Winning Space Race with Data Science

Anubhav Bhatt
27 May 2024

Outline

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

Executive Summary

- The goal of this project is to analyze the effect of various parameters on the landing success rate of the first stage of the SpaceX Falcon 9 rocket
- The main steps of the project involve -
  - Data collection - Collecting SpaceX launch data from SpaceX API by making a GET request and Web Scraping
  - Data Wrangling - Performing basic data wrangling and formatting to clean up the data
  - Exploratory Data Analysis using:-
    - SQL(sqlite3) to explore the dataset
    - Seaborn to visualize the dataset
    - Feature Engineering for one hot encoding on columns-Orbit, LaunchSite, LandingPad, and Serial
  - Creating Interactive dashboards using:-
    - Folium for Launch Site Location Analysis and proximity of Launch Sites to highways, cities, rail-lines, and coastlines
    - Plotly dash to visualize the SpaceX launch records- Launch site success, Correlation between payload range(kg) and success rate
  - Predictive analysis for first-stage landing using Machine learning:- best method out of Logistic Regression, SVM, Decision Tree, and KNN
- Summary -
  - Features like Payload Mass, Launch Site, Launch Orbit, etc. affect the outcome of the launch
  - Decision Tree is found to be the best algorithm to predict if the first stage of Falcon 9 will land successfully

Introduction

- The goal of the project is to predict what features successfully affect the success rate of the landing of first-stage booster rocket using the publicly available data for SpaceX Falcon 9 launches and the data science principles acquired through the IBM Data Science course
- Each Falcon 9 launch is advertised at $62 million(USD). Comparing this to the $165 million charged by the competitors, SpaceX provides a significant cost advantage. The cost savings are due to the re-usability of the first stage of the rocket
- If one can determine if the first stage of the rocket will land, the cost of the launch can be estimated. This can help in preparing a competitive bid
- Based on the data gathered from the API, factors like Payload Mass, Orbit Type, Launch Site, etc. can affect the launch outcome. Given this set of features, the project hopes to uncover vital correlations between the features and the launch outcomes
- Tools like exploratory data analysis and predictive analysis help establish an efficient prediction model based on publicly shared historical data

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Requesting data from SpaceX API using the URL: *https://api.spacexdata.com/v4/launches/past*
  - Using GET request
- Perform data wrangling
  - Cleaning data:- replacing missing/NaN values(with mean for PayloadMass)
  - Finding the training labels (using *.value_counts()* on LaunchSite, Orbits and Outcome columns)
  - Creating classification variable 'Class' that represents the outcome of each launch
- Perform exploratory data analysis (EDA) using visualization (seaborn, pandas) and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using Logistic Regression, SVM, Decision Tree, and KNN to predict the first stage landing success. Found the best method as Decision Trees, by comparing the accuracy

Data Collection

- Data was collected using SpaceX REST API at URL:
  https://api.spacexdata.com/v4/launches/past
  - Using GET request to get JSON data
- Data collected using BeautifulSoup(Web scraping)
  - Extract tables from HTML data and convert the tables to Pandas Dataframes

Data Collection – SpaceX API

- Request JSON data from SpaceX API using URL: *https://api.spacexdata.com/v4/launches/past*
- Use GET response and normalize the data
- GitHub URL of the completed SpaceX API calls notebook: Link to Jupyter Notebook!

Use URL: *https://api.spacexdata.com/v4/launches/past*

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

Use GET request

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

Decode the response using *.json()*
and Normalize the JSON data using *.json_normalize()*

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

[9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

[10]:
```
response.status_code
```

[10]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

[11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url)
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

[12]:
```
# Get the head of the dataframe
data.head()
```

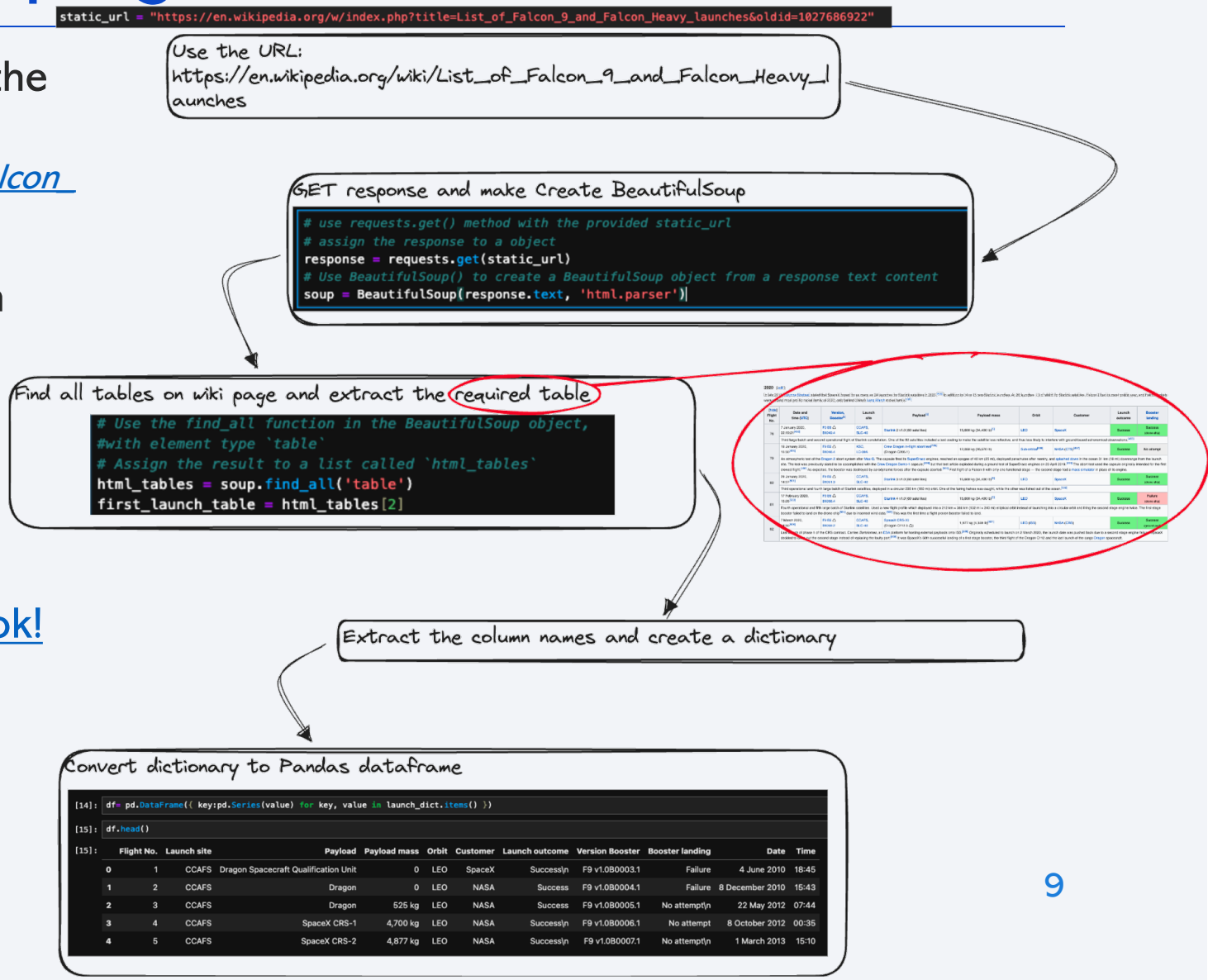
[12]:

	tic_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules	payloads	launchpad	auto_
	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e4502f5090995de566f86	
	None	NaN	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover	[]	[]	[]	[5eb0e4b6b6c3bb0006eeb1e2]	5e9e4502f5090995de566f86	

Screenshot of cells from Jupyter Notebook

Data Collection - Scraping

- Extract the SpaceX launch data table from the Wikipedia page using beautiful soup: *https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922*
- Use GET request, create BeautifulSoup from HTML response
- Extract the third table using *.find_all()* and create dataframe
- Add the GitHub URL of the completed web scraping notebook: Link to Jupyter notebook!



Data Wrangling

- The missing values were found in some of the rows of the dataset using *.isnull().sum() as:*

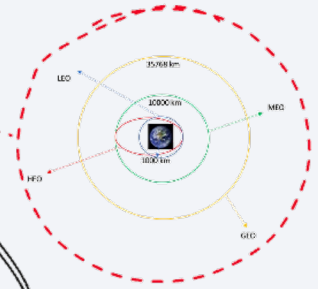
Column	Missing/none values
Payload Mass	5
LandingPad	26

- Replace missing values as follows:
  - Calculate the mean 'Payload Mass' and replace missing values with the mean
  - The none value in the 'LandingPad' indicates that the launch site wasn't used
- Creating a classification variable called '**Class**' that represents the outcome of each launch
- Add the GitHub URL of the completed data wrangling notebook: [Link to the Jupyter Notebook!]

Calculate the number and occurrence of mission outcome of different orbits

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes.keys()
```

```
Index(['True ASDS', 'None None', 'True RTLS', 'False ASDS', 'True Ocean',
       'False Ocean', 'None ASDS', 'False RTLS'],
      dtype='object', name='Outcome')
```



Create a landing outcome label from the outcomes where the element is 0 if a row in 'Outcome' is in set bad_outcomes

```
landing_class=[]
for temp in df['Outcome']:
    if temp in bad_outcomes:
        landing_class.append(0) # landing_class = 0 if bad_outcome
    else:
        landing_class.append(1) # landing_class = 1 otherwise
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

Assigning landing_class as the classification variable
Here 0 -> Unsuccessful first stage landing
1 -> Successful first stage landing

The class variable helps with further predictive analysis

```
df['Class']=landing_class
df[['Class']].head(5)
```

	Class
0	0
1	0
2	0
3	0
4	0

EDA with Data Visualization

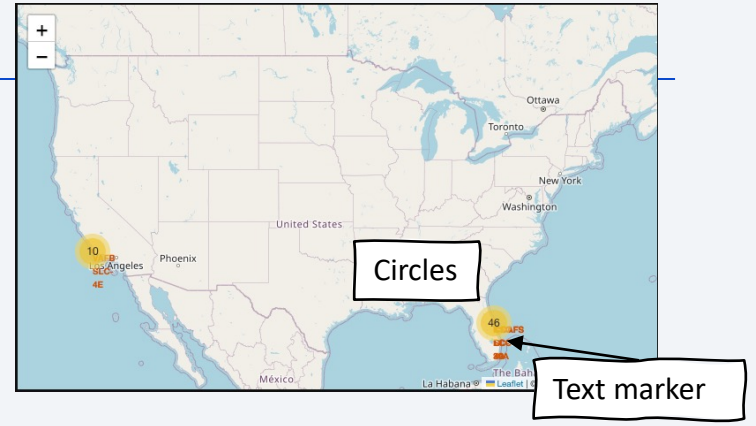
- The goal of the EDA of launch outcomes of Flacon 9 is to visualize launch success through different seaborn plots-
  - catplot: effect of FlightNumber vs PayloadMass and overlay launch outcome(categorical)
  - scatterplot: to visualize the relationship between variables
    - FlightNumber vs LaunchSite
    - PayloadMass vs LaunchSite
    - Orbit type vs FlightNumber
    - PayloadMass vs Orbit type
  - barplot: visualize the relationship between the success rate and Orbit type
  - lineplot: visualize yearly trend of launch success
- GitHub URL of the completed EDA Visualization notebook: Link to the completed Jupyter Notebook!

