IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rajit Sanghvi
19.10.2021

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection
  - Data wangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics results
  - Predictive analysis results (results from the best machine learning model)

Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- We will predict if the Falcon 9 first stage will land successfully or not using machine learning techniques.

Section 1

Methodology

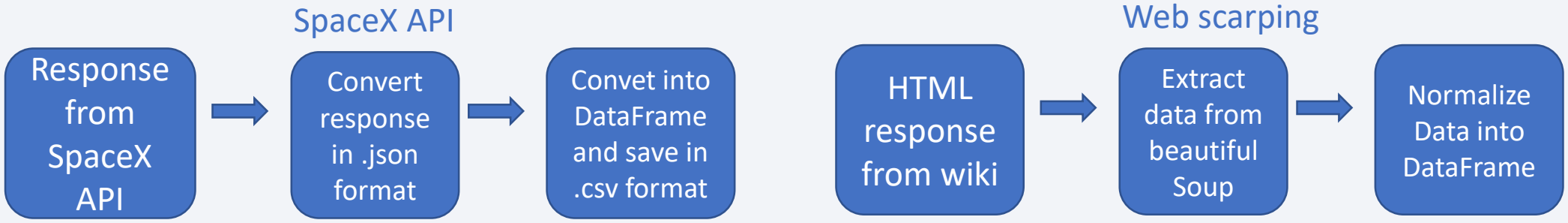
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API.
  - Web scraping from Wikipedia
- Perform data wrangling
  - One hot encoding was used to convert the categorical values to numerical.
  - Outcomes were converted into Training labels with 1 means the booster successfully landed and 0 means it was unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data were collected using two methods.
  - SpaceX REST API
  - Web scarping from Wikipedia page
- SpaceX REST API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- A get request was made to the SpaceX API using SpaceX url `https://api.spacexdata.com/v4/launches/past`
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using Beautiful Soup.



Data Collection – SpaceX API

1 .Getting Response from API

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

response = requests.get(spacex_url)
```

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

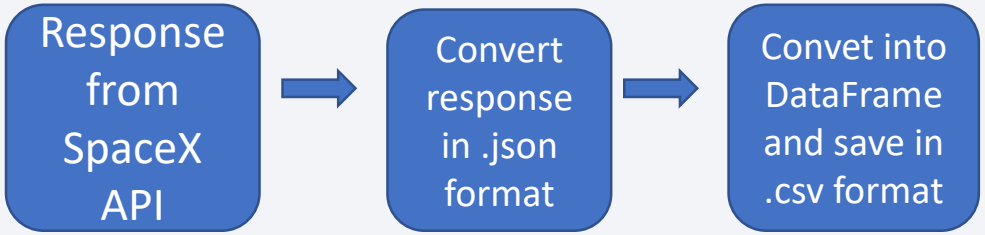
```
getBoosterVersion(data)
getPayloadData(data)
getLaunchSite(data)
getCoreData(data)
```

4. Assign list to dictionary then dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

SpaceX API



GitHub URL to Notebook

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df[df['BoosterVersion'] != 'Falcon 1']
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

1 .Getting Response from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36'}
data = requests.get(static_url, headers = headers, timeout=5).text
```

2. Creating BeautifulSoup Object

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

3. Extracting all the tables

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

4. Create a data frame by parsing the launch HTML tables

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

5. Dataframe to .CSV

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

Web scarping

GitHub URL to Notebook

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

GitHub URL to Notebook

EDA with Data Visualization

- **Scatter charts:** Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation.
  - Flight Number VS. Payload Mass
  - Flight Number VS. Launch Site
  - Payload VS. Launch Site
  - Flight Number VS Orbit type
  - Payload VS. Orbit Type

- **Bar Graph**: A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes.
  - Class Mean VS. Orbit

- **Line Graph:** Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded
  - Year VS Success Rate

GitHub URL to Notebook

EDA with SQL

Performed SQL queries to gather information about the dataset.

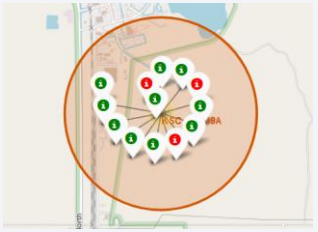
Following questions were answered about the dataset using SQL.

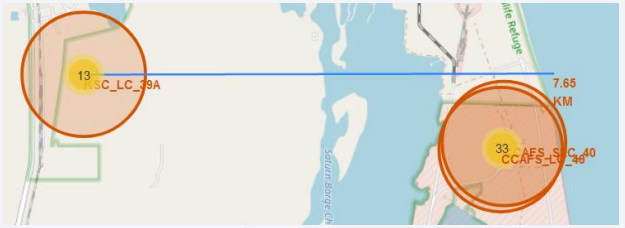
- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

GitHub URL to Notebook

Build an Interactive Map with Folium

- **To visualize the Launch Data into an interactive map.** We took the Latitude and Longitude Coordinates at each launch site and added a *Circle Marker around each launch site with a label of the name of the launch site.*
- We assigned the dataframe launch_outcomes (failures, successes) to *classes 0 and 1* with Green and Red markers on the map in a MarkerCluster()
- **Using Haversine's formula we calculated the distance** from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns.
- **PolyLines** are drawn on the map to measure distance to landmarks
- **All launch sites are located close to the Equator.** This is because rockets launched from sites near the Equator get an additional natural boost that helps save the cost of putting in extra fuel and boosters.
- **All launch sites are also located close to the coast**, because if there is an issue with the rocket after lift off, space flight operators can safely put it down in the Atlantic Ocean without endangering the public.



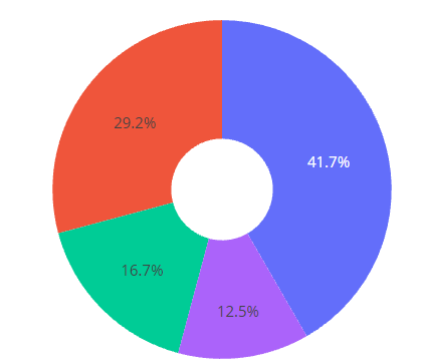


1. All launch sites are close proxmity to railways.
2. All launch sites are close proxmity to highways.
3. All launch sites are close proxmity to coastlines.
4. All launch sites are away from cities.

