IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Simran Pal Singh
08-01-2022

Outline

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

Executive Summary

Summary of methodologies:

- Collection of Data Via API and Web Scraping using BeautifulSoup
- Data Wrangling
- Exploratory Data Analysis Visualization and SQL
- Interactive Visual analytics using Folium
- Predictive analysis using machine learning models.

Summary of all results:

- EDA results and interactive visualization dashboard.
- Comparison of predictive analysis model.

Introduction

SpaceX has falcon 9 rocket series, which has a special ability that its first stage can be reused again, provided that it lands successfully. This unique ability allows SpaceX to be much more cost efficient ($ 62 million) than its competitors ($ 165 million).

The problem I try to answer in this project is how can we determine, whether the first stage of the Falcon 9, will land successfully or not. This is a big challenge since millions of dollars can be saved via this research.

Section 1

Methodology

Methodology Executive Summary

1. Data collection methodology:

- The data was collected via SpaceX REST API and Web scraping from Wikipedia.

2. Perform data wrangling

- Using the API we got the JSON file, which we later normalized into a data frame.
- In order to fill the null values, ‘np.nan’ and ‘.replace()’ function were used.

3. (EDA) using visualization and SQL

- Exploring the dataset in sql
- Visualizing different relationships between different variables.

4. Interactive visual analytics using Folium and Plotly Dash

- Built Visual maps using Folium.

5. Predictive analysis using classification models

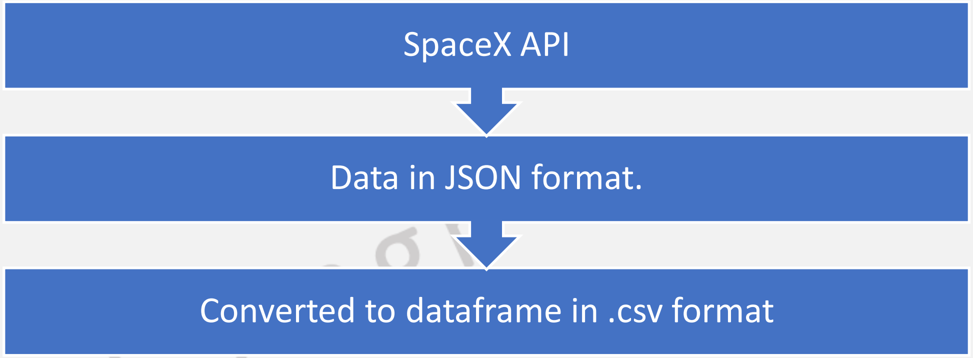
- Different models were trained and tested using a list of parameters.
- Models include: Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors.

Data Collection

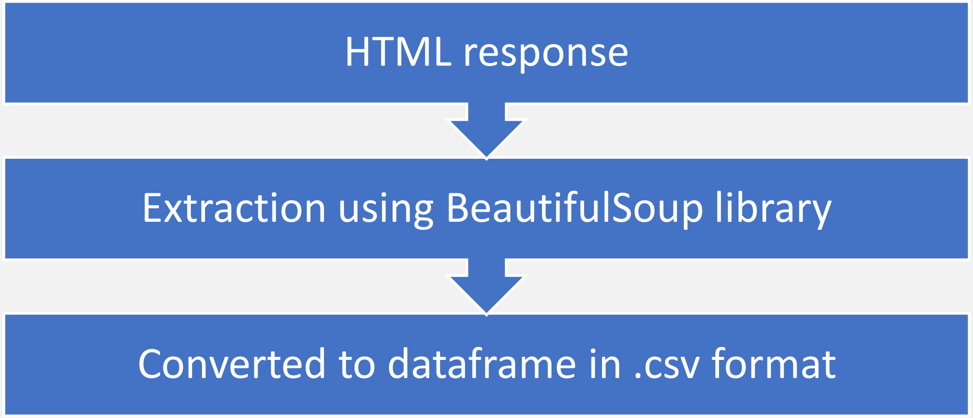
Describe how data sets were collected.

- Obtained Data from API and Web pages.
- Made it into a data frame.
- Filtered unnecessary data by formatting the data frame.
- Converting the data frame into .csv file for further use
- **You need to present your data collection process use key phrases and flowcharts.** **LINK: Web Scraping, SpaceX API**

API method:



Web scrapping method:


HTML response
Extraction using BeautifulSoup library
Converted to dataframe in .csv format

Data Collection – SpaceX API

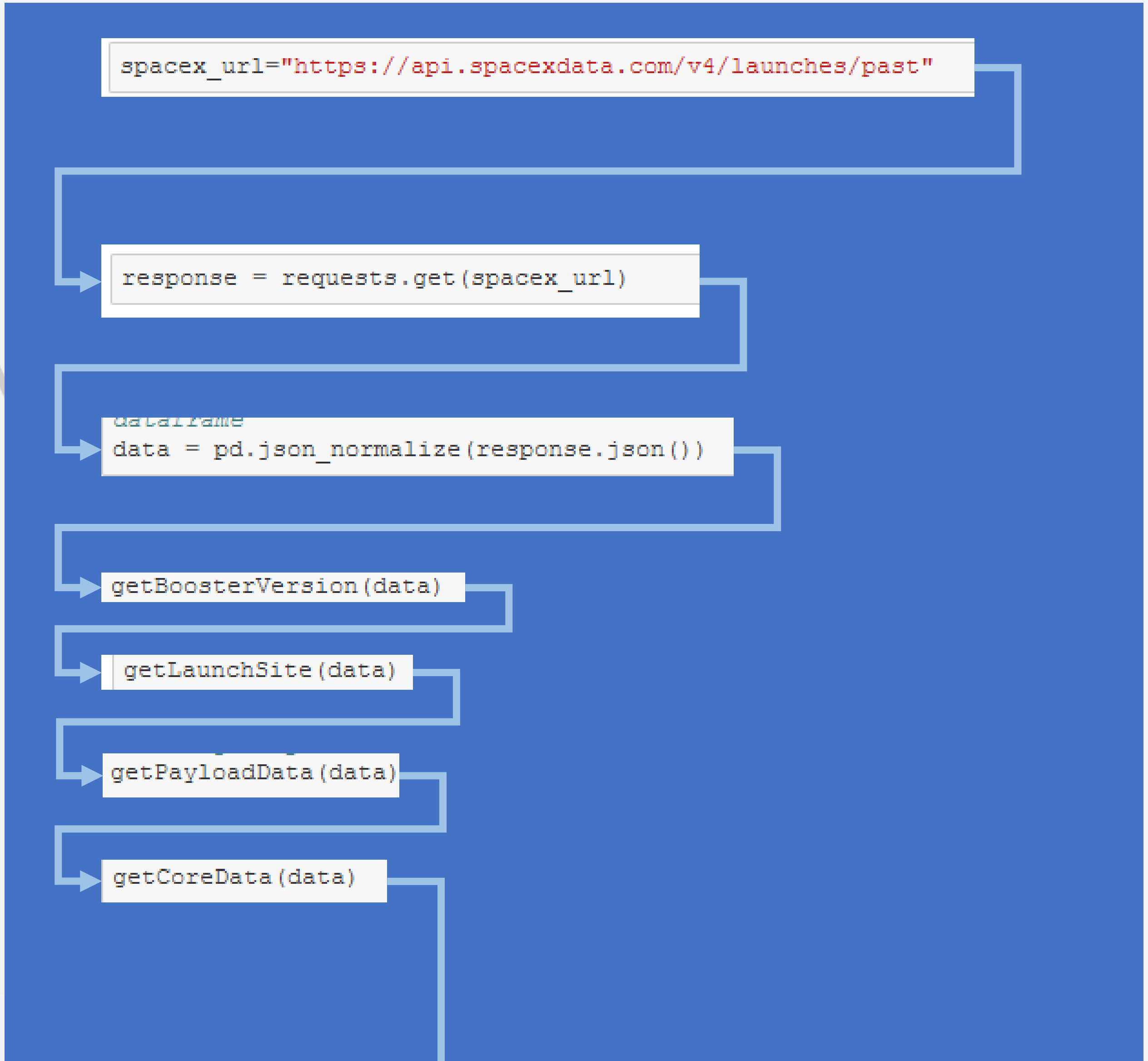
1. First we got the URL to the SpaceX REST API.