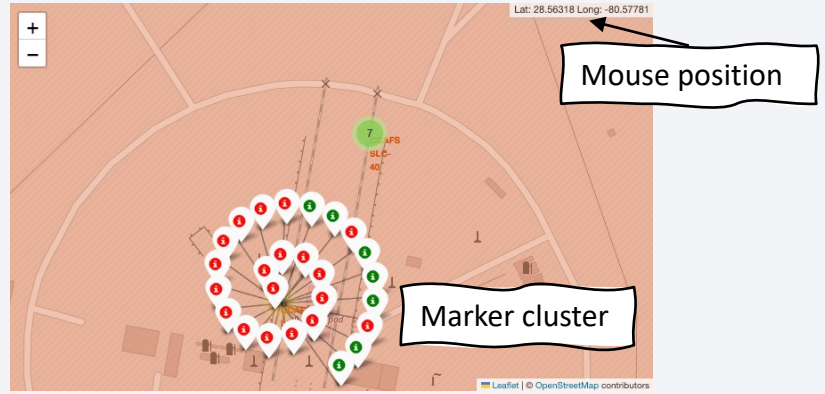
EDA with SQL

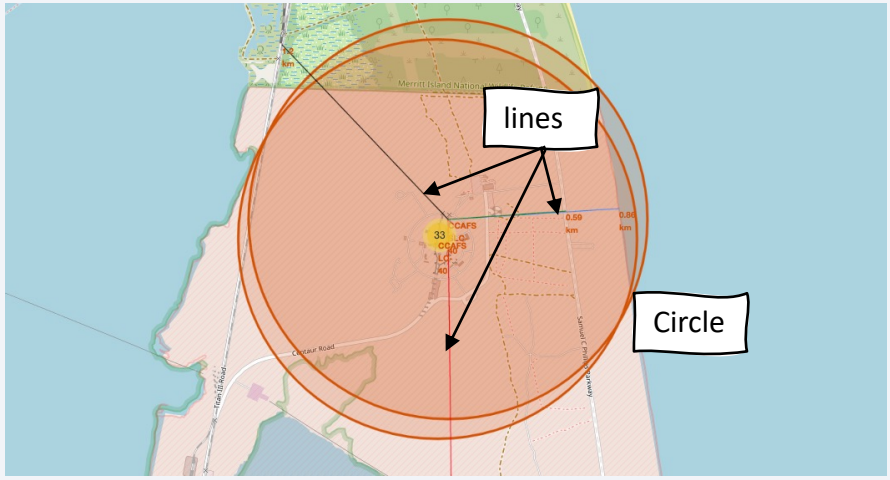
- Following SQL queries are performed:
  - distinct - to show unique launch sites
  - where Launch_Site like 'CCA%' - where clause with wildcard character '%' to find launch sites beginning with the string 'CCA'
  - limit 5 - to print only 5 records
  - sum() – to find the total payload mass
  - min() – to find the first successful landing outcome on the ground pad
  - and – True when both conditions are met
  - between – to find boosters that have been successful in drone ship landings and have payload mass between 4000 and 6000
  - count(*) – to find the total number
  - group by – to find the successful/failed mission outcomes, ranking landing outcomes
  - order by – sort result in descending order
  - subquery – (*PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)*) – find booster versions that carried max payload
  - substr() – select substring
- Add the GitHub URL of the completed EDA with SQL notebook – Link to Jupyter Notebook!

Build an Interactive Map with Folium

- The following map objects and the reason they were added are listed below –
  - Circles – to add highlighted circle area
  - Markers - add text label on coordinates
  - Marker Cluster (Icons) – to add icon labels for success/failure launch outcome
  - Mouse Position – coordinates for the position of the mouse pointer on the map
  - lines – to illustrate distance of launch site to the coastline, rail line, highway and the nearest city
- GitHub URL of the completed interactive map with Folium map – [Link to completed Jupyter Notebook!]







Build a Dashboard with Plotly Dash

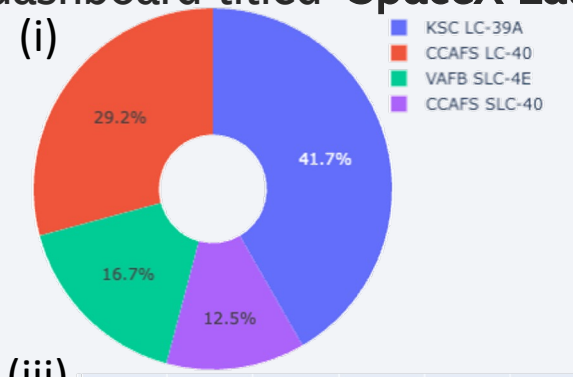
- The plots/graphs along with explanations are added to the dashboard titled **'SpaceX Launch Records Dashboard'** –

Plot	Explanation of plot
Pie Chart	Successful launches at all launch sites (i)
	Launch success at specific launch site (ii)
Scatter Plot	Correlation between payload and success with each booster version category (iii)

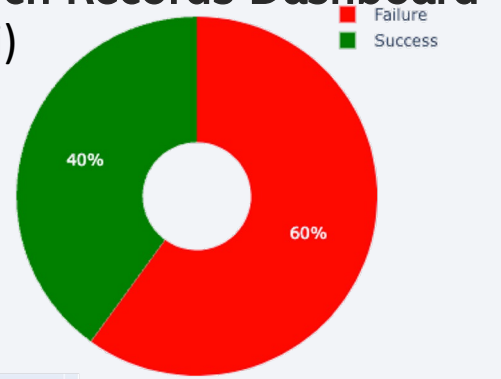
- The explanation for the interactions

Interaction	Purpose
Dropdown	Select between all launch sites or a specific site (iv)
Slider	Set a payload range (range on x-axis) for the scatter plot (v)

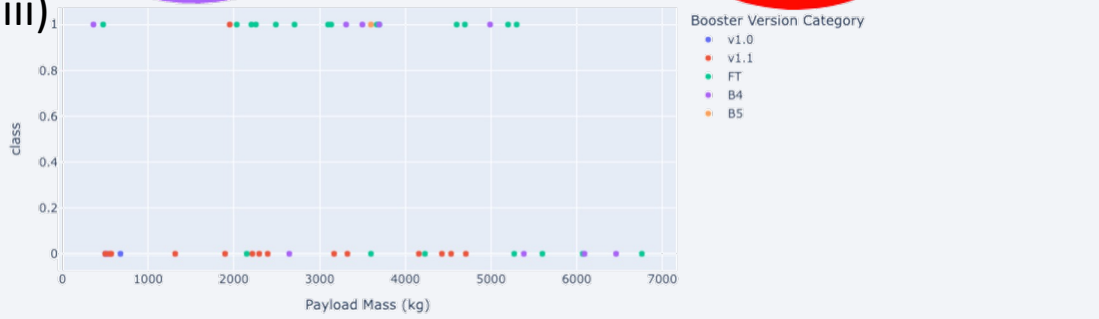
(i)



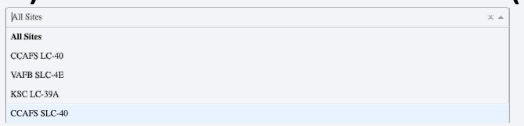
(ii)



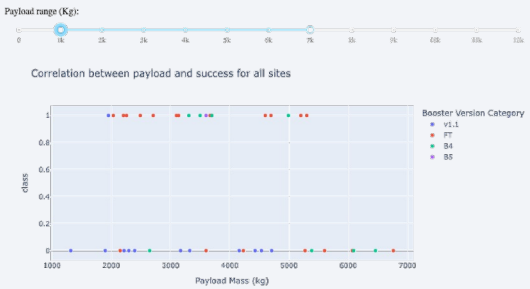
(iii)



(iv)



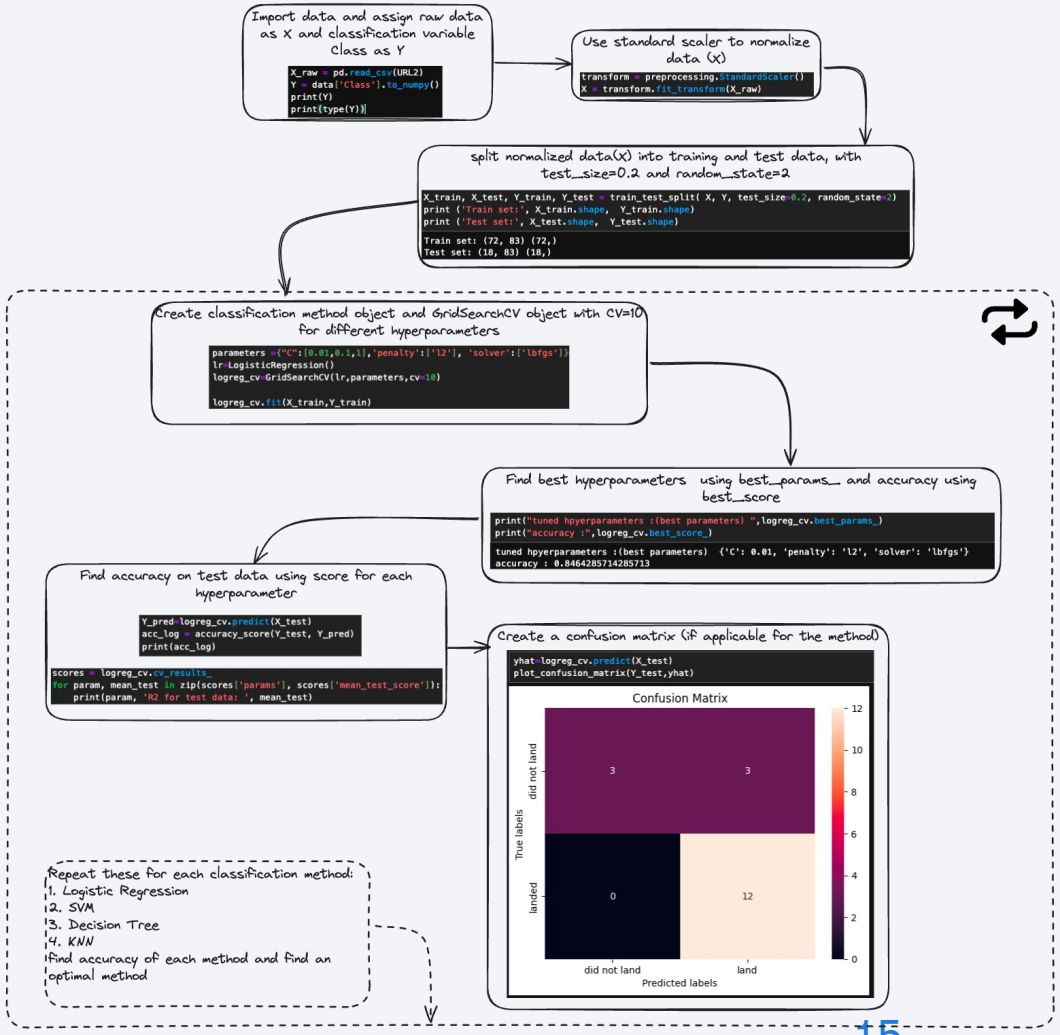
(v)



- GitHub URL of the completed Plotly Dash lab - [Link to the python code!]

