Winning Space Race with Data Science

Kgotso Bruce Moepye
23rd February 2022

Outline

Executive Summary

Summary of methodologies

- Data collection via SQL, Web Scraping and API
- Data Wrangling and Analysis
- Interactive and Data visualization with Folium
- Predictive Analysis with various Classification models

Summary of all results

- Data Analysis with Interactive Visualizations
- Best Model for Predictive Analysis

Introduction

❑ Project background and context

In this Project, we will predict if the Falcon9 will land Successfully. SpaceX advertises Falcon 9 on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars, much of the saving from SpaceX is because they can reuse the first Stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

❑ Problems you want to find answers

- The relationship between Mass. payload and Launch Site
- Considerations for successful Launch and achieve best results

Section 1

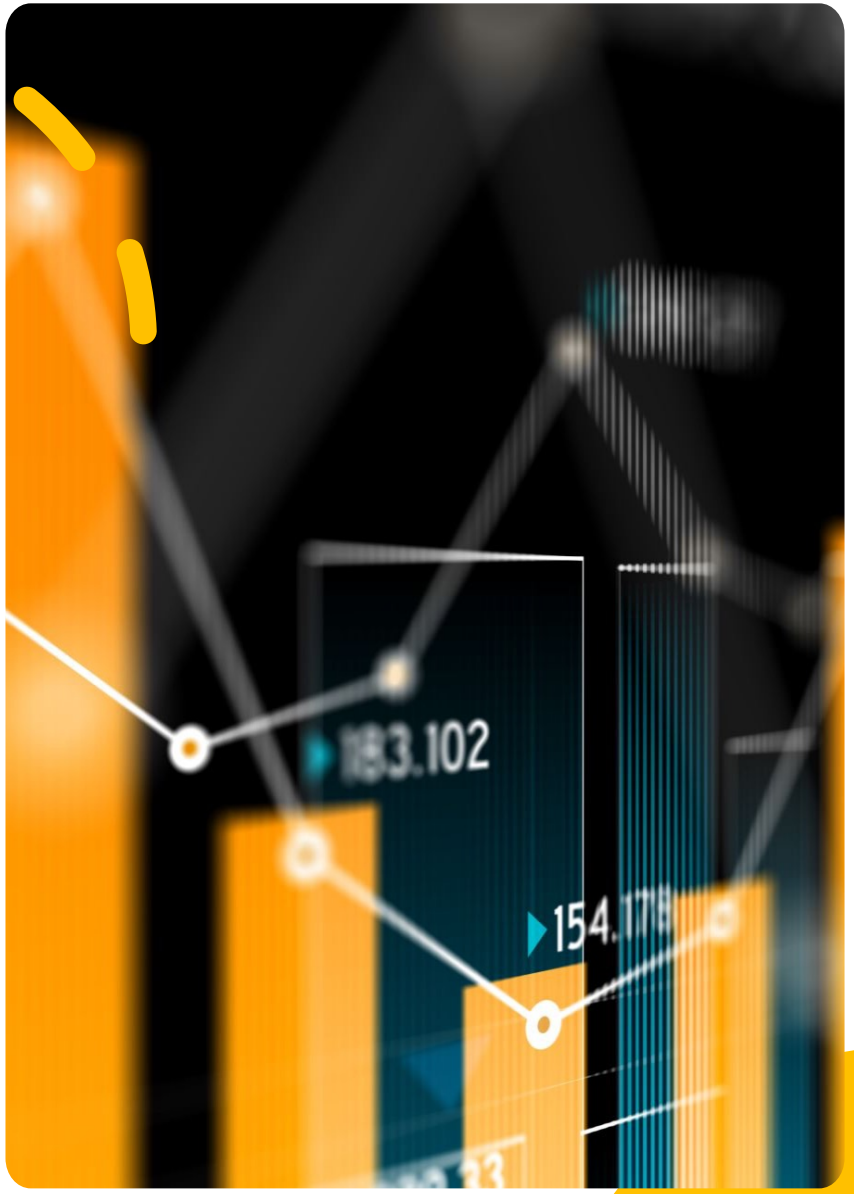
Methodology

Methodology

- **Executive Summary**
- Data collection methodology:
  - The secondary Data Collection method was used to gather the SpaceX launch data, gathered from an API as the data collection tool. The result will be viewed calling the .json() method, we worked on endpoint api.spacexdata.com/v4/launches/past. Using webscrapping, data source for Falcon 9 Launch was obtained scrapping related Wiki pages. Python BeautifulSoup package was used to web scrape some HTML tables that contain valuable Falcon 9 Launch data.
- Perform data wrangling
  - To gain insights, visualize and analyze the data, we transformed the raw data into clean datasets, data wrangling using an API was performed on the raw data. Attributes were reviewed, namely: Flight number, date, Booster Version, PayLoadMass, Orbit, LaunchSite, Outcome, GridFins, Launching Pads etc.

Methodology…(cont)

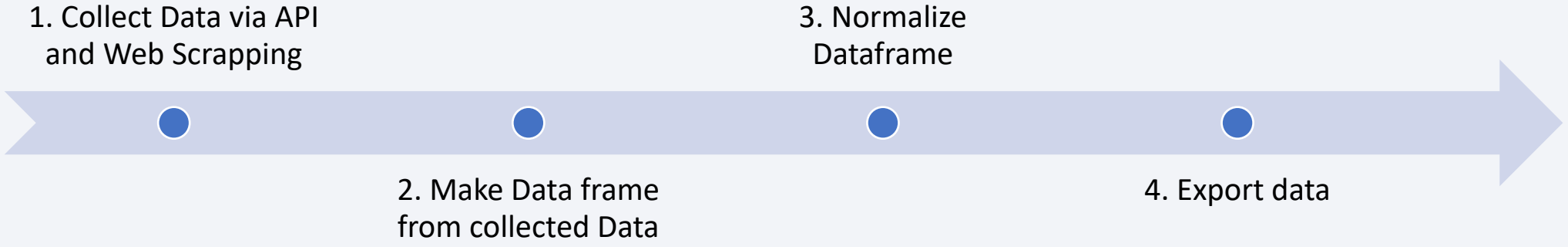
- ❑ Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and Bar graphs used to show data patterns
- ❑ Perform interactive visual analytics using Folium and Plotly Dash
  - Folium and Plotly Dash visualizations were used
- ❑ Perform predictive analysis using classification models
  - Various classifications used, SVM, Classification Trees, and Logistic Regression.



Data Collection

How data sets were collected.

Used URL to target a specific endpoint of the API, web scraping related wiki pages, converted collected data to Pandas dataframe for visualization and analysis. Normalized the data by dealing with null values and Falcon 1 data.



Data Collection – SpaceX API

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

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

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

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

Requesting rocket launch data from SpaceX API → Converting response to .json file → Get Booster versions → Assign Dictionary → Filter Data Frame

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

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

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None

GitHub URL

Data Collection - Scraping

Response from HTML

```
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
```

Creating Beautifulsoup Object

```
html_file = BeautifulSoup(data.text, "html.parser")

print(html_file.prettify())
```

Extract from HTML Tables

```
html_tables = html_file.find_all('table')
```

Getting column names

```
temp = html_file.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

Creating a dictionary and appending data to keys

Converting dictionary to dataframe

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

GitHub URL

Dataframe to .CSV

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Data was processed through Data Wrangling, which is a process of transforming and mapping data from raw data to a format that allows us to perform Analytics.

- Below we assign element 0 equals bad outcome, else it it a successful outcome.

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

1. Load data
2. Calculate number of launches on each site
3. Calculate occurrence of each orbit
4. Calculate mission outcome per each orbit type
5. Create outcome label column
5. Export to .csv

GitHub

Data Wrangling …(cont)

1

Calculate number of launches on each site

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

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

2

Calculate occurrence of each orbit

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

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
GEO        1
HEO        1
SO         1
ES-L1      1
Name: Orbit, dtype: int64
```

3

Calculate mission outcome per each orbit type

```
landing_outcomes = df['Outcome'].value_counts()
print("The number of Landing Outcomes : ")
landing_outcomes
```

```
The number of Landing Outcomes :

True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
None ASDS        2
False Ocean      2
False RTLS       1
Name: Outcome, dtype: int64
```

4

Create outcome label column

```
df['Class']=landing_class
df[['Class']].head(8)
```

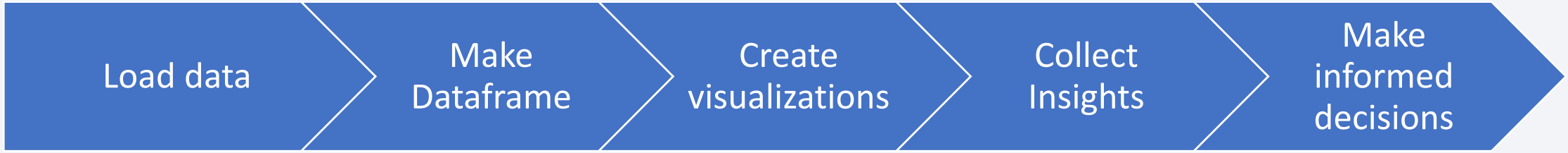
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA with Data Visualization

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, using data visualization methods and statistical graphics to show data.