GitHub URL to Notebook

Build a Dashboard with Plotly Dash

- **The dashboard is built with Plotly Dash.**
- A **Pie chart** showing total successful launches for different launch sites and the **Scatter plot** showing the relationship with Payload Mass (Kg) and Outcome for the different launch sites for different booster versions were plotted on a Dashboard.
- Following question were answered using the interactive Pie chart:
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest



[GitHub URL to Notebook]

Predictive Analysis (Classification)

- **BUILDING MODEL**
  - Load our dataset into NumPy and Pandas
  - Transform Data
  - Split our data into training and test data sets
  - Check how many test samples we have
  - Decide which type of machine learning algorithms we want to use
  - Set our parameters and algorithms to GridSearchCV
  - Fit our datasets into the GridSearchCV objects and train our dataset.
- **EVALUATING MODEL**
  - Check accuracy for each model
  - Get tuned hyper-parameters for each type of algorithms
  - Plot Confusion Matrix
- **IMPROVING MODEL**
  - Feature Engineering
  - Algorithm Tuning
- **FINDING THE BEST PERFORMING CLASSIFICATION MODEL**
  - The model with the best accuracy score wins the best performing model
  - In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

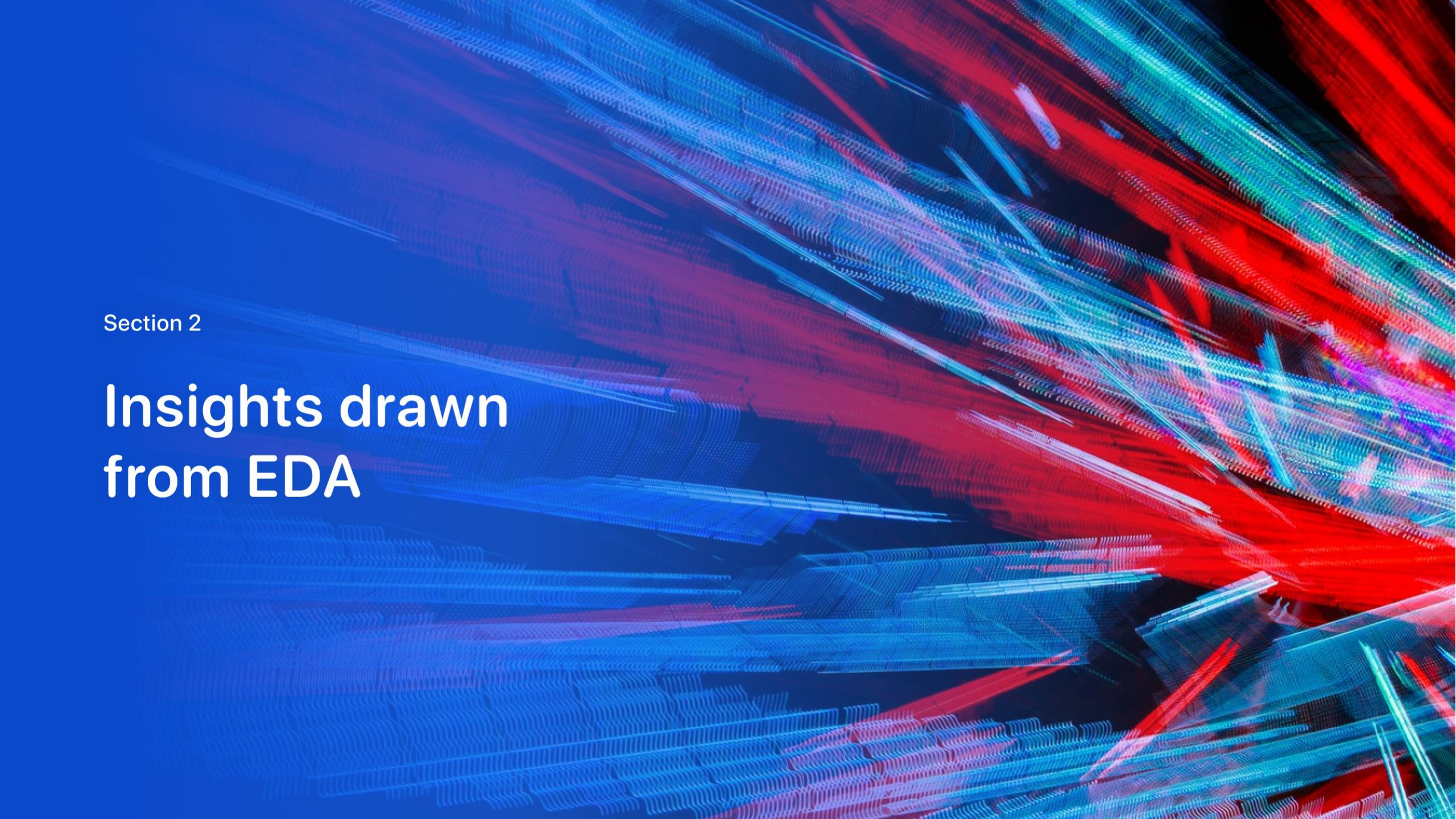
GitHub URL to Notebook

Results

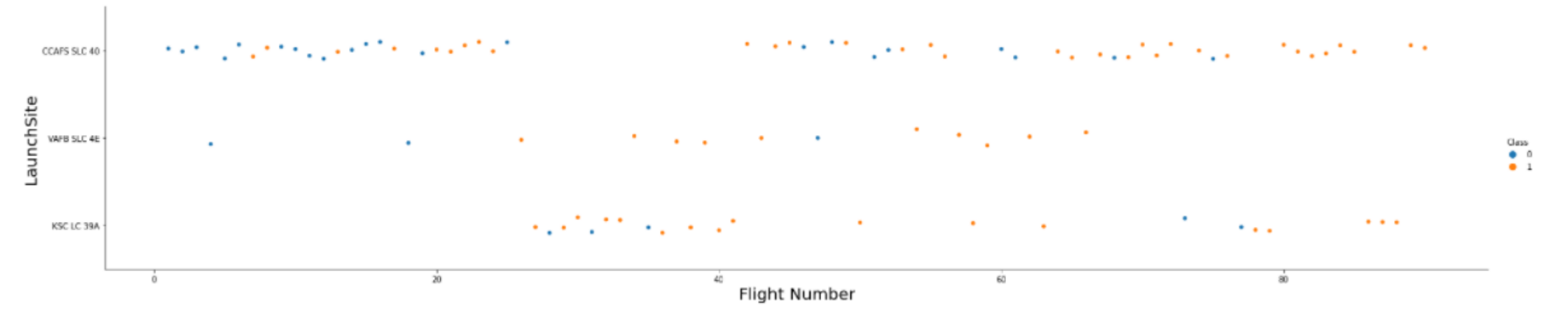
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

