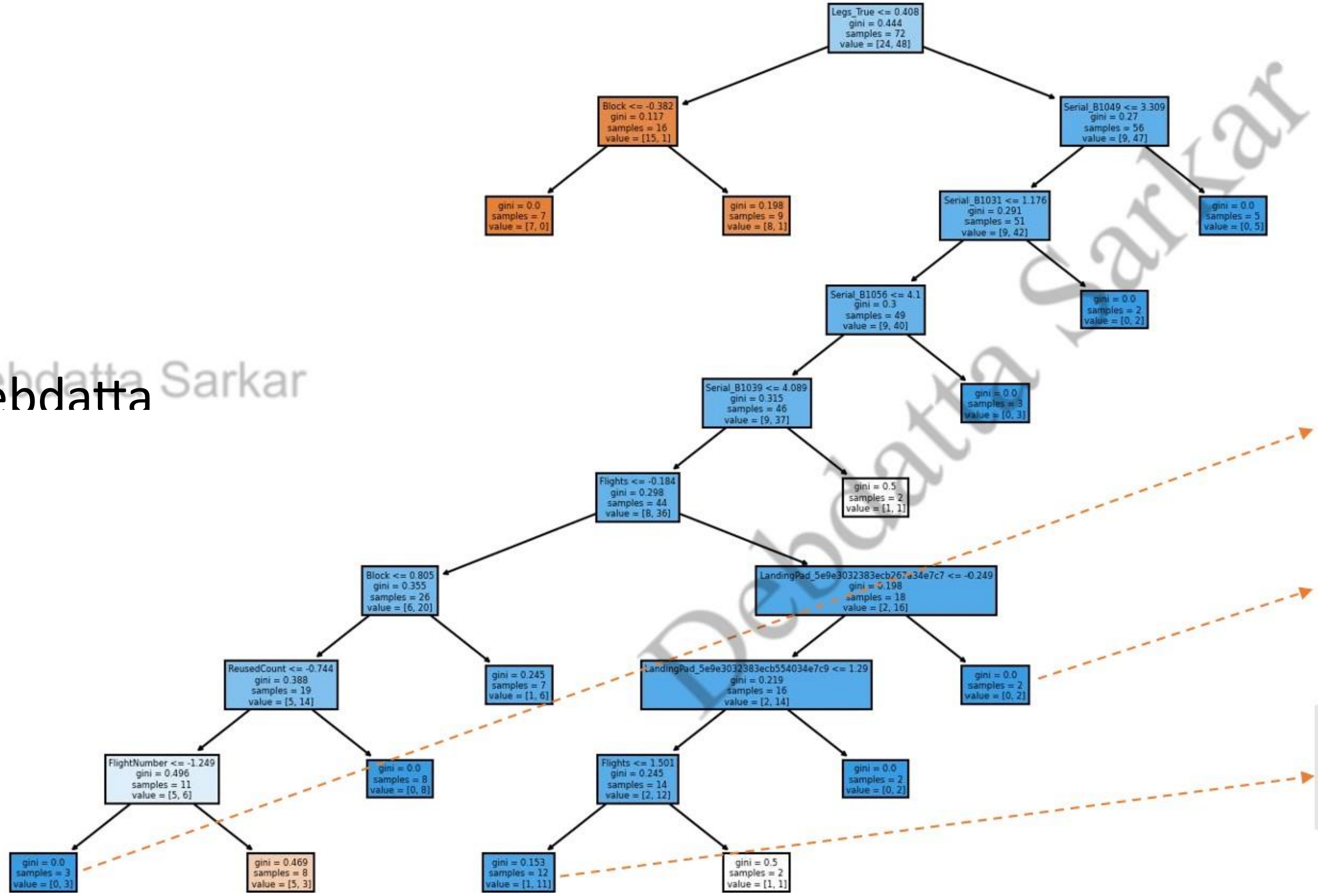
Inmarsat

Iridium Commnetims

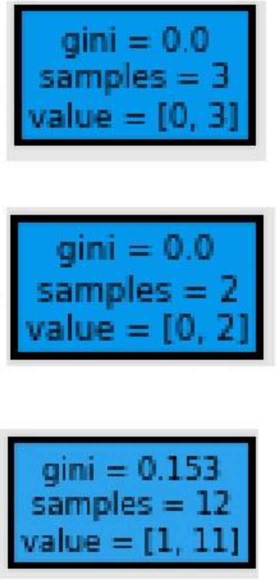
alta Sarkar

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

Code

Debdatta Sarkar

Thank You!