GitHub URL

https://github.com/KBMoepye/Applied-Data-Science-Capstone/blob/Master/EDA%20using%20Pandas%20and%20Matplotlib.ipynb

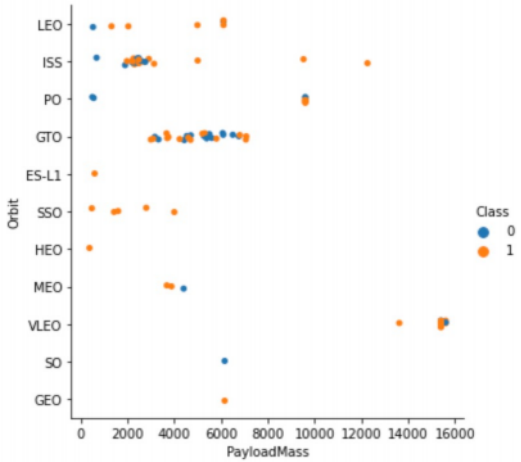


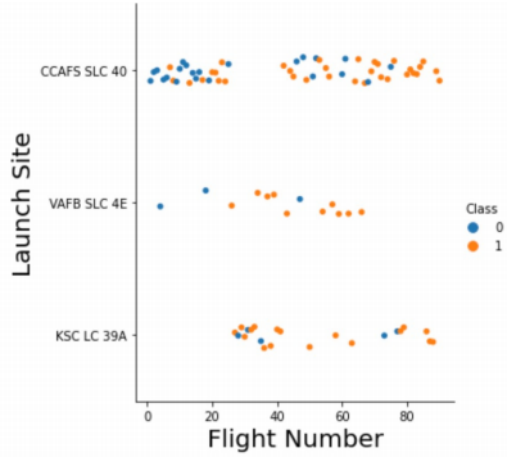
EDA with Data Visualization… (cont.)

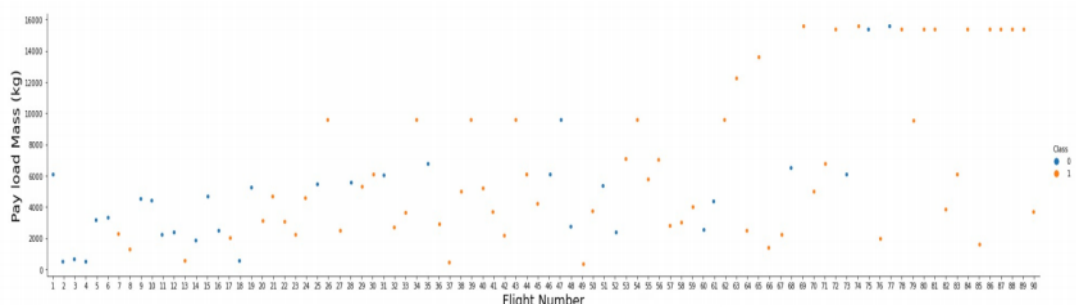
Scatter Graphs:

- 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, its very easy to predict which factors will lead to maximum probability of success in both outcome and landing.



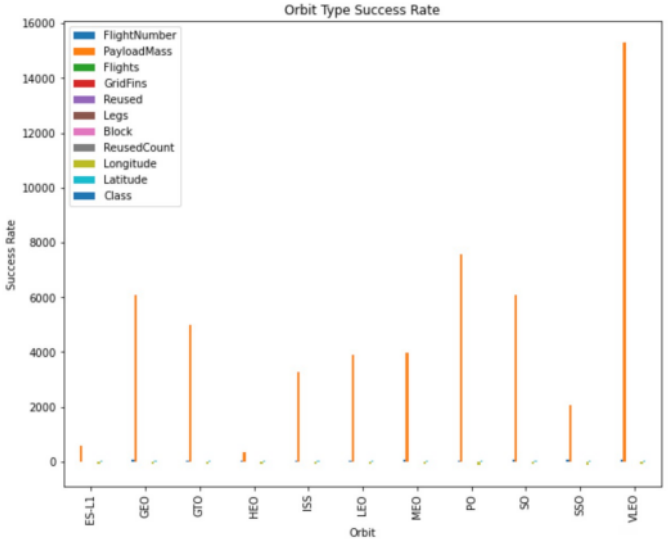




Bar Graph:

- Success Rate vs Orbit Type

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



Line Graph:

- Launch Success Yearly Trend

Line graphs are useful in that they show trends over time and can aide in future predictions.

[GitHub-link]

EDA with SQL

SQL queries you performed

- *Display the names of the unique launch sites in the space mission:*
  - **%sql** select distinct(LAUNCH_SITE) from SPACEXTBL
- *Display records where launch sites begin with the string 'CCA'*
  - **%sql** select * from SPACEXTBL where LAUNCH_SITE LIKE 'CCA%' limit 10
- Display the total payload Mass carried by boosters launched by NASA(CRS)
  - **%sql** select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = 'NASA (CRS)'
- *Display average payload mass carried by booster version F9 V1.1*
  - **%sql** select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
- *List date where group pad Launch was successful*
  - **%sql** select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
- *Listing booster versions where payload mass greater than 4000 but less than 6000*
  - **%sql** select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ > '4000' AND PAYLOAD_MASS__KG_ < '6000')
- *Listing total number of successful and failure mission outcomes*
  - **%sql** select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME LIKE ('Success%')

EDA with SQL…(cont.)

Listing names of booster versions that carried maximum payload mass

- **%sql** select BOOSTER_VERSION from SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

Listing the failed landing outcomes in drone ship, their versions and where date between 2015-01-01 and 2015-12-31

- **%sql** select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and (Date >= '2015-01-01' and Date <= '2015-12-31')

Ranking the count of landing outcomes, where landing outcome is 'Success (group pad)' between (Date >= '2010-06-04' and Date <= '2017-03-20')

- **%sql** select * from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)' and (Date >= '2010-06-04' and Date <= '2017-03-20') order by Date desc

GitHub URL

Build an Interactive Map with Folium

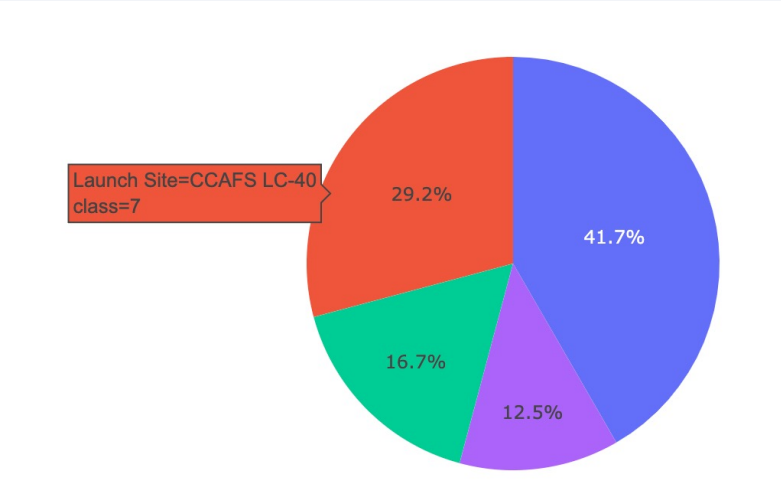
Folium Maps make it possible to visualize data on an interactive Map. Using Latitude and Longitude coordinates, we can label listed Launch sites using Circle Marker, mark distance from Launch sites to i.e Major roads etc. We would successfully visualize number of success and failure launches with different colour markers.

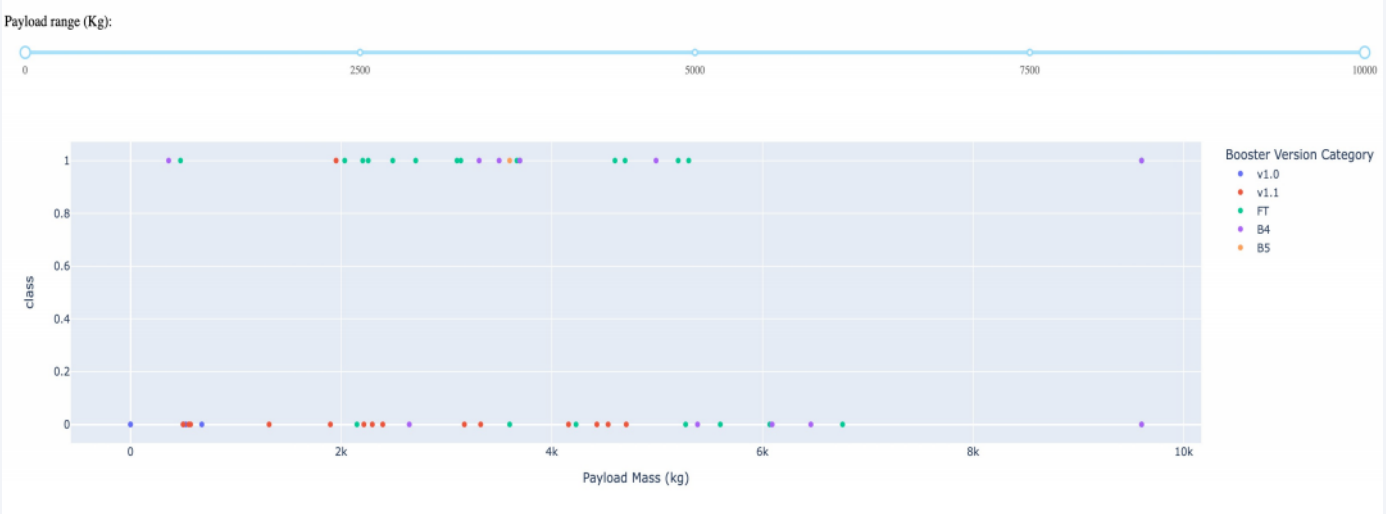
Map Objects	Code used	Result
Map Marker	folium.map.Marker()	Simple leaflet style location marker
Icon Marker	folium.Icon()	Convenience function to enable location marking
Circle Marker	folium.circle()	Leaflets Circle and CircleMarker, helps users interactively browsing the map
Polyline	folium.PolyLine()	Easily create line between two or multiple points.
Marker Cluster Object	MarkerCluster()	Good way to simplify a map containing many markers.
Marker Popups	Folium.Popup()	Location Marker with Popup

GitHub link:

Build a Dashboard with Plotly Dash

Pie Chart showing all four lunch sites and percentages, and a scatter plot showing correlation between Payload mass and Class.