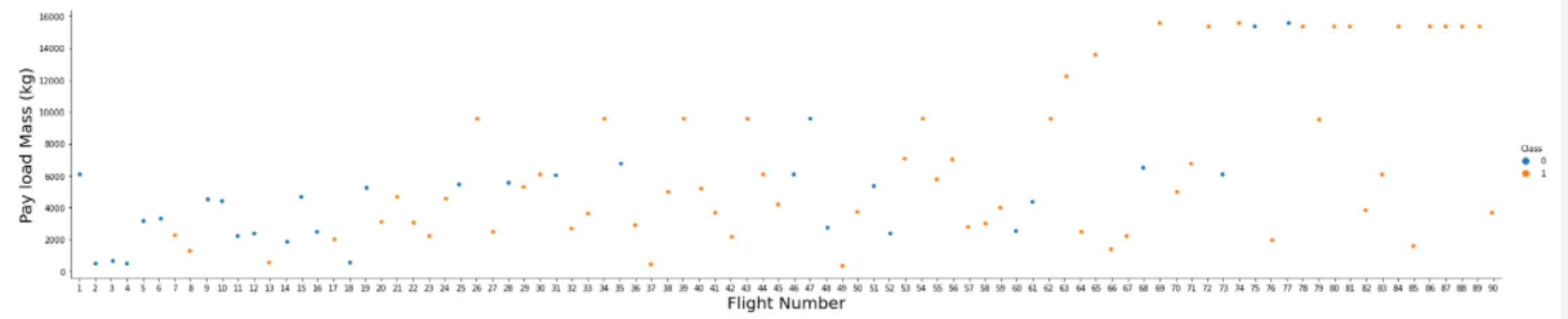


Flight Number vs. Launch Site



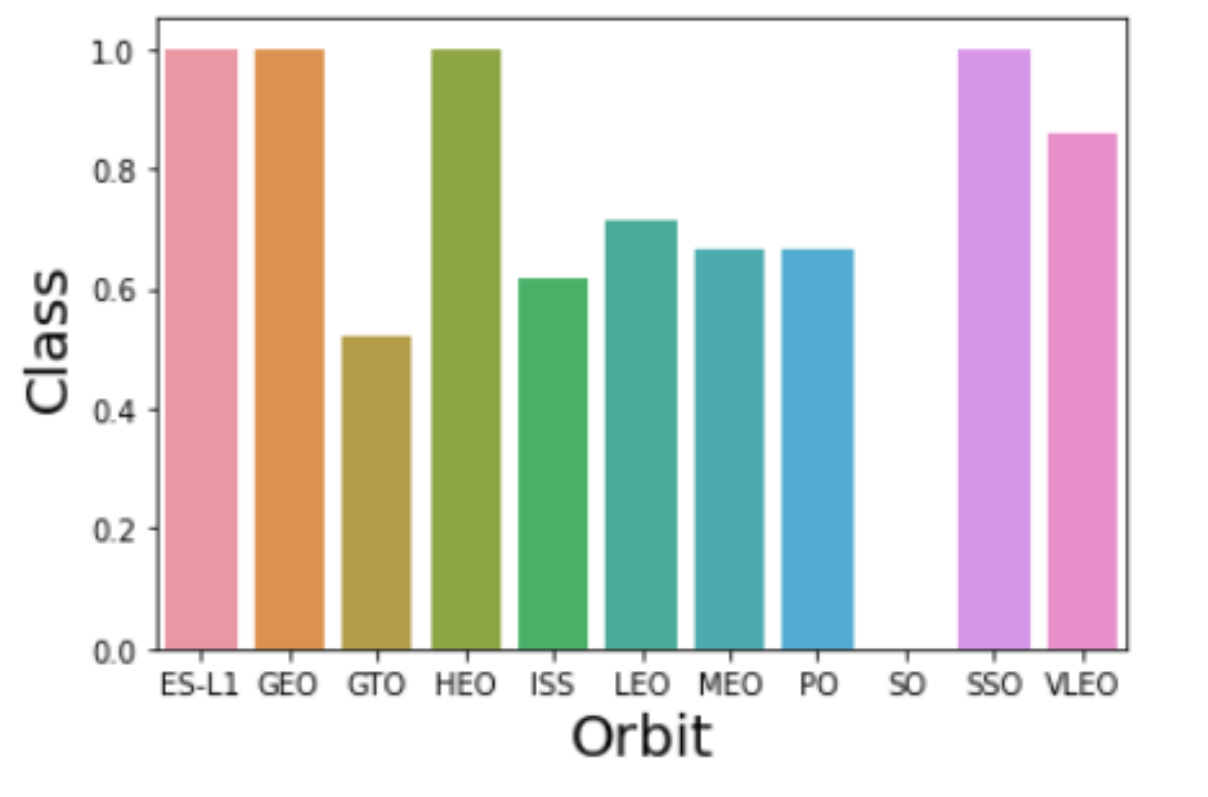
We see that as the flight number increases, the first stage is more likely to land successfully. The LaunchSite is also important; it seems the CCAFS SLC 40 launchsite has higher failure compare to other launchsites

Payload vs. Launch Site



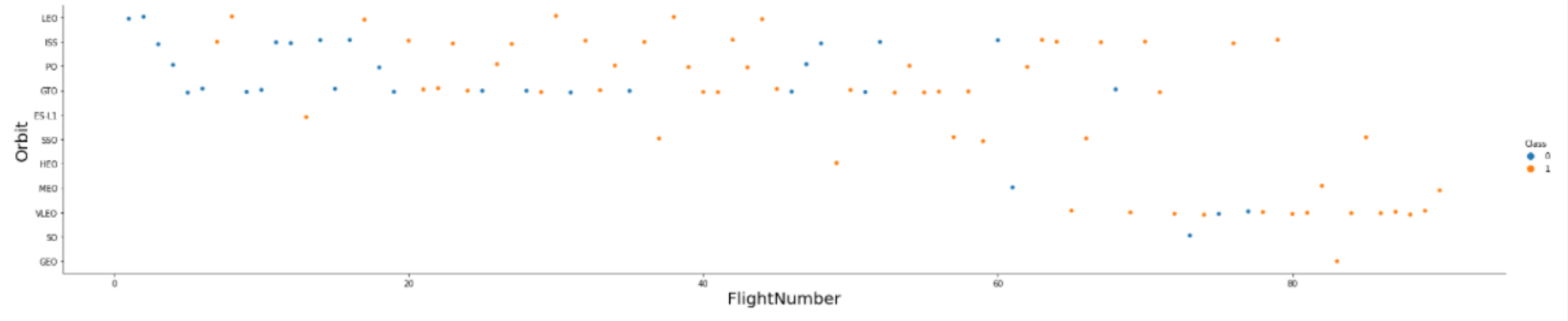
We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Success Rate vs. Orbit Type