2. Getting a response File using the requests.get().

3. Converting it into a data frame.

4. Applying functions to format the data according to our need.

Link: SpaceX API



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

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

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

```
getBoosterVersion(data)
```

```
getLaunchSite(data)
```

```
getPayloadData(data)
```

```
getCoreData(data)
```

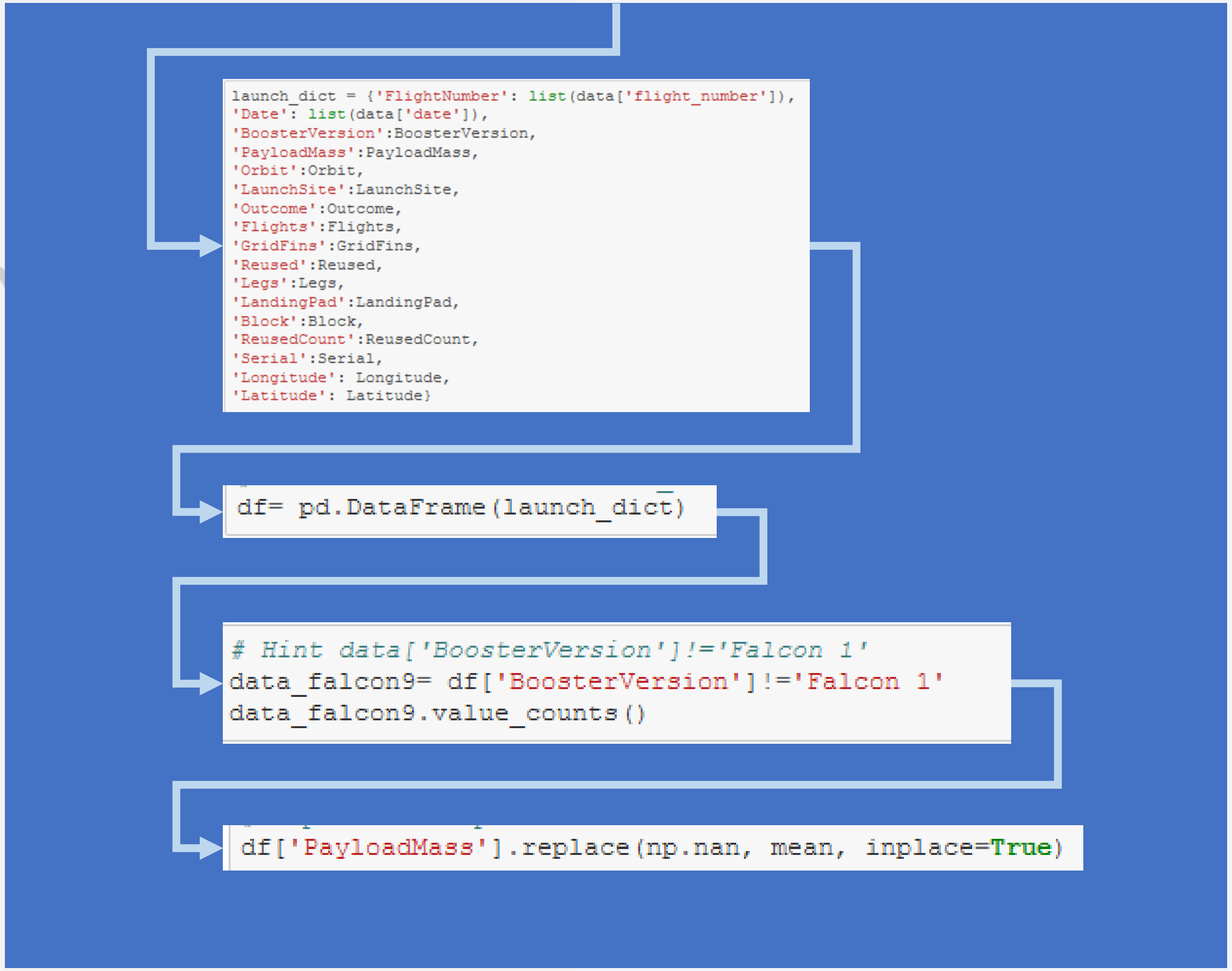
Data Collection – SpaceX API

5. Combining columns into a dictionary.

6. Creating a data frame using that dictionary.

7. Filtering Dataframe to only include Falcon-9 launches

8. Dealing with Null Values.



```
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(launch_dict)
```

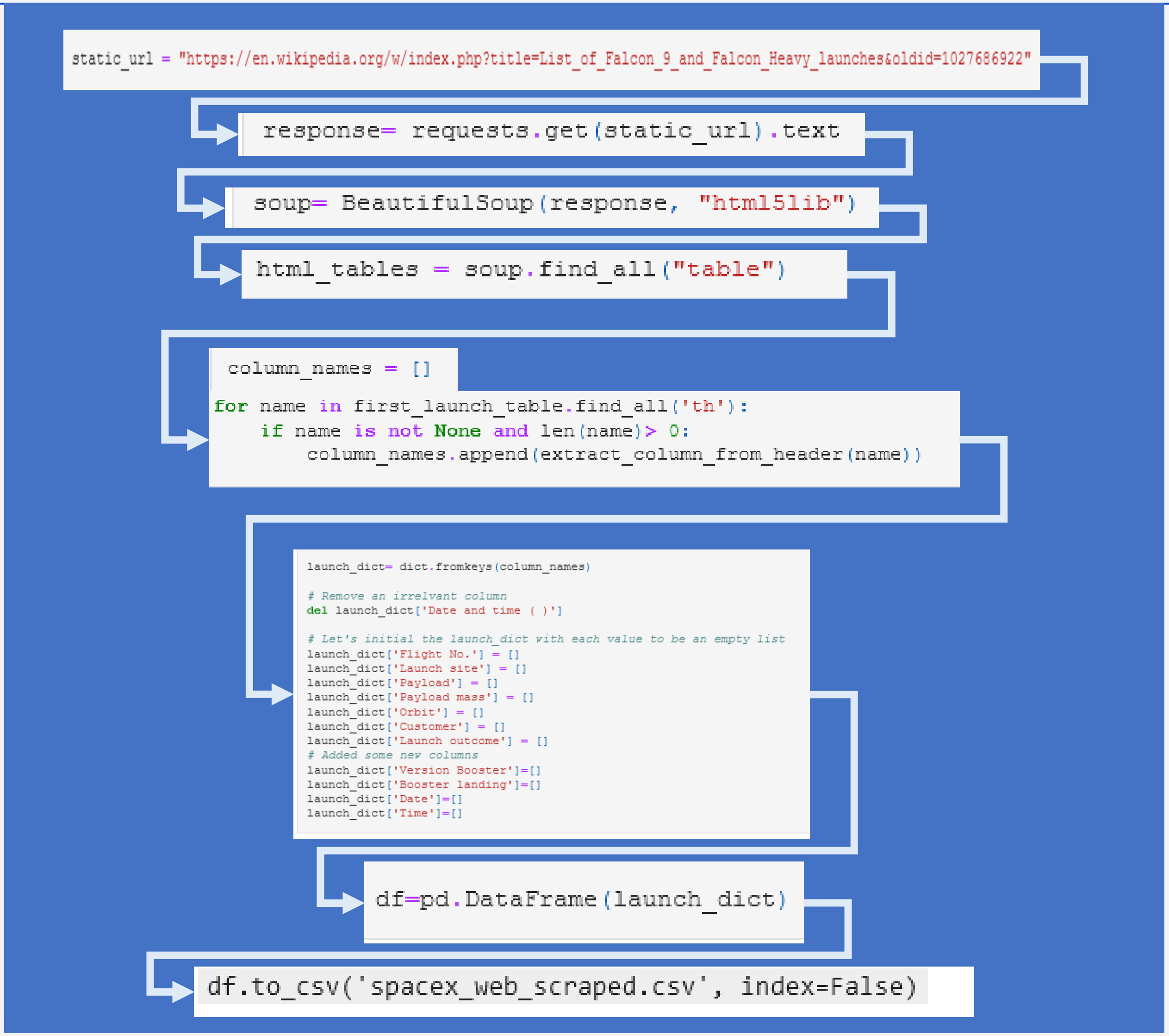
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9= df['BoosterVersion']!='Falcon 1'
data_falcon9.value_counts()
```

