

# Winning Space Race with Data Science

Habib Ahmad

26.12.2024



# Outline

---

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

# Executive Summary

## Summary of methodologies

1. Data Collection - gathering data
2. Data Wrangling – cleaning & preparing data
3. Exploratory Data Analysis
4. Interactive Visual Analytics
5. Predictive Analysis

## Summary of results

1. Exploratory Data Analysis results of SQL and visualization
2. Interactive analytics results of Folium and Dash
3. Predictive analysis results

# Introduction

---

The commercial space age is here, companies are making space travel affordable for everyone. Talking about affordability, the agency that we can first think of is SpaceX which was founded in 2002 by Elon Musk with the goal of reducing space transportation costs. It's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access and many more.

One reason SpaceX can do this is the rocket launches are relatively inexpensive, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

As a Data Scientist at SpaceY Corp., I will be predicting if the SpaceX Falcon 9 first stage will land successfully, if the first stage can be reused and as a result to determine the price of each launch. I will achieve this by training a machine learning model and use public information to make the required prediction.

Section 1

# Methodology

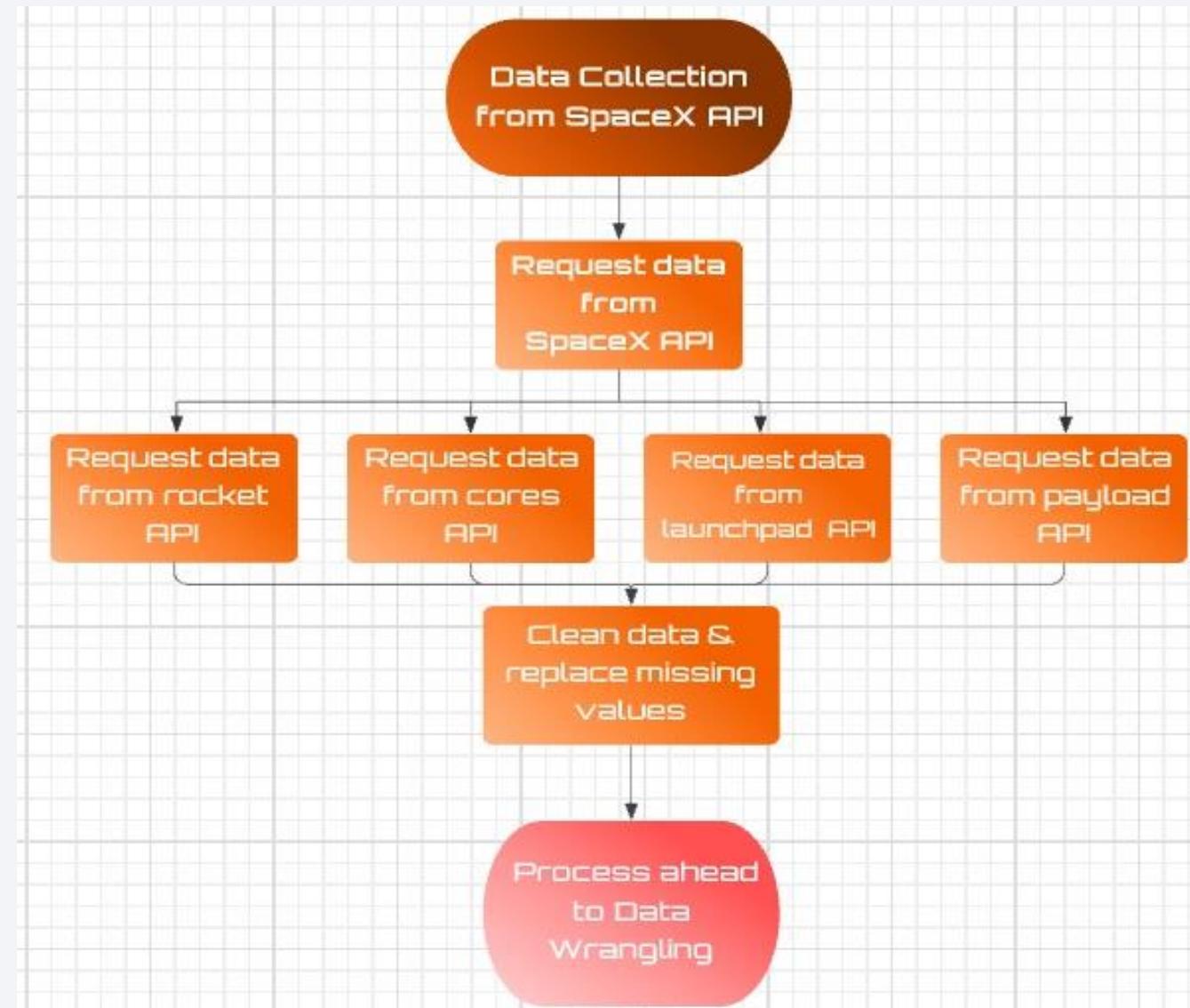
## Summary

- Data collection methodology:
  - Obtained the launch data from SpaceX REST API & also by Web Scrapping with Beautiful Soup.
- Perform data wrangling:
  - The booster landings outcomes are converted to training labels using Exploratory data analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Performed visualization with pandas & matplotlib and executed SQL queries for EDA.
- Perform interactive visual analytics using Folium and Plotly Dash:
  - Used Folium maps to find geographical patterns and built a dashboard with the Plotly Dash to get more insights.
- Perform predictive analysis using classification models:
  - Built machine learning models to make the prediction and find best model.

- ▶ The initial step of Collecting SpaceX Falcon 9 launch data is fulfilled using two methodologies as follows:
  - 1) From API:- using the get() method from requests library the launch data is collected from SpaceX REST API.
  - 2) From Web Page:- web scrapping Falcon 9 historical launch records from Wikipedia webpage titled ‘List of Falcon 9 and Falcon heavy launches’ with BeautifulSoup.