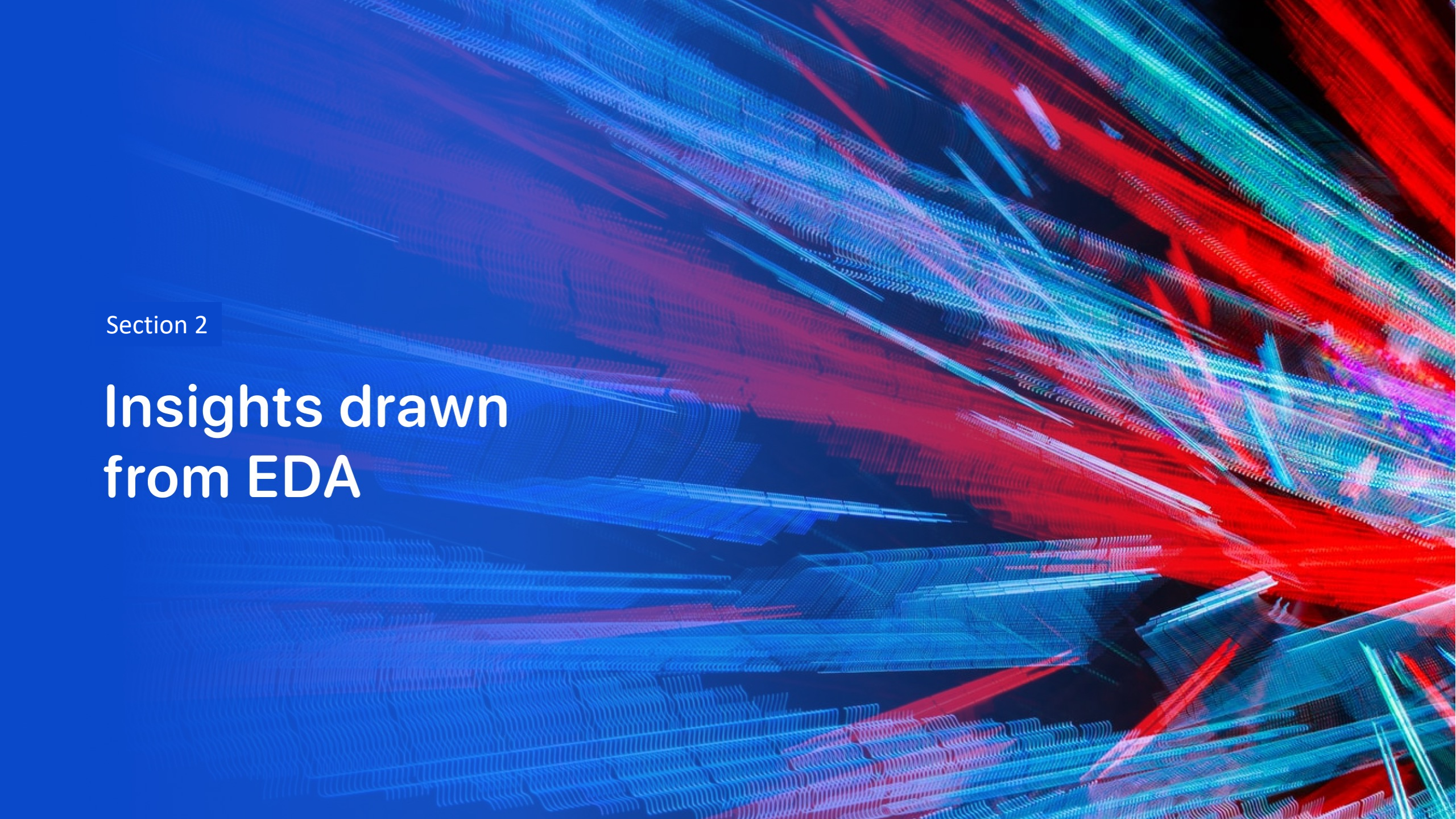
Predictive Analysis (Classification)

- The following steps are followed to create and evaluate a classification model-
  - Clean the classification variable, 'Class' in the dataset (*.to_numpy()*)
  - Standardize the data using *.StandardScaler()* and use *.fit_transform()*
  - Split data in training and testing data using *train_test_split() function*
  - Create the classification method object using sklearn module
  - Create a gridsearch object with CV=10
  - Fit the object using *.fit()* method in the training data
  - Find optimum hyperparameters using *.best_params* and accuracy of method using *.best_score_*
  - Use test data to find the predicted value and calculate accuracy using *accuracy_score()* method. Also find the confusion matrix wherever applicable
- The **Logistic Regression** method, **SVM, Decision Tree,** and **KNN** classification methods are evaluated and their accuracies are compared
- The GitHub URL of the completed predictive analysis lab – [Link to Jupyter notebook!]



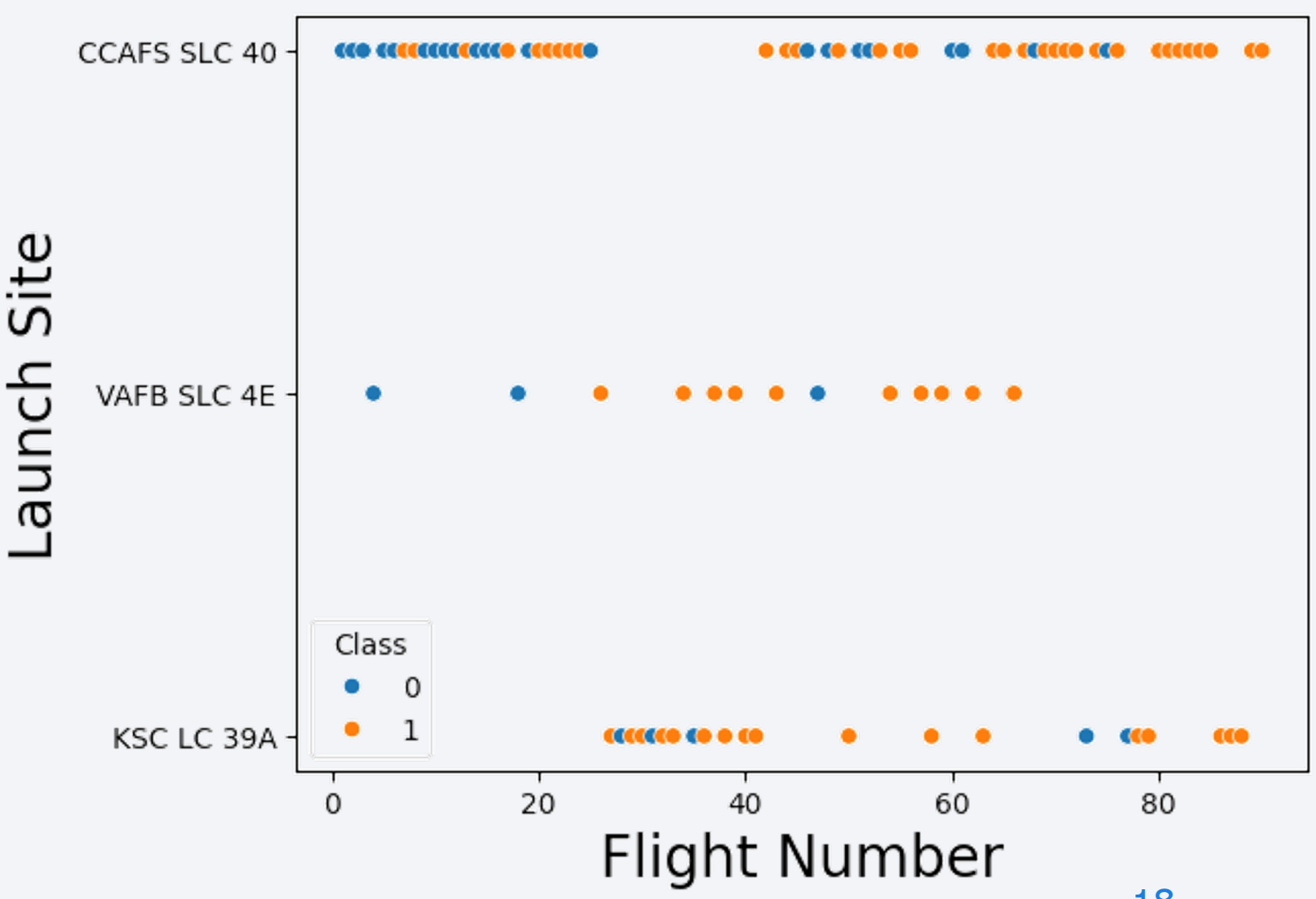
Results

- Exploratory data analysis results
  - Through Visualizations using seaborn and pandas to understand different features
  - Through SQL to understand the dataset and different properties
- Interactive analytics demo in screenshots
  - Analyzing the proximity of launch sites to different accessibility points like highways, rail lines, coast, and the nearest major city using Folium maps and map objects
  - Plotly dashboard highlighting SpaceX launch records and the correlation between Class and Payload Mass
- Predictive analysis results
  - Logistic Regression, SVM, Decision Tree, and KNN are used
  - Decision Tree is found to be the best for prediction
- For all subsequent plots, Class = 0 (Failure) and Class = 1 (Success)


Section 2
Insights drawn from EDA

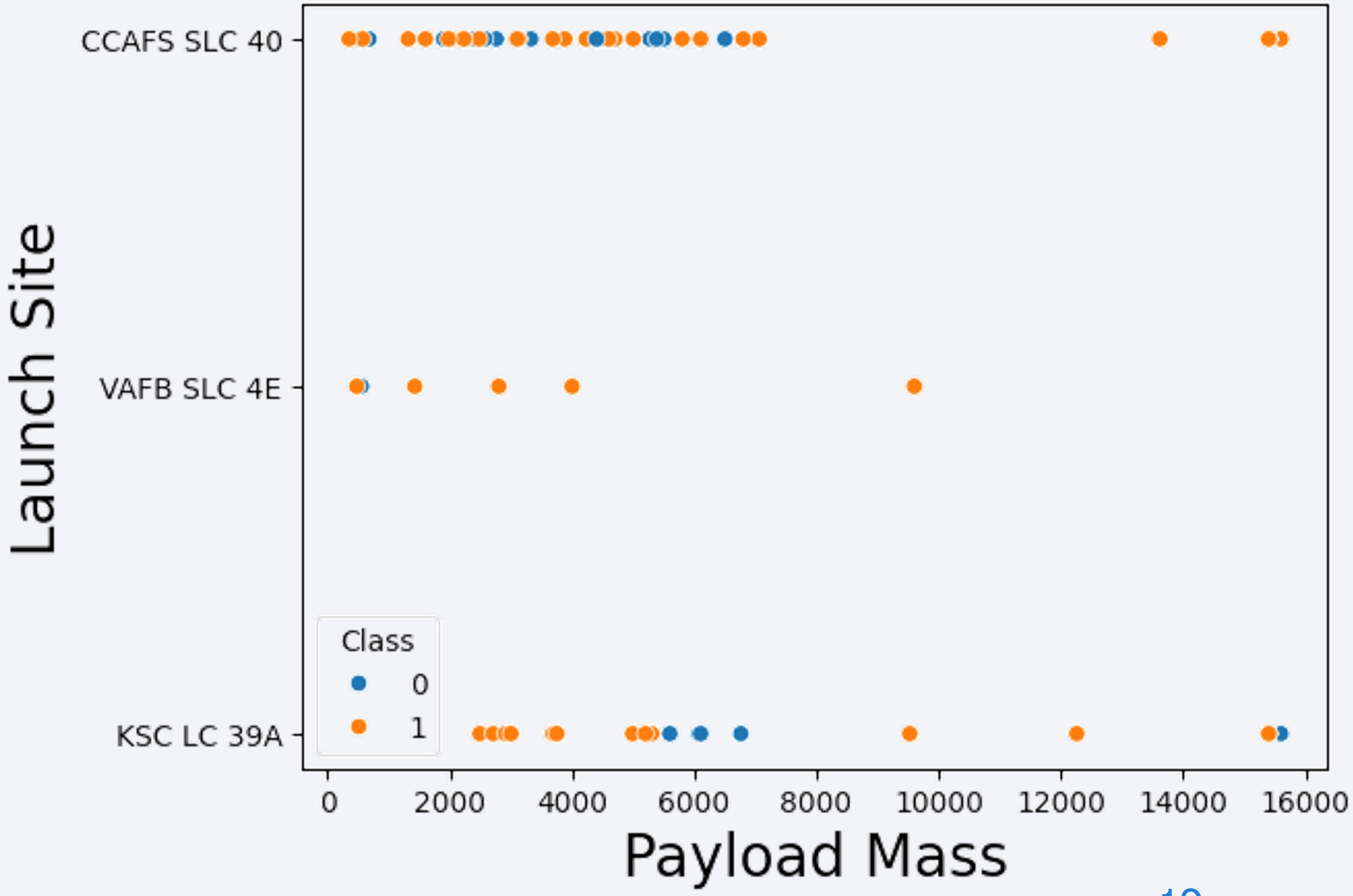
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- The scatter plot indicates -
  - Considerable failed launches from CCAFS SLC 40 for flight numbers between 0-30
  - The success rate increases for CCAFS SLC 40 after flight number 60
  - Between flight numbers 30-40, only VAFB SLC 4E and KSC LC 39A had launches. Also, all launches on VAFB SLC 4E between these flight numbers are successful
  - The success rate for a launch site increases with the greater flight number. Beyond flight number 80 success increases significantly