```
df['PayloadMass'].replace(np.nan, mean, inplace=True)
```

Data Collection – SpaceX API

1. Getting response from HTML.
2. Creating a BeautifulSoup object.
3. Finding Tables.
4. Extracting Columns
5. Creating a dictionary
6. Appending data to keys
7. Converting dictionary to dataframe.
8. Exporting dataframe into .csv format.

Link: Web scrapping



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
response= requests.get(static_url).text
```

```
soup= BeautifulSoup(response, "html5lib")
```

```
html_tables = soup.find_all("table")
```

```
column_names = []
for name in first_launch_table.find_all('th'):
    if name is not None and len(name)> 0:
        column_names.append(extract_column_from_header(name))
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

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

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

Data Wrangling

Imported the dataset and libraries into the notebook.

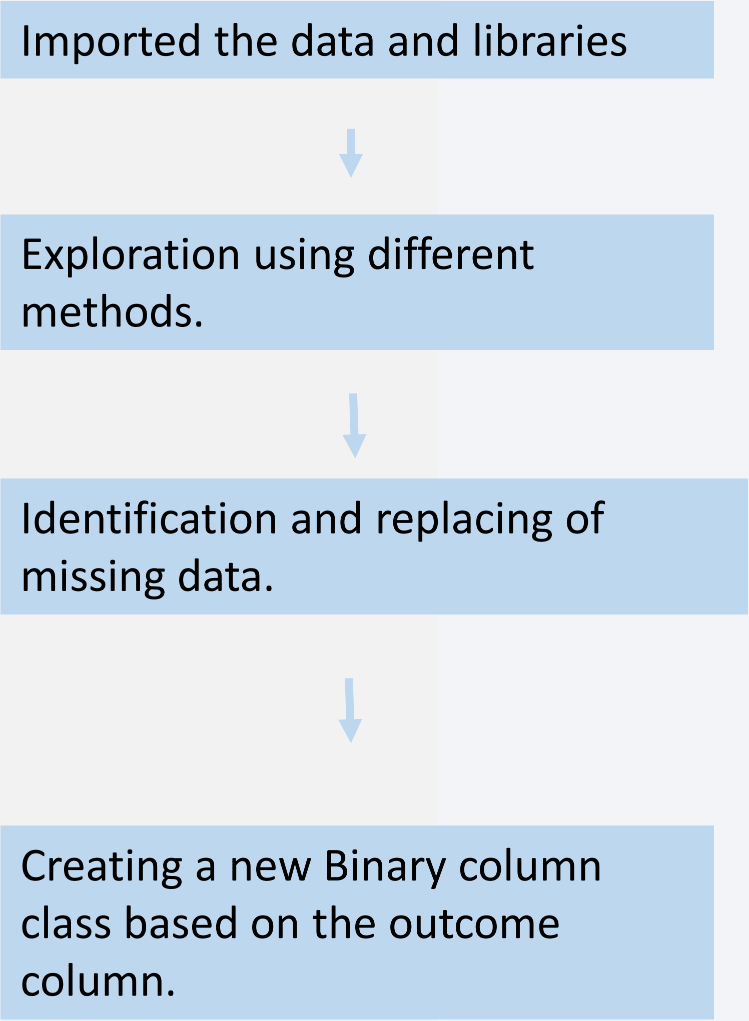
Before making changes to the data,

- Explored the dataframe, using .head(), dtypes and .value_count()

Data analysis:

- Identified and calculated the percentage of missing values in each column of the dataframe.
- Replaced the null data in the dataframe.
- Made a binary value column based on landing outcome, called ‘class’. Which indicates if the landing was successful or ended in failure.

Link: EDA (Data Wrangling)



Imported the data and libraries

Exploration using different methods.

Identification and replacing of missing data.

Creating a new Binary column class based on the outcome column.

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts.

1. Plotted the relationship between **Flight Number** and **Launch Sites**. This allowed me to observe the number of launches from each site and also observe the number of successful and failed attempts as well.

2. Plotted the relationship between the **Payload Mass** and **Launch Sites**, this enabled me to observe, what's the payload mass on average and which sites have a better success rate with heavier payloads and which ones are used for the lighter ones.