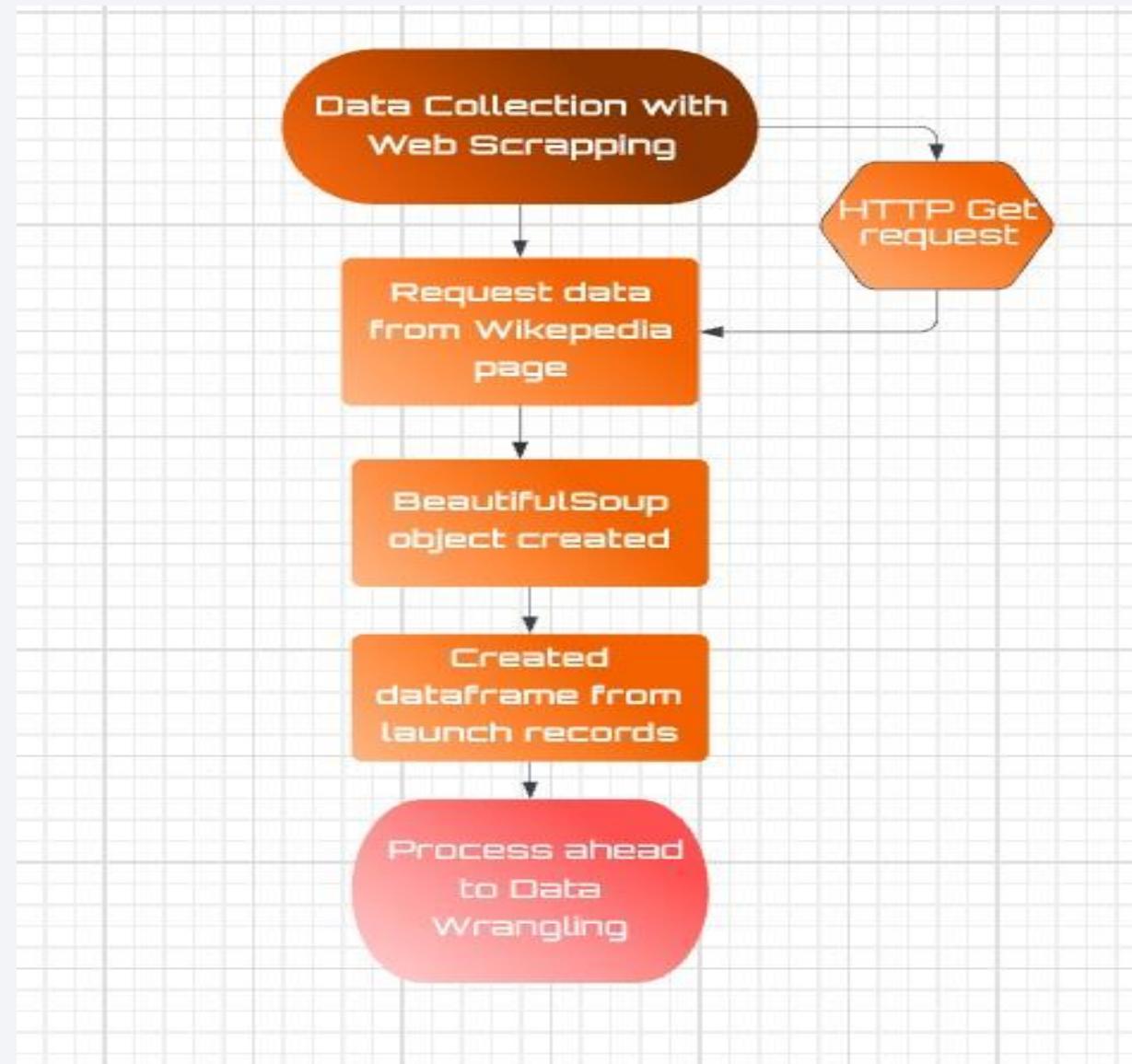
# Data Collection – SpaceX API

- ▶ Requested rocket launch data from SpaceX API using `get()` method from `requests` library.
- ▶ Parsed the launch data to a JSON and then converted it to a dataframe.
- ▶ Removed irrelevant features and removed all launches data except the Falcon 9 launches.
- ▶ Replaced missing values with the mean of respective column.



# Data Collection - Scraping

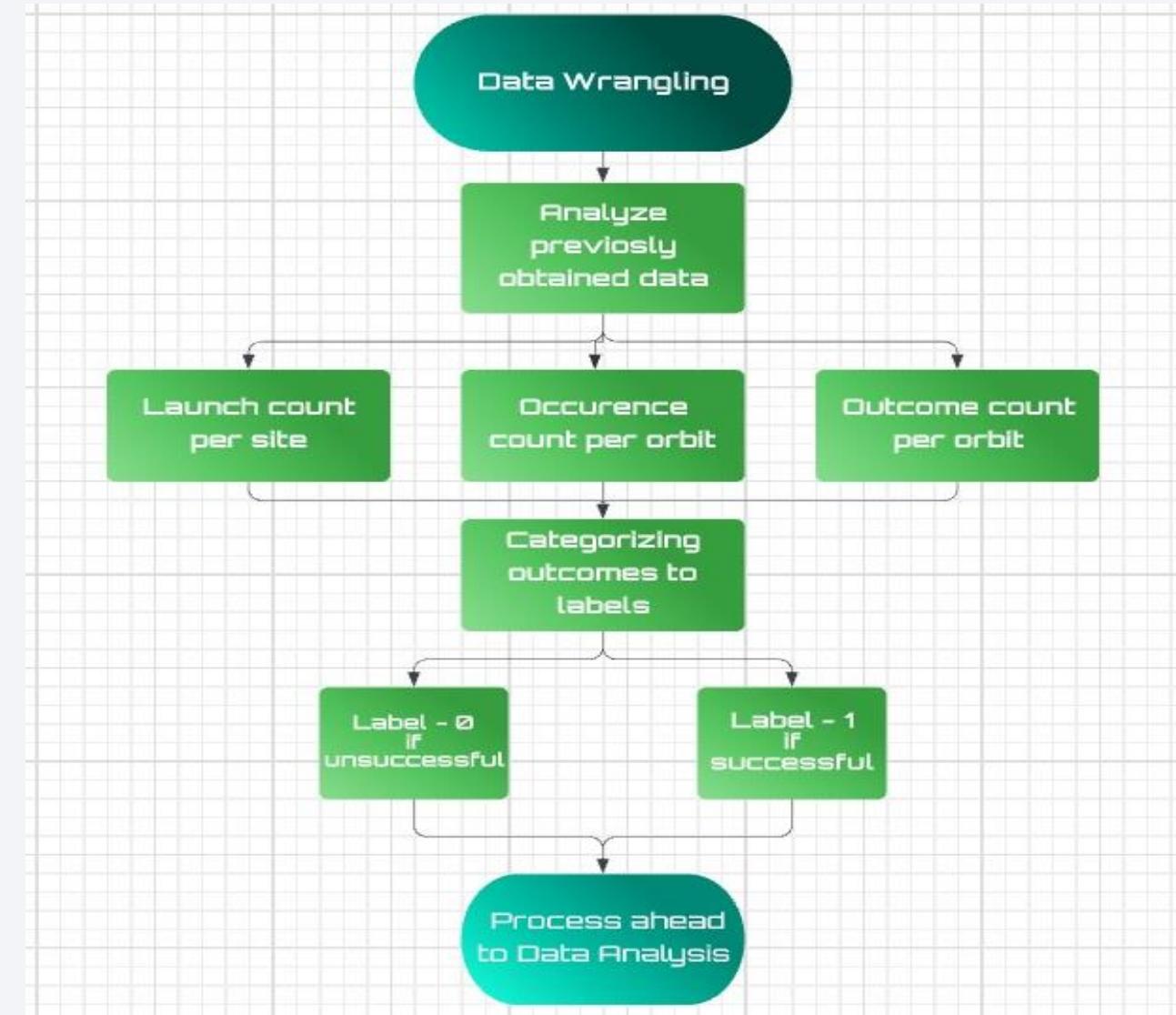
- ▶ Performed HTTP GET request to the Falcon9 Launch HTML page.
- ▶ Created a BeautifulSoup object using response text content.
- ▶ Created dataframe from parsed launch record values.



# Data Wrangling

10

- ▶ Analyzed the number of launches on each site, each orbit and outcome.
- ▶ Categorized outcomes as landing labels, where 0 represents unsuccessful landing and 1 for successful landing.
- ▶ Created a new dataframe with an added Class column containing the landing labels.



- ▶ Plots used for exploratory data analysis to visualize valuable insights.
- ▶ Made scatter plots of
  1. Flight Number vs Payload Mass
  2. Flight Number vs Launch Site
  3. Payload Mass vs Launch Site
  4. Flight Number vs Orbit Type
  5. Payload Mass vs Orbit Type, each with Class overlay.
- ▶ Created bar plot to visualize success rate of each orbit.
- ▶ Used line plot to visualize launch success yearly trend.

- ▶ Executed SQL queries for exploratory data analysis.
- ▶ Displayed following SQL queries:-
  1. Unique launch site names
  2. 5 launch sites starting with 'CCA'
  3. Total payload mass carried by NASA(CRS) boosters
  4. Average payload mass carried by F9 v1.1 version
  5. Date of first successful landing ground pad
  6. Boosters that had success in drone ship & payload mass between 4000 & 6000
  7. Total number of successful and failure mission outcomes
  8. Booster Versions that carried maximum payload mass
  9. Failed landing outcomes in drone ship, booster versions & launch sites for the year 2015
  10. Rank in descending the landing outcome count between 2010-06-04 and 2017-03-20

# Build an Interactive Map with Folium

13

- ▶ Folium map objects used to build an Interactive Map are markers, circles and polylines.
- ▶ Created a map with center location as NASA Johnson Space Center at Houston, Texas, to visualize and draw insights of SpaceX launch sites.
- ▶ Marked the success and failed launches for each site on the map using markers.