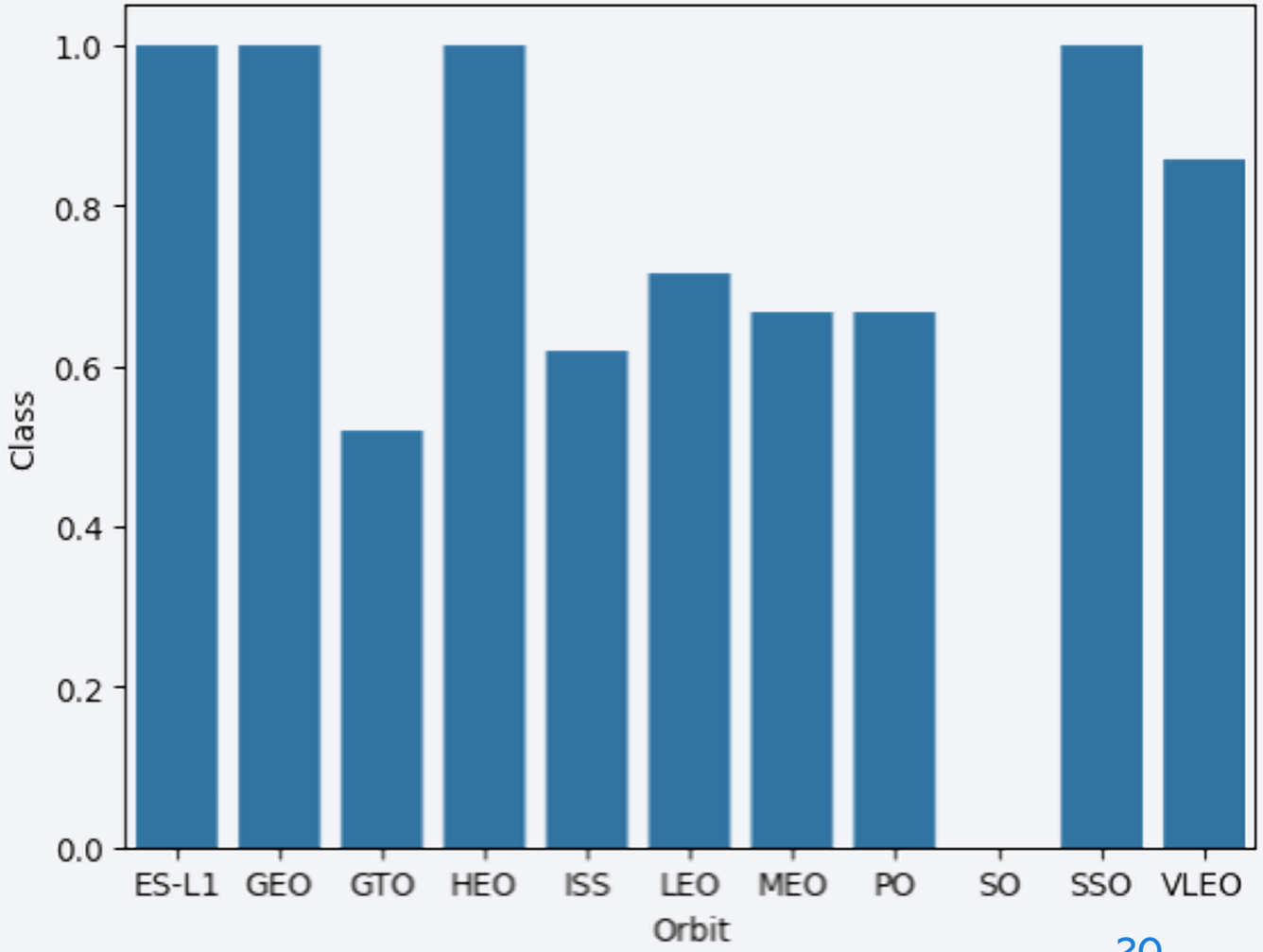
Payload vs. Launch Site

- Show a scatter plot of Payload Mass vs. Launch Site
- The scatter plot indicates –
  - CCAFS SLC40 and KSC LC 39A can handle both light and heavy payloads. VAFB SLC 4E can handle light to medium payloads
  - At less than 5000 kg, KSCLC 39A has the highest success rate, followed by VAFB SLC 4E
  - 5000-7000 kg payload masses have the lowest success rate
  - CCAFS has a 100% success rate for payload masses > 12000 kg
  - No relationship between payload mass and success for KSC LC 39A



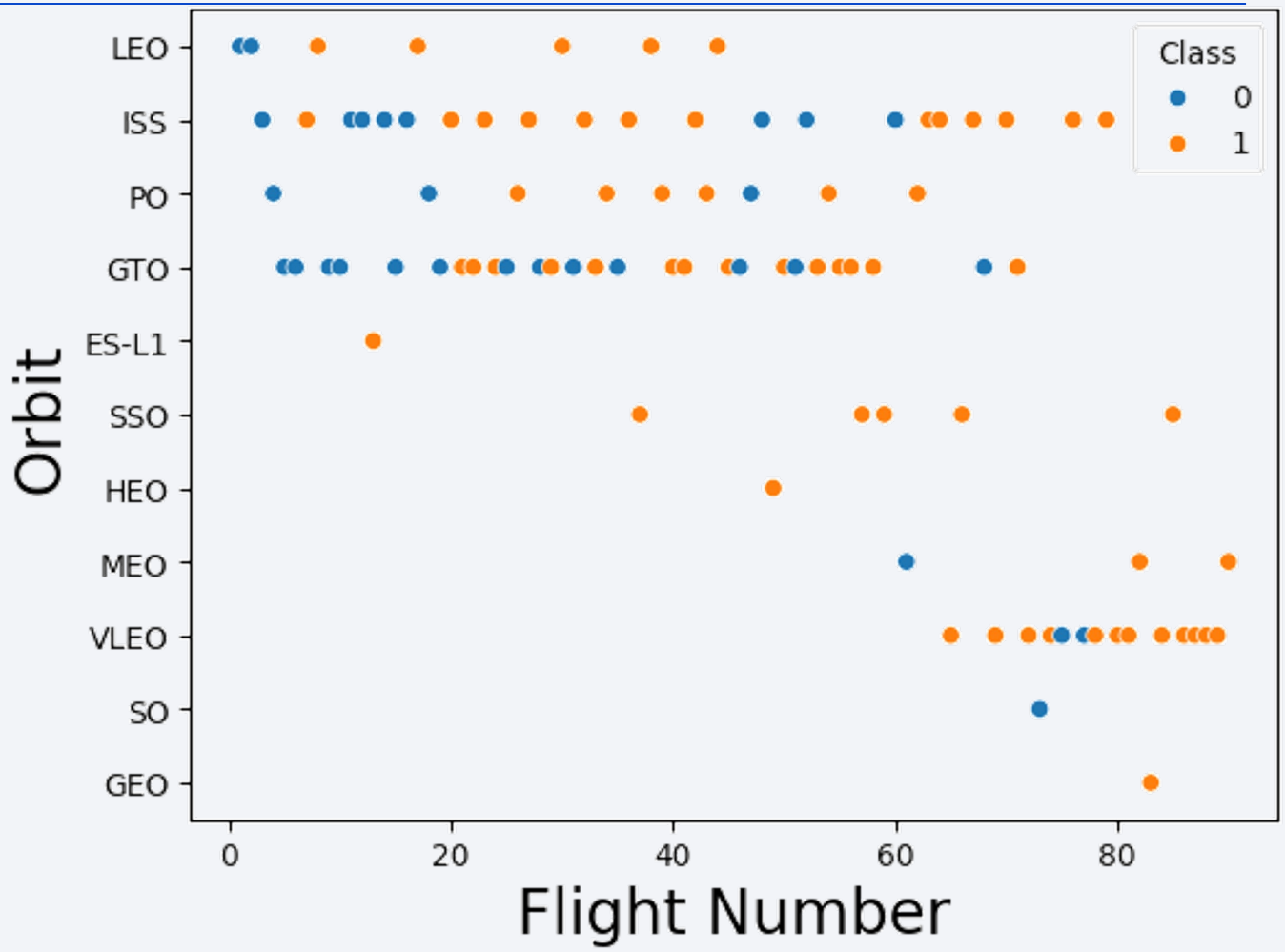
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- The bar plot indicates –
  - ES-L1, GEO, HEO, and SSO have the highest success rate(1)
  - GTO orbit has the lowest success rate. No launches in SO



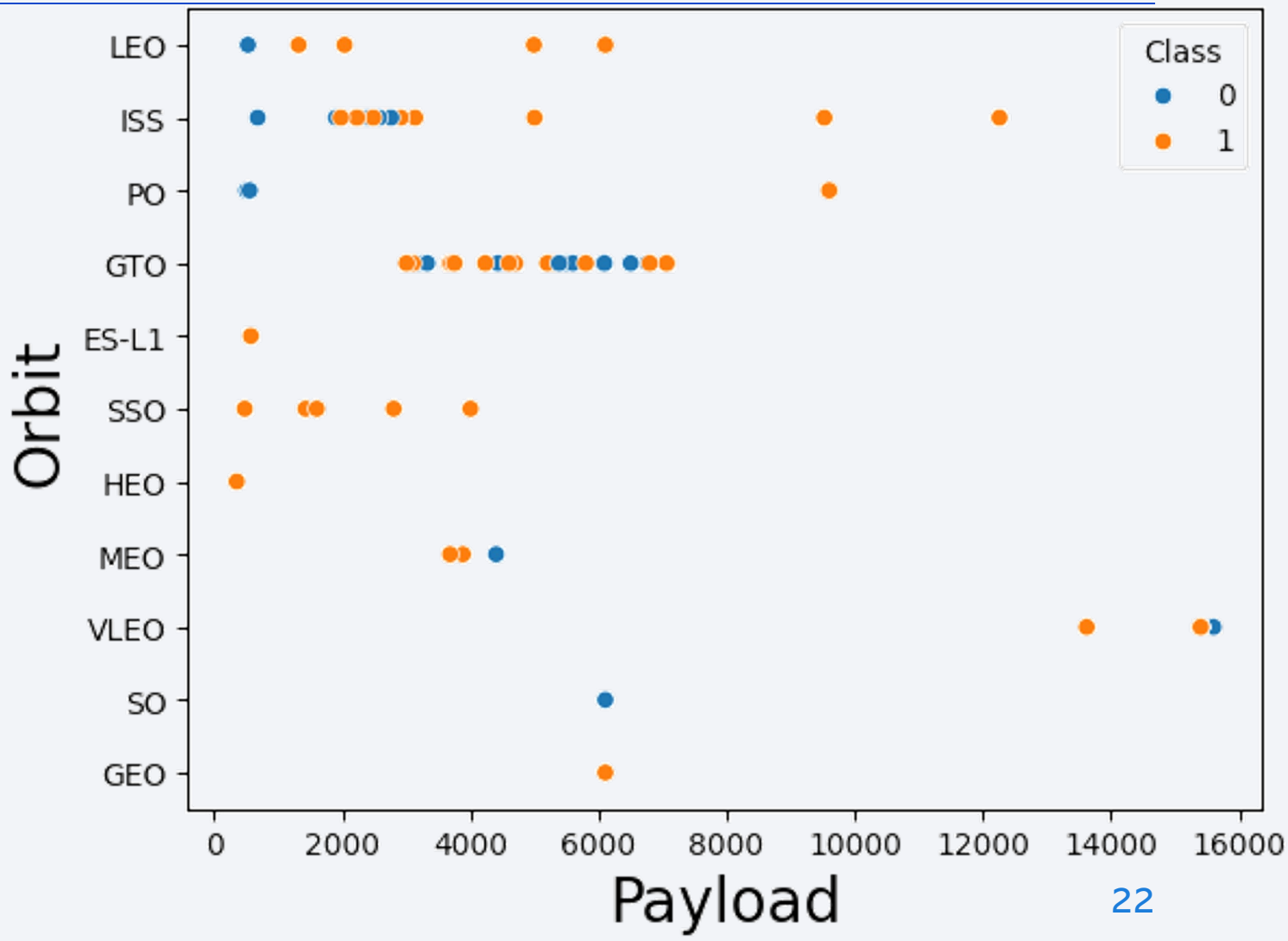
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- The scatter plot indicates –
  - In LEO, the success is related to flight number
  - For ISS, PO and GTO orbits, there is no obvious relationship between success and flight number
  - ES-L1, SSO, and HEO launches have all been successful
  - VLEO launches have over 85% success rate