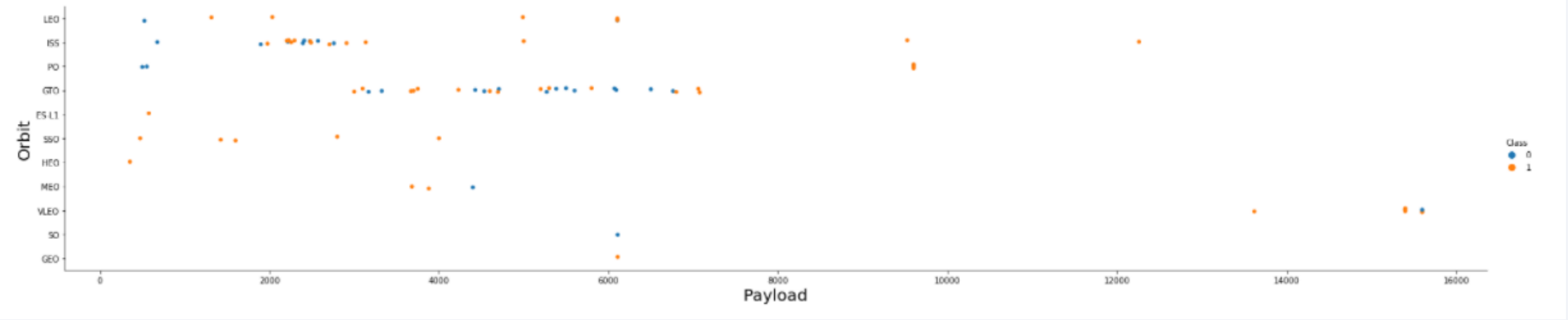
ES-L1, GEO, HEO, SSO have high success rate

Flight Number vs. Orbit Type



we can 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.

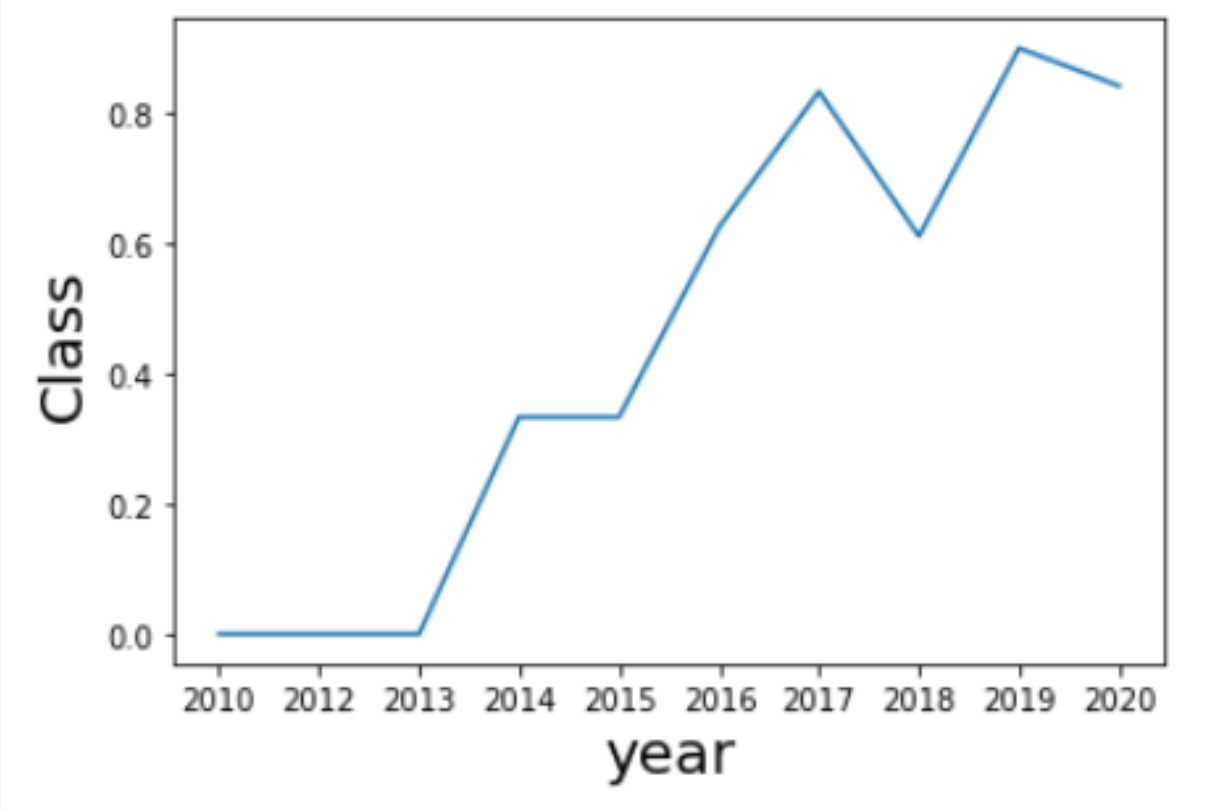
Payload vs. Orbit Type



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

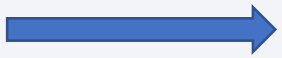


we can observe that the success rate since 2013 kept increasing till 2020

All Launch Site Names

- **SQL QUERY**

```
%sql select DISTINCT(launch_site) from SPACEXTBL
```



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

- **QUERY EXPLAINATION**

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***launch_Site*** column from ***SpaceXtbl***

Launch Site Names Begin with 'CCA'

- **SQL QUERY**

```
select * from SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5
```

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

- **QUERY EXPLAINATION**

Using the word ***TOP 5*** in the query means that it will only show 5 records from ***SpaceXtbl*** and ***LIKE*** keyword has a wild card with the words ***'CAA%'*** the percentage at the end suggests that the Launch_Site name begin with CAA.

Total Payload Mass

- **SQL QUERY**

```
%sql select SUM(payload_mass__kg_) as Total_payload from SPACEXTBL WHERE customer = 'NASA (CRS)'
```

total_payload
45596

- **QUERY EXPLAINATION**

Using the function ***SUM*** summates the total in the column ***PAYLOAD_MASS_KG_***
The ***WHERE*** clause filters the dataset to only perform calculations on ***Customer NASA (CRS)***

Average Payload Mass by F9 v1.1

- **SQL QUERY**

```
%sql select AVG(payload_mass__kg_) as Average from SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

average
2928

- **QUERY EXPLAINATION**

Using the function ***AVG*** works out the average in the column ***PAYLOAD_MASS_KG_***
The ***WHERE*** clause filters the dataset to only perform calculations on ***Booster_version F9 v1.1***

First Successful Ground Landing Date

- **SQL QUERY**

```
%sql select min(DATE) as fisrt_success_date from SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

fisrt_success_date
2015-12-22

- **QUERY EXPLAINATION**

Using the function ***MIN*** works out the minimum date in the column ***Date***
The ***WHERE*** clause filters the dataset to only perform calculations on ***Landing_Outcome Success (ground pad)***

Successful Drone Ship Landing with Payload between 4000 and 6000

- **SQL QUERY**

```
%sql select booster_version from SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass_
_kg_ < 6000
```

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

- **QUERY EXPLAINATION**

Selecting only ***booster_version***
The ***WHERE*** clause filters the dataset to ***Landing_Outcome = Success (drone ship)***
The ***AND*** clause specifies additional filter conditions
Payload_MASS_KG_ > 4000 AND ***Payload_MASS_KG_ < 6000***

Total Number of Successful and Failure Mission Outcomes

- **SQL QUERY**

```
%sql select COUNT(*) as successful_Mission_Outcomes from SPACEXTBL WHERE landing__outcome LIKE '%Success%'
```

successful_mission_outcomes
61

```
%sql select COUNT(*) as Failure_Mission_Outcomes from SPACEXTBL WHERE landing__outcome LIKE '%Failure%'
```

failure_mission_outcomes
10

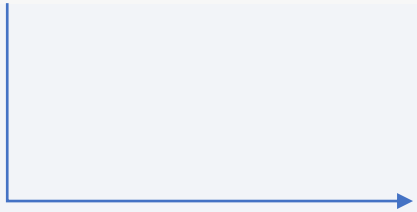
- **QUERY EXPLAINATION**

COUNT clause will count the total number of rows
The ***WHERE*** clause filters the dataset to ***Landing_Outcome*** *and* ***LIKE*** keyword has a wild card with the words ***'%Success%' and '%Failure%' ,*** the percentage in the start and at the end suggests that the ***Landing_Outcome*** should have word success and Failure.