3. Plotted the **success rate** of each **orbit** . This enabled me to see which orbit missions are easier to accomplish and which ones are harder to pull off. This can be an important factor in our machine learning model later.

Link: EDA with Visualisation

Cont..

EDA with Data Visualization

4. Visualized the relation in **Flight Number** and **Orbit t**ype to check whether there's a correlation between them, there's a trend that suggest the more the number of flight in an orbit the higher are the chances of success but there's nothing solid to support it.

5. Visualized the relation between **Payload mass** and **Orbit**, to check if there's any correlation in between the two.

6. Plotted the **success rate** against the years of launches to see if there's been any improvement in the success rate and there was a positive trend observed.

Link: EDA with Visualisation

EDA with SQL

- Displayed the names of the unique launch sites in the space mission.
- Displayed 5 records where launch sites begin with the string 'CCA’.
- Displayed the total payload mass carried by boosters launched by NASA (CRS).
- Displayed average payload mass carried by booster version F9 v1.1.
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listed the names of the booster versions which have carried the maximum payload mass. Using a subquery.
- Listed the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Ranked 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.

LINK: EDA with SQL

cont..

Build an Interactive Map with Folium

- **Circles**: To pin the **Important locations** on the map.
- **Makers**: To highlight **Launch sites**.
- **Red/Green Icons**: To highlight if the attempt resulted in **Success** or **Failure**.
- **Cluster**: To highlight the N**umber of launches** from a particular facility.
- **Lines**: To highlight the **Distance** between any two locations.

The success rate of a launch can also be influenced by the location and proximities of the launch sites. Plotting these also helps the user understand and discover correlation between various sites and launches that might not be explainable by data only.

Link: Interactive Visual Analytics with Folium

Build a dashboard with Plotly Dash.

We used Plotly Dash to create an interactive dashboard:

Dropdown