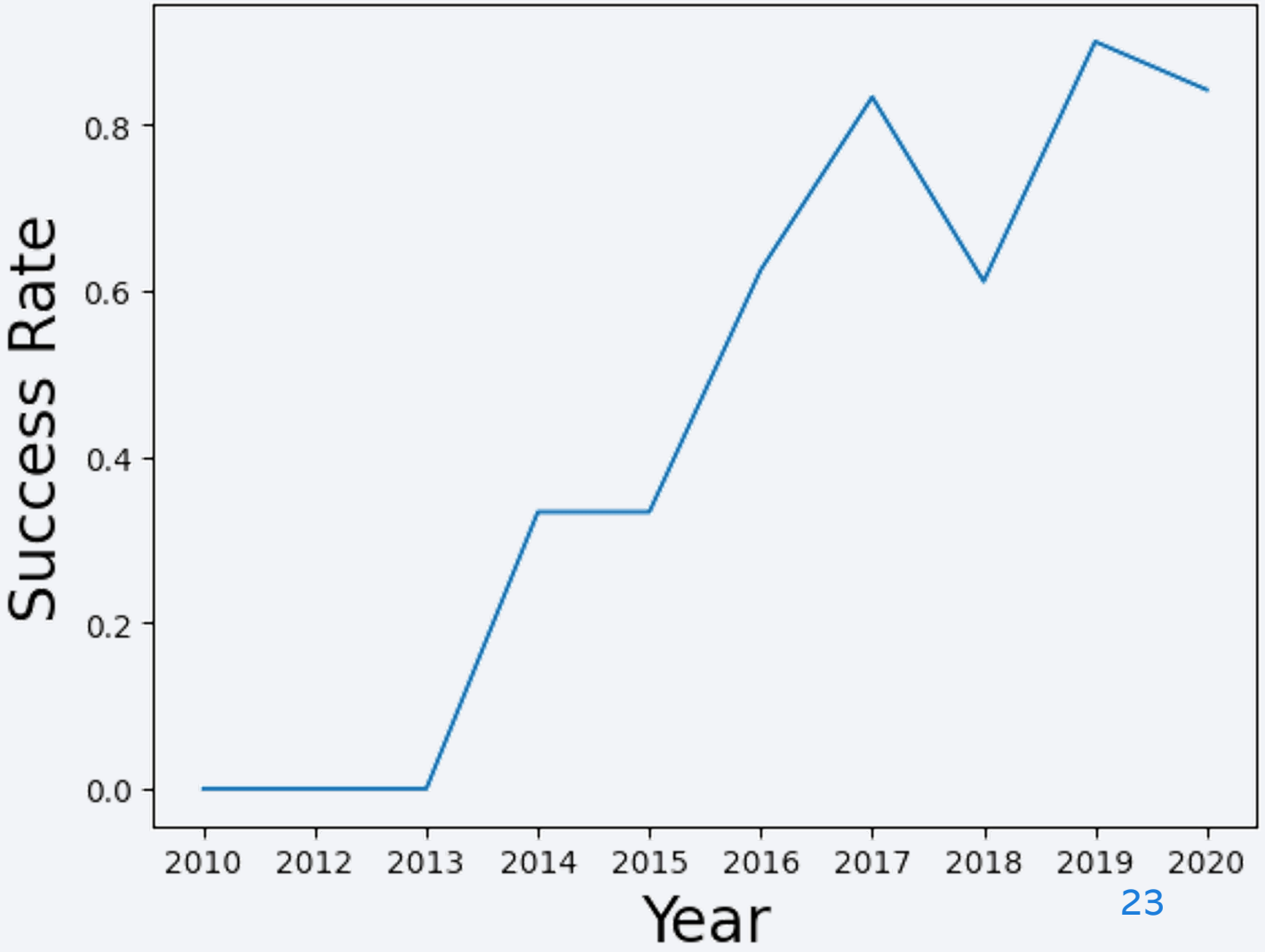
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- The scatter plot indicates-
  - With light payloads (< 5000kg), 100% landing success exists for ES-L1, SSO and HEO orbits
  - Heavier the payload, the higher the landing success rate for the Polar(PO), ISS, and LEO orbits
  - For GTO there is no obvious relationship between success rate and payload mass



Launch Success Yearly Trend

- Show a line chart of the yearly average success rate
- The line plot indicates –
  - Year-on-year success rate increases between 2013 - 2017 (same rate between 2014-15)
  - The success rate dipped between 2017-2018, and recovered from 2018-19



All Launch Site Names

- Find the names of the unique launch sites
- The query result is presented below-

```
%sql select distinct Launch_Site FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

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

- distinct – selects unique Launch_Site names in table SPACEXTABLE

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- The query result is presented below-

```
%sql select * FROM SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- where Launch_Site like ‘CCA%’ - where clause with wildcard character ‘%’ to find launch sites beginning with the string ‘CCA’
- limit 5 - to print only 5 records

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- The query result is presented below – 45596 kg

```
%sql select sum(PAYLOAD_MASS__KG_) AS 'Total Payload Mass Carried by Boosters(launched by NASA)' from SPACEXTABLE where Customer='NASA (CRS)'
 * sqlite:///my_data1.db
Done.
Total Payload Mass Carried by Boosters(launched by NASA)
                                                  45596
```

- sum(PAYLOAD_MASS__KG_) - find the total payload mass
- AS – renames the query result to **'Total Payload Mass Carried by Boosters(launched by NASA)**'
- where – checks if Customer = 'NASA (CRS)'

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The query result is presented below – 2534.67 kg

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average Payload Mass carried by booster F9 v1.1'  from SPACEXTABLE where Booster_Version like "F9 v1.1%"
 * sqlite:///my_data1.db
Done.
Average Payload Mass carried by booster F9 v1.1
                                    2534.6666666666665
```

- avg(PAYLOAD_MASS__KG_) – find the average of payload mass
- like – checks for Booster_Version string staring with ‘F9 v1.1%’ , where ‘%’ is the wildcard character

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- The query result is presented below – 2015-12-22

```
%sql select min(Date) as 'first succesful landing outcome in ground pad' from SPACEXTABLE where Mission_Outcome like '%Success%' and Landing_Outcome like '%ground pad%'
 * sqlite:///my_data1.db
Done.
 min(Date)
2015-12-22
```

- min(Date) – finds the minimum Date
- like '%Success%' – Find the string 'Success' in the Mission_Outcome
- Like '%ground pad%' - Find the string 'ground pad' in the Landing_Outcome
- and – is True only when there is a 'Success' in Mission_Outcome and 'ground pad' in the Landing_Outcome

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- The query result is presented below –

```
[41]: %sql select Booster_Version from SPACEXTABLE where Mission_Outcome like '%Success%' and Landing_Outcome like '%drone ship%' and PAYLOAD_MASS__KG_ between 4000 and 6000
 * sqlite:///my_data1.db
Done.
[41]: Booster_Version
      F9 FT B1020
      F9 FT B1022
      F9 FT B1026
      F9 FT B1021.2
      F9 FT B1031.2
```

- and – is True only when there is a ‘Success’ in Mission_Outcome and ‘drone ship’ in the Landing_Outcome
- between – selects payload mass between the range 4000-6000

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- The query result is presented below –

```
%sql select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome
 * sqlite:///my_data1.db
Done.
```

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

- count(*) – find the total number of items in the group
- group by – to find groups of successful/failed mission outcomes

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- The query result is presented below –

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- subquery – (*PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)*) – find booster versions that carried max payload (15600 kg)

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- The query result is presented below –

```
%sql select substr(Date, 6,2) as 'Month', Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where substr(Date,0,5)='2015' and Landing_Outcome like '%Failure (drone ship)%'
 * sqlite:///my_data1.db
Done.
```

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

- substr() – select substring from Date to get Month(substr(Date, 6,2)) and Year(substr(Date,0,5)='2015')
- like – find '%Failure (drone ship)%' string in Landing_Outcome
- Only showing Month(substr(Date, 6,2)), Landing_Outcome, Booster_Version and Launch_Site

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- The query result is presented below –

```
%sql select Landing_Outcome, count(Landing_Outcome) from SPACEXTABLE group by Landing_Outcome order by count(Landing_Outcome) desc
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count(Landing_Outcome)
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

- order by – sort result in descending order of Landing_Outcomes
- group by – group the Landing_Outcomes in the dataset

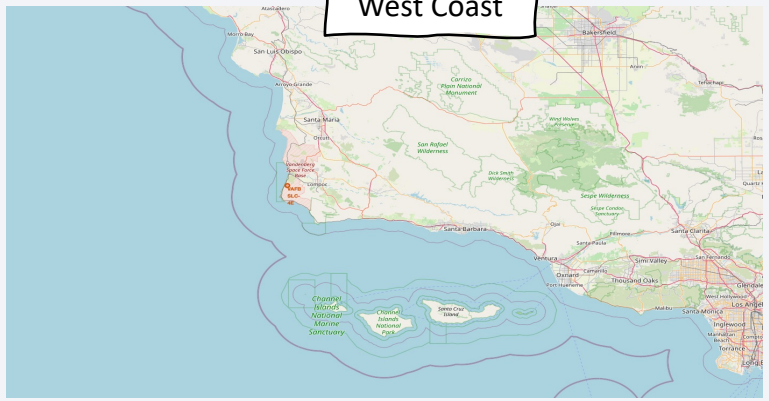

Section 3
Launch Sites
Proximities Analysis

Location of All SpaceX Launch Sites

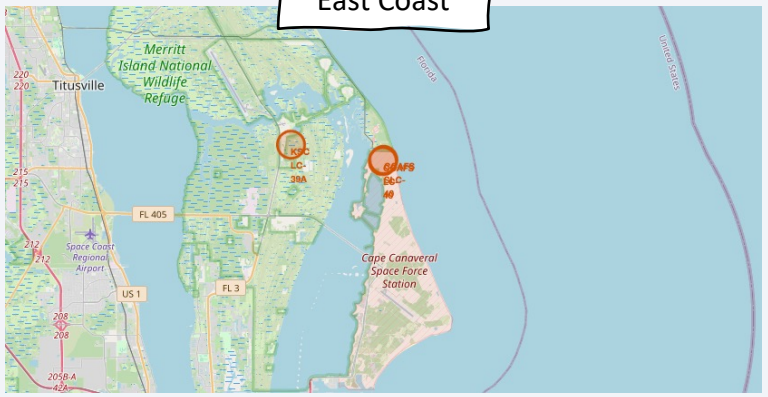
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