Boosters Carried Maximum Payload

- **SQL QUERY**

```
%sql SELECT DISTINCT booster_version, MAX(payload_mass__kg_) AS max_payload_mass__kg_ FROM SPACEXTBL GROUP BY booster_version ORDER BY ma
x_payload_mass__kg_ DESC
```



booster_version	max_payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600
.	
.	
F9 FT B1038.1	475
F9 B4 B1045.1	362
F9 v1.0 B0003	0
F9 v1.0 B0004	0

- **QUERY EXPLAINATION**

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***booster_version*** column from ***SpaceXtbl***
GROUP BY puts the list in order set to a certain condition.
DESC means its arranging the dataset into descending order

2015 Launch Records

- **SQL QUERY**

```
%sql select booster_version, launch_site, landing__outcome from SPACEXTBL WHERE landing__outcome = 'Failure (drone ship)' AND DATE LIKE
'%2015%'
```

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

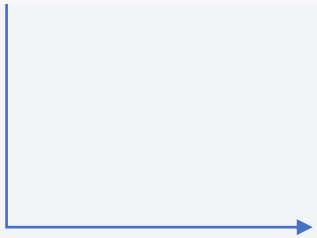
- **QUERY EXPLAINATION**

Selecting the **booster_version, launch_site,** and **landing_outcome**
The ***WHERE*** clause filters the dataset to ***Landing_Outcome = 'Failure (drone ship)'*** *and* ***Date Like '%2015%', where*** the percentage in the start and at the end suggests that the ***Date*** should have **2015**.

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

- **SQL QUERY**

```
%sql SELECT landing__outcome, COUNT(landing__outcome) as count FROM SPACEXTBL WHERE (Date >'2010-06-04') AND (Date < '2017-03-20') GROUP
BY landing__outcome ORDER BY count DESC
```



landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

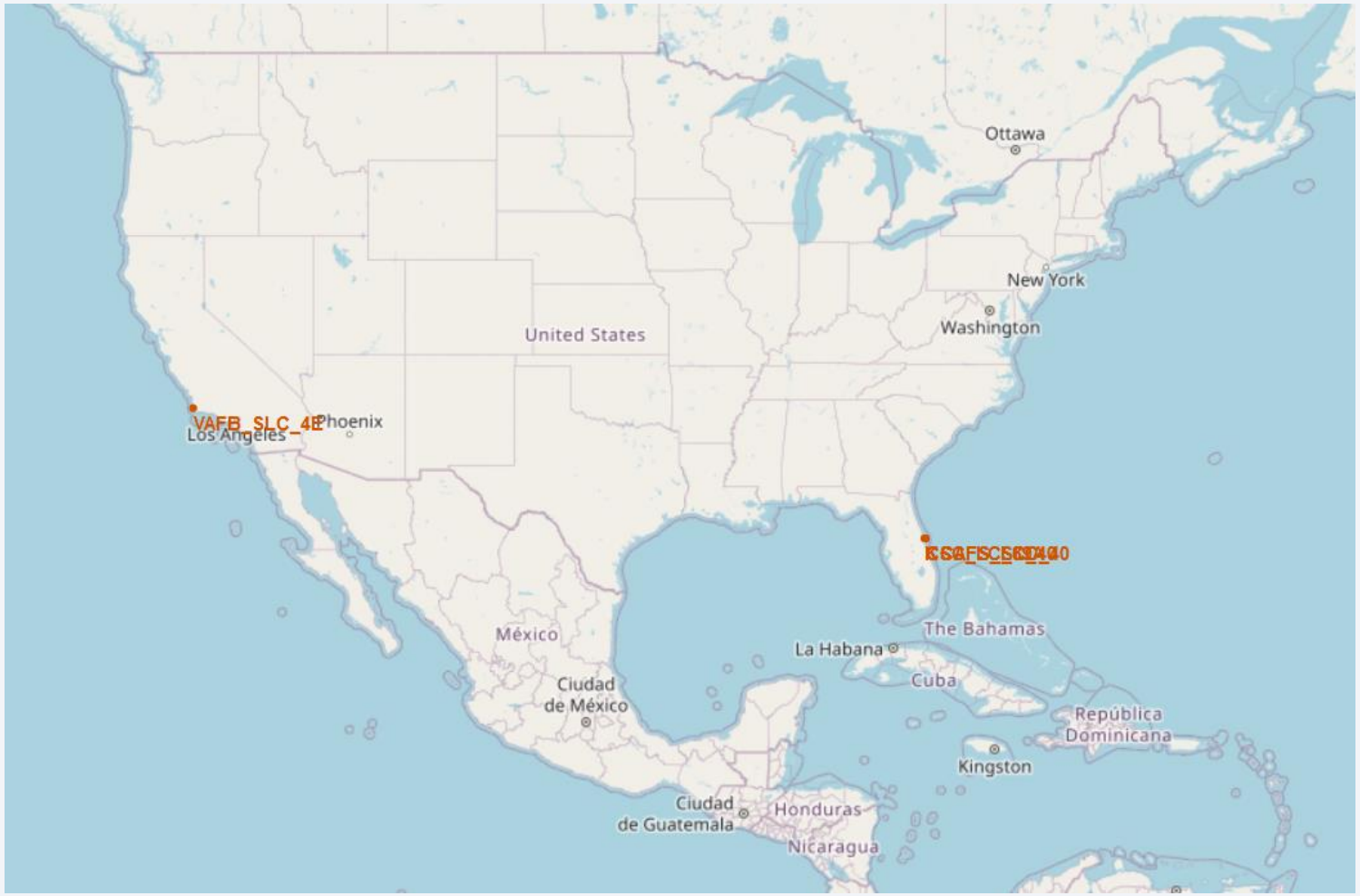
- **QUERY EXPLAINATION**

Selecting the **landing_outcome, Count clause** calculates the number of rows
The ***WHERE*** clause filters the dataset to ***Date > '2010-06.04' AND < '2017-03-20'***
GROUP by *clause groups the outcome on the basis of landing_outcome* .
ORDER by clause sorts the count in decreasing order