- For choosing different launch sites.

Range Slider

- For choosing a payload range(0-10000kg)

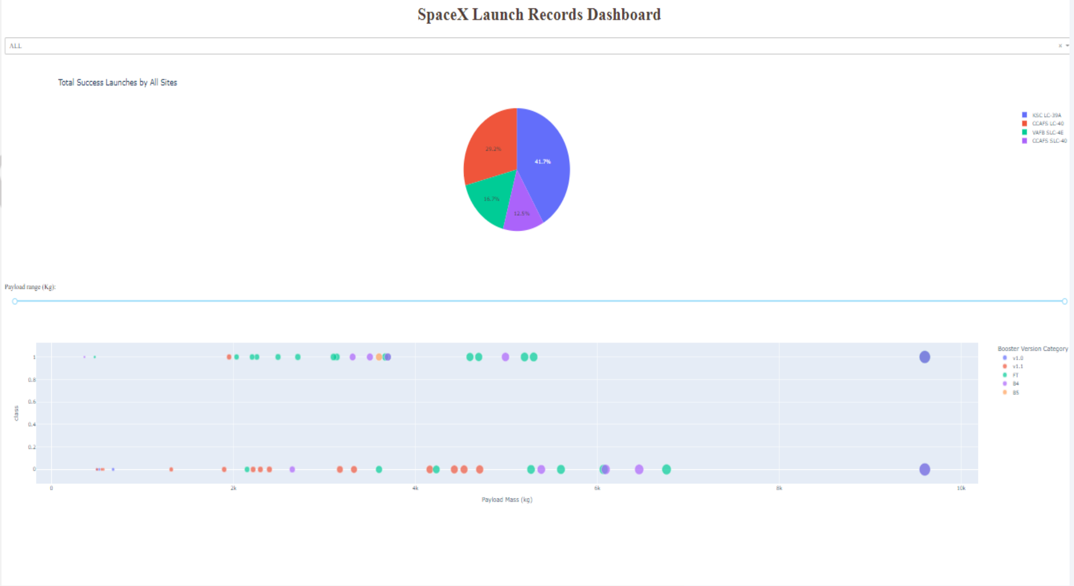
Pie Chart

- When 'ALL' sites is chosen, it shows the distribution of success launches per site.
- When a specific site is chosen, it shows the success rate of that site.

Scatter Plot

- We can choose a payload range using the slider and the booster version categories depending on payload mass and class.

Link: [Plotly_dash]()



SpaceX Launch Records Dashboard

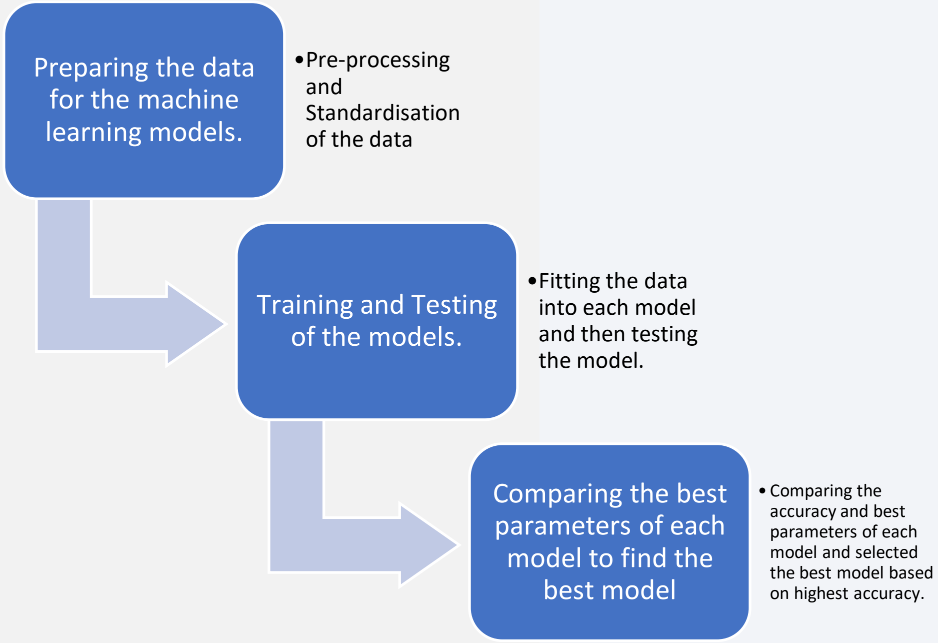
ALL

Payload range (Kg):

Predictive Analysis (Classification)

- Imported the necessary data and libraries and then standardized the data for fitting into the different methods.
- Created objects for Logistic Regression, SVM, Decision Tree and KNN models and fitted each model with GridSearchCV object and ran the models using training set.
- Afterwards Testing set was fit into each model and the results were compared with the ‘Y_test’ set each time by calculating the accuracy between ‘yhat’ and ‘Y_test’.
- The best parameters were then selected using the .score() method.
- These parameters were then selected based on the model that had the highest score.

Link: [Machine Learning Prediction]()



Preparing the data for the machine learning models.

- Pre-processing and Standardisation of the data

Training and Testing of the models.

- Fitting the data into each model and then testing the model.

Comparing the best parameters of each model to find the best model

- Comparing the accuracy and best parameters of each model and selected the best model based on highest accuracy.

Results

Exploratory data analysis results:

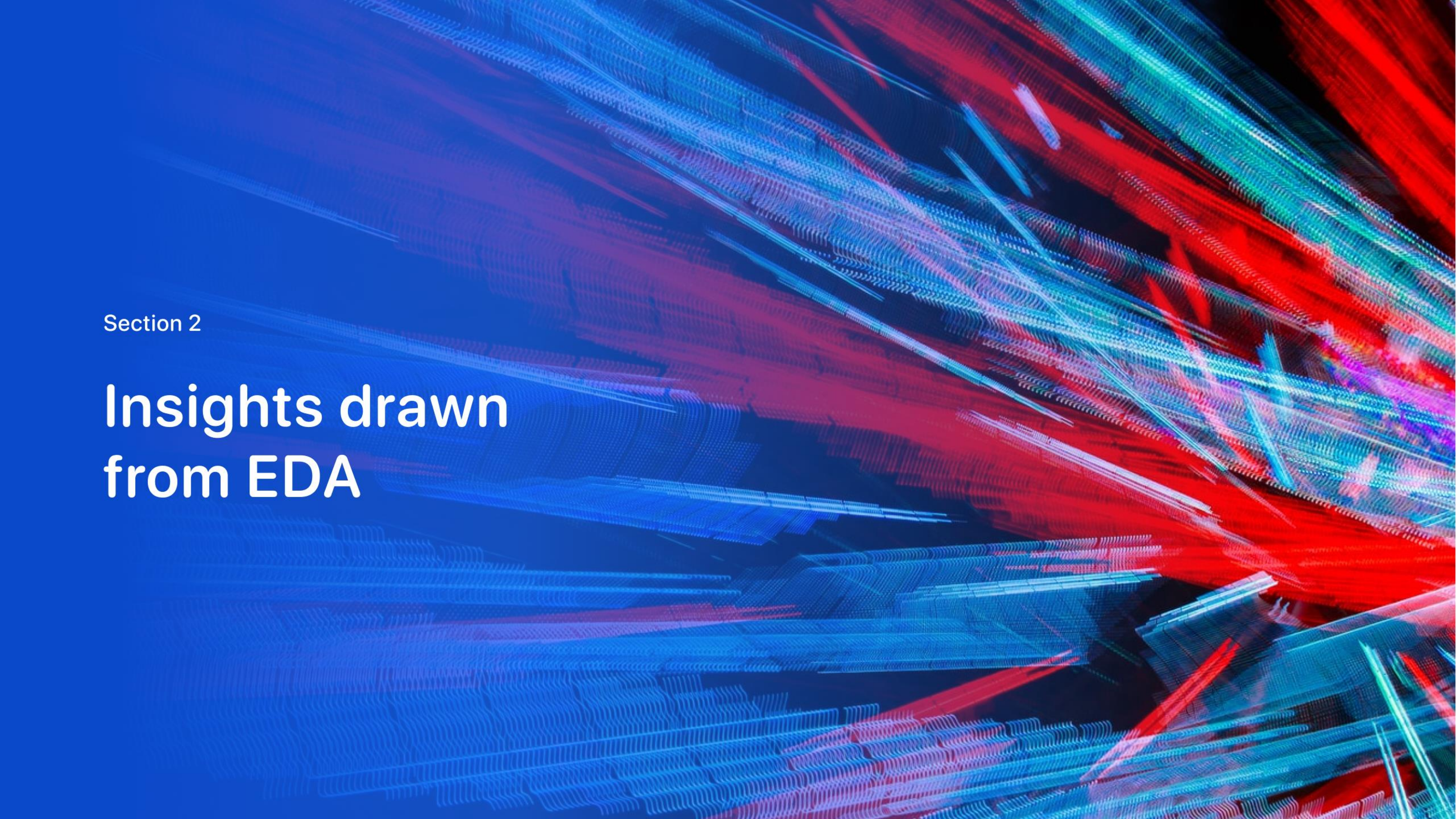
- EDA with visualization
- EDA with SQL

Interactive analytics demo in screenshots:

- Folium
- Dashboard using Plotly Dash.

Predictive analysis results:

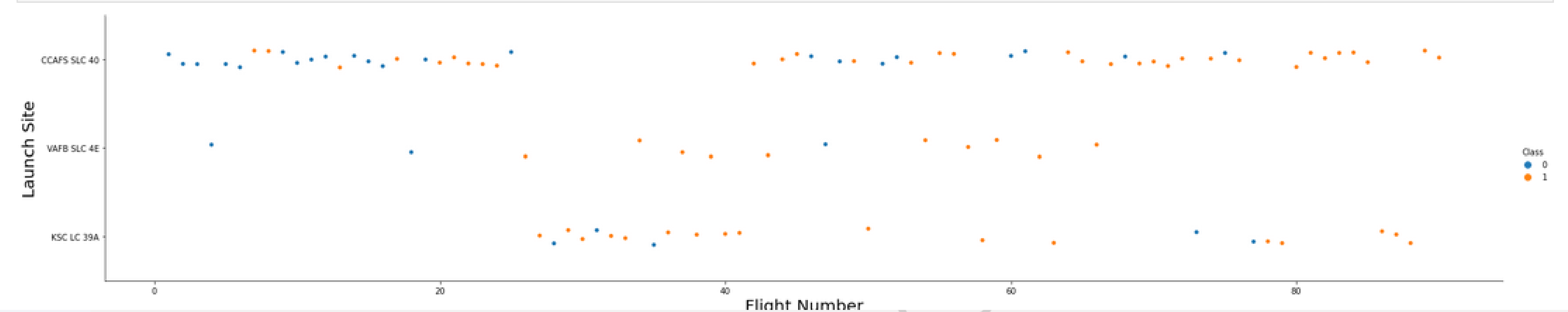
- Predictive Analysis (Classification)



Section 2

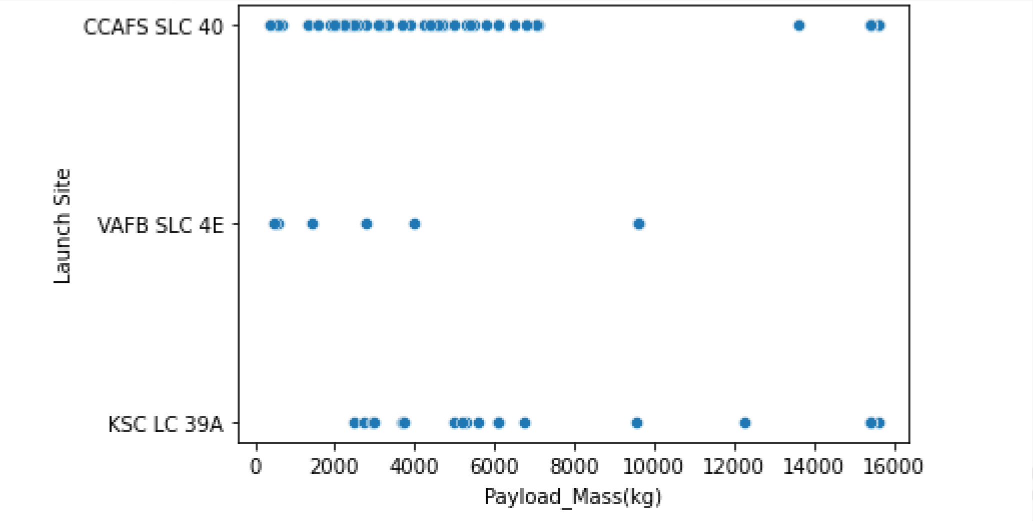
Insights drawn from EDA

Flight Number vs. Launch Site



- Here each dot represents a Flight.
- Each Blue dot represents a failed attempt and each Orange dot represents a successful attempt.
