IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

William Scarlett

November 29th, 2023

Executive Summary

- **Summary of methodologies**
  - Data Collection via API, Web Scarping
  - Exploratory Data Analysis with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Predictive Analysis
- **Summary of all results**
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- SpaceX is a revolutionary company who has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving thanks to SpaceX astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price down even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

Problems you want to find answers:

- What are the main characteristics of a successful or failed landing?
- What are the effects of each relationship of the rocket variables on the success or failure of a landing?
- What are the conditions which will allow SpaceX to achieve the best landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Webscraping from Wikipedia
- Perform data wrangling
  - Resolved unnecessary columns
  - One Hot encoding for classification models
- 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 collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia
- For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values and fill with whatever needed.
- For web scrapping, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

- Get data, normalize method, and data clean

Now let's start requesting rocket launch data from SpaceX API with the following URL:

[8]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[9]:
```
response = requests.get(spacex_url)
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

- From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

Data Collection - Scraping

- Request the Falcon9 wiki page, created a BeautifulSoup from the html response, extracted all column/variables

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html5lib')
```

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
response.status_code
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])
            launch_dict['Flight No.'].append(flight_number)

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            launch_dict['Time'].append(time)
            #print(time)

            # Booster version
            # TODO: Append the bv into launch_dict with key `Version Booster`
            bv=booster_version(row[1])
            if not(bv):
                bv=row[1].a.string
                launch_dict['Version Booster'].append(bv)
            print(bv)
```

- From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

Data Wrangling

- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).
- We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type.
- We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

- From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

EDA with Data Visualization

Scatter Graphs:

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

Line Graph:

- Success rate vs. Year
- Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.

Bar Graph:

- Success rate vs. Orbit
- Bar graphs show the relationship between numeric and categoric variables.

From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

EDA with SQL

- We performed SQL queries together and understand data from dataset:
- Displaying the names of the unique lauunch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass.
- List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
- • Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

Build an Interactive Map with Folium

- FoliummapobjectisamapcenteredonNASAJohnsonSpaceCenteratHouson,Texas
- Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.Marker).
- Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
- The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
- Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, folium.Icon).
- Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them . (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- • Theseobjectsarecreatedinordertounderstandbettertheproblemandthedata.Wecanshoweasily all launch sites, their surroundings and the number of successful and unsuccessful landings.
- Explain why you added those objects

From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components • Dropdownallowsausertochoosethelaunchsiteoralllaunchsites
- (dash_core_components.Dropdown).
- Piechartshowsthetotalsuccessandthetotalfailureforthelaunchsitechosenwiththe dropdown component (plotly.express.pie).
- Rangesliderallowsausertoselectapayloadmassinafixedrange (dash_core_components.RangeSlider).
- Scatterchartshowstherelationshipbetweentwovariables,inparticularSuccessvs Payload Mass (plotly.express.scatter).

From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

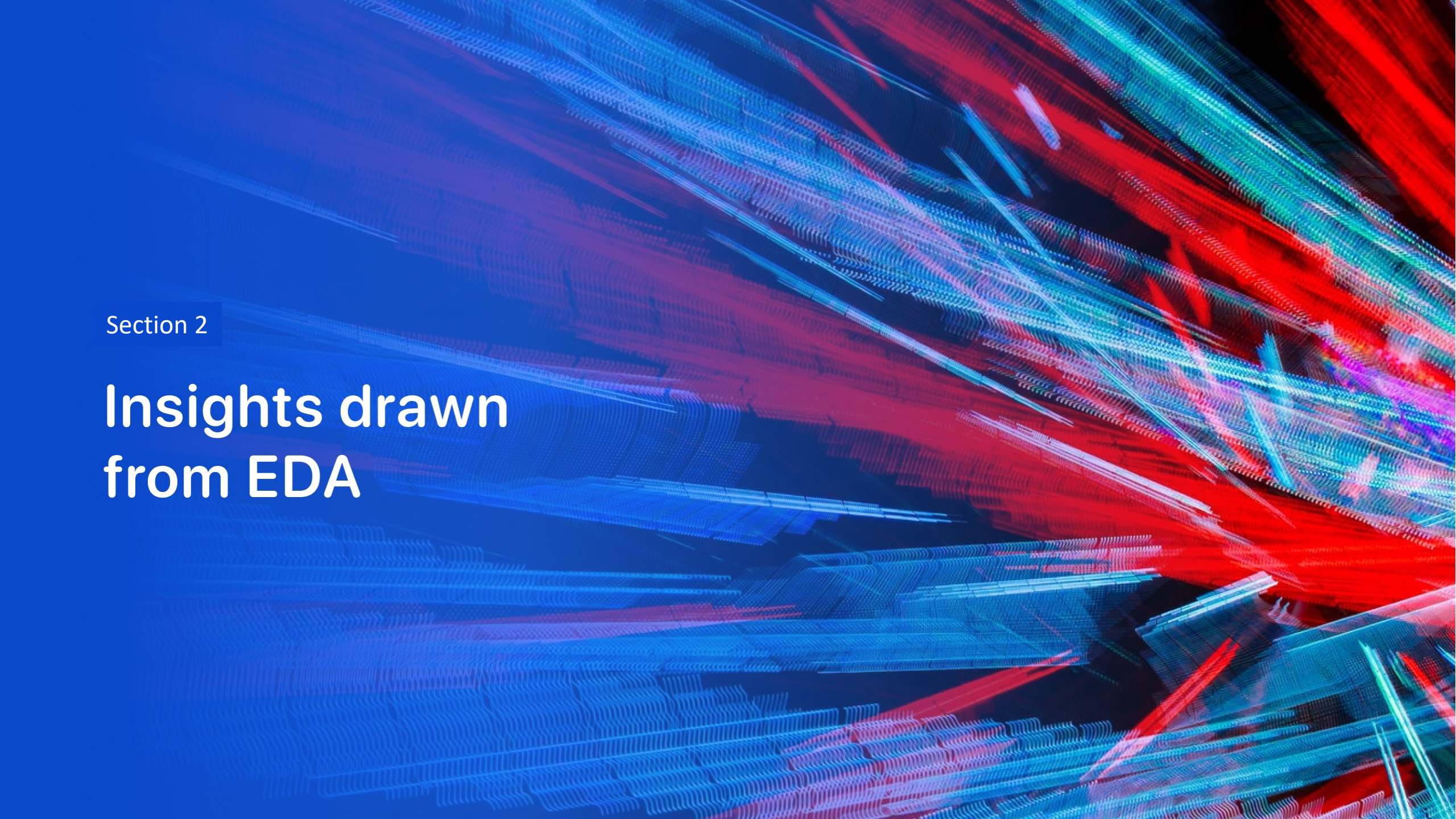
Predictive Analysis (Classification)

- Data preparation
- Load dataset
- Normalize data
- Split data into training and test sets.