Section 4

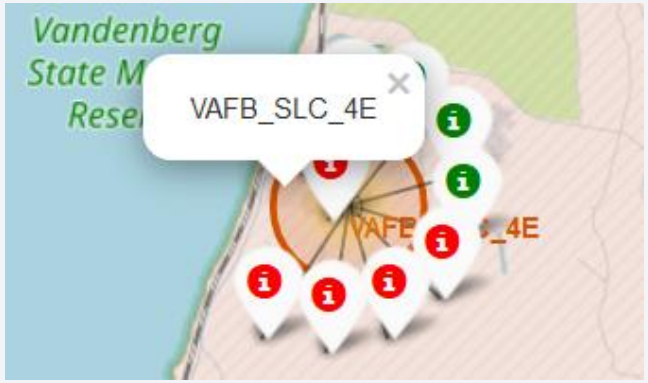
Launch Sites Proximities Analysis

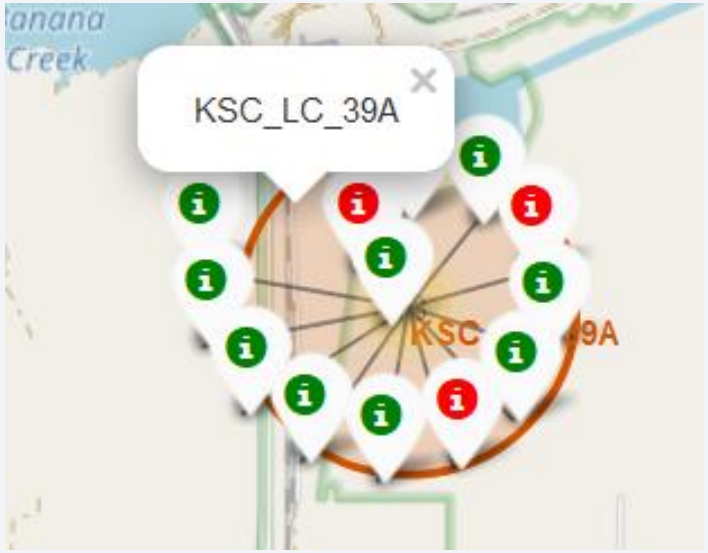
All launch sites global map markers

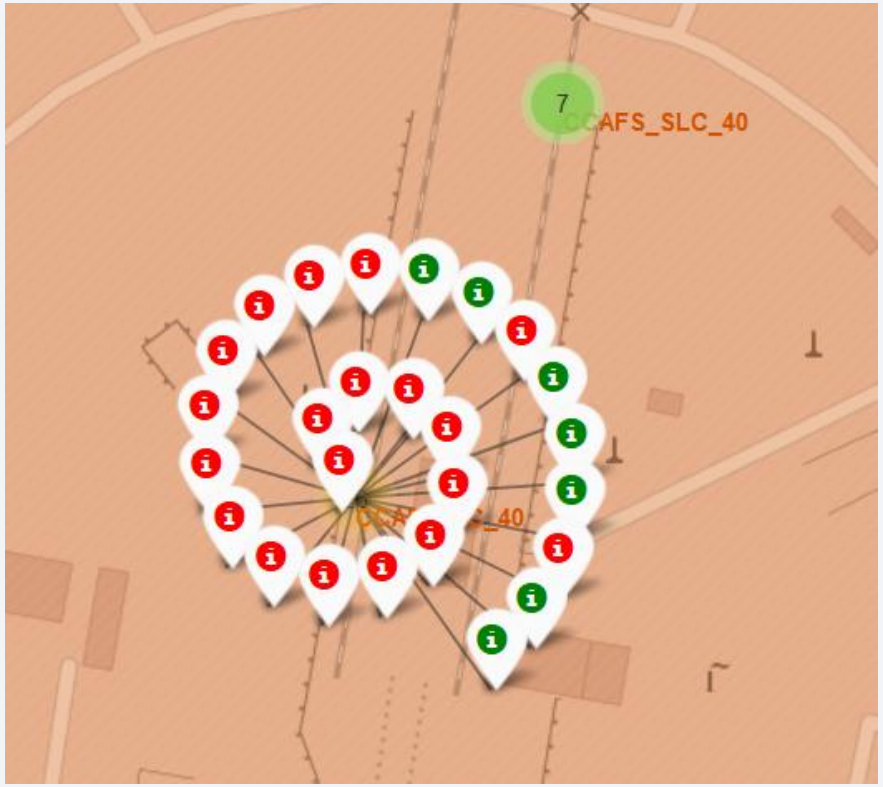


We can see that the SpaceX launch sites are in the United States of America coasts

Colour Labelled Markers

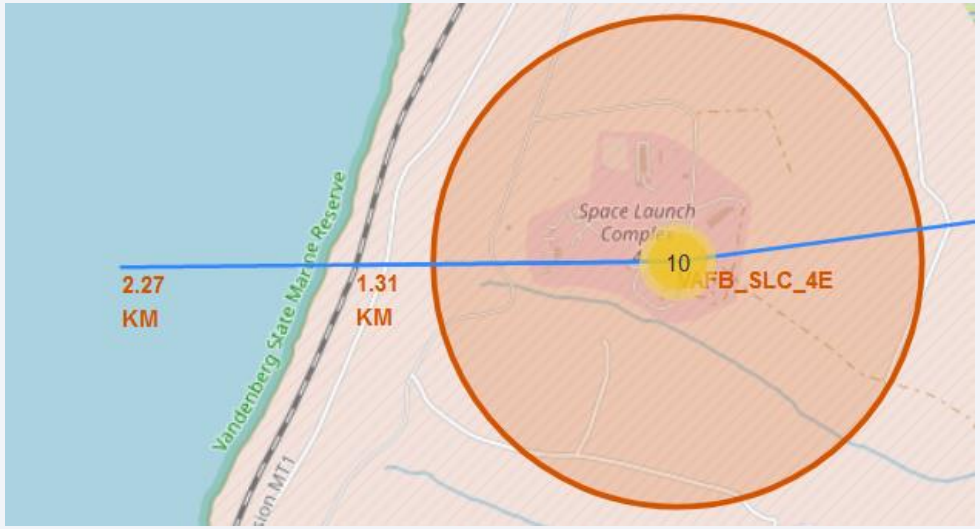






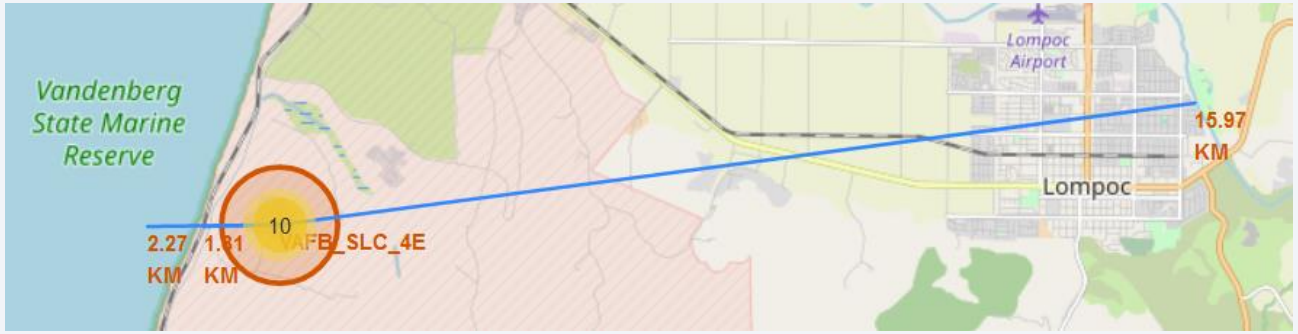
Green Marker shows successful Launches and Red Markershows Failures