1

# Build a Dashboard with Plotly Dash

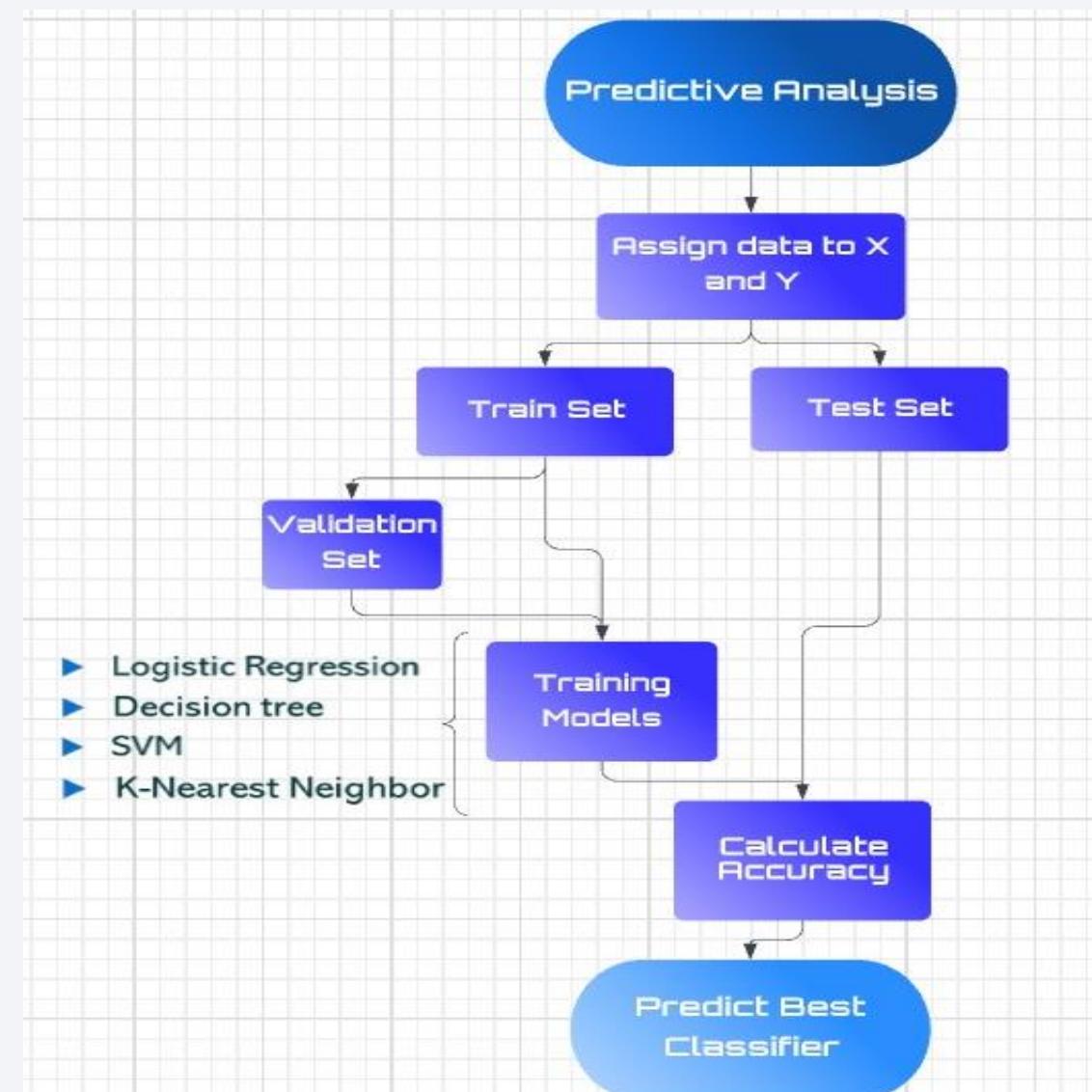
14

- ▶ Added a dropdown list of Launch Site names to enable selection.
- ▶ Added a pie chart to the Dashboard showing comparison between success Vs failed launches of selected site or count of successful launches of all sites.
- ▶ Added a slider to select payload mass range.
- ▶ Added a scatter plot to show the correlation between payload mass and launch success.

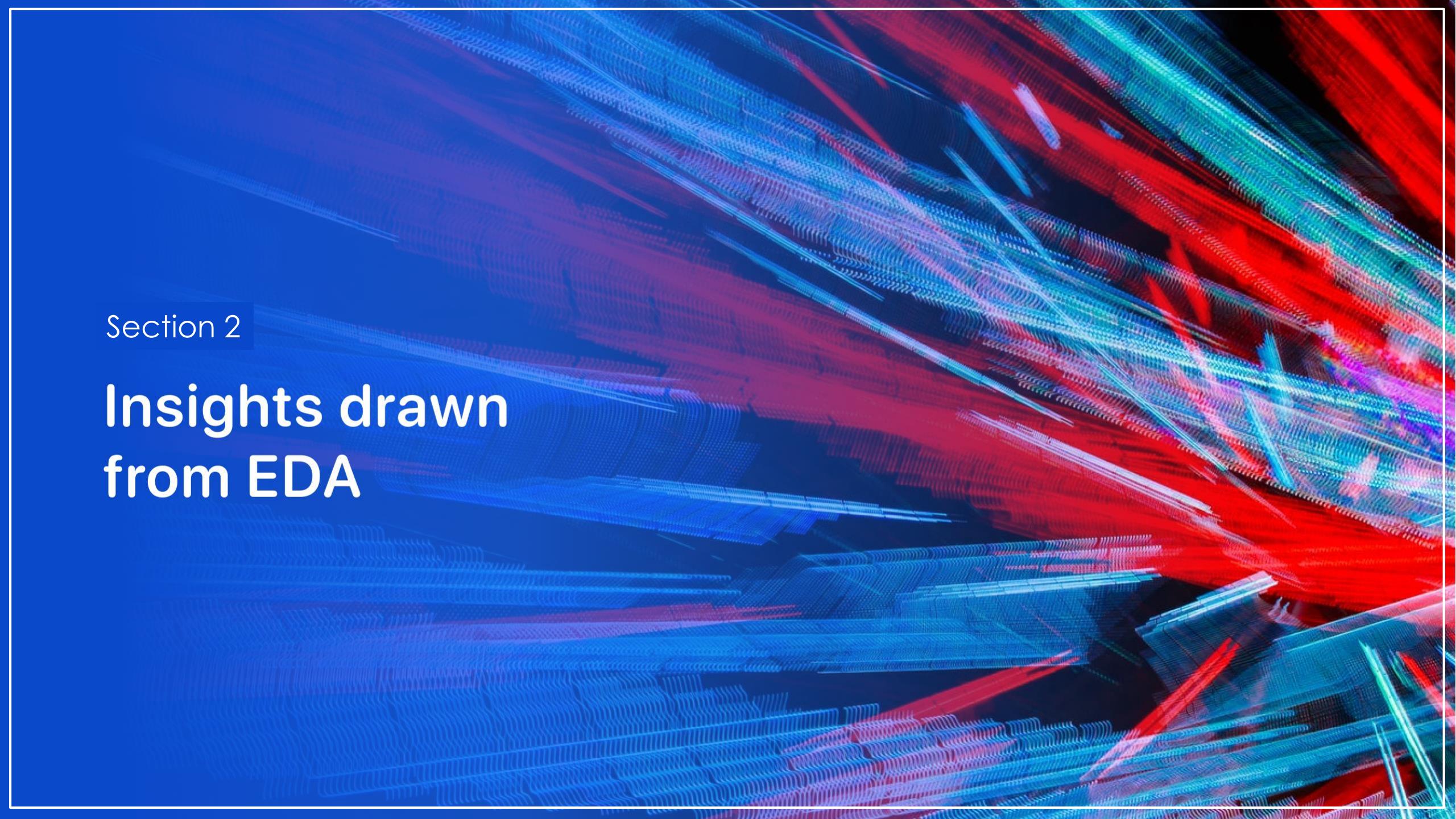
# Predictive Analysis (Classification)

15

- ▶ Obtained and standardized feature data in X and assigned Class data to Y.
- ▶ Data spilt into train & test data and training data further divided to validation data.
- ▶ Created models of Logistic Regression, Support Vector Machine, Decision tree classifier and K-Nearest Neighbor and calculated validation set accuracy and test set accuracy of each model.
- ▶ Plotted Confusion Matrix of each model.
- ▶ Observed that all four models have same accuracy score of 83.3% and hence cannot determine best classifier.



- Exploratory data analysis results:
  - ❖ As the flight number increases, the first stage is more likely to land successfully.
  - ❖ Rockets with payload mass less than 8000kg are highly preferred for launches.
  - ❖ Orbits ES-L1, GEO, HEO, SSO have the highest success rates.
  - ❖ Since 2013, the launch success rate kept increasing till 2020.
  - ❖ First successful landing outcome on ground pad was on 2015-12-22.
- Interactive analytics results:
  - ❖ Site KSC LC-39A has the highest success rate & CCAFS LC-40 has lowest.
  - ❖ Launch sites are in close proximity to railways, highways and coastline but located far away from the cities.
- Predictive analysis results:
  - ❖ All the four classification models have the same Accuracy Score which is 0.833.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