- • Model preparation
- Selection of machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Training GridSearchModel models with training dataset
- • Model evaluation
- Get best hyperparameters for each type of model
- Compute accuracy for each model with test dataset
- Plot Confusion Matrix
- • Model comparison
- Comparison of models according to their accuracy
- The model with the best accuracy will be chosen (see Notebook for result)

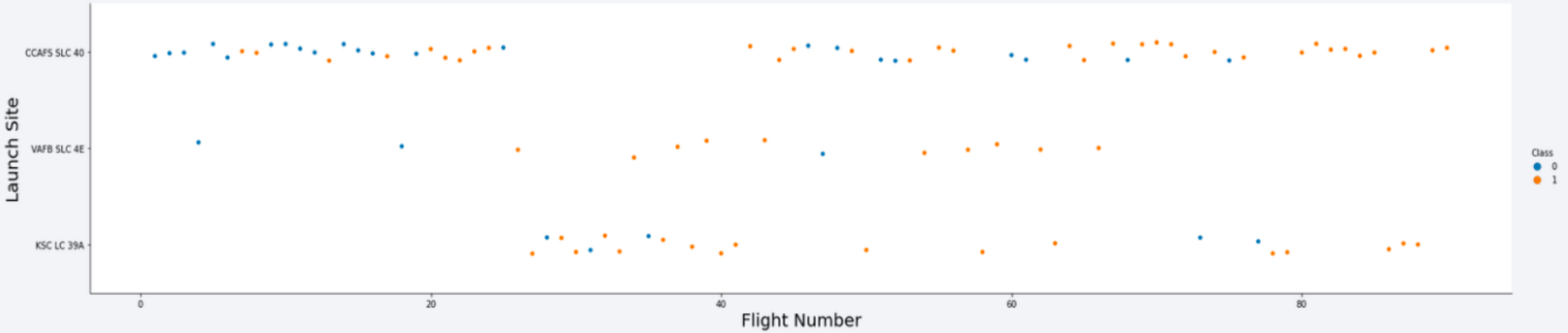
From: https://github.com/willscarlett23/Applied-Data-Science-Capestone

Results

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

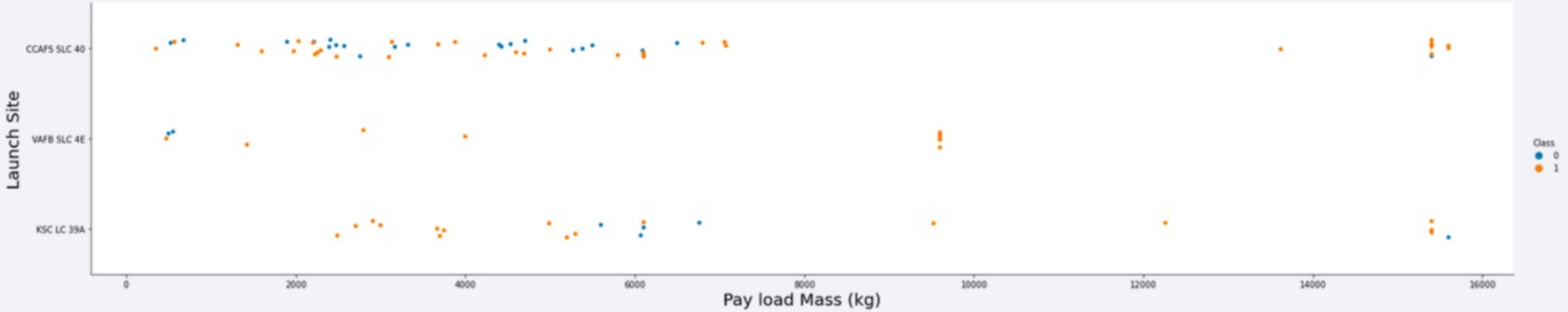

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- We observe that, for each site, the success rate is increasing.