- As we can see the frequency of Blue dots decreased as the number of flights increased.
- Some Launch sites have a better overall Success rate than the others.

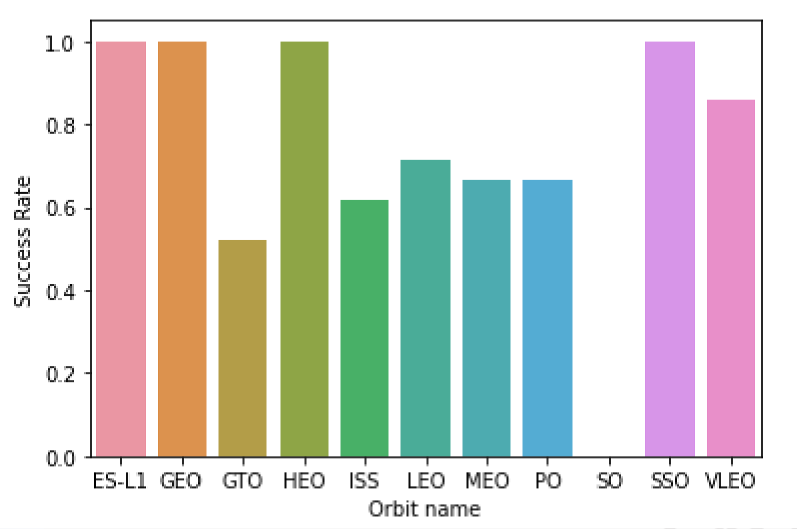
Payload vs. Launch Site



If you observe the relationship between the launch sites and Payload mass you'll observe that:

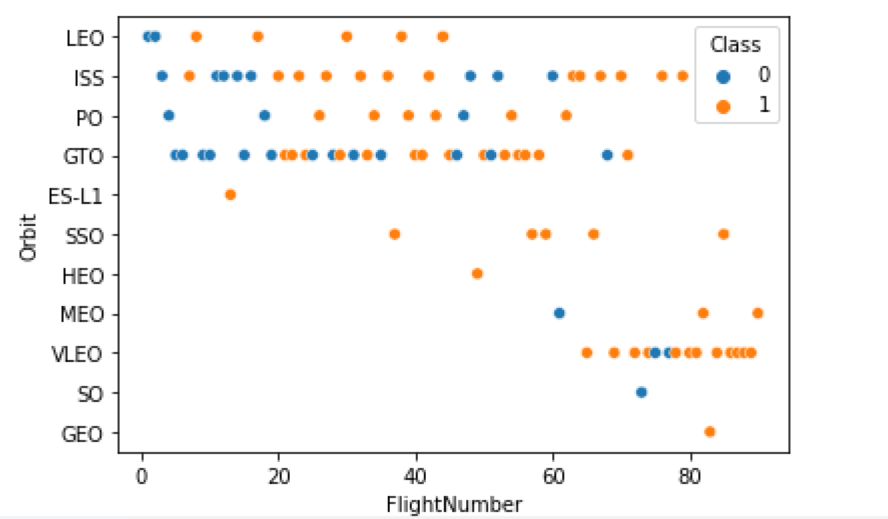
- Most of the flights' payload mass in lies in the range of 2000 kg to 8000 kg.
- Some sites have a higher range of frequency of lighter loads relative others.
- VAFB SLC 4E isn't used for payloads heavier than 10000 kg.

Success Rate vs. Orbit Type



- If we observe the relationship of different orbit types and their success rate we observe that different orbits have different success rate.
- This could be due to the fact that some orbits have are further away and require more precision in the technology. Therefore increasing the chances of failiure.
- Some orbits with the highest success rate are: ES-l1, SSO, HEO, GEO..
- Orbit with the lowest success rate: GTO.

Flight Number vs. Orbit Type



When we plot the relationship between Flight number and orbits in a scatter plot we get to observe the following.

- As the number of flights increase success rate has also risen, but this is not entirely true in each orbit's case.

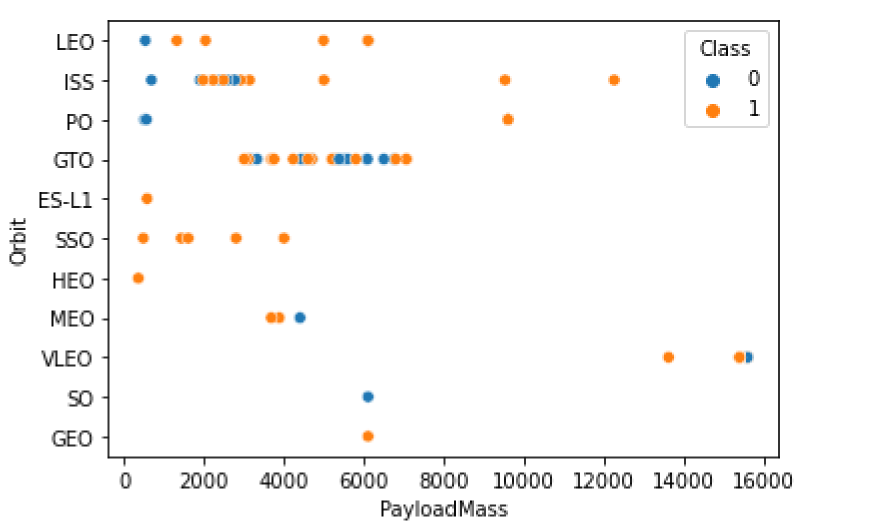
Eg: GTO orbit's success rate hasn't risen with the number of flights.

- Some orbits have a better success rate than the others.

Payload vs. Orbit Type

If we plot the relationship of Payload mass with different orbits we get to see that:

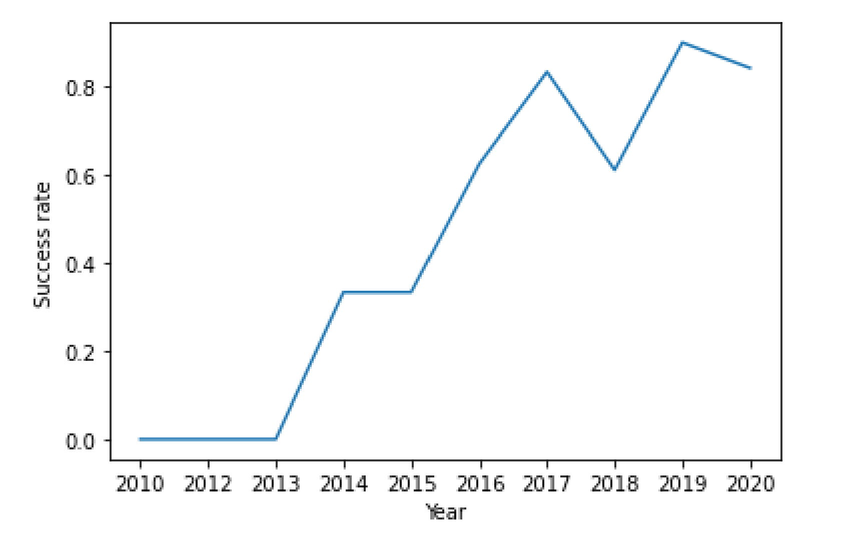
- Heavier payloads have a better success rate in some orbits. Eg: ISS, PO, LEO etc..
- There's no correlation of higher mass and better success rate in the GTO orbit.



Launch Success Yearly Trend

If we plot the data of date of the launch and successful landing. We get to observe the following.

- There has been an increase in the success rate of the landings.
- The probability of success rate rose to 0.4 in 2014 to 0.8 in 2017.
- There has been a slight decrease in 2018 but in 2019 the success rate is around 0.9.

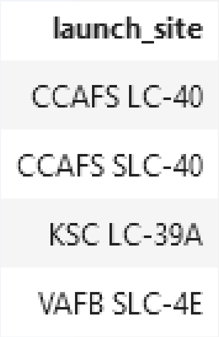


All Launch Site Names

Names of the unique launch sites:

```
Query: %sql select unique(launch_site) from spacextbl;
```

Present your query result with a short explanation here:

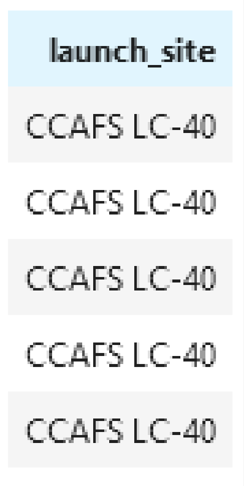


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

The unique() command allows you to find all the unique observations in a particular column of the table. This help when you only require the name of the different categories, irrespective of their the number of times they occurrence in the table.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- **Query: %sql select launch_site from spacextbl where (launch_site) like 'CCA%' limit 5;**
- Present your query result with a short explanation here:



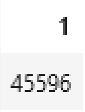
launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Here we used **like** operator, which allows one to search the database using ‘**%**’ sign. If you put ‘**%**’ it after a character of words eg: ‘**abc%**’, the like operator will start looking for observations that start with the letters ‘abc’. Similarly you can put it in front of a set of strings to find observations that end the same way. It can also be used to find the middle of the strings.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA.
- **Query: %sql select sum (payload_mass__kg_) from spacextbl where customer= 'NASA (CRS)';**
- Present your query result with a short explanation here:

Result of the query:

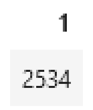


1
45596

Here we used multiple queries, like ‘**sum()**’ and ‘where’. The **sum()**, as the name suggest is used to add integer or float values. **where**, is used to put multiple conditions in a single query to obtain better results. Here ‘45596’ is the sum of payload mass of NASA (CRS).

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- **Query: `%sql select avg (payload_mass__kg_) from spacextbl where booster_version like 'F9 v1.1%';`**
- Present your query result with a short explanation here
- Result of the query:

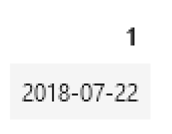


1
2534

Here we used **avg()** to find the mean of the observations we are working with.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- **Query: %sql select min(DATE) from spacextbl where landing__outcome = 'Success’;**
- Present your query result with a short explanation here
- Result: 2018-07-22

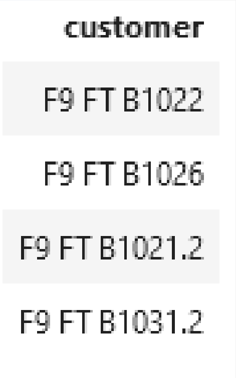


1
2018-07-22

Here we used the **min()** command to find the first date of successful landing.

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
- **Query: %sql select booster_version customer from spacextbl where landing__outcome='Success (drone ship)' and payload_mass__kg_ between '4000' and '6000’;**
- Present your query result with a short explanation here

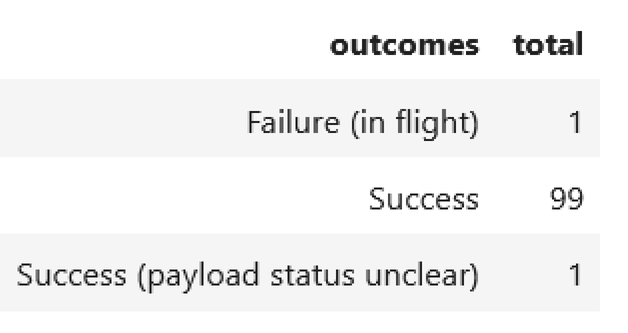


customer
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