## Insights drawn from EDA

# Flight Number vs. Launch Site

18

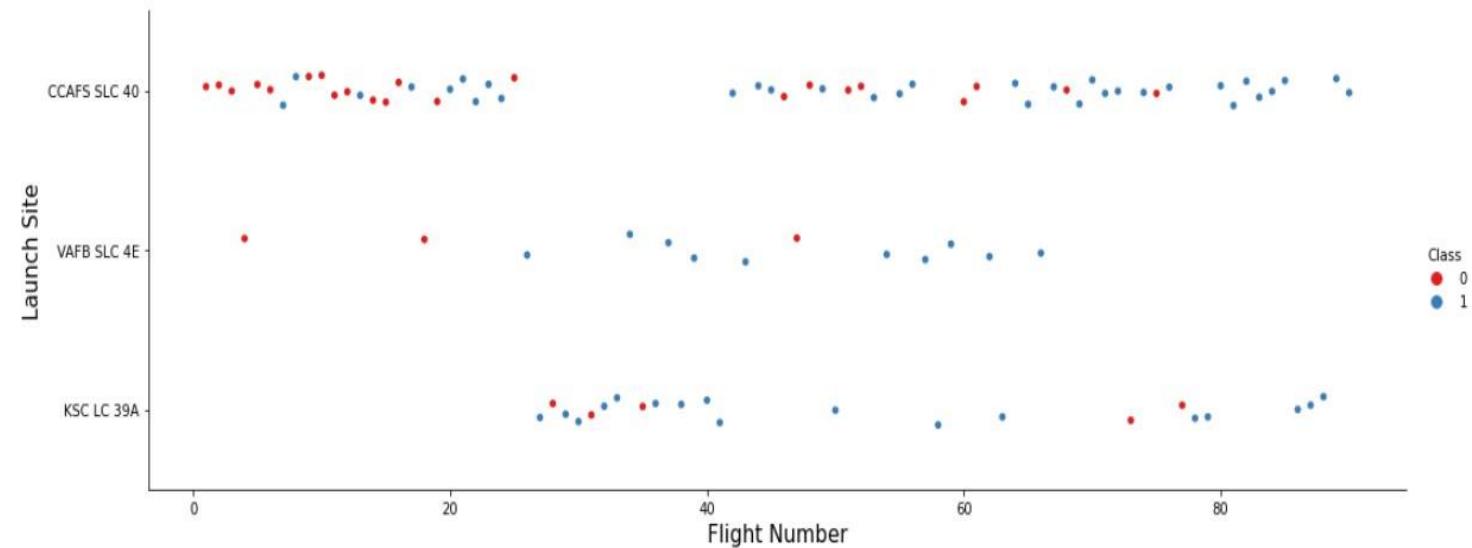
- ▶ Observations of scatter plot of Flight Number vs. Launch Site with Class as overlay:-

1. Launch sites KSC LC-39A and have higher success rates than VAFB SLC 4E and CCAFS LC-40.
2. As the flight number increases, the first stage is more likely to land successfully

## TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function catplot to plot FlightNumber vs LaunchSite, set the parameter x parameter to FlightNumber, set the y to Launch Site and set the parameter hue to 'class'

```
In [36]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the Launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=3, palette="Set1")
plt.xlabel("Flight Number", fontsize=15)
plt.ylabel("Launch Site", fontsize = 15)
plt.show()
```



# Payload vs. Launch Site

19

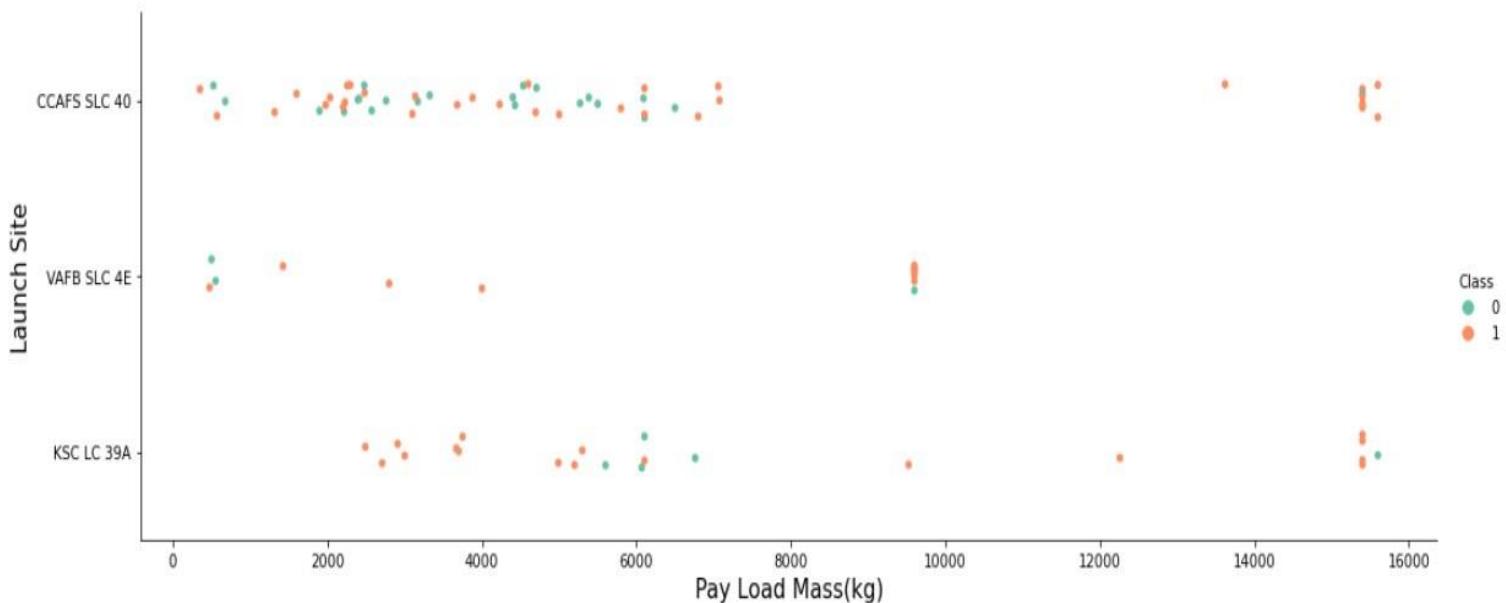
- ▶ Observations of scatter plot of Payload Mass vs. Launch Site with Class as overlay:-

1. Rockets with payload mass less than 8000kg are highly preferred for launches.
2. No rockets launched from VAFB SLC 4E launch site having payload mass greater than 10000kg

## TASK 2: Visualize the relationship between Payload and Launch Site

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

```
In [37]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value  
sns.catplot(y="LaunchSite" , x="PayloadMass" , data=df , hue="Class" , aspect=3 , palette="Set2")  
plt.xlabel("Pay Load Mass(kg)" , fontsize=15)  
plt.ylabel("Launch Site" , fontsize=15)  
plt.show()
```



# Success Rate vs. Orbit Type

20

## ► Observations of bar plot of Success Rate vs. Orbit Type:-

1. Orbit types ES-L1, GEO, HEO, SSO have the highest success rates.
2. Approximately no success rate for orbit type 'SO'

### TASK 3: Visualize the relationship between success rate of each orbit type

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

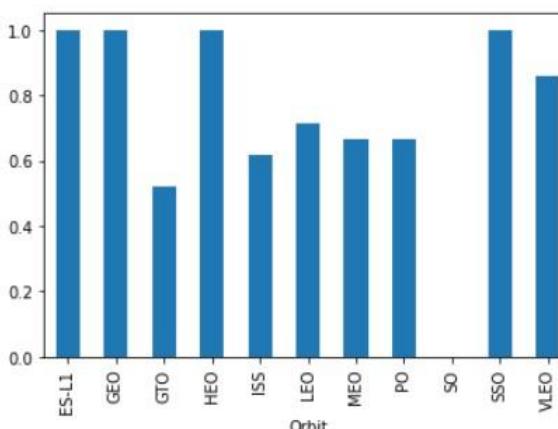
Let's create a bar chart for the success rate of each orbit

```
In [38]: # HINT use groupby method on Orbit column and get the mean of Class column  
df_grp = df.groupby(['Orbit']).mean()['Class']  
  
df_grp.head()
```

```
Out[38]: Orbit  
ES-L1    1.000000  
GEO     1.000000  
GTO     0.518519  
HEO     1.000000  
ISS     0.619048  
Name: Class, dtype: float64
```

```
In [39]: df_grp.plot(kind='bar')
```

```
Out[39]: <AxesSubplot:xlabel='Orbit'>
```