Pie Chart: Shows percentage of success in relation to launch site

Scatter Plot Graph: It shows the relation between Success rate and Booster version category

GitHub URL

Predictive Analysis (Classification)

Building Model

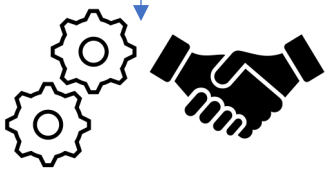
- Load feature engineered data into dataframe
- Transform feature into Numpy arrays
- Normalize and Transform data
- Split data into training and test data sets
- Check number of samples created
- Make a list of learning algorithms we will use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our model

Finding Best Performing Classification Model

- The best scoring model is selected as best performing.

Evaluate Model

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



GitHub URL

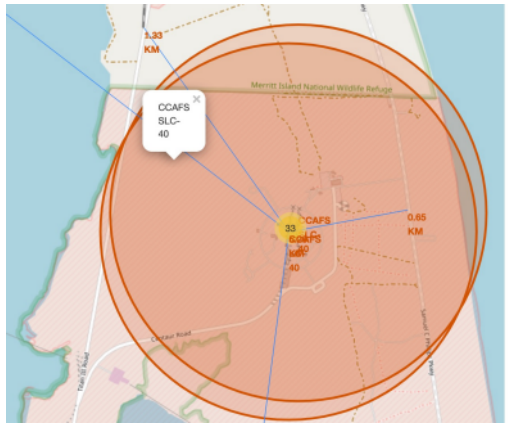
Results

EXPLORATORY DATA ANALYSIS RESULTS

Performed EDA using Pandas, Matplotlib and SQL, successfully performed EDA and data feature engineering, generated graphs and exported results to .csv file.

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

Performed analysis with folium, marked all sites on map, mark all success / failure launches, calculated distances between launch site to its proximities.

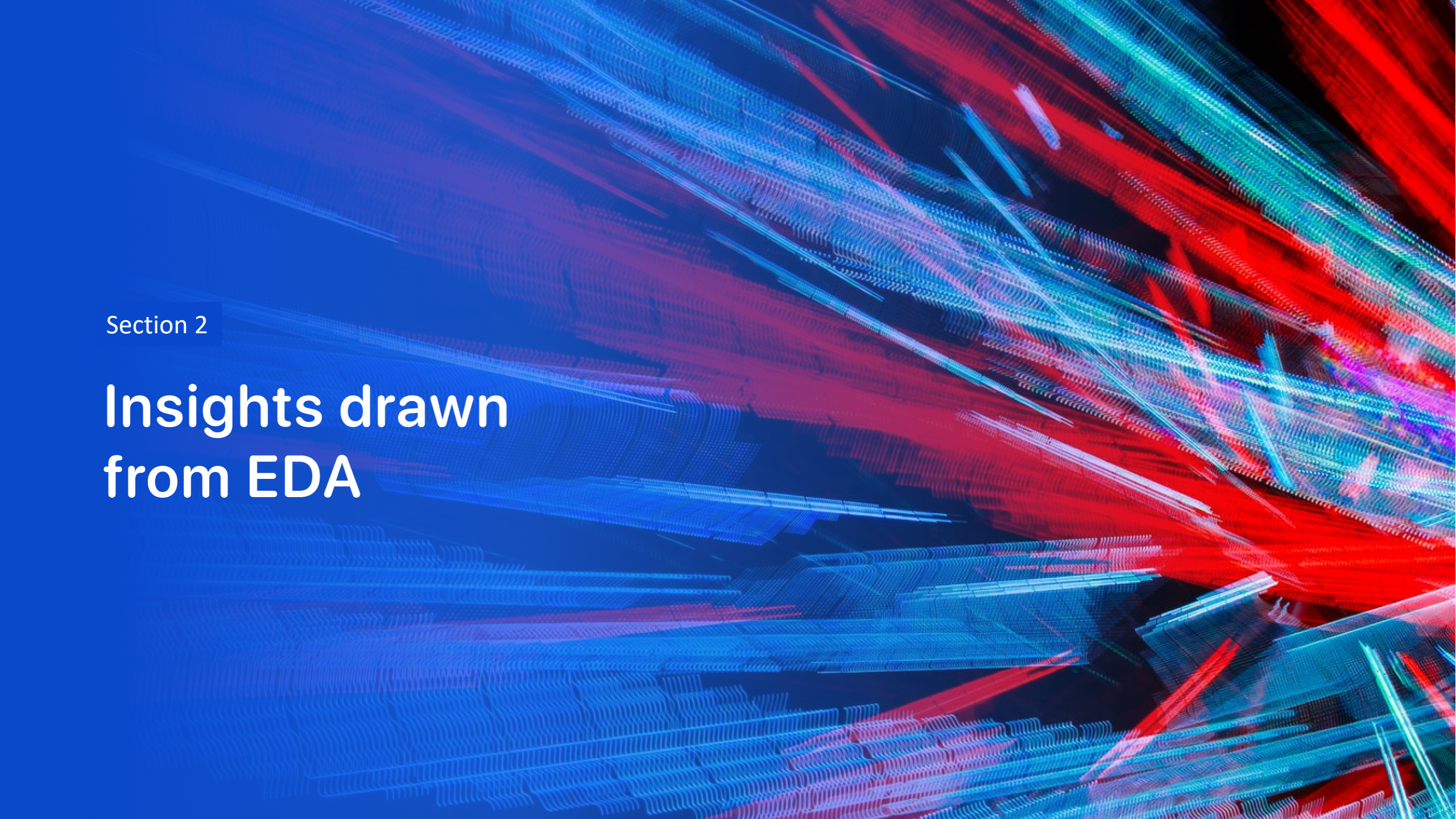


PREDICTIVE ANALYSIS RESULTS

Successfully created Class Column, standardized the data, split data into training and test data, found best performing method using test data

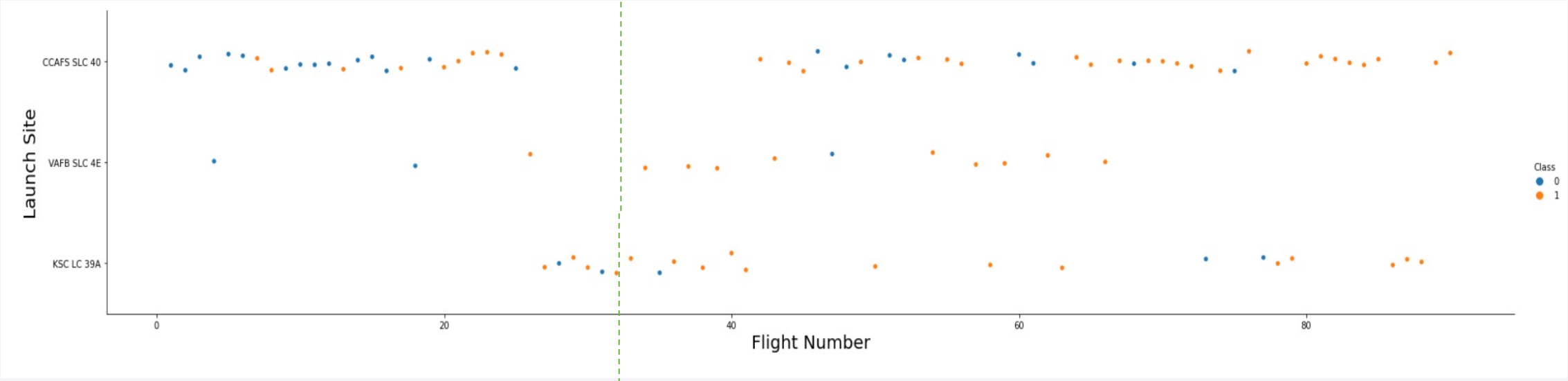
Decision tree classifier :

```
accuracy : 0.8785714285714284
```


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

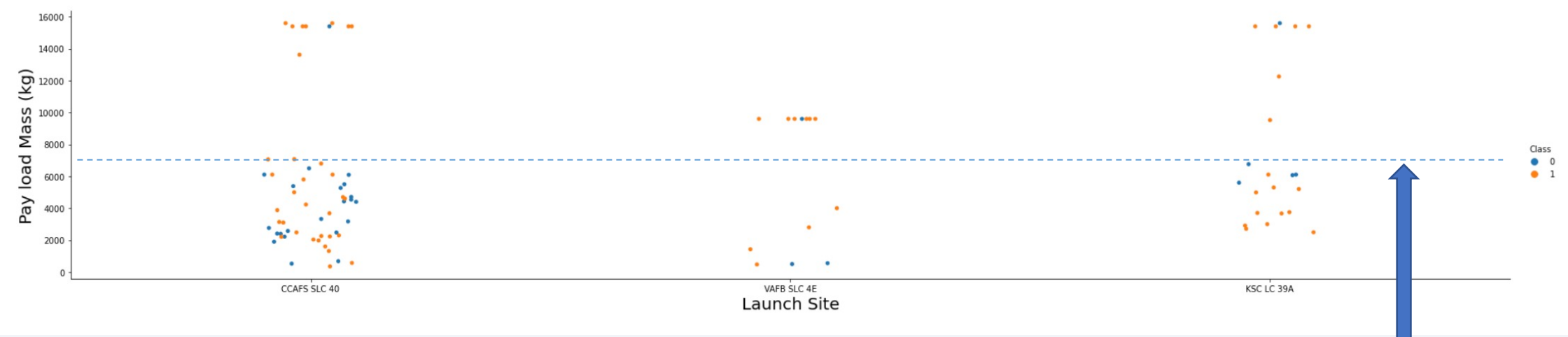


- Scatter plot with explanations

- The success rate is increasing with higher flight numbers, *there is higher success rate from about thirty (30)* across all launch sites