All Sites

West Coast



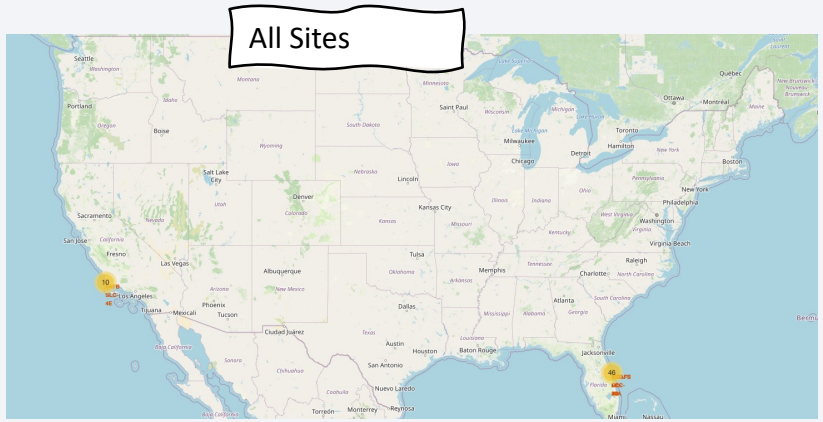
East Coast

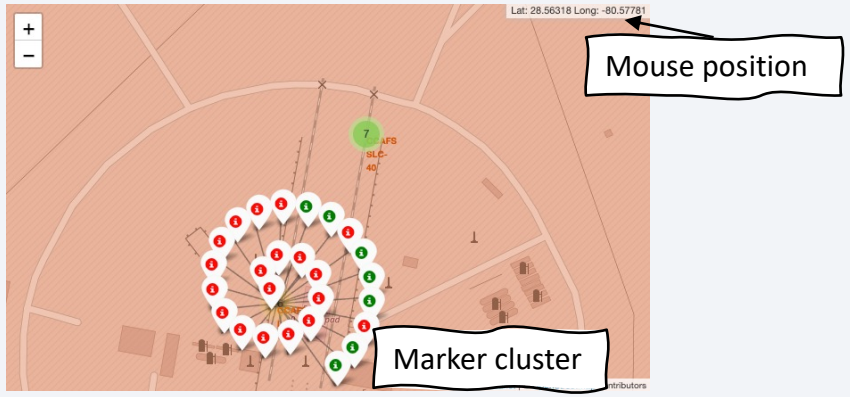


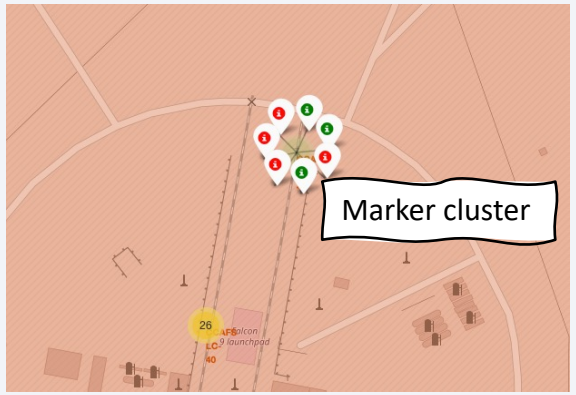
- Important elements –
  - Circle - adds highlighted circle area to the coordinates for the four launch sites
  - Label – name of the launch sites in red (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E)

Launch Outcomes at Each Launch Site

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map







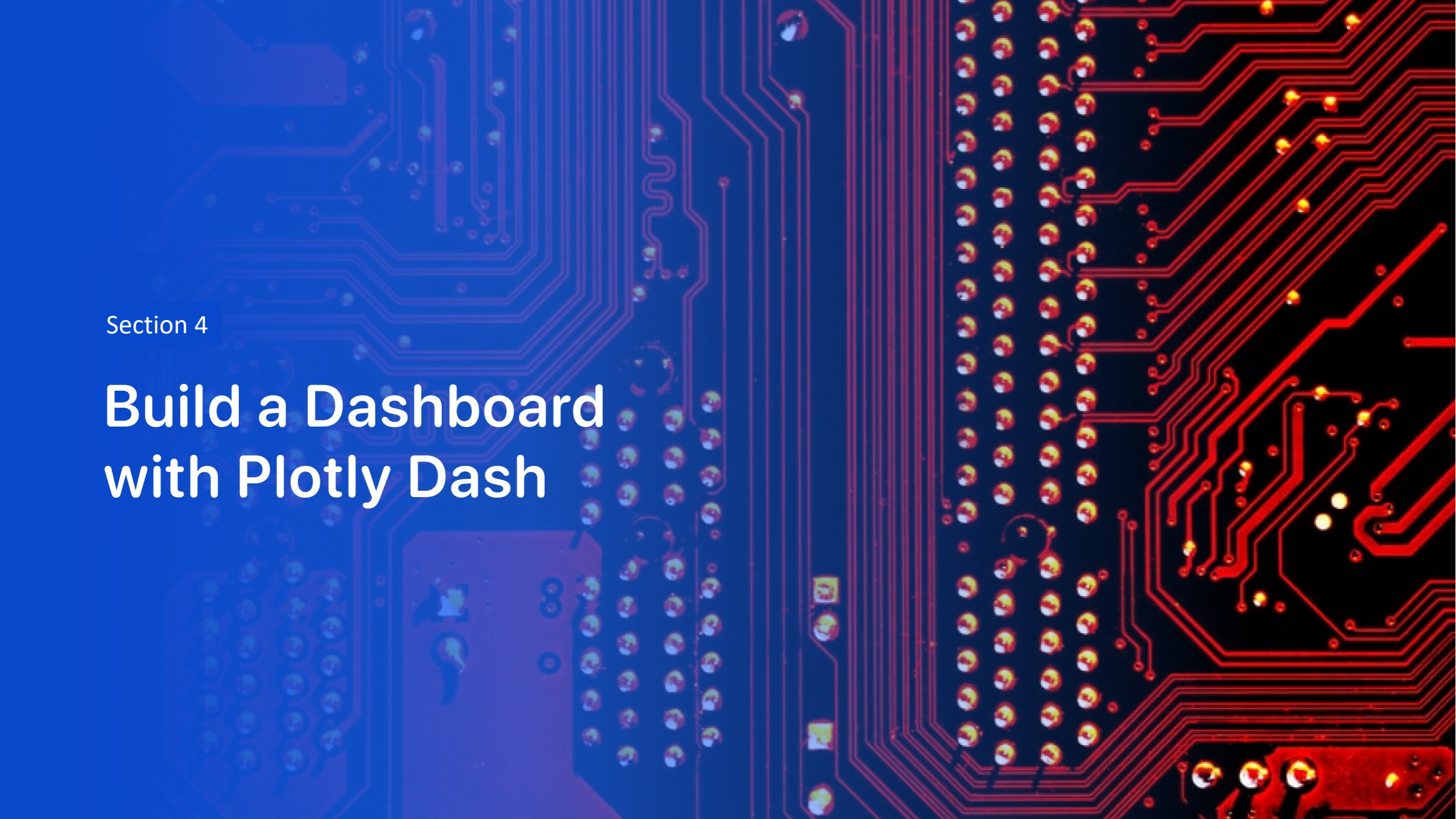
- Important elements –
  - Marker Cluster (Icons) – icon labels for launch outcome. Red Label – Failure; Green Label – Success
  - Number on icons – Total launches; 10 launches from the West Coast, 46 from the East Coast
  - Mouse Position - coordinates for the position of the mouse pointer on the map

Accessibility of Launch Sites

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

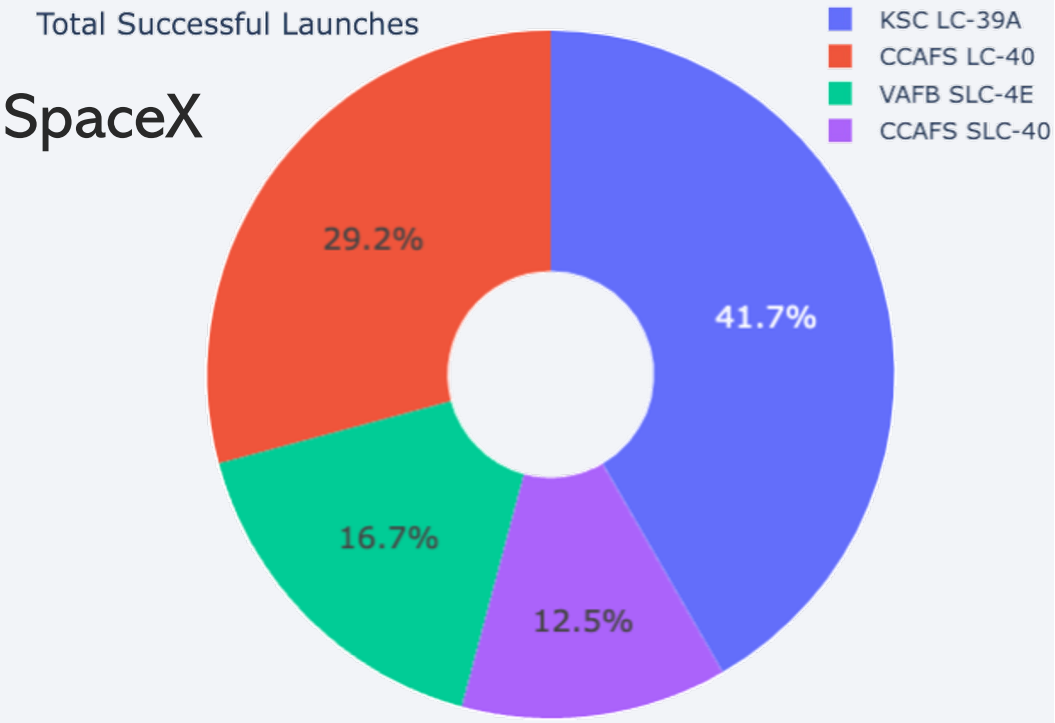
- Important elements –
  - lines – to illustrate the distance of the launch site to the coastline(blue), rail line(black), highway(green), and the nearest city(Melbourne, red)
  - labels – showing distance to:
    - Rail line (black line) = 1.2 km
    - Highway (green line) = 0.59 km
    - Coastline (blue line) = 0.86 km
    - Melbourne(city, red line) = 52 km

- Launch sites are close to Rail lines, highways and coastline for ease of logistics
- Sufficient distance from large cities for safety


Section 4
Build a Dashboard with Plotly Dash

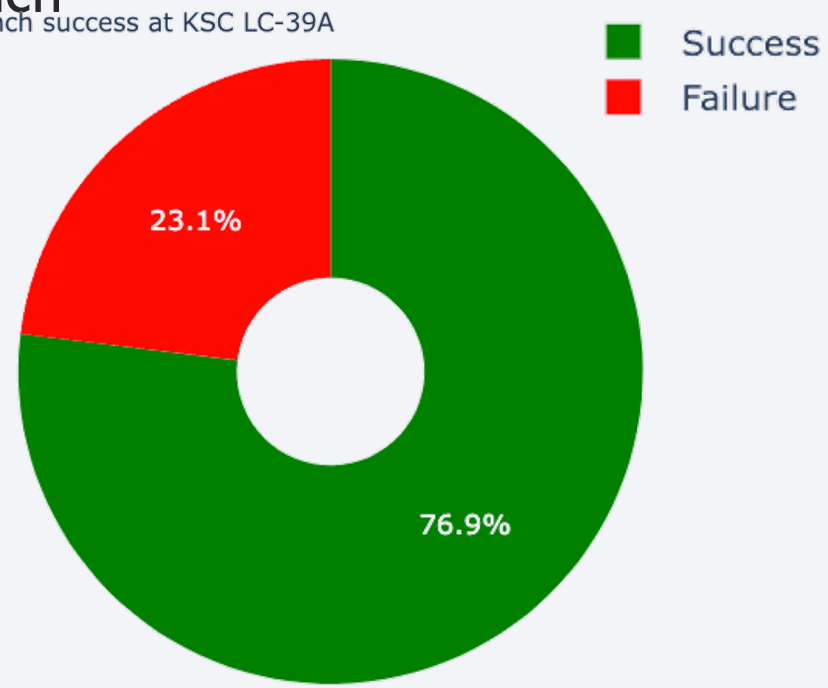
Successful Launches at all SpaceX Launch Sites