# Flight Number vs. Orbit Type

21

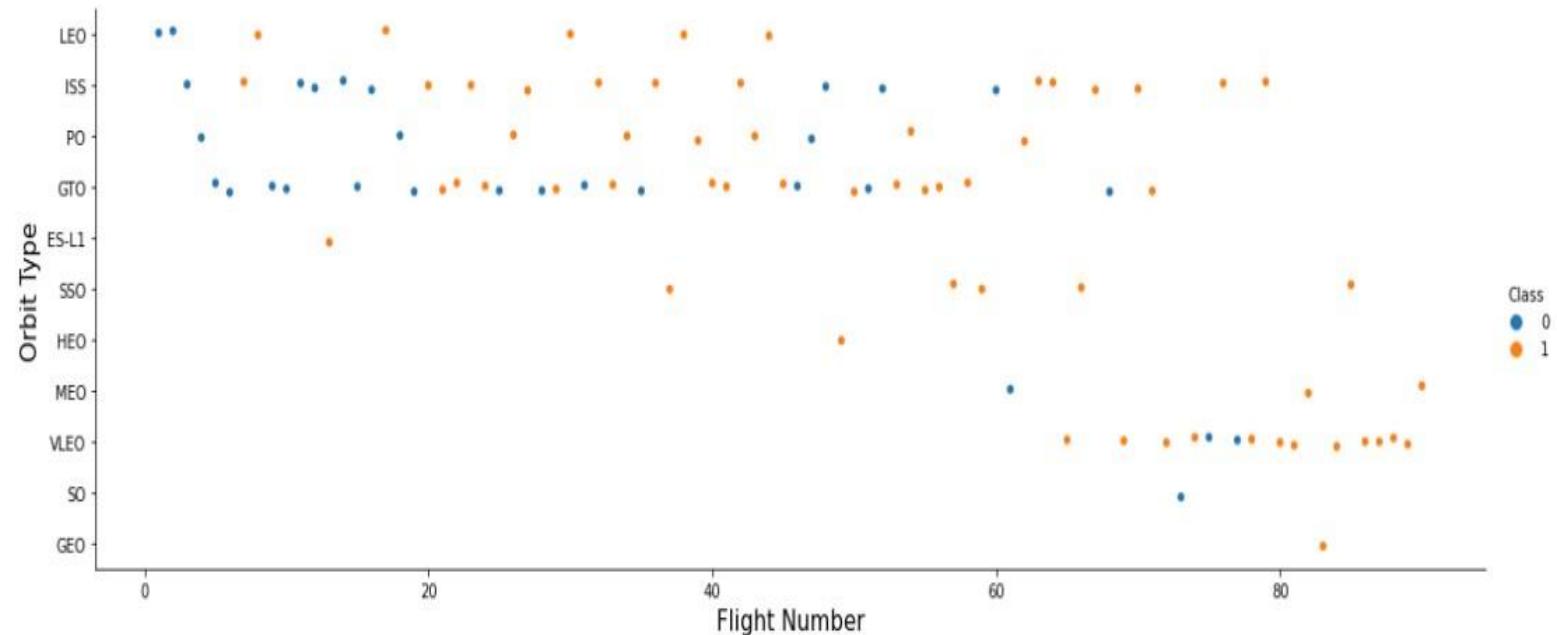
## ► Observations of scatter plot of Flight Number vs. Orbit Type:-

1. Success rate in LEO orbit increases with increase in Flight number.
2. Orbit VLEO has higher success rate when flight number is more than 60.
3. Orbits SSO, HEO, GEO and ES-L1 have 100% success rate, irrespective of number of flights.

## TASK 4: Visualize the relationship between FlightNumber and Orbit type

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

```
In [40]: # Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit" , x="FlightNumber" , data=df , hue="Class" , aspect=3)
plt.xlabel("Flight Number" , fontsize=15)
plt.ylabel("Orbit Type" , fontsize=15)
plt.show()
```



# Payload vs. Orbit Type

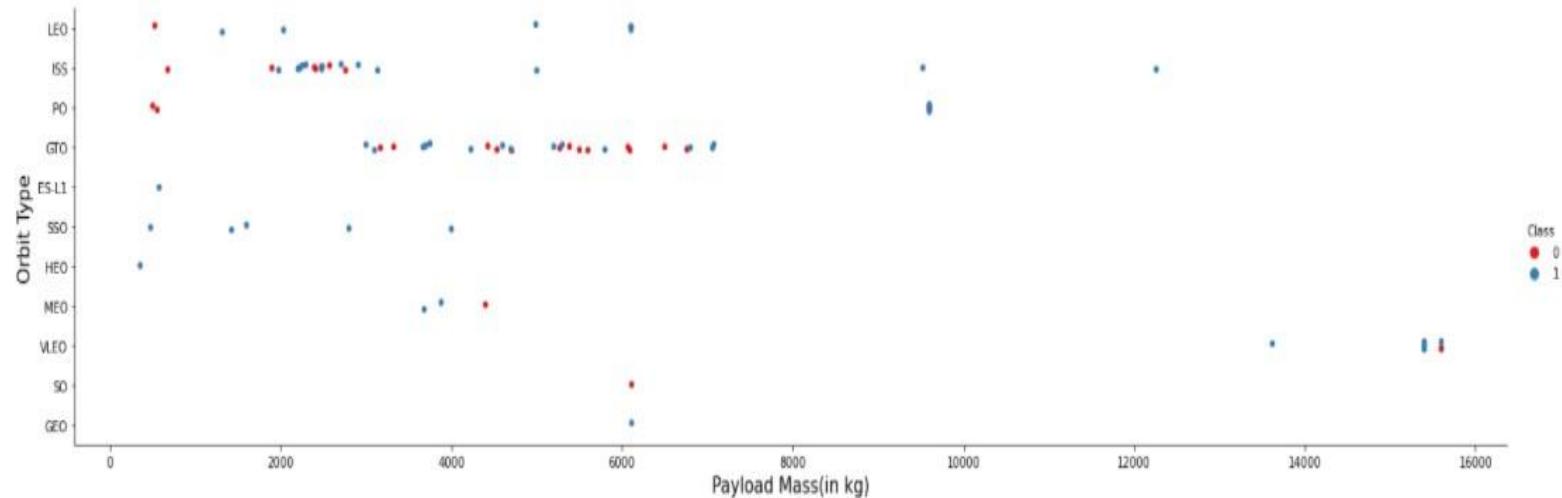
22

- ▶ Observations of scatter plot of Payload Mass vs. Orbit Type:-
  1. Success rates of orbits LEO, ISS, PO increases with heavy payloads.
  2. SSO, HEO, ES-L1 have successful landings with lighter payloads.
  3. For orbit GTO no relationship observed.

## TASK 5: Visualize the relationship between Payload and Orbit type

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

```
In [41]: # Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value  
sns.catplot(y="Orbit" , x="PayloadMass" , data=df , hue="Class" , aspect=4 , palette="Set1")  
plt.xlabel("Payload Mass(in kg)" , fontsize=15)  
plt.ylabel("Orbit Type" , fontsize=15)  
plt.show()
```



# Launch Success Yearly Trend

23

## ► Observations of line plot of Launch Success yearly trend:-

1. Since 2013, the launch success rate kept increasing till 2020, with a drastic downfall in 2018.

### TASK 6: Visualize the launch success yearly trend

You can plot a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend.

The function will help you get the year from the date:

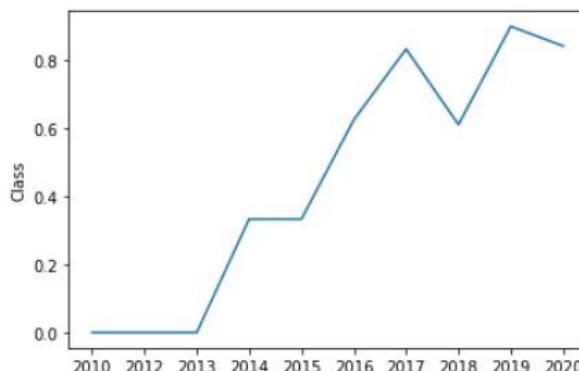
```
In [42]: # A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

```
In [43]: # Plot a Line chart with x axis to be the extracted year and y axis to be the success rate
Year = Extract_year(df['Date'])

df_yr = pd.DataFrame(Year , columns=['ExtractedYear'])
df_yr['Class'] = df['Class']

sns.lineplot(y=df_yr.groupby(['ExtractedYear'])['Class'].mean(), x=np.unique(df_yr['ExtractedYear']))
```

Out[43]: <AxesSubplot:ylabel='Class'>



- ▶ Used DISTINCT with SELECT statement to find unique launch site names.

## Task 1

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

In [34]: %sql select distinct(launch\_site) from SPACEXDATASET

```
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[34]:

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

# Launch Site Names Begin with 'CCA'

25

- Used LIMIT with SELECT statement to get first 5 rows WHERE launch site starting with 'CCA'

## Task 2

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

```
In [35]: %sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

out[35]:

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- ▶ Used SUM() function with SELECT statement to get total payload mass of NASA(CRS)

## Task 3

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

In [36]: %sql select sum(payload\_mass\_kg\_) as Total\_Payload\_Mass from SPACEXDATASET where customer= 'NASA (CRS)'

\* ibm\_db\_sa://hkq84699:\*\*\*@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.

Out[36]:

total_payload_mass
45596

- ▶ Used AVG() function with SELECT statement to get average payload mass carried by booster version F9 v1.1

## Task 4

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

```
In [37]: %sql select avg(payload_mass_kg_) as Average_Payload_Mass from SPACEXDATASET where booster_version= 'F9 v1.1'  
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od81cg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[37]:

average_payload_mass
2928

- ▶ Used MIN function with SELECT statement to get date of the first successful landing outcome on ground pad

## Task 5

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

*Hint: Use min function*

```
In [38]: %sql select min(DATE) as First_Successful_Landing from SPACEXDATASET where landing__outcome= 'Success (ground pad)'  
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[38]:

first_successful_landing
2015-12-22

- ▶ Used AND to satisfy 2 conditions in 1 query and obtain boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

## Task 6

*List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000*

```
In [39]: %sql select booster_version as Booster_Names from SPACEXDATASET where landing_outcome='Success (drone ship)' and payload_mass_kg_ between 4000 and 6000
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

Out[39]:

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

# Total Number of Successful and Failure Mission Outcomes

30

- ▶ Used COUNT function with SELECT statement to total number of mission outcomes, grouped by type of mission outcome.