Working out Launch Sites distance to landmarks to find trends



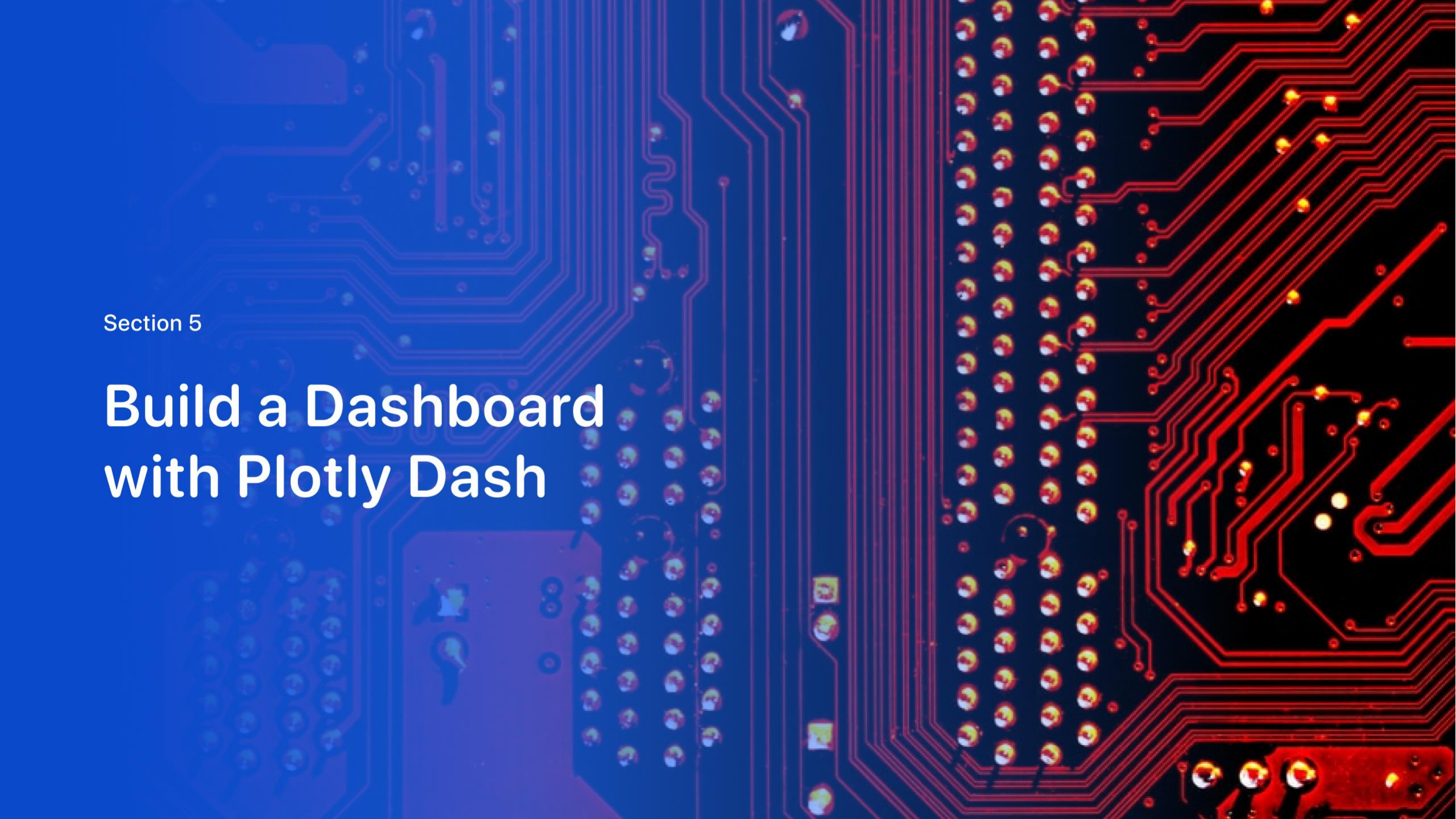
1. All launch sites are close proxmity to coastlines.
2. CCAFS_SLC_40 is within 1 km of highway.
3. VAFB_SLC_4E is wintin 1 km of railways.
4. VAFB_SLC_4E is approx 16 km away from city lompoc.

1. All launch sites are close proxmity to railways.
2. All launch sites are close proxmity to highways.
3. All launch sites are close proxmity to coastlines.
4. All launch sites are away from cities.