- The piechart depicting successful launches at all the SpaceX launch sites
- Important Elements -
  - The success ratio of each launch site is depicted through percentages and colors
  - The chart legend with color labels is also shown
  - The chart title is 'Total Successful Launches'
- Findings:
  - KSC LC-39A has the highest success ratio at 41.7%
  - CCAFS SLC-40 has the lowest success ratio at 12.5%



Highest Launch Success Rate Launch Site

- Piechart depicting Launch success ratio at KSC LC-39A, the launch site with the highest launch success ratio
- Explain the important elements and findings on the screenshot-
  - The success ratio for the site KSC LC-39A is depicted through percentages and colors
  - Title of the chart is 'Launch Success at KSC LC-39A'
  - The legend labels are included as Success(Green) and Failure(Red)
- Findings-
  - KSC LC-39A has a high success rate of 76.9%
  - This can be partly due to high launch success ratio for Payloads weighing <5000kg (see PayloadMass vs LaunchSite scatter plot)

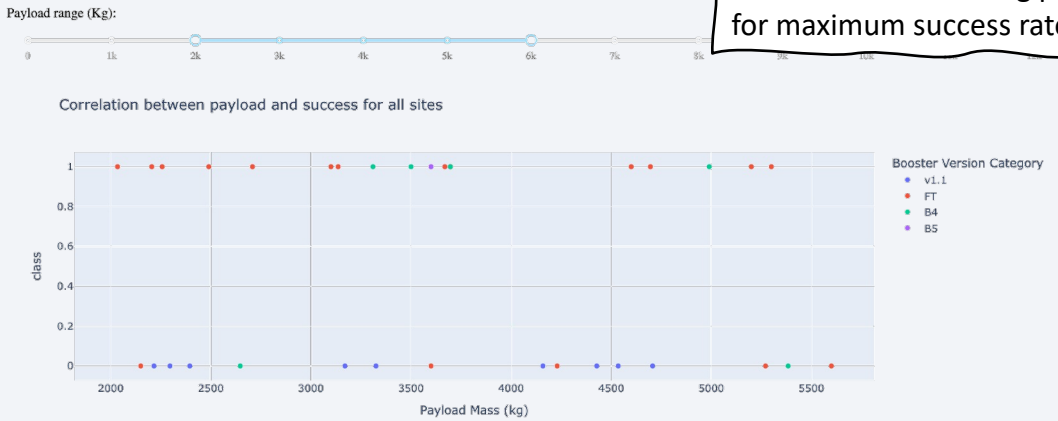


Scatter Plot for Payload Mass vs Launch Outcome

- Scatter plot depicting Payload vs. Launch Outcome for all sites, with different payload selected in the range slider

Full range

Between 2000-6000 kg payload for maximum success rate

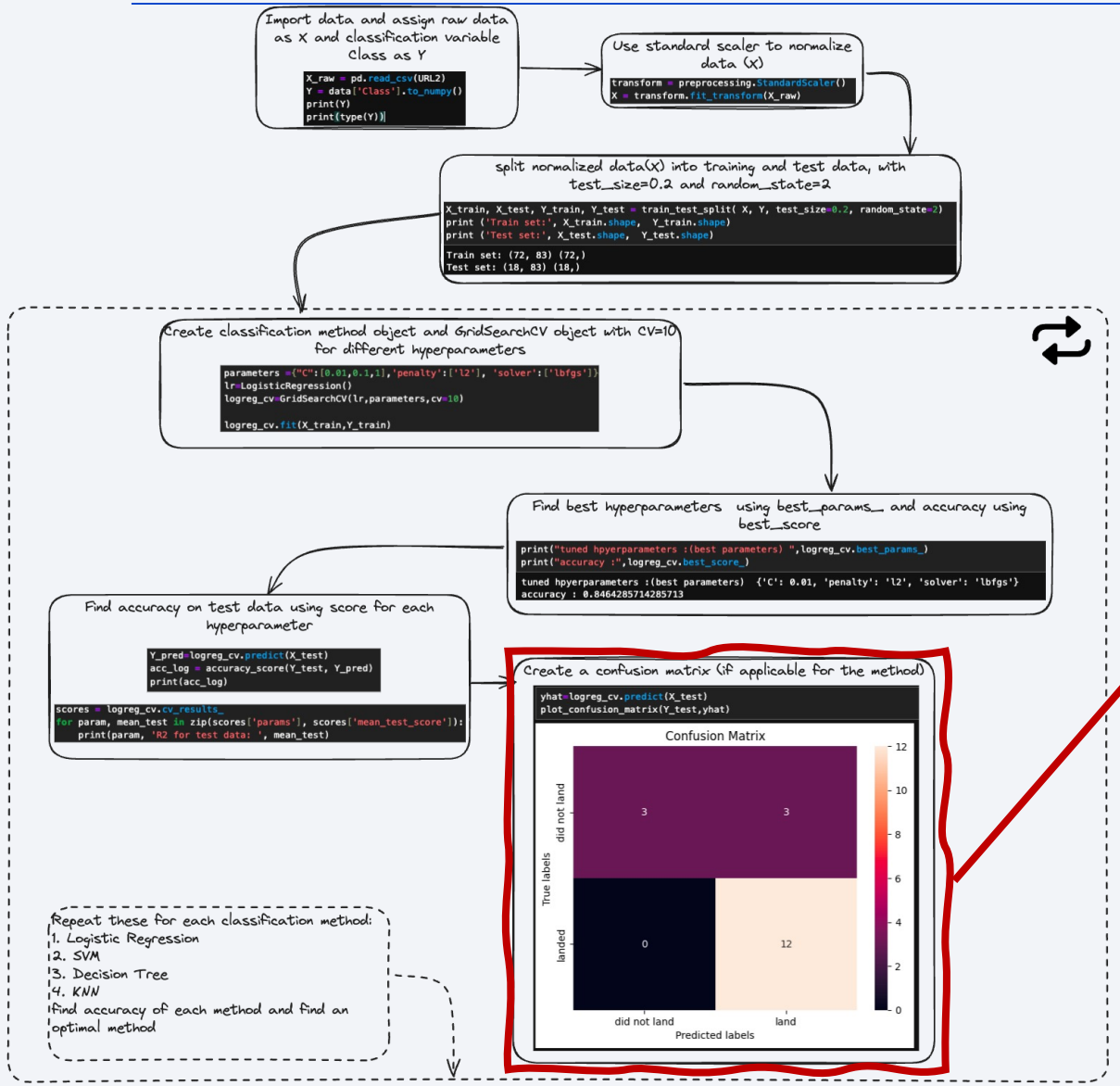


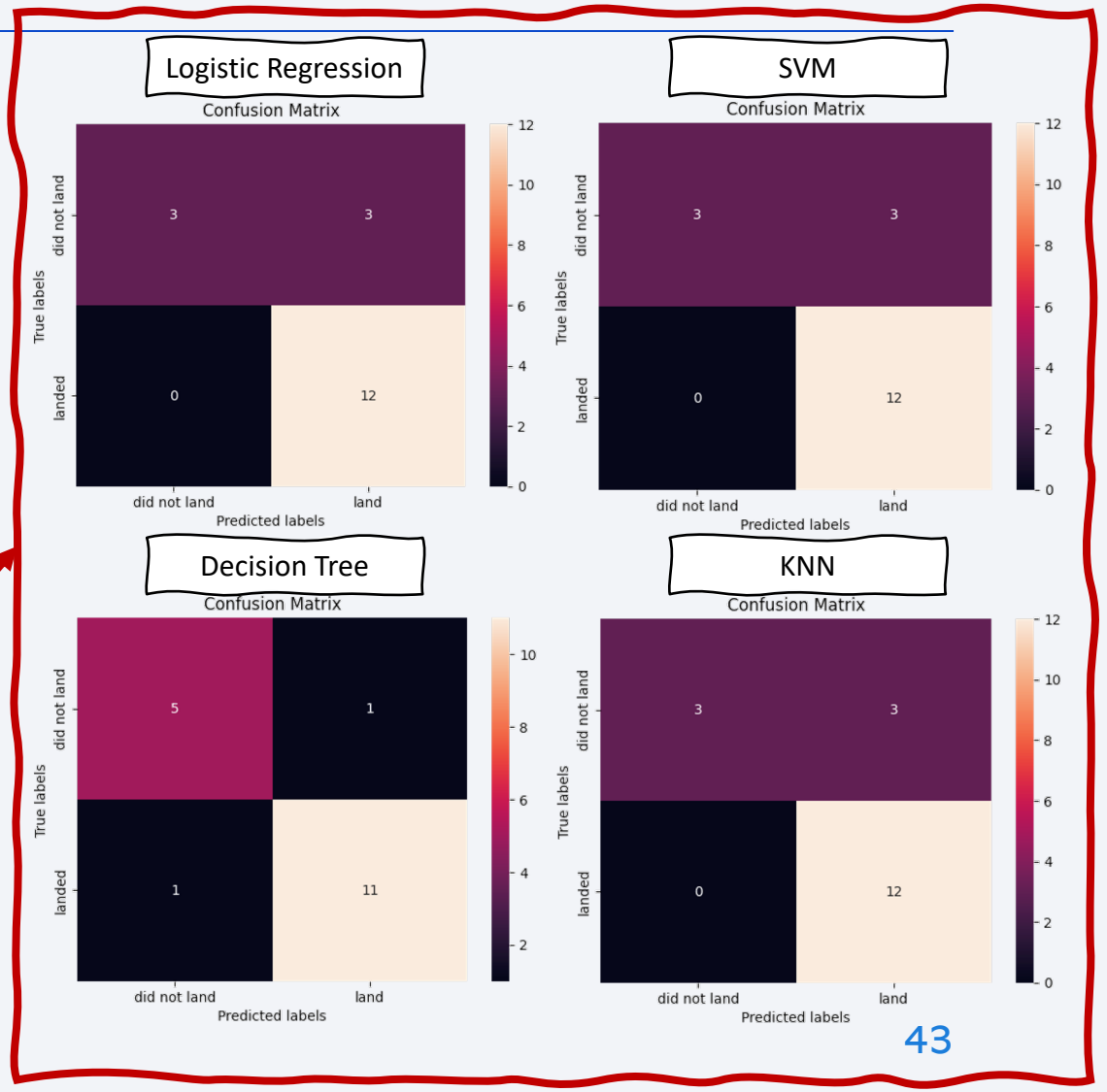
- Important elements –
  - Slider to adjust Payload Range which changes the x-axis of the scatter plot
  - Scatter plot with classification variable ‘Class’ on y-axis and Payload Mass on x-axis for different booster rocket version
- Findings –
  - The highest success rate occurs when the Payload Mass range is between 2000-6000 kg (medium weight)
  - Booster Version ‘FT’ has the highest success rate
  - Booster Version ‘v1.1’ has the highest failure rate

Section 5

Predictive Analysis (Classification)