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



Scatter plot with explanations

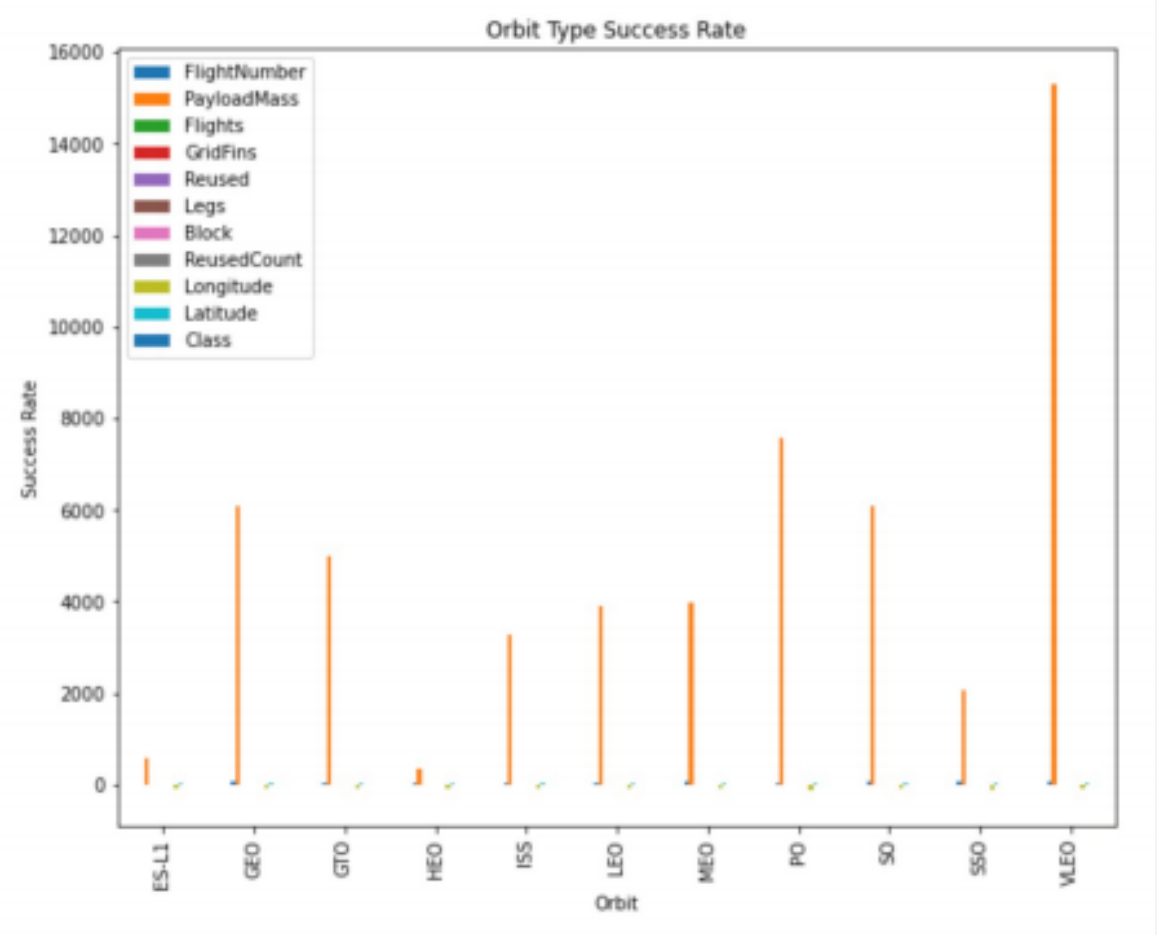
- Observing PayLoad vs Lanch Site scatter plot, there is higher success rate for rockets launched for heavy payload mass greater than 10000

Success Rate vs. Orbit Type

- GEO, PO, GTO and VLEO have a higher sucesss rate based on Payload Mass

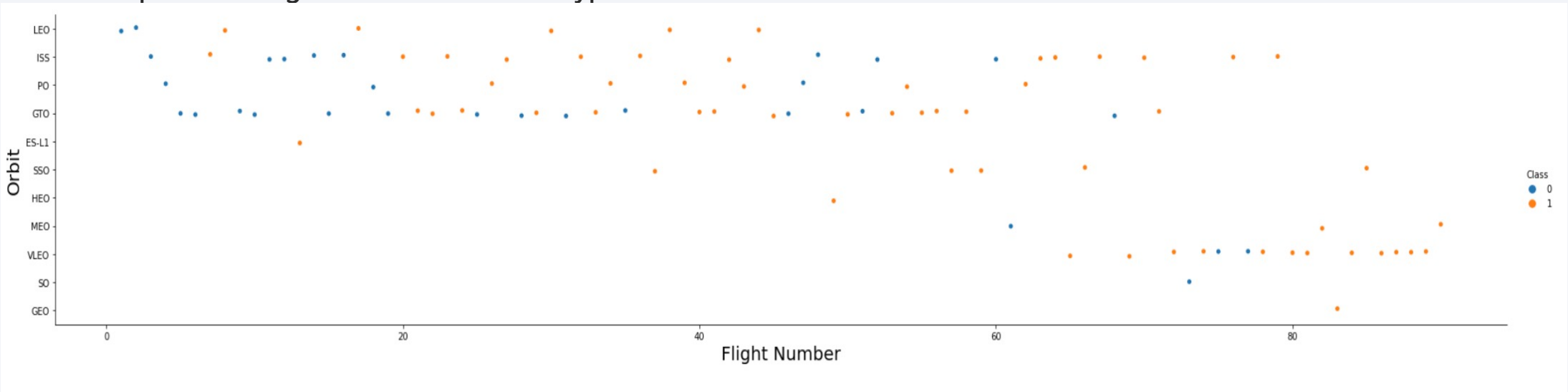
- Scatter plot with explanations

- Observations are, VLEO has a higher success rate with PayloadMass of over 15000



Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type



- Scatter plot with explanations

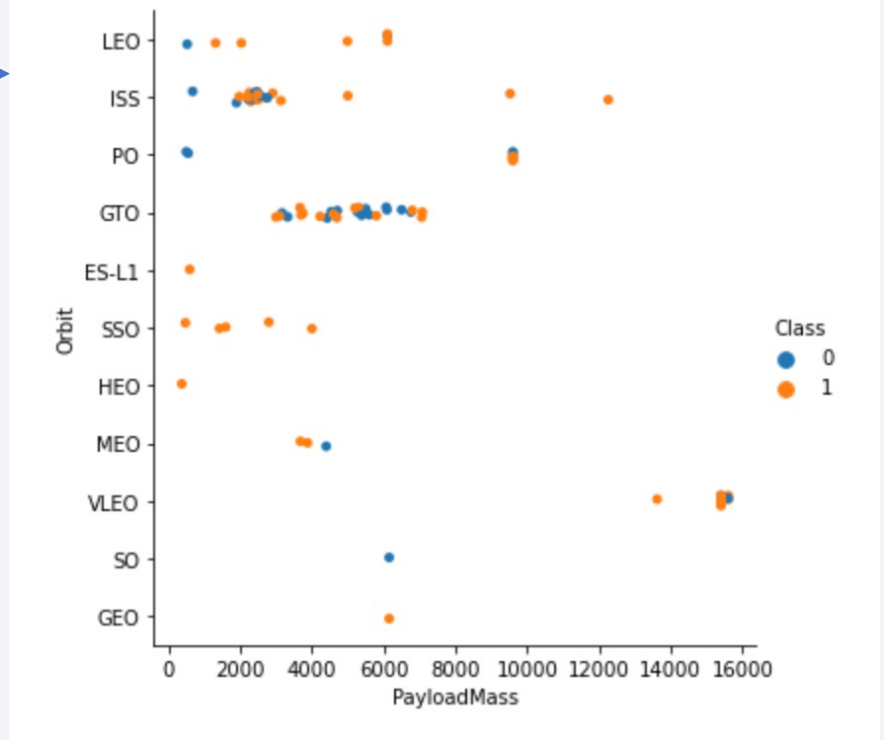
- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