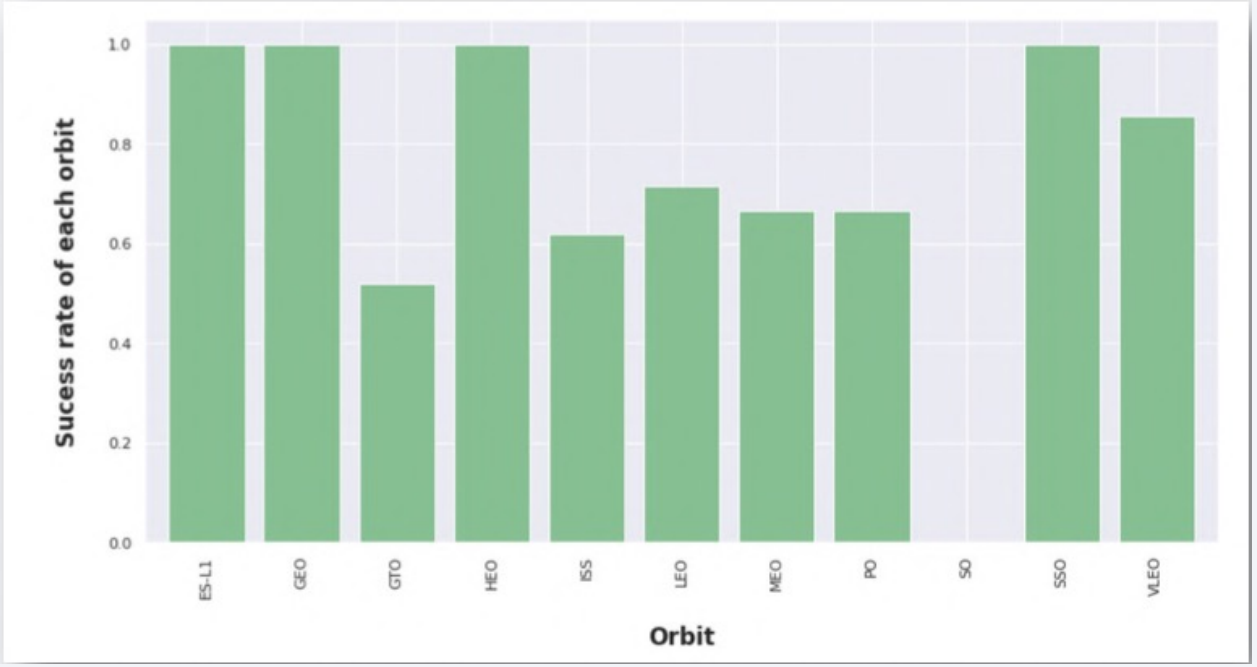
Payload vs. Launch Site



- Depending on the launch site, a heavier payload may be a consideration for a successful landing. On the other hand, a too heavy payload can make a landing fail.

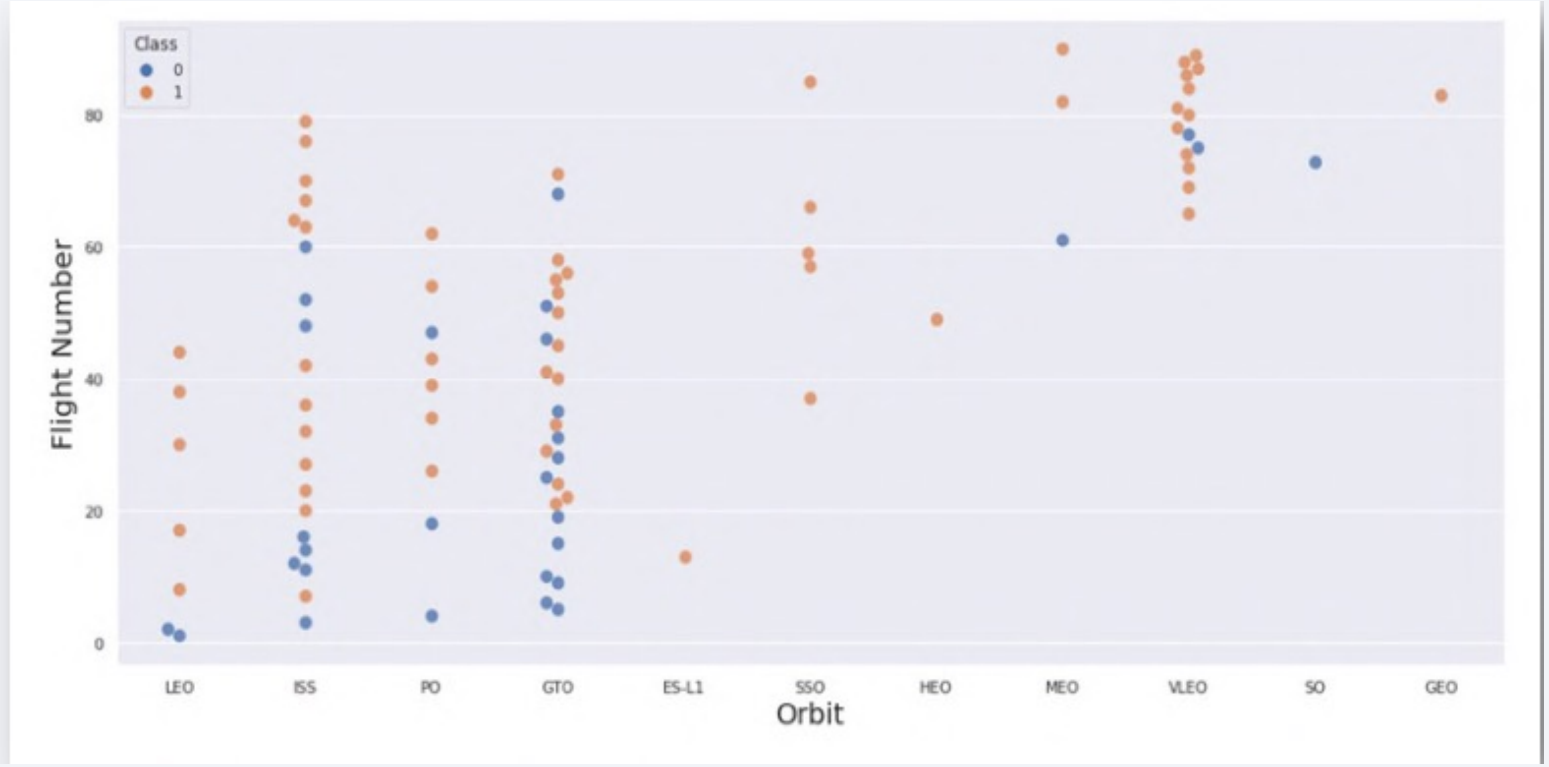
Success Rate vs. Orbit Type

- This figure depicted the possibility of the orbits to influences the landing outcomes as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success.