Predictive Analysis Flowchart

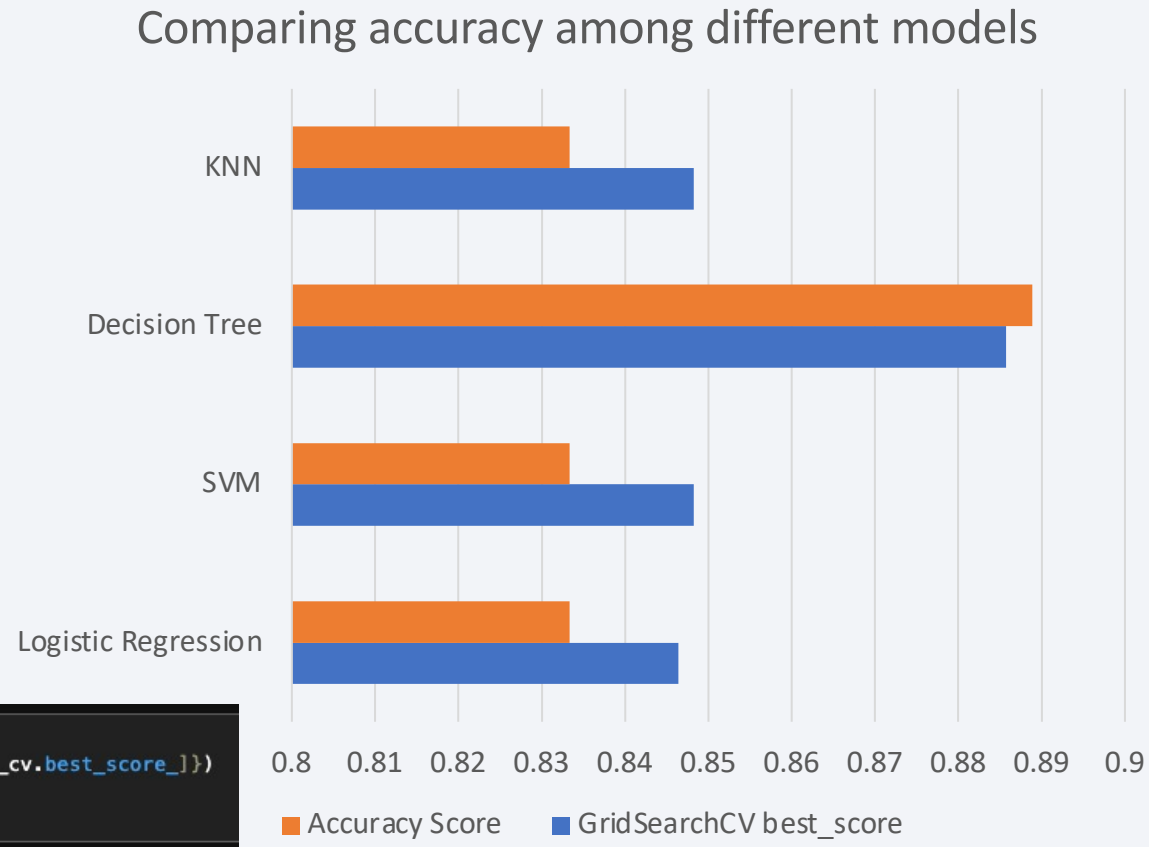




Classification Accuracy

- The built model accuracy for all built classification models is visualized in a bar chart

- Based on the GridSearchCV best scores, the Decision Tree model has the highest classification accuracy

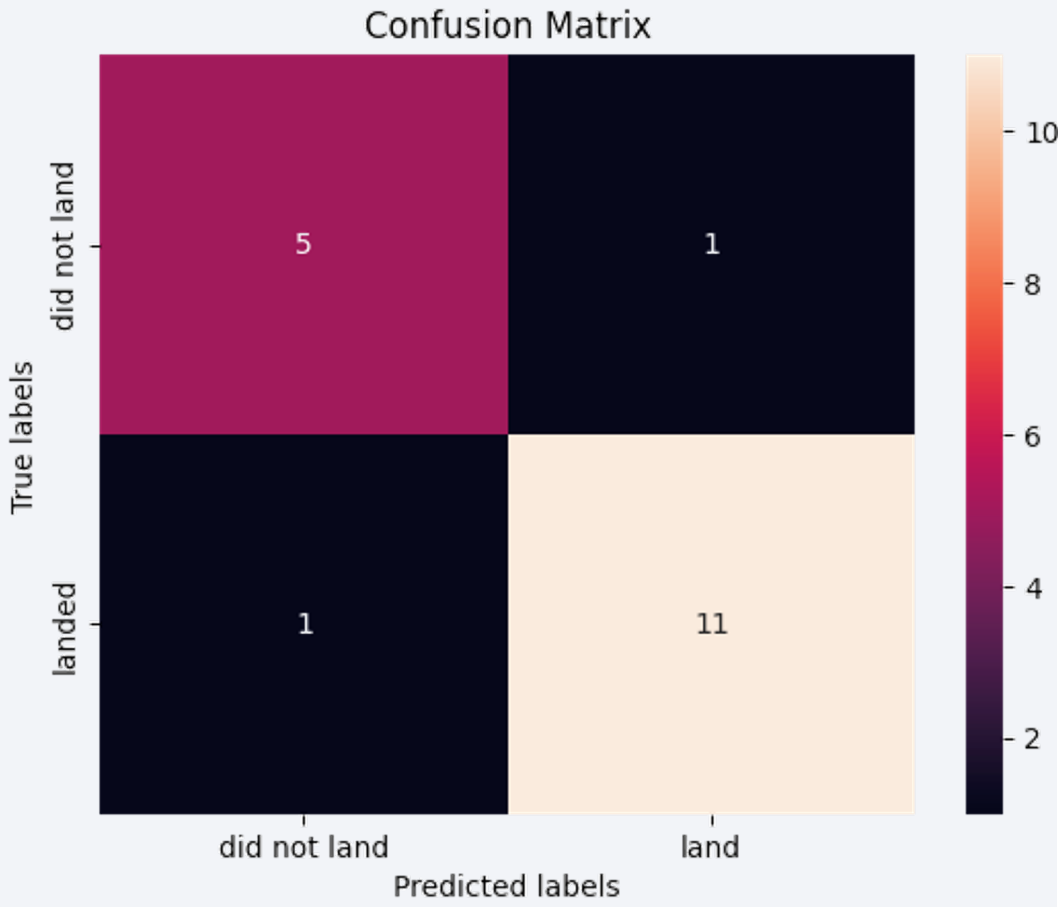



```
acc = pd.DataFrame({'Method':['Logistic Regression','SVM','Decision Tree','KNN'],
                    'accuracy':[logreg_cv.best_score_,svm_cv.best_score_,tree_cv.best_score_,knn_cv.best_score_]})
print(acc.to_string(index=False))
print('Decision Tree appears to be the most accurate')
```

```
              Method  accuracy
Logistic Regression  0.846429
                 SVM  0.848214
       Decision Tree  0.885714
                 KNN  0.848214
Decision Tree appears to be the most accurate
```

Confusion Matrix

- The Confusion Matrix for the Decision Tree is shown
- Important conclusion –
  - Best score = 0.8857
  - Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{11+5}{11+5+1+1} = 0.8889$
  - Precision = $\frac{TP}{TP+FP} = \frac{11}{12} = 0.9167$
  - Sensitivity = $\frac{TP}{TP+FN} = 0.9167$
  - Specificity = $\frac{TN}{TN+FP} = \frac{5}{5+1} = 0.8333$
  - F-score = $2 \times \frac{Precision * Sensitivity}{Precision+Sensitivity} = 0.9167$
  - Best Parameter – Criterion – 'gini' ; Max depth = 6 ; Max features – sqrt ; Minimum samples leaf = 1 ; Minimum samples split = 10 ; Splitter = best

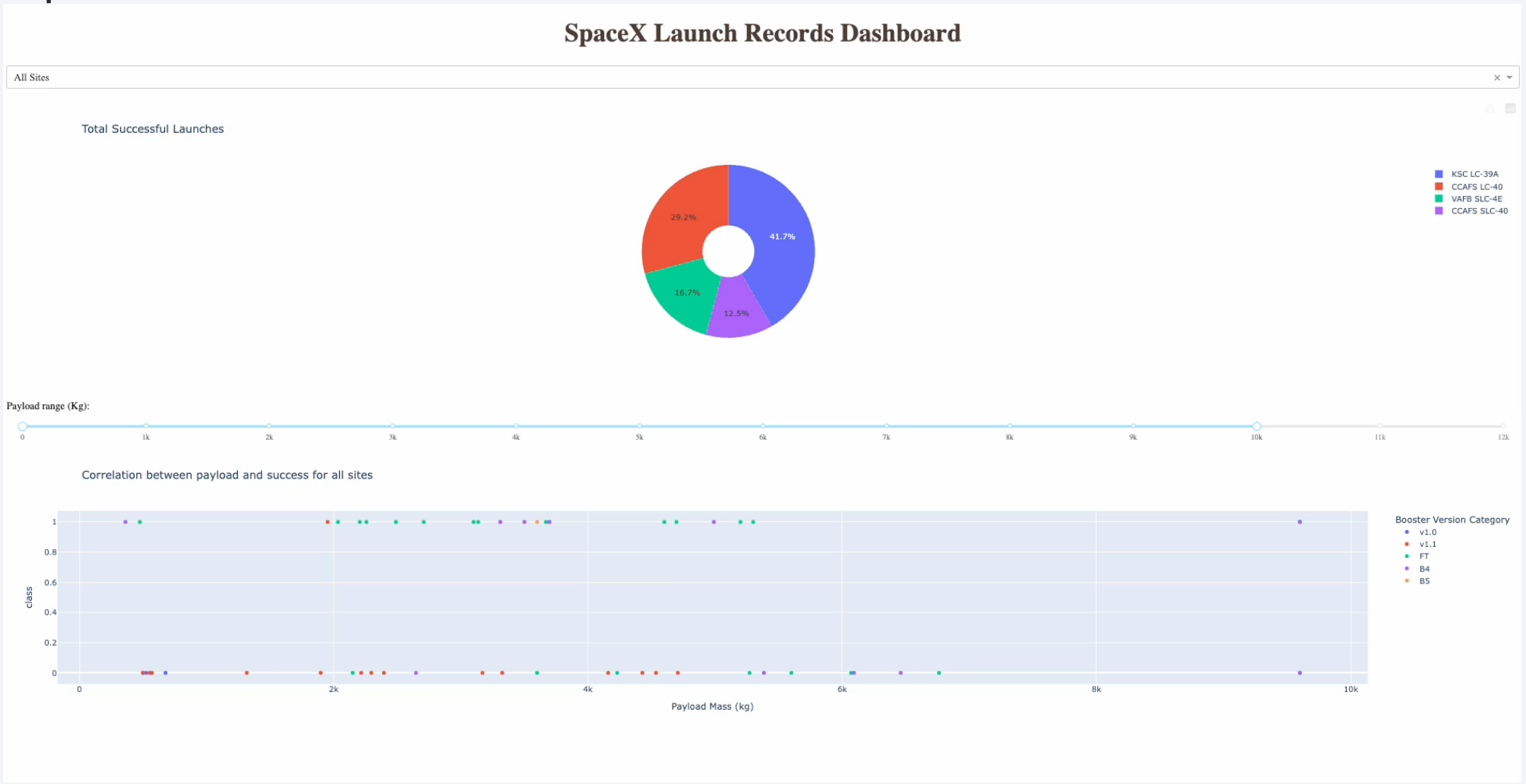


Conclusions

- Publicly available SpaceX launch data was successfully extracted and scraped through the SpaceX API to create a dataset. The collected dataset was cleaned and prepped (wrangling). Through feature engineering, a classification variable ‘Class’ was added to the dataset
- Payload Mass and orbit type features affect the success of the launch outcome
- KSC LC-39A has the highest launch success rate. This can be partly attributed to the high launch success ratio for Payloads weighing <5000kg at the launch site
- Most launch sites have excellent proximity to accessibility points like highways, rail lines, and coastlines
- Maximum success rate occurs in the Payload Mass range between 2000-6000 kg, on all sites. FT booster version has the highest success rate, v1.1 booster has the highest failure rate and BT booster successfully carries the highest payload capacity(total)
- The predictive analysis using different machine learning algorithms reveals that the model created by the ‘Decision Tree Algorithm’ performed the best among the 4 algorithms

Appendix

- Snapshot of the interactive dashboard:



Appendix

- Links to all the notebooks are summarized here:
  - Data Collection
  - Web Scraping
  - Data Wrangling
  - EDA Visualization
  - EDA SQL
  - Folium Maps
  - Plotly Dashboard (Python Code)
  - Predictive Analysis(Machine Learning)

Thank you!