- Scatter plot with explanations

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

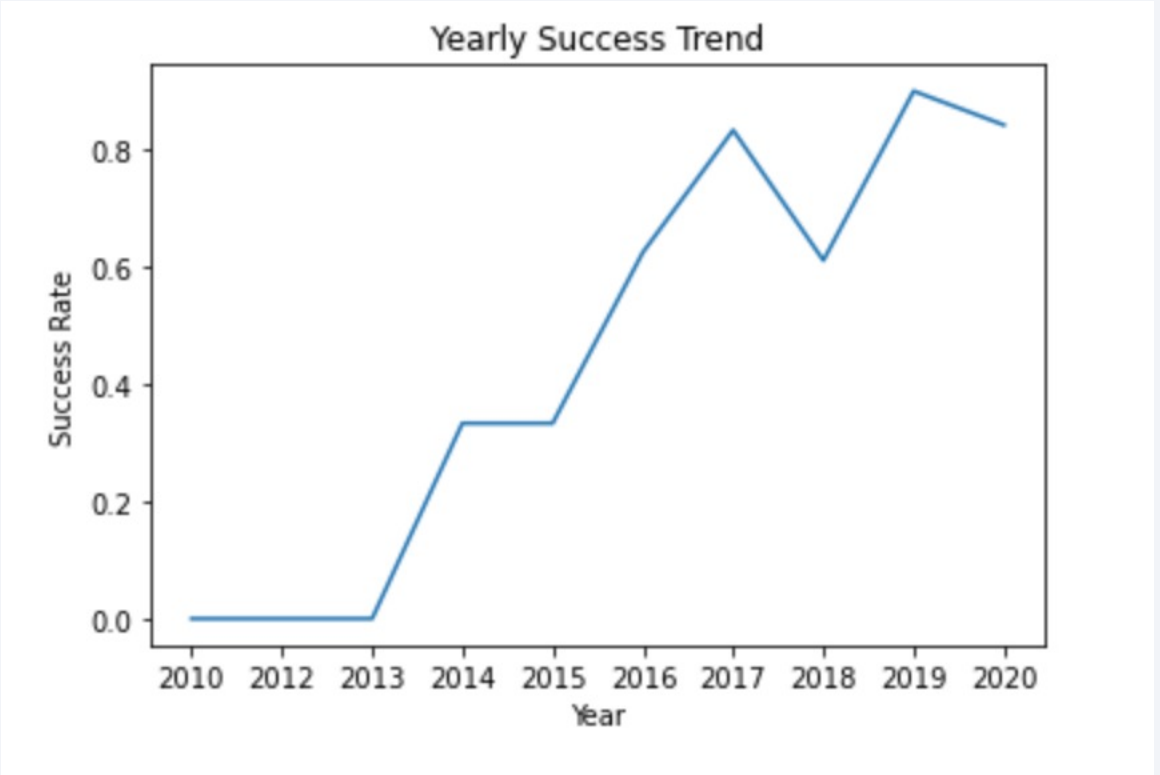


Launch Success Yearly Trend

- Line chart of yearly average success rate

Scatter plot with explanations

- You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- Find the names of the unique launch sites
  - CCAFS SLC-40
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
- Query unique launch sites in the space mission.
- **The following SQL statement was used:**
- result with a short explanation here
- Above are all the **%sql** select distinct(LAUNCH_SITE) from SPACEXTBL

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin 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 (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

- Present your query result with a short explanation here

The results above show different Payloads being used and landing outcomes.

I have used the below SQL statement to run the query.

- **%sql** select * from SPACEXTBL where LAUNCH_SITE LIKE 'CCA%' limit 5

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

The following Query was used to calculate the total payload:

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = 'NASA (CRS)'
```

- Present your query result with a short explanation here

- The result of the above query is as follows:

1
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The following query was used to calculate the average payload mass.

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

- Present your query result with a short explanation here

1
2928

First Successful Ground Landing Date

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

1
2015-12-22

- Present your query result with a short explanation here

- The below Query was used to get the first successful landing outcome.

```
%sql select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
```

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 list of names are:

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

- Present your query result with a short explanation here
- **The Query used is the following:**
- **%sql** select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ > '4000' AND PAYLOAD_MASS__KG_ < '6000')

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

- Query for Successful and Failure mission outcomes:

```
%sql select count(MISSION_OUTCOME) as MISSION_COUNT, MISSION_OUTCOME from SPACEXTBL GROUP BY MISSION_OUTCOME
```

- Present your query result with a short explanation here

- Result for mission outcomes:

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

Boosters Carried Maximum Payload

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

- Present your query result with a short explanation here

- The below Nested SQL Query was used to get the above result:

➢ **%sql** select BOOSTER_VERSION from SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

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

2015 Launch Records

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

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

- Result with a short explanation here

The SQL query that was used is the following:

➢ **%sql** select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and (Date >= '2015-01-01' and Date <= '2015-12-31')

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

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

- SQL Query

➢ %sql select Count(LANDING__OUTCOME) as Landing_outcome_count, LANDING__OUTCOME from SPACEXTBL where (Date >= '2010-06-04' and Date <= '2017-03-20') AND (LANDING__OUTCOME LIKE('Failure (d%)') or LANDING__OUTCOME LIKE('Success(g%)')) GROUP BY LANDING__OUTCOME

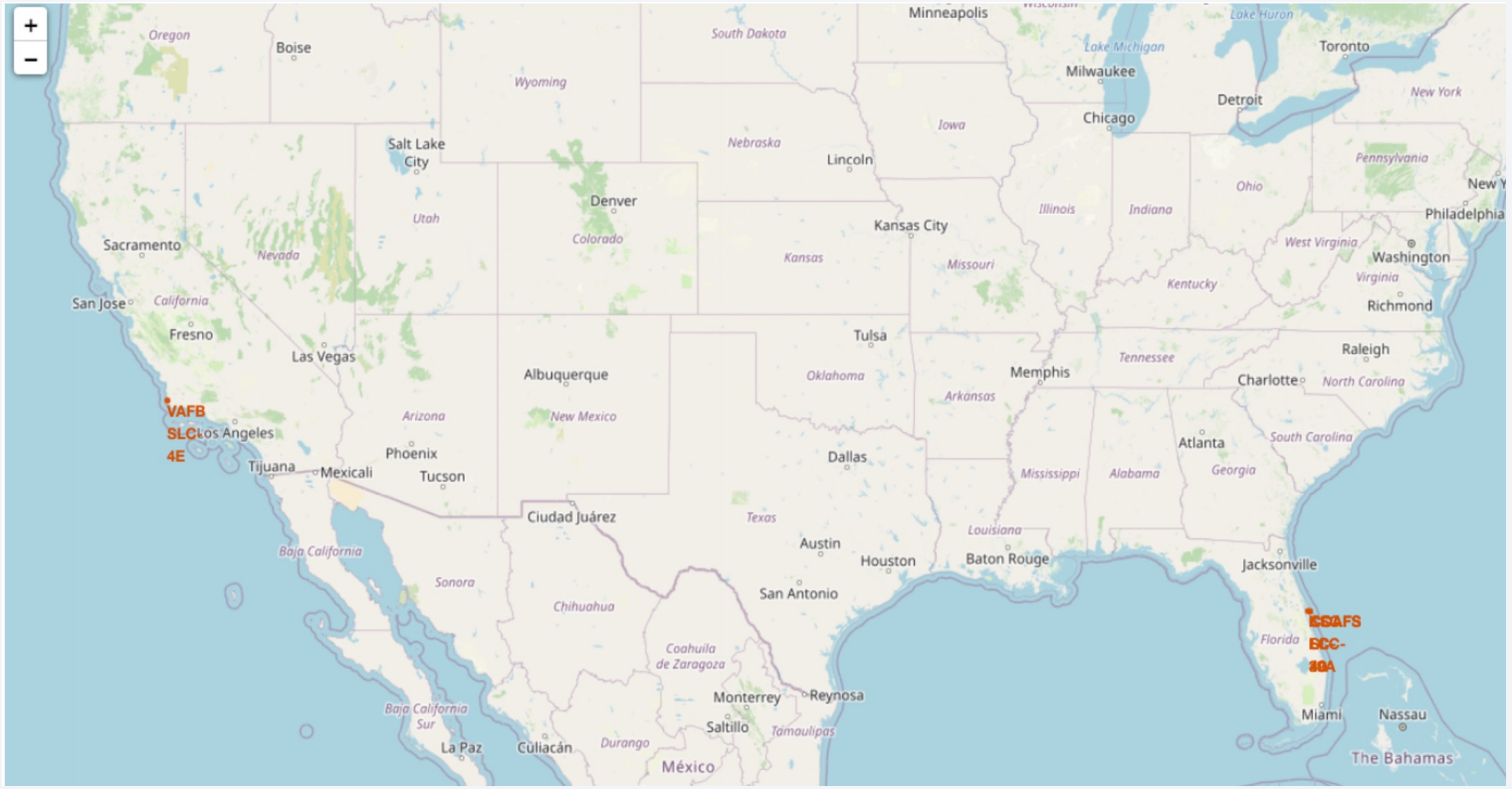
- Result with a short explanation here

landing_outcome_count	landing__outcome
5	Failure (drone ship)


Section 3
Launch Sites
Proximities Analysis

Launch sites Marked on map

- Generated folium map screenshot



- Explain the important elements and findings on the screenshot

Launch sites are in Proximity to the Coast

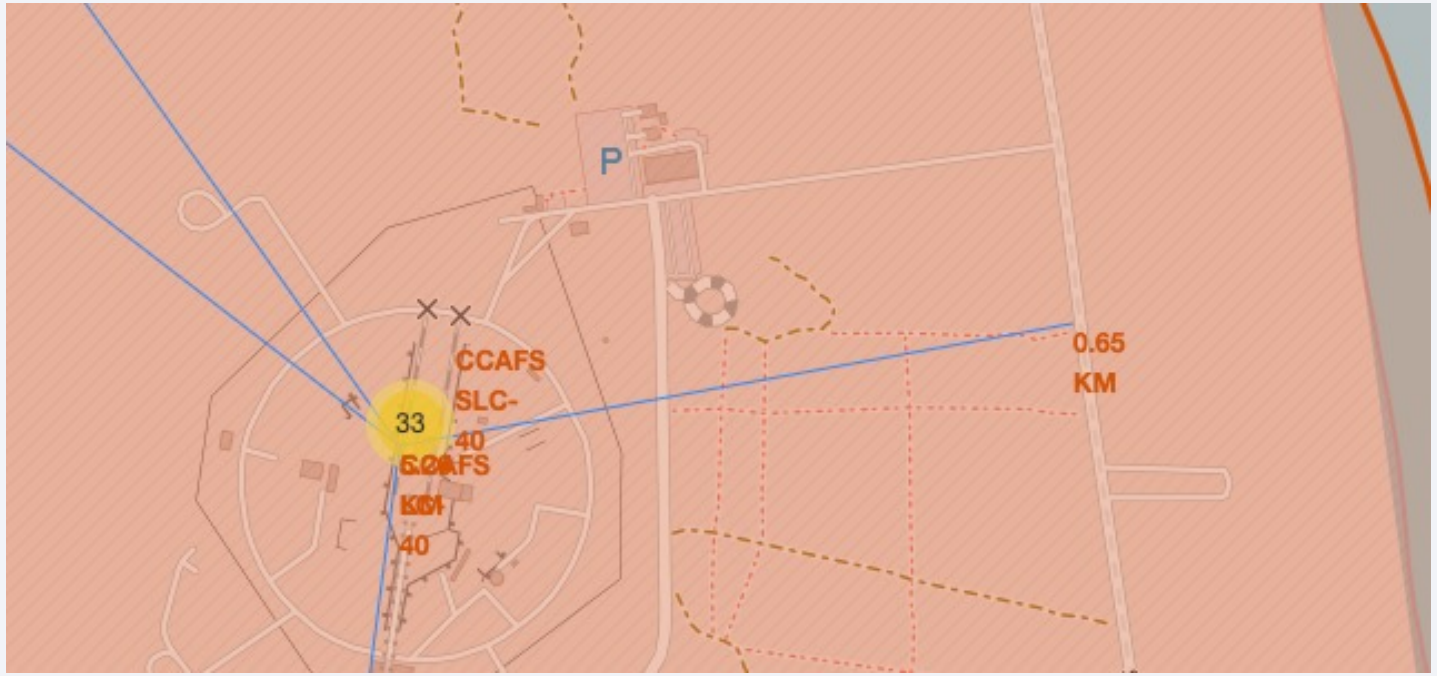