- However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.

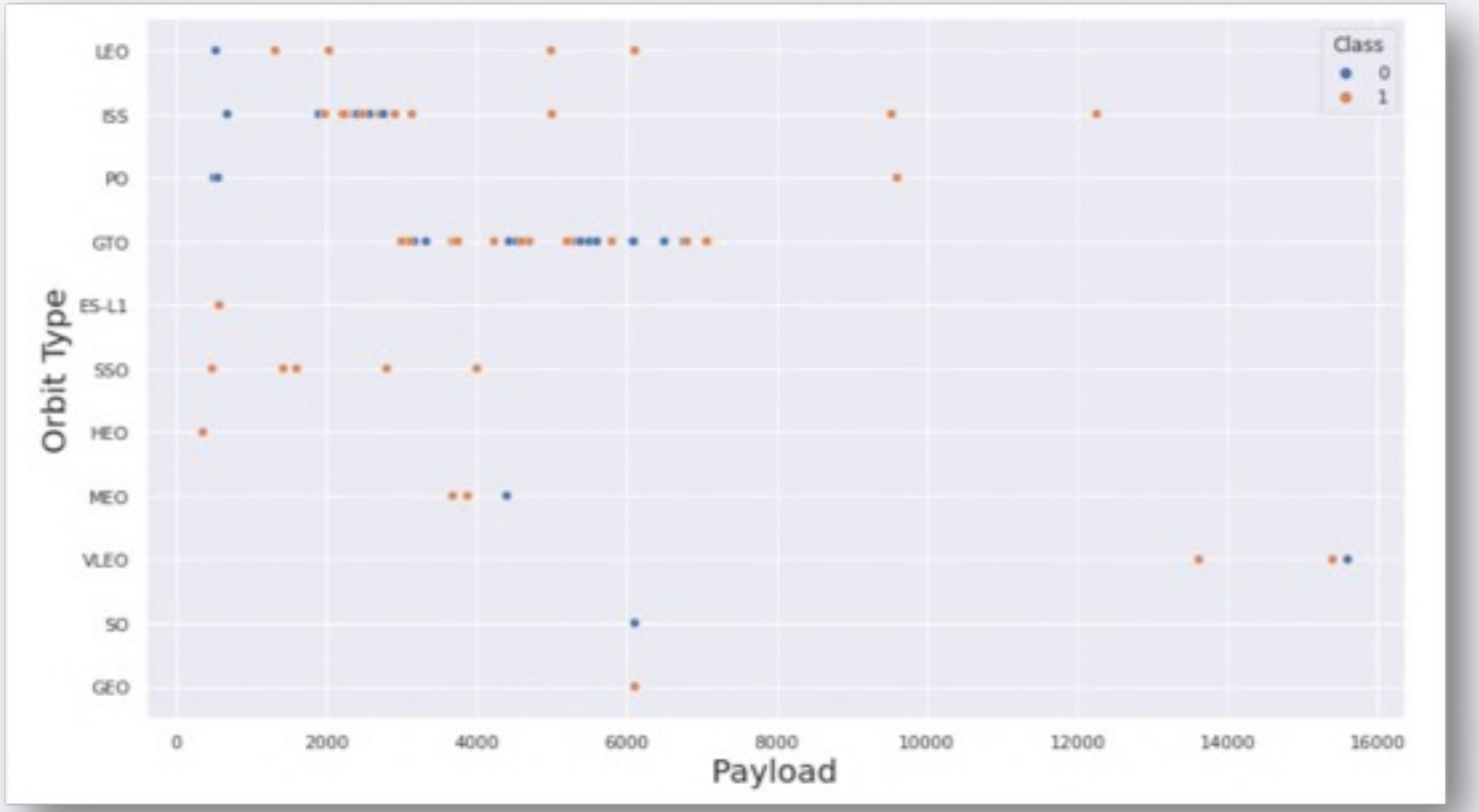


Flight Number vs. Orbit Type

- This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.
- Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset



Payload vs. Orbit Type



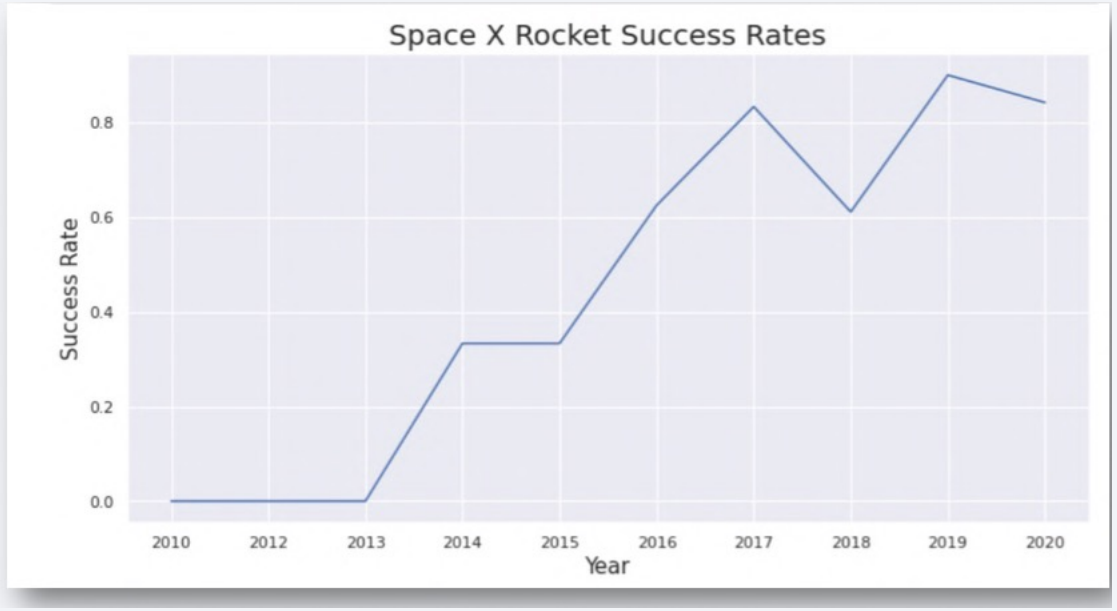
Heavier payload has positive impact on LEO, ISS and PO orbit. However, it has negative impact on MEO and VLEO orbit.