## Task 7

*List the total number of successful and failure mission outcomes*

```
In [40]: %sql select mission_outcome , count(mission_outcome) as Total_Number from SPACEXDATASET group by mission_outcome  
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[40]:

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

# Boosters Carried Maximum Payload

31

- Used a subquery and MAX() function to list booster names which have carried the maximum payload mass.

## Task 8

*List the names of the booster\_versions which have carried the maximum payload mass. Use a subquery*

In [41]: `%sql select booster_version from SPACEXDATASET where payload_mass_kg_ = (select max(payload_mass_kg_) from SPACEXDATASET)`  
\* ibm\_db\_sa://hkq84699:\*\*\*@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.

Out[41]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- ▶ Used AND to satisfy 2 conditions in 1 query and failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 using YEAR() function.

## Task 9

*List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015*

```
In [42]: %sql select landing_outcome, booster_version, launch_site from SPACEXDATASET where landing_outcome='Failure (drone ship)' and YEAR(DATE)=2015
```

```
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[42]:

landing_outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Used AND to satisfy 2 conditions in 1 query and rank the count of landing outcomes grouped by type of outcome between the date 2010-06-04 and 2017-03-20 using GROUP BY clause, in descending order using ORDER BY clause.

## Task 10

*Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order*

```
In [43]: %sql select landing_outcome , count(landing_outcome) as Count from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20' group by landing_outcome order by count(landing_outcome) desc
```

```
* ibm_db_sa://hkq84699:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od81cg.databases.appdomain.cloud:30120/bludb  
Done.
```

Out[43]:

landing_outcome	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

The background image is a nighttime satellite view of Earth from space. It shows the curvature of the planet against the dark void of space. City lights are visible as glowing yellow and white dots, primarily concentrated in the lower right quadrant where the United States and Mexico are located. The atmosphere appears as a thin blue layer, and the horizon shows a faint green glow from the aurora borealis or aurora australis.

Section 3

# Launch Sites Proximities Analysis

# Visualize Launch Sites

35

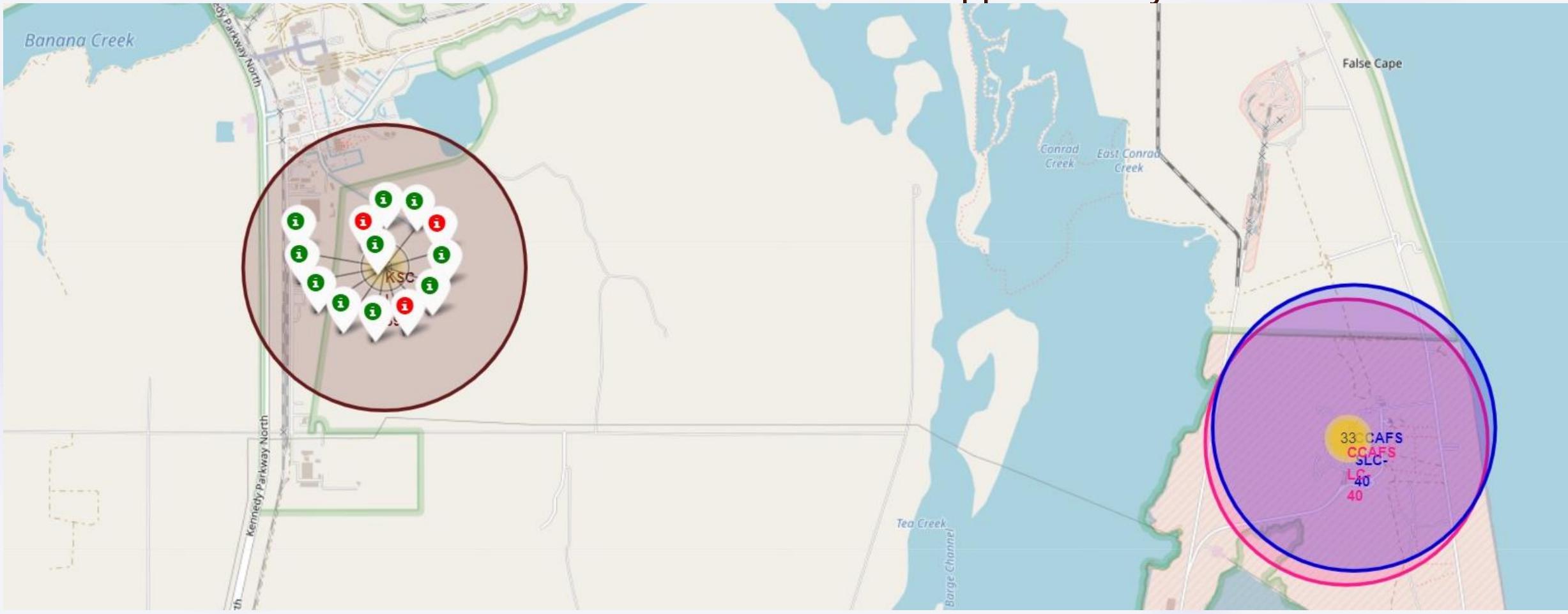
- ▶ Launch Site VAFB SLC-4E is located on farther west as compared to the other 3 sites, which are in proximity.
- ▶ The launch sites are in proximity to the coast but far away from the Equator line.



# Visualize launch outcome of each site

36

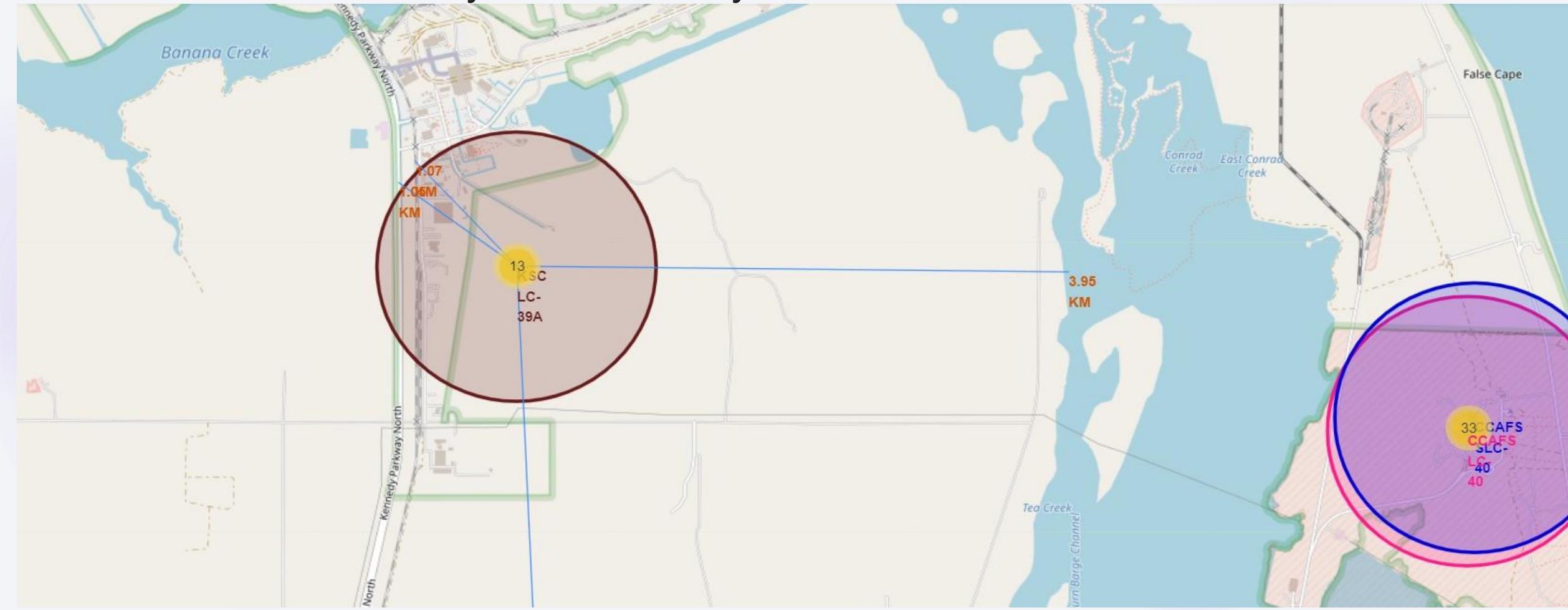
- ▶ Launch Site KSC LC-39A has the highest success rate of approximately 77%.
- ▶ Launch Site CCAFS LC-40 has the lowest success rate of approximately 27%