Success/Failed Launches Marked

- Success/Failed Launches Marked on each site

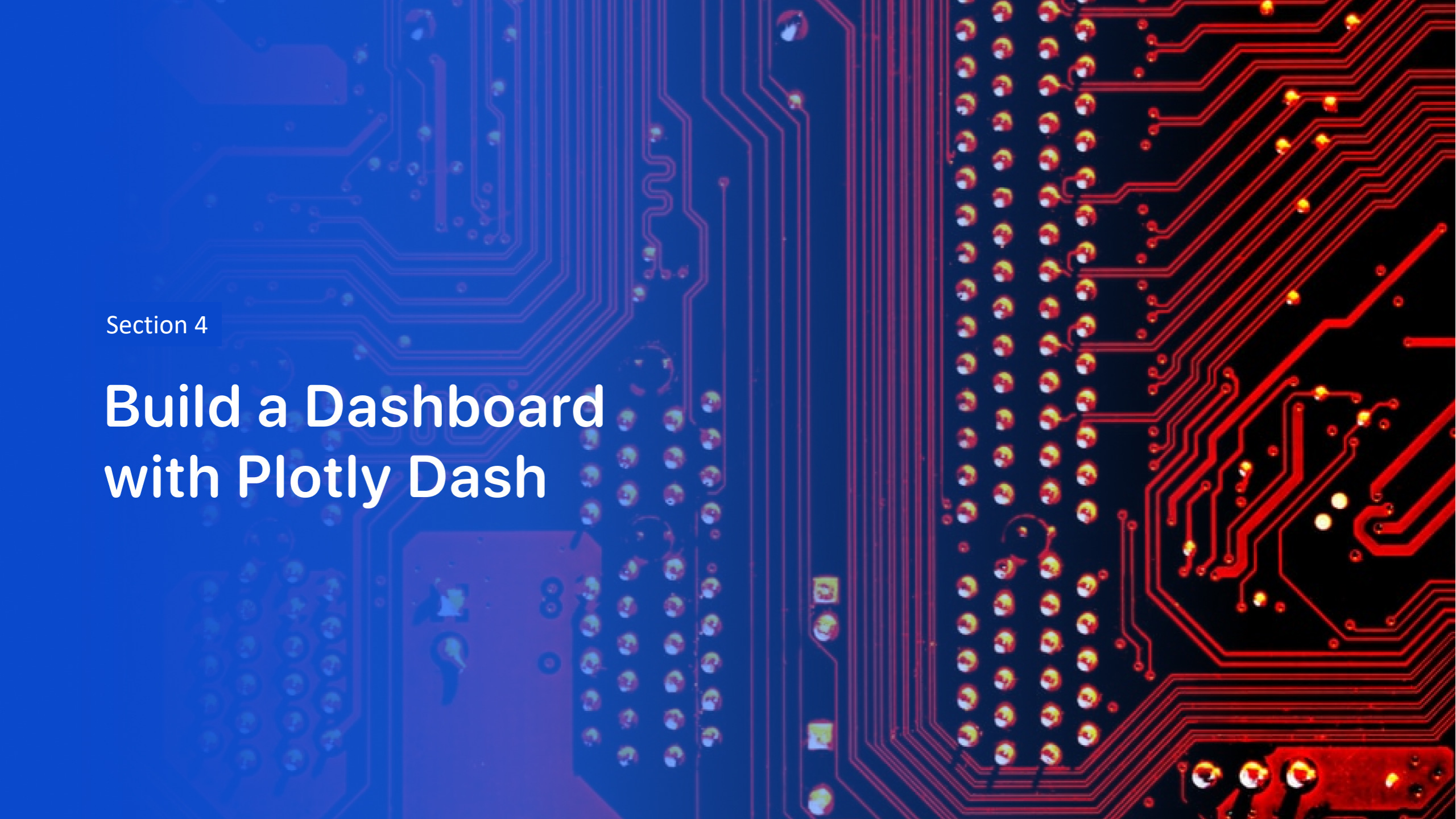
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot:
  - Added launch outcomes for each site, we want to see which site has high success rate. From the colour-labeled markers, when can identify sites with high success rate, in this case Launch site **CCAFS SLC-40** has high success rate.

Distance between Launch sites

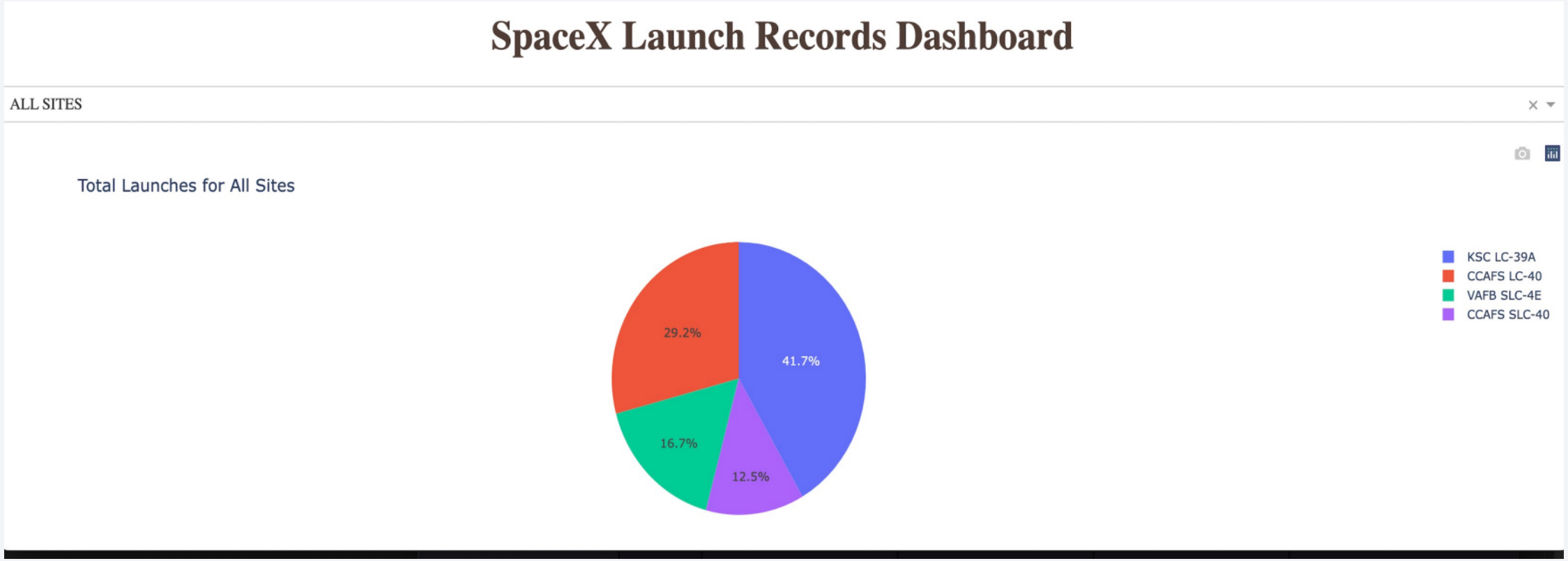
Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



- Explain the important elements and findings on the screenshot
- The Map shows Launch Site CCAFS SLC-40 being 0.65KM from the major highway


Section 4
Build a Dashboard with Plotly Dash

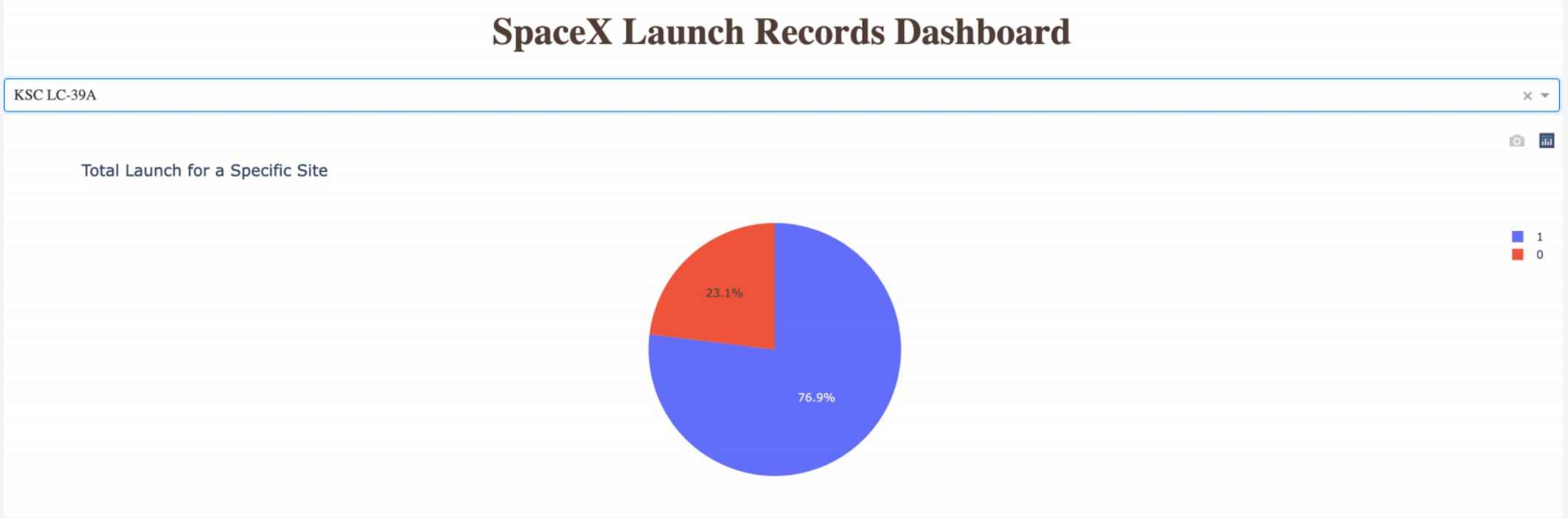
Total Launches for All Sites



Based on the above results, launch site KSC LC-39A has the most successful launches.

Highest Launch Success

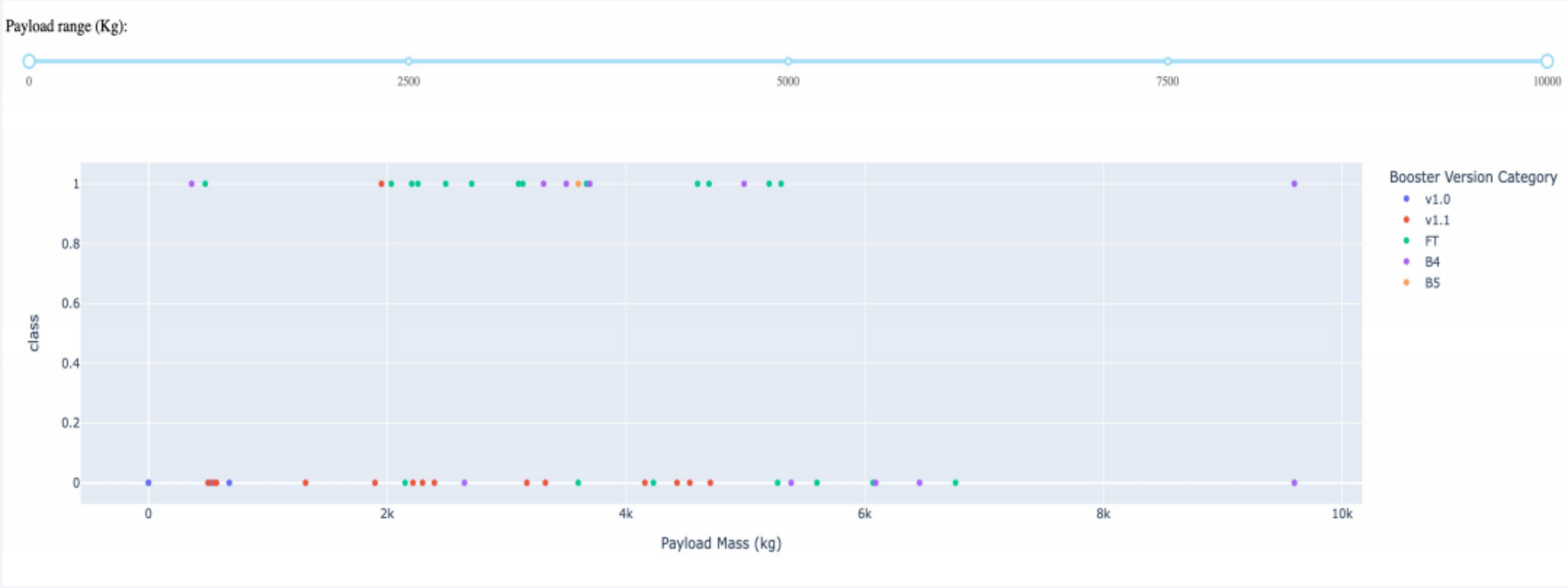
- Pie chart for the launch site with highest launch success ratio



Based on the above results, KSC LC-39A achieved 76.9% success rate with a 23.1% failure rate.

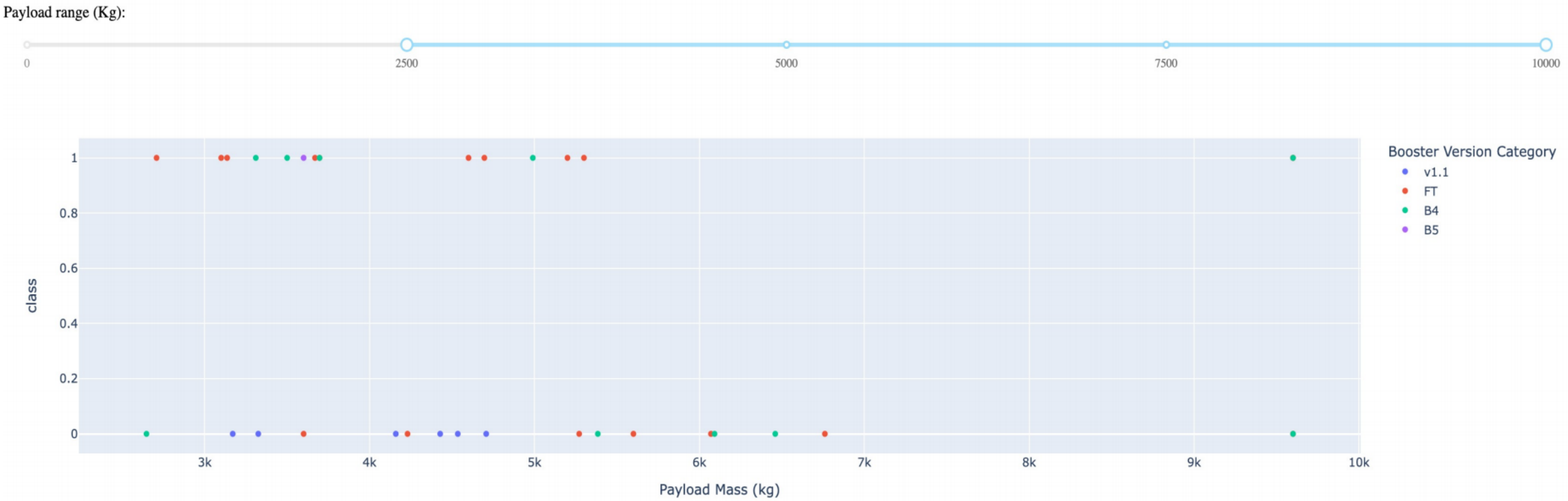
Payload vs Launch Outcome Scatter Plot – All sites

Payload range(kg) @ 0



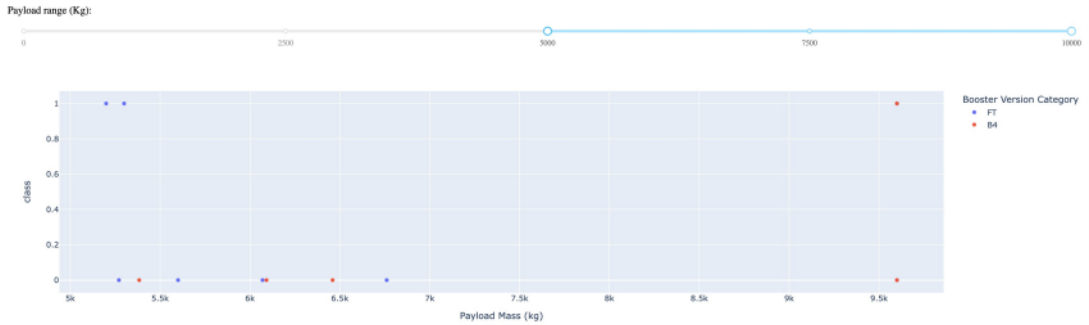
Based on the above scatter plot, there is more success rate on the Payload mass < 4K compared to Payload Mass > 4K.

Payload range(kg) @ 2500

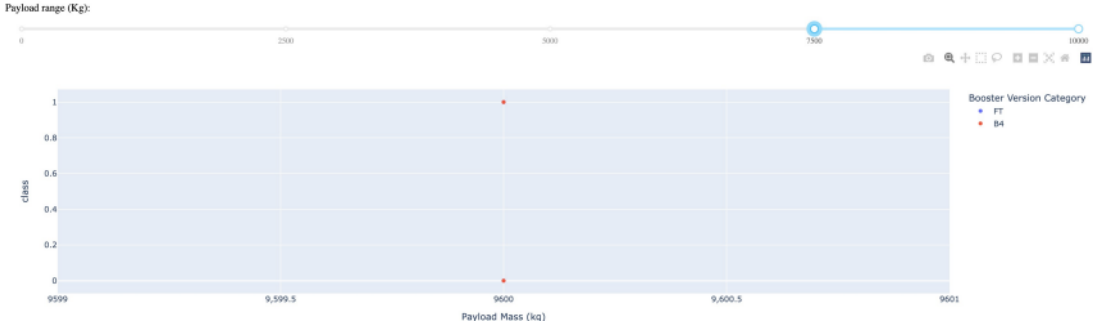


Payload vs Launch outcome…(cont.)

Payload range(kg) @ 5000

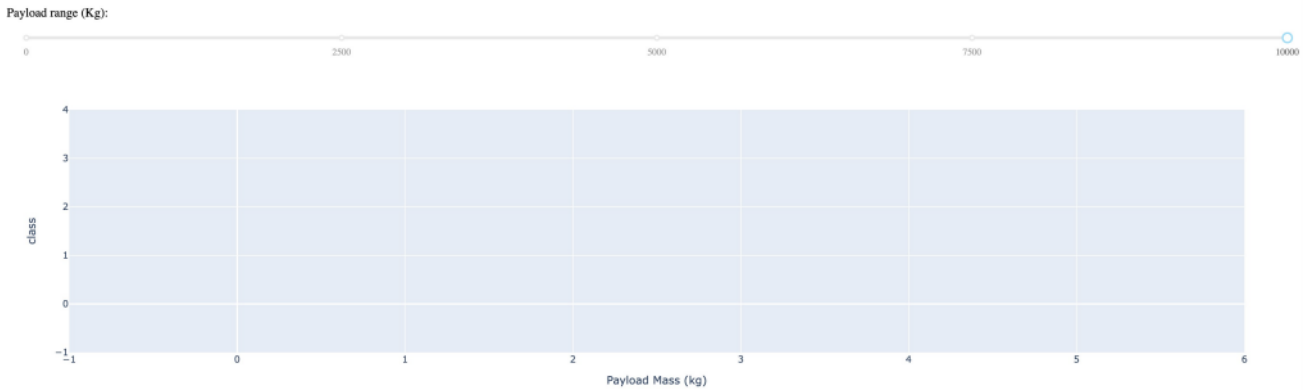


Payload range(kg) @ 7500



Payload vs Launch outcome…(cont.)

Payload range(kg) @ 10000



Section 5

Predictive Analysis (Classification)

Classification Accuracy

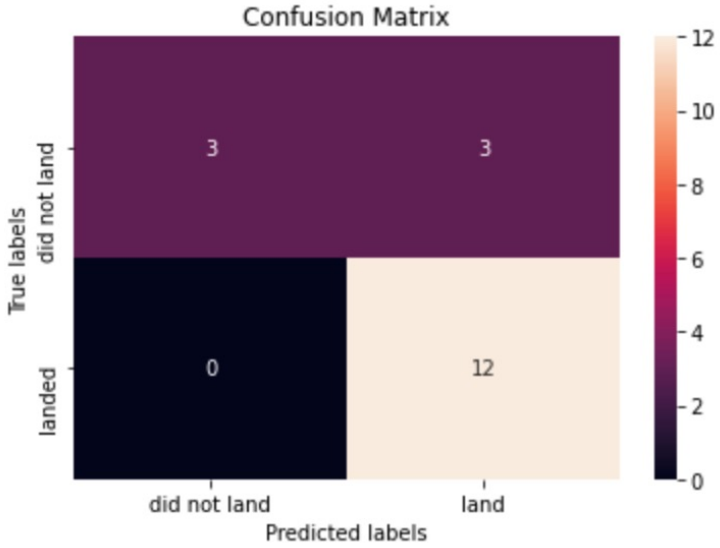