GTO orbit seem to depict no relation between the attributes.

Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.

Launch Success Yearly Trend

- This figures clearly depicted and increasing trend from the year 2013 until 2020.
- If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
In [5]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

- The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

- This query returns the sum of all payload masses where the customer is NASA (CRS).

SUM("PAYLOAD_MASS__KG_")
45596

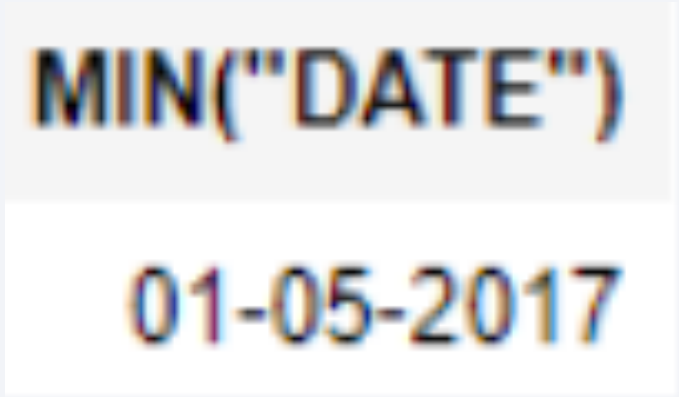
Average Payload Mass by F9 v1.1