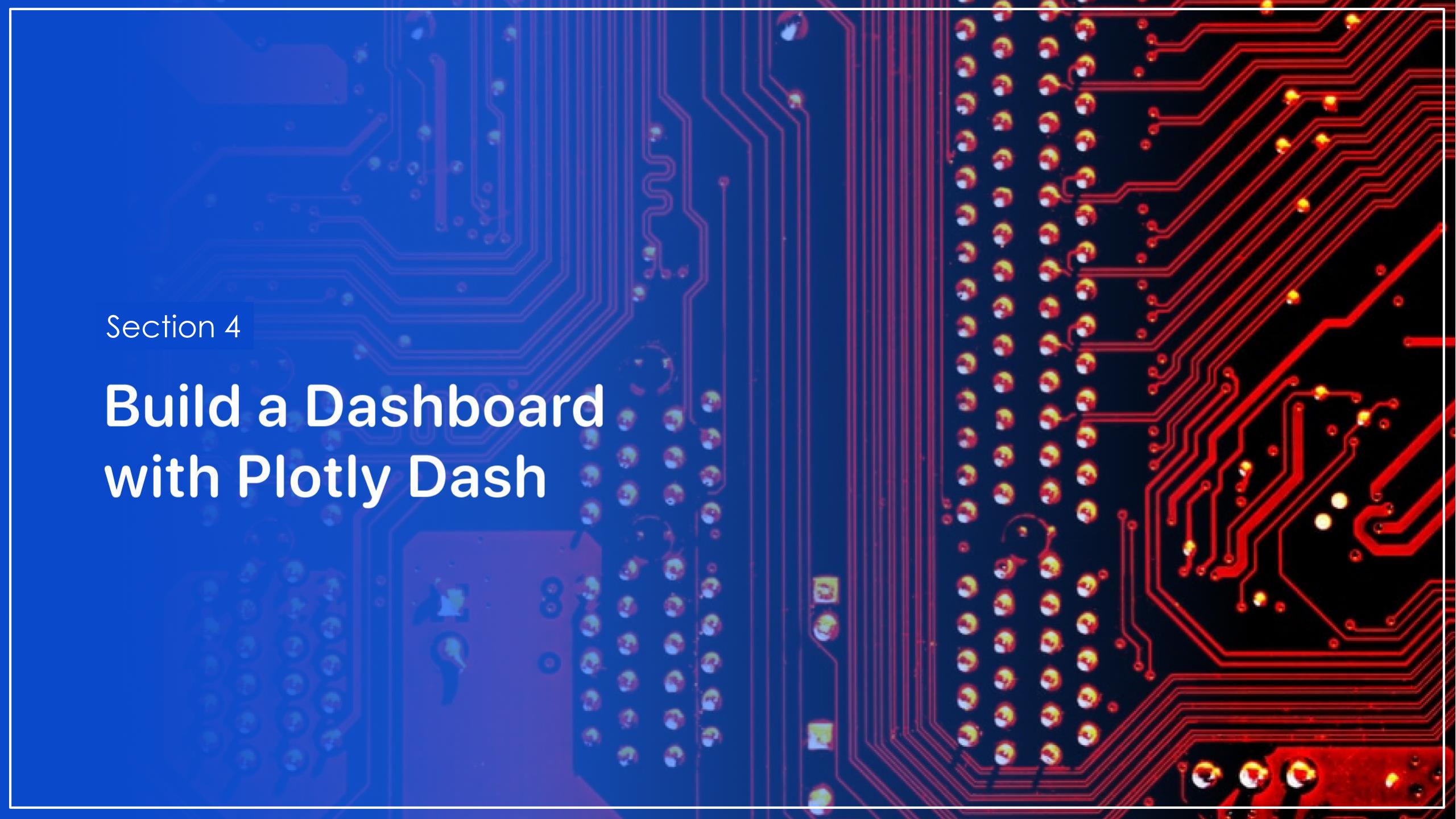


# Calculate distance of sites to its proximities

37

- ▶ The launch sites are in close proximity to railways, highways and coastline.
- ▶ The launch sites are usually located far away from the cities.





Section 4

# Build a Dashboard with Plotly Dash

# Compare success rates of all sites

39

- ▶ Selecting All Sites from dropdown, shows pie chart of total successful launch for all four sites.
- ▶ KSC LC-39A has the highest number of success launches.

## SpaceX Launch Records Dashboard

All Sites

Success Launches for all Sites



# Success launches of chosen site

40

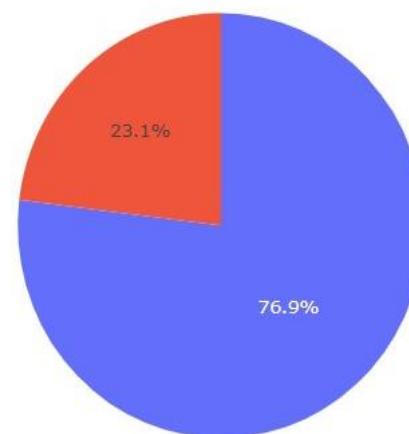
- ▶ Success rate of launch site KSC LC-39A is 76.9%, which denotes out of all the launches from this site, the first stage of most launches land successfully and hence can be reused.

## SpaceX Launch Records Dashboard

KSC LC-39A

x ▾

Total Success launches for the chosen site



# Correlation between payload and success rate

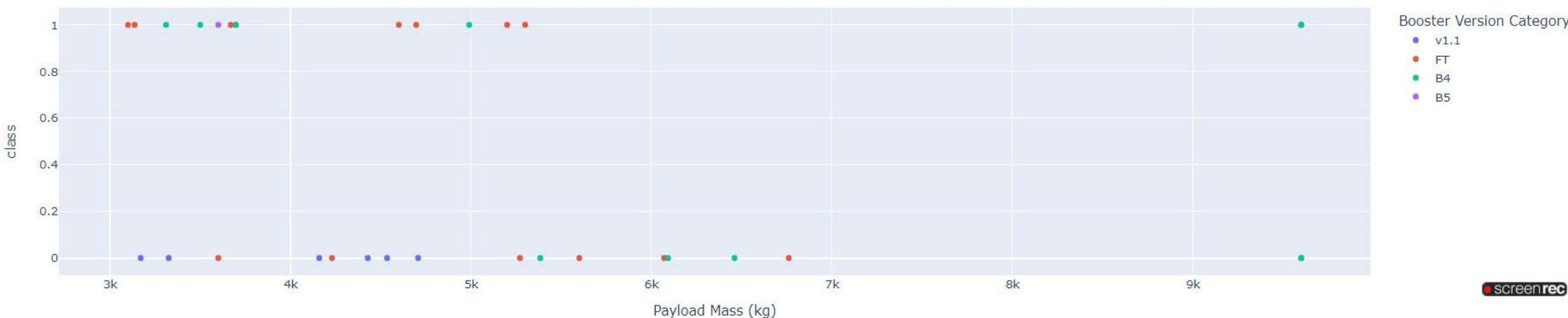
41

- ▶ Altered the range of payload mass by sliding to 3000kg-10000kg.
- ▶ From scatter plot we observe, booster version FT has a high success rate within selected range of payload mass whereas no success with booster version v1.1.