- Decision Tree with an Accuracy score of 0.878571428571 has the best performing score. The other three models are performing close to one another.

Algorithm	Accuracy	Accuracy on Test Data	Tuned Hyperparameters
Decision Tree	0.878571428571	0.83333333333334	{'criterion': 'gini', 'max_depth': 18, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
SVM	0.84821428571	0.83333333333334	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
KNN	0.848214285714	0.83333333333334	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
Logistic Regression	0.846428571	0.83333333333334	{‘c’:0.01, ‘penalty’:’12’, ‘solver’: ‘lbfgs’}

Confusion Matrix

- Decision Tree
- SVM
- KNN
- Logistic Regression

❑ All All models generated an identical confusion matrix



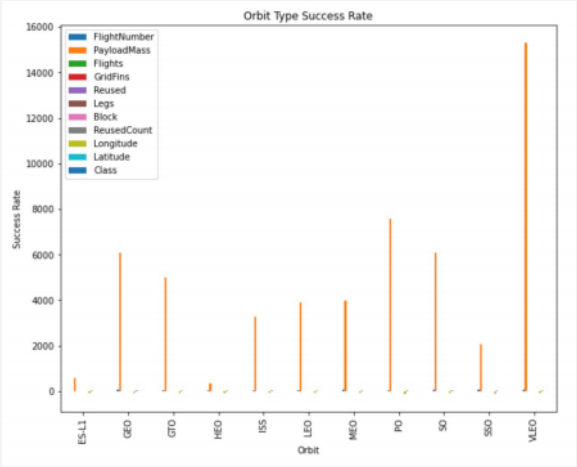
Predicted Values Matrix

	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

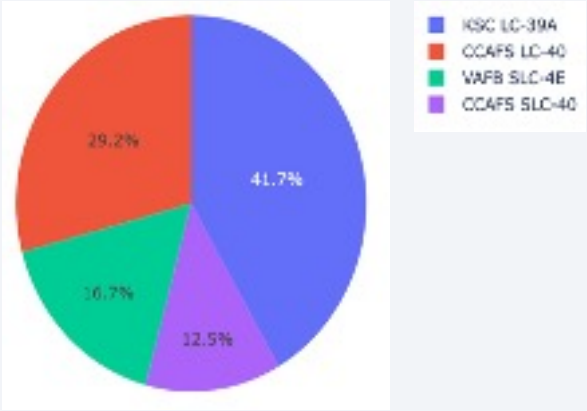
Label	Formula
Accuracy	(TP+TN)/Total =(12+3)/18=0.833333
Misclassification Rate	(FP+FN)/Total=(3+0)/18 = 0.1667
True Positive Rate	TP/Actual Yes = 12/12 =1
False Positive Rate	FP/Actual No = 3/6 = 0.5
True Negative Rate	TP/Predicted Yes = 12/15 = 0.8
Precision	Actual Yes/Total = 12/18 = 0.6667

Conclusions

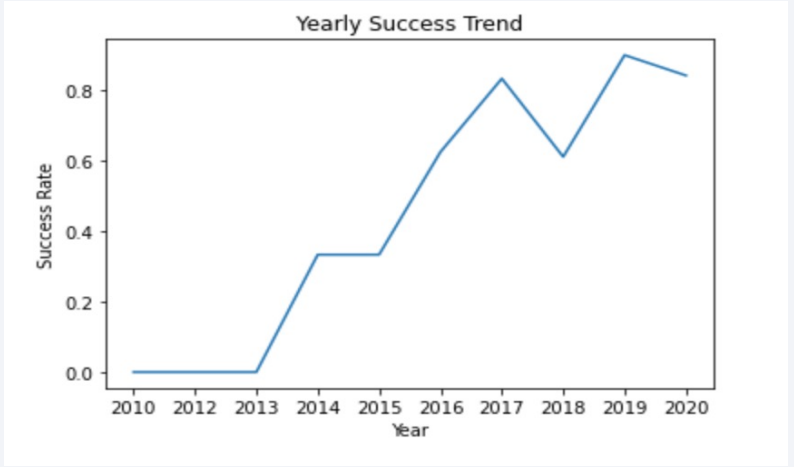
- Orbits type GEO, PO, GTO and VLEO have highest success rates



- KSC LC-39A has had the most successful launches



The success rate of launches have been increasing since 2013, with 2019 being the highest



Decision Tree Classifier algorithm is the best model provided dataset

Algorithm	Accuracy
Decision Tree	0.878571428571
SVM	0.84821428571
KNN	0.848214285714
Logistic Regression	0.846428571

Appendix

- All related SQL, Pandas, Snippets and Visualizations are provided for in GitHub links.

Thank you!