- This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- With this query, we select the oldest successful landing.
  The WHERE clause filters dataset in order to keep only records where landing was successful.
  With the MIN function, we select the record with the oldest date.



Successful Drone Ship Landing with Payload between 4000 and 6000

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

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

Total Number of Successful and Failure Mission Outcomes

- With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

- We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

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

- This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

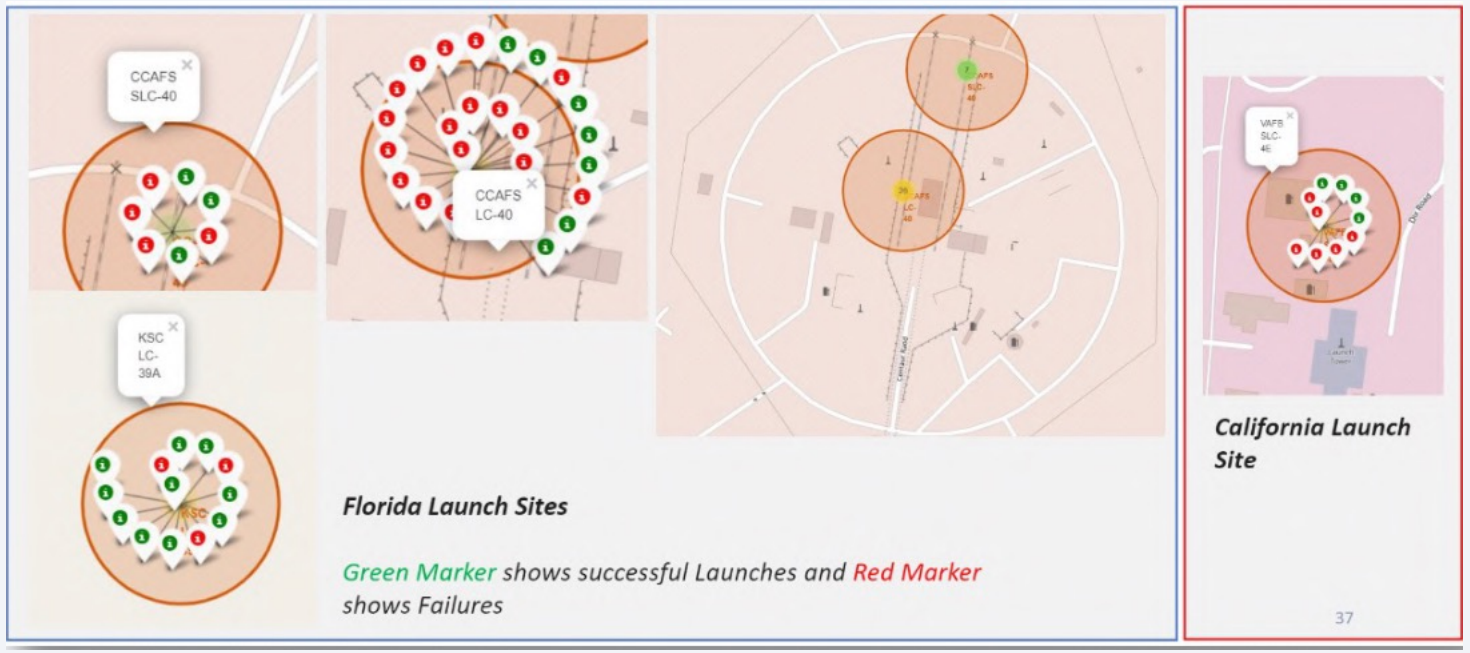
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6


Section 3
Launch Sites
Proximities Analysis

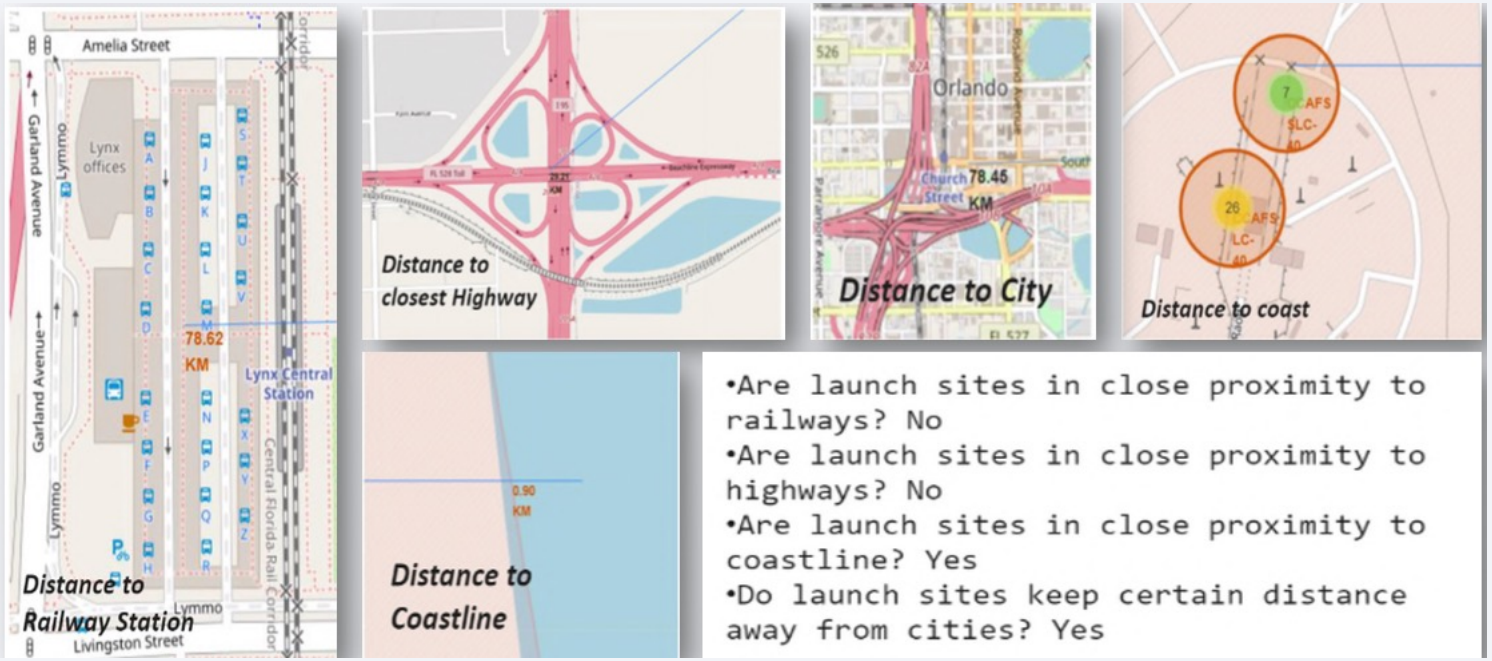
<Folium Map Screenshot 1>

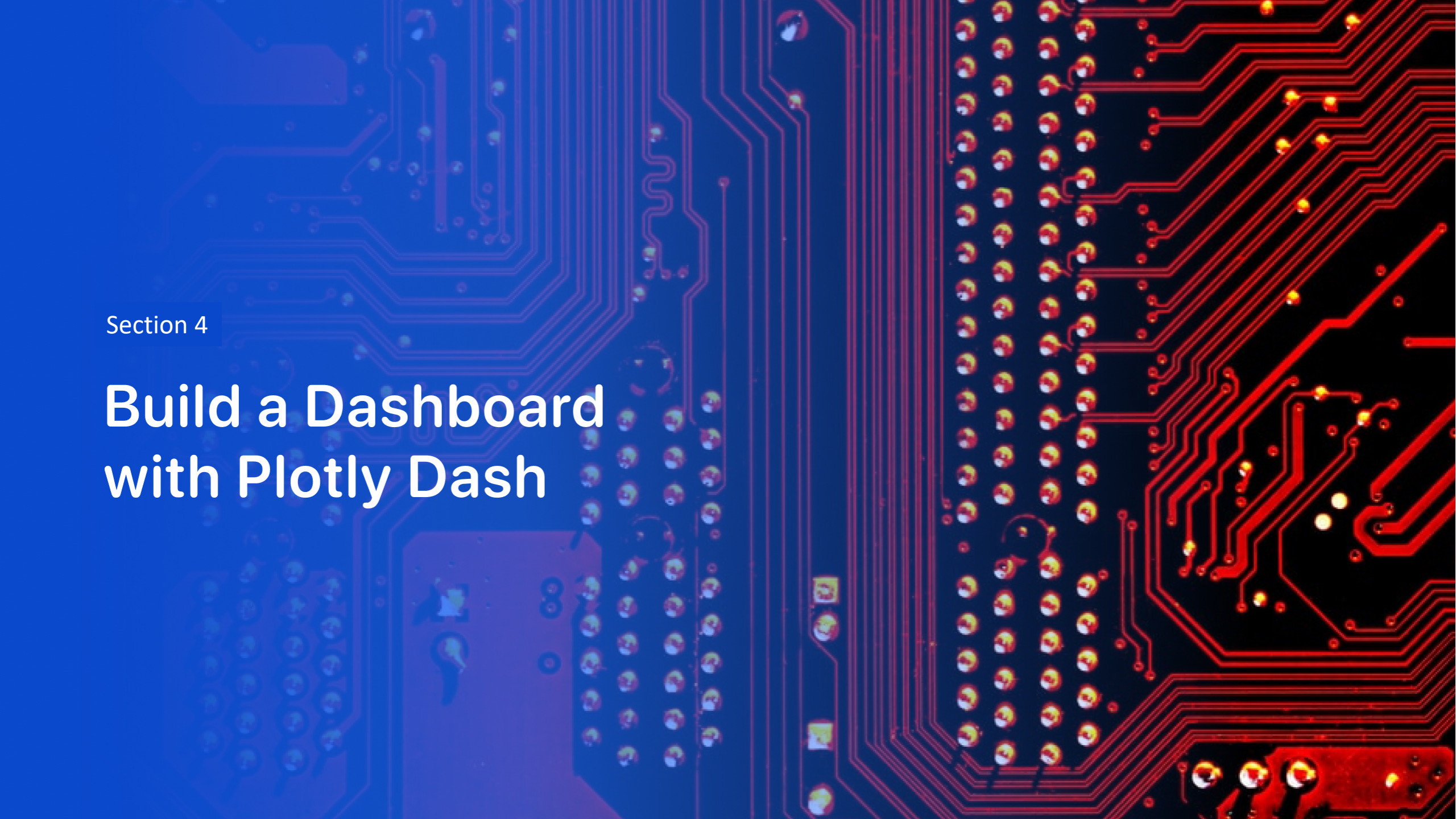
- We can see that all the SpaceX launch sites are located inside the United States

<Folium Map Screenshot 2>

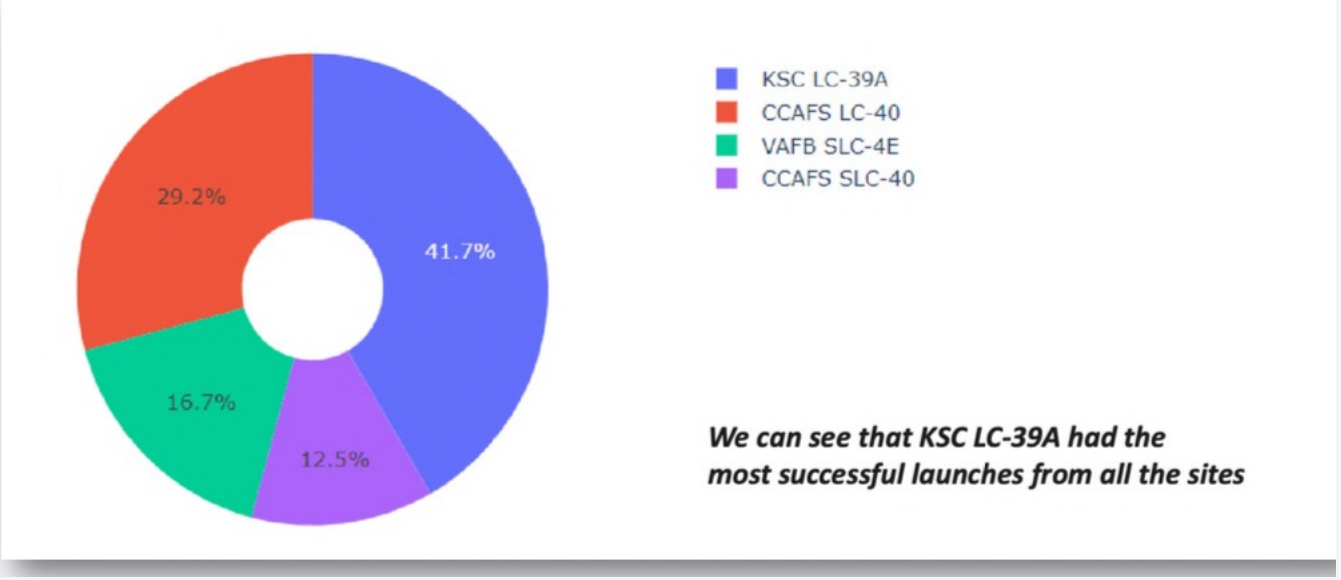


<Folium Map Screenshot 3>




Section 4
Build a Dashboard with Plotly Dash

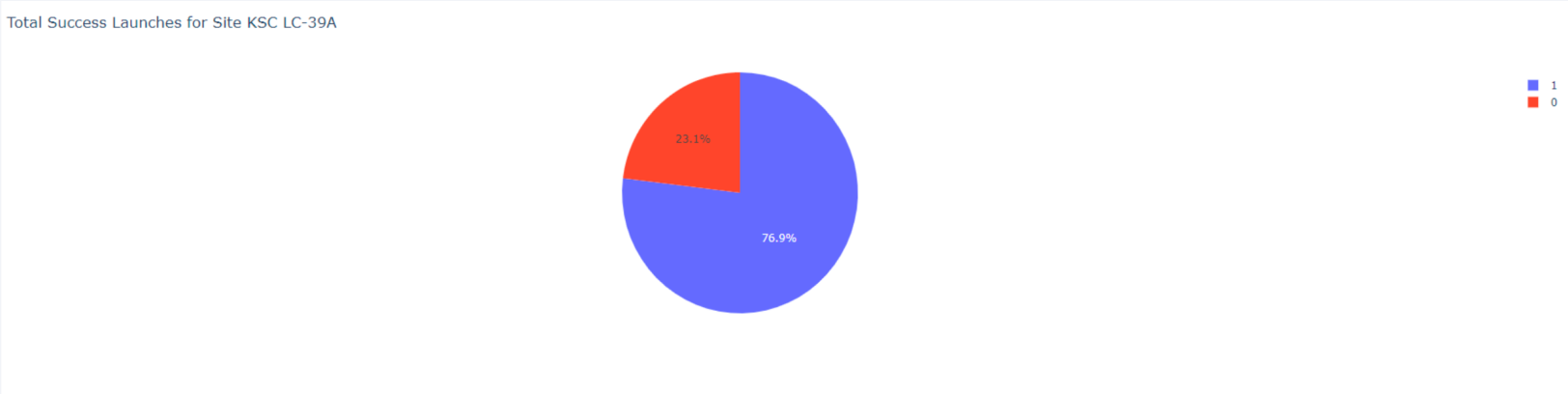
The success percentage by each sites.