Payload range (Kg):



Correlation Between payload and success for all sites :



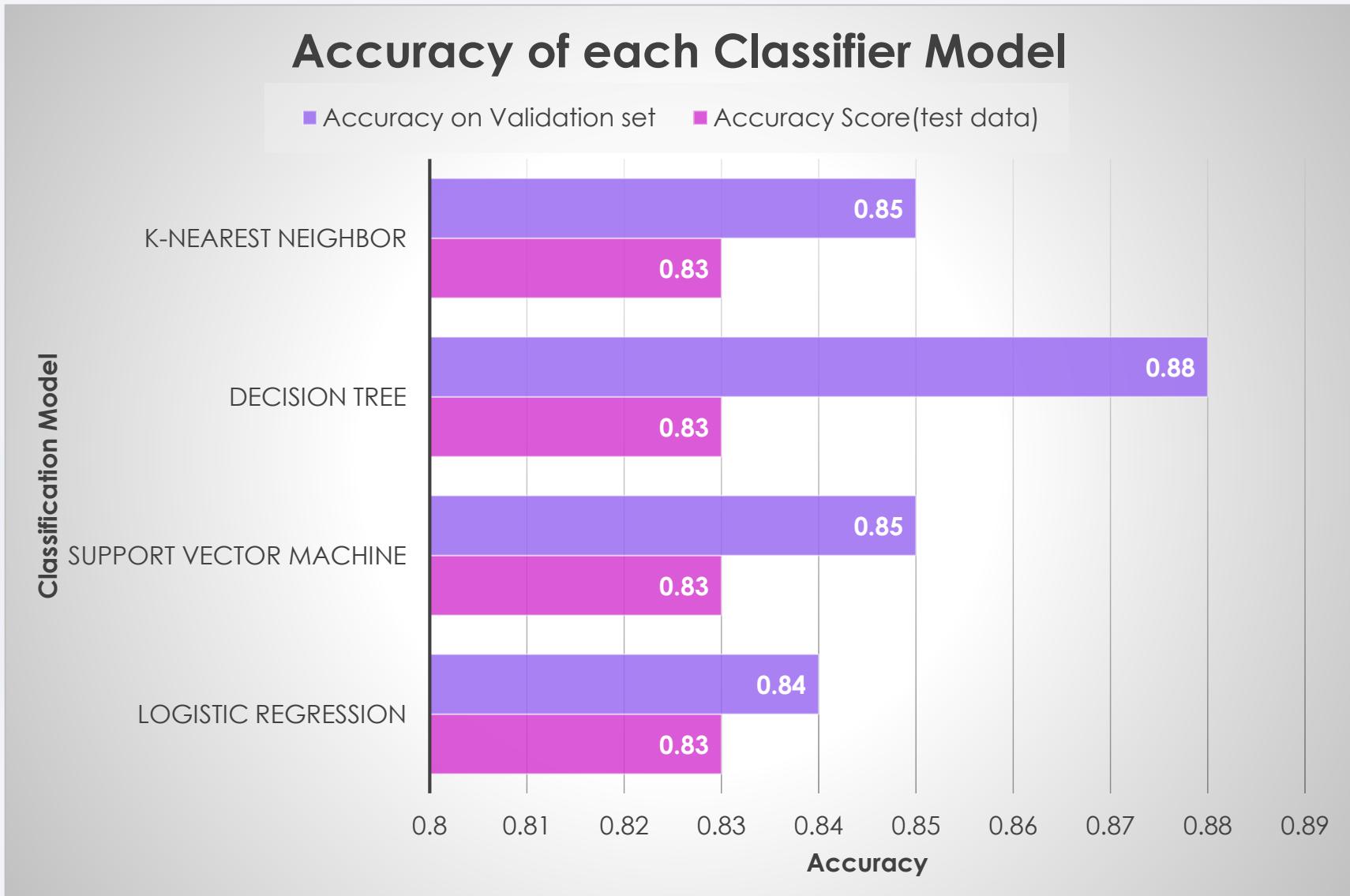
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

43

- ▶ All the four classification models have the same Accuracy Score which is 0.833.
- ▶ Hence cannot determine the best out of the four models that has highest classification accuracy.

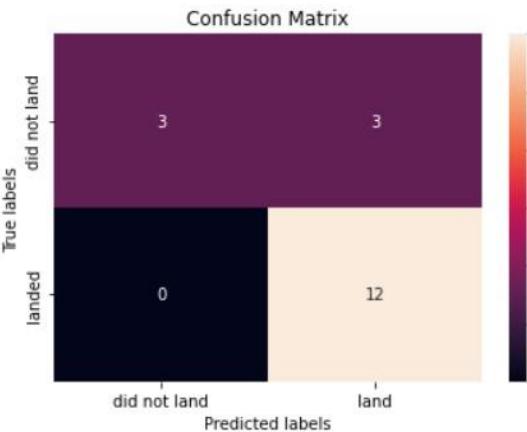


# Confusion Matrix

44

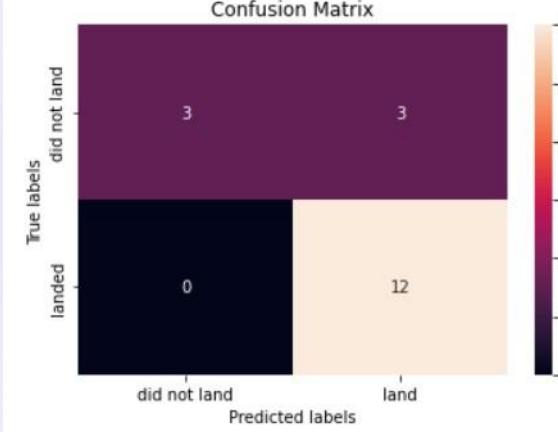
- ▶ As observed in previous results, all the four models have the same Accuracy Score and its difficult to determine the best classification model.
- ▶ The Confusion Matrices of each model depicts similar performance of each model.

```
: yhat_lr=logreg_cv.predict(X_test)  
plot_confusion_matrix(Y_test ,yhat_lr)
```



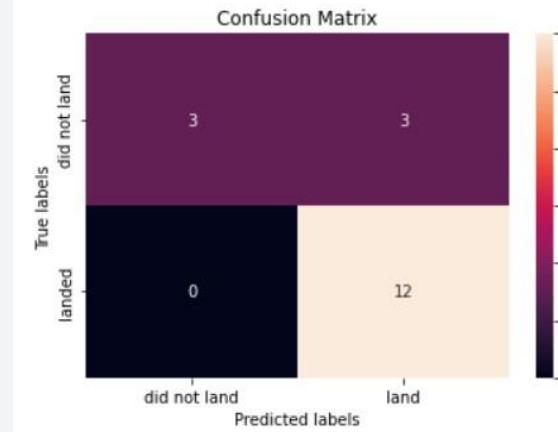
Logistic Regression

```
yhat_svm =svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test, yhat_svm)
```



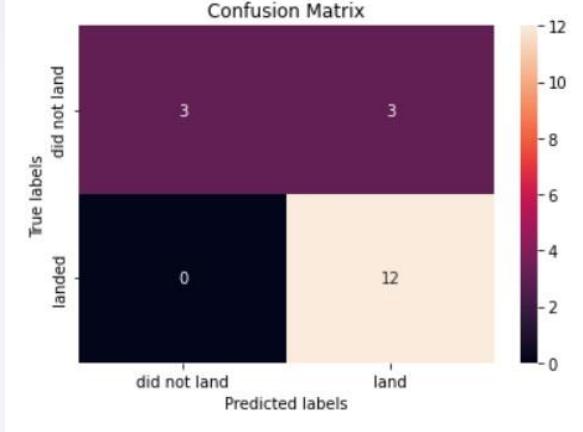
SVM

```
yhat_tree = svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test, yhat_tree)
```



Decision Tree

```
yhat_knn = knn_cv.predict(X_test)  
plot_confusion_matrix(Y_test, yhat_knn)
```



K-Nearest Neighbor

- ▶ Launch Site KSC LC-39A has the highest success rate of landing and is more preferable for launches where first stage needs to be reused.
- ▶ To have more successful landings, orbits ES-L1, GEO, HEO, SSO can be chosen.
- ▶ As the number of flight increases, the first stage is more likely to land successfully, thereby more chances of it to be reused.
- ▶ The best accuracy of classification model is 0.833.
- ▶ Starting from 2013, there has been an increase in successful landings and so with the classification model we may predict an increase in successful first stage landings which will lead to cost reduction of each launch.

- ▶ The Python code to obtain the Test Accuracy Score using the method `accuracy_score()` and comparing with the models created is as below:

Find the method performs best:

```
: from sklearn import metrics
print("Accuracy score of LR model is:" , metrics.accuracy_score(Y_test , yhat_lr))
print("Accuracy score of SVM model is:" , metrics.accuracy_score(Y_test , yhat_svm))
print("Accuracy score of Decision Tree model is:" , metrics.accuracy_score(Y_test , yhat_tree))
print("Accuracy score of KNN model is:" , metrics.accuracy_score(Y_test , yhat_knn))
```

Accuracy score of LR model is: 0.833333333333334

Accuracy score of SVM model is: 0.833333333333334

Accuracy score of Decision Tree model is: 0.833333333333334

Accuracy score of KNN model is: 0.833333333333334



# Thank you!