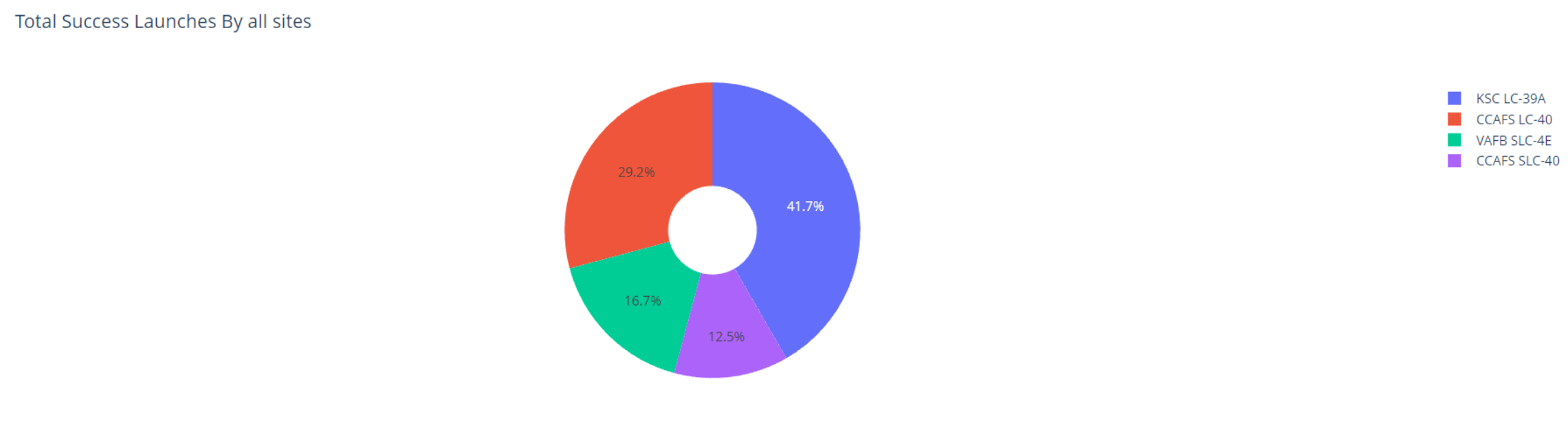


Section 5

Build a Dashboard with Plotly Dash

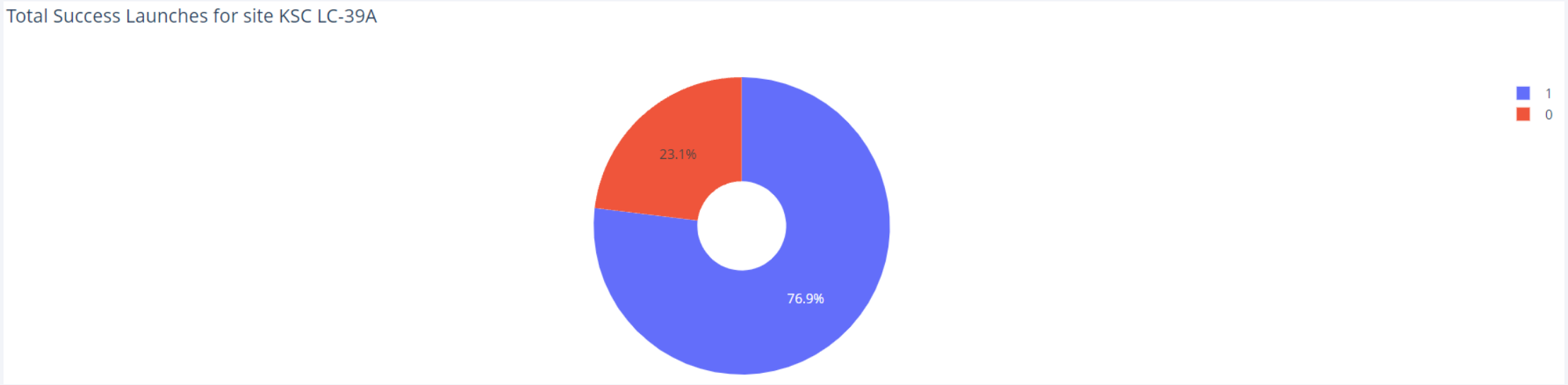


Dashboard – Pie Chart



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

Dashboard – Pie chart for launch site with highest success rate



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Dashboard – Scatter Plot for the launch site with highest success rate

Scatter plot showing the relationship with Payload Mass (Kg) for the ***KSC LC-39A*** launch sites for different booster version. Success rate is higher for lower weighted payloads.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- 4 Machine learning Algorithms logistic regression, Decision Tree, Support vector machine and KNN were compared.

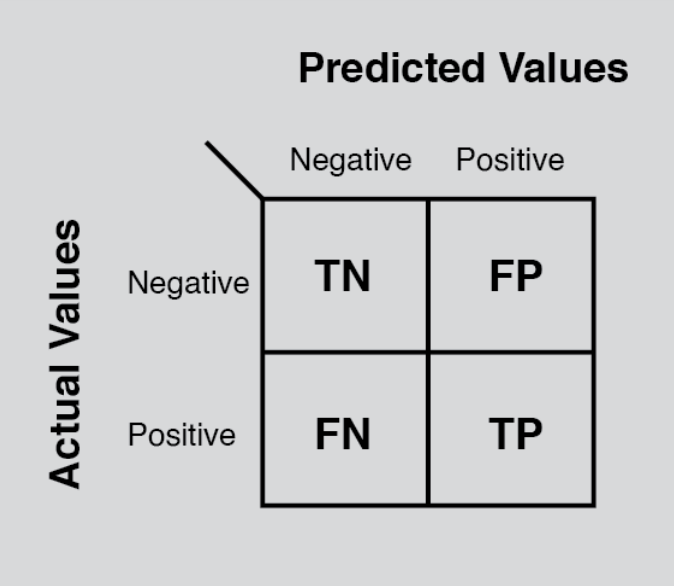
Algorithm	Training accuracy
Logistic Regression	0.8464
SVM	0.8482
Decision Tree	**0.8910**
KNN	0.84821

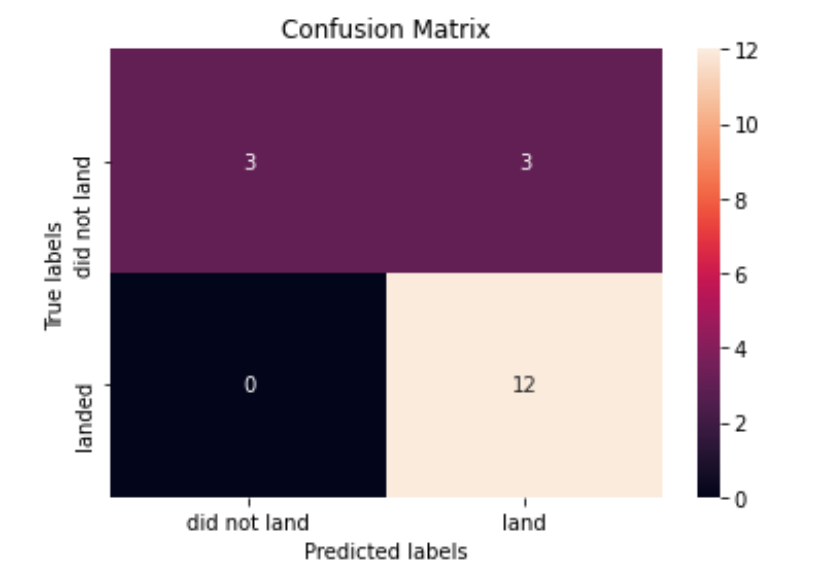
- We clearly have a winner. **Decision tree** gives the best accuracy on the training Dataset.

```
Best Algorithm on Train dataset is DecisionTree with a score of 0.8910714285714285
Best Params is : {'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitte
r': 'random'}
```

Confusion Matrix

- A confusion matrix is **a summary of prediction results on a classification problem**. The number of correct and incorrect predictions are summarized with count values and broken down by each class.
- We see that the major problem is false positives.





Conclusions

- The **Tree Classifier Algorithm** is the best for Machine Learning for this dataset
- **Low weighted payloads** perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- **KSC LC-39A** had the most successful launches from all the sites
- Orbit **GEO, HEO, SSO, ES-L1** has the best Success Rate

Thank you!