<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

Total success launches for site KSC LC-39A



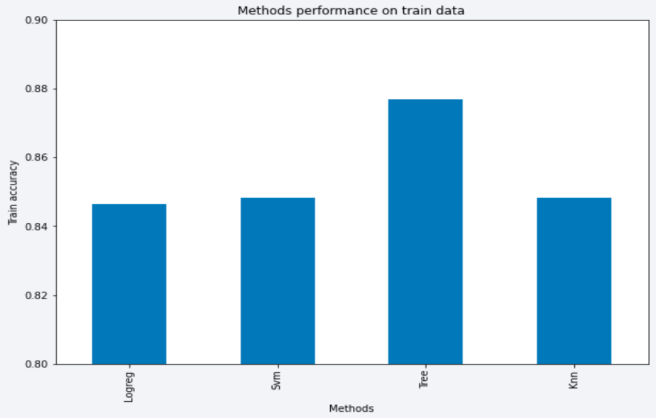
We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

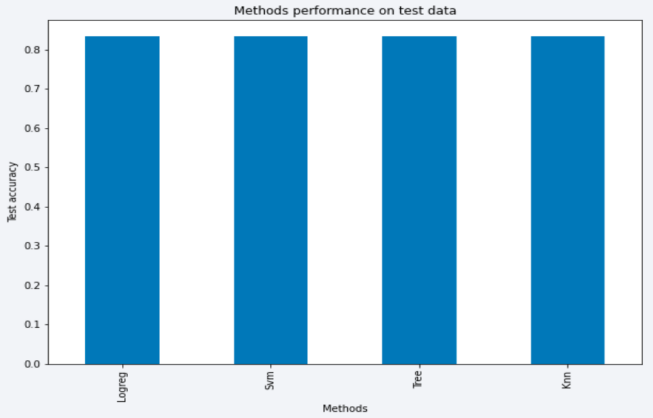
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

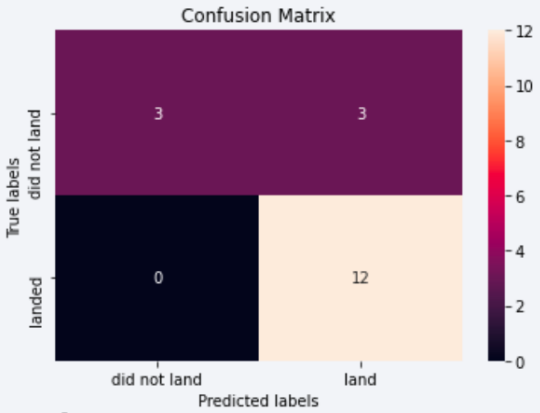




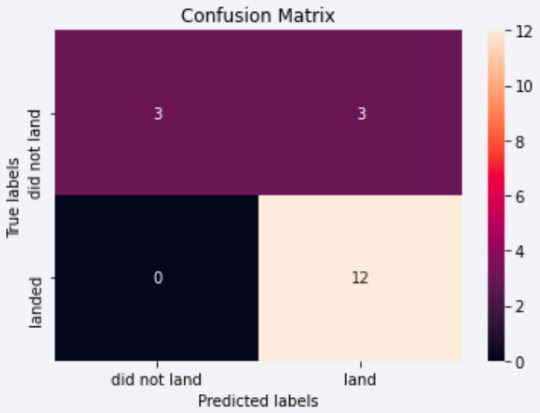
- For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

Confusion Matrix

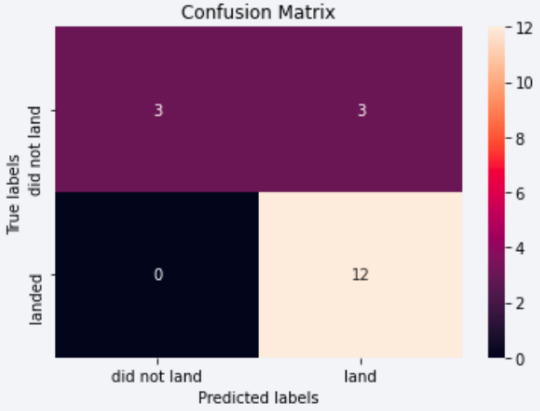
Logistic regression



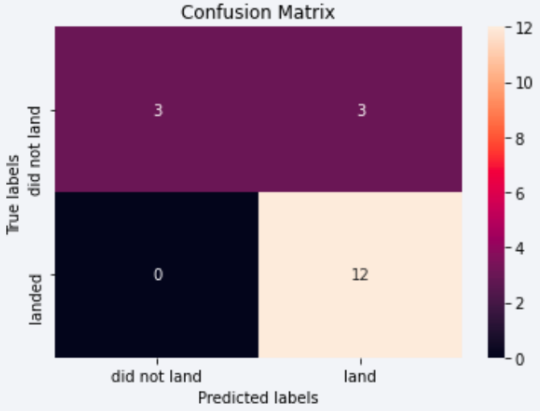
Decision Tree



kNN



SVM



- As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Conclusion

- We can conclude that:
- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!