As we can observe here we used the **between** query. This allows one to find the observations within certain range. Here the range was payload mass between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- **Query: `%sql select mission_outcome as Outcomes, count(mission_outcome) as Total from spacextbl group by mission_outcome;`**
- Present your query result with a short explanation here:



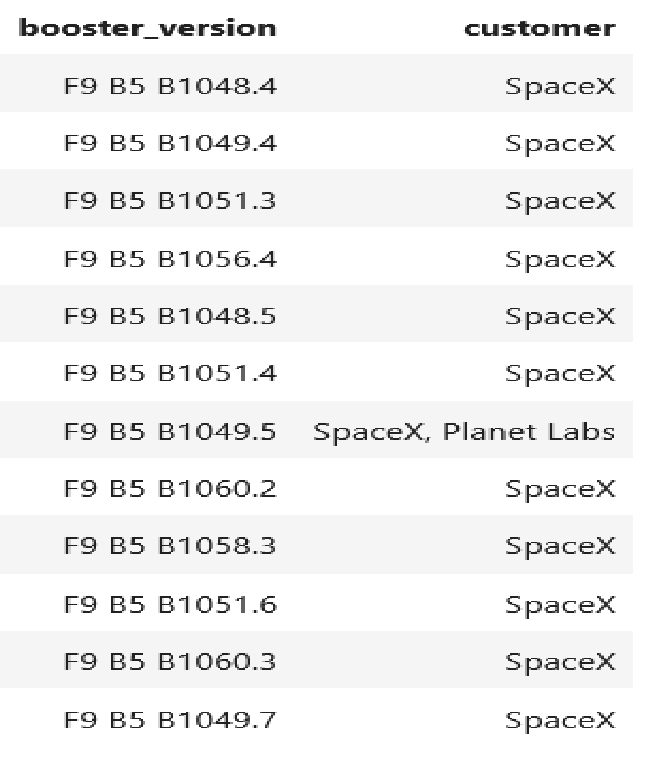
outcomes	total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Here combination of commands were used to find/build a specific result. Here we creating a column in result using **‘as’**. We also used **group by** to see the number of success and failed attempts.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass.
- **Query: `%sql select booster_version, customer from spacextbl where payload_mass__kg_=( select max (payload_mass__kg_) from spacextbl);`**
- Present your query result with a short explanation here:

Explanation: here we used a query within a query to specify the order of the result table. Its also known as nested queries.



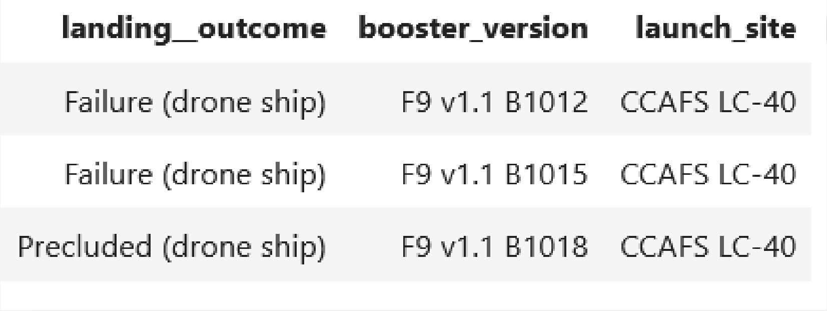
booster_version	customer
F9 B5 B1048.4	SpaceX
F9 B5 B1049.4	SpaceX
F9 B5 B1051.3	SpaceX
F9 B5 B1056.4	SpaceX
F9 B5 B1048.5	SpaceX
F9 B5 B1051.4	SpaceX
F9 B5 B1049.5	SpaceX, Planet Labs
F9 B5 B1060.2	SpaceX
F9 B5 B1058.3	SpaceX
F9 B5 B1051.6	SpaceX
F9 B5 B1060.3	SpaceX
F9 B5 B1049.7	SpaceX

2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.

- **Query:**
```
%sql select landing__outcome, booster_version, launch_site
from spacextbl where DATE like'2015%' and landing__outcome like
'%(drone ship)%’;
```

- Present your query result with a short explanation here

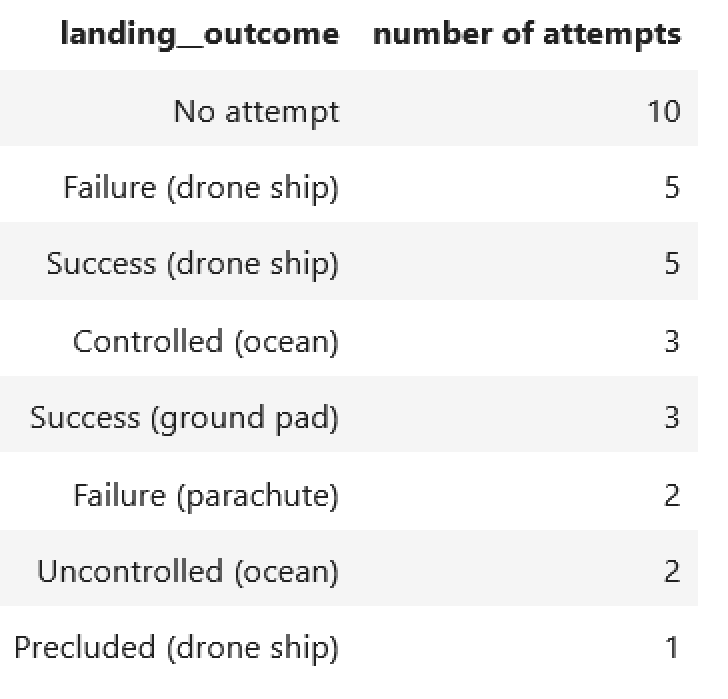


landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

Here we used **and** which allowed us to put multiple conditions like ‘**where’** and ‘**like**’.

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.
- **Query: %sql select landing__outcome, count(landing__outcome) as "number of attempts" from spacextbl where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by count(landing__outcome) desc;**
- Present your query result with a short explanation here
- Here we used the combination of several queries explained in previous slides, this allowed us to make a custom table listing each landing outcome and its frequency.

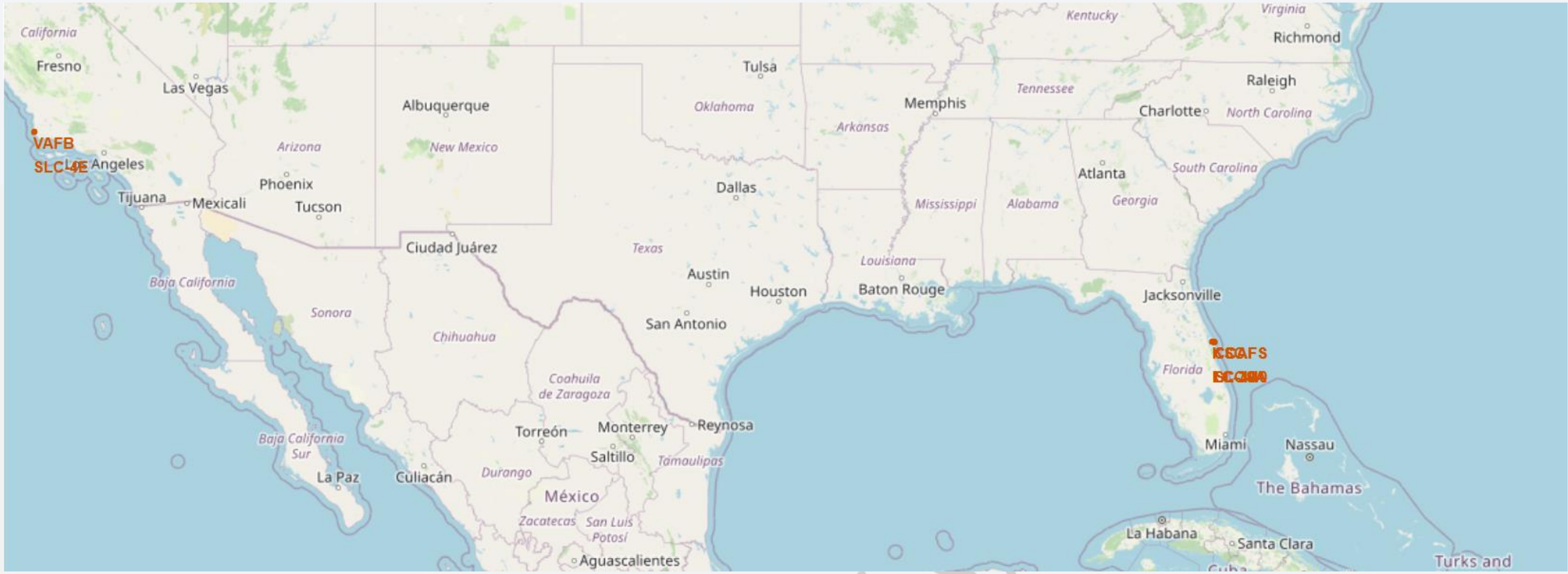


landing__outcome	number of attempts
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

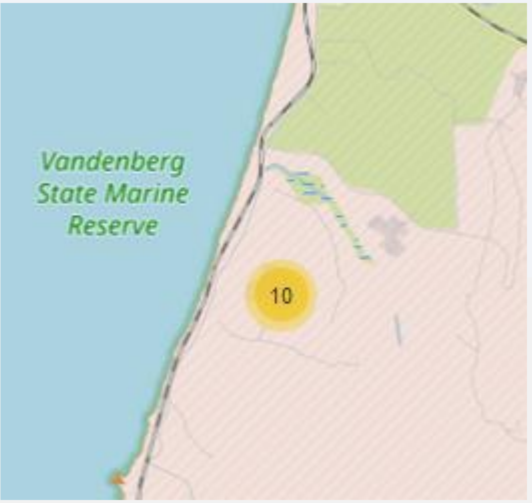
Launch Sites Proximities Analysis

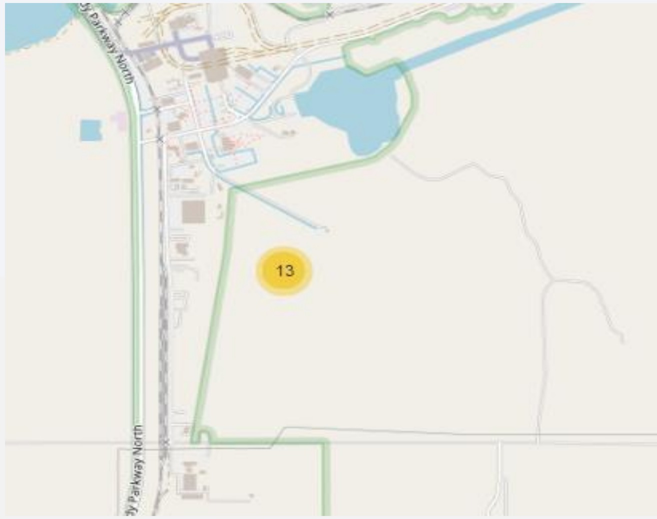
Launch Sites

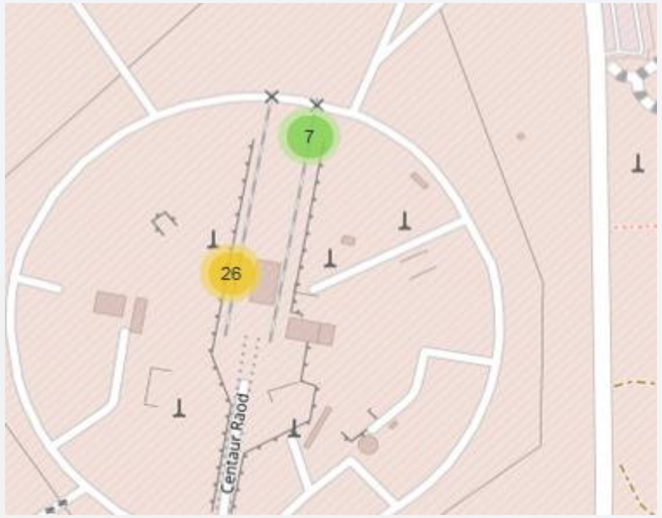


- Overall there are four launch sites, three of which are on the east coast, Florida. Fourth one is in the state of California, west coast.

Launch Clusters and Outcomes

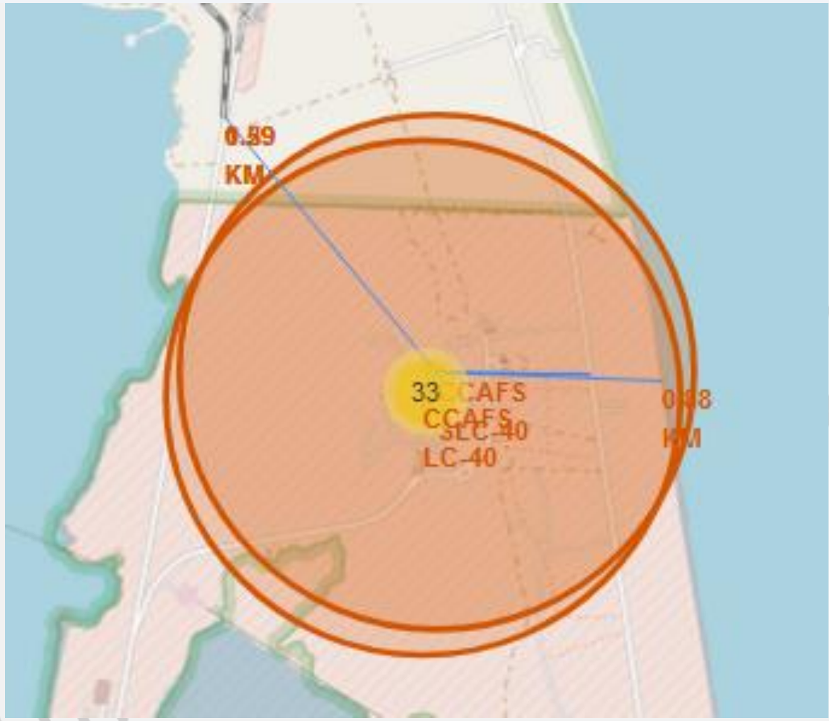






- There have been 10 Launches from California and 46 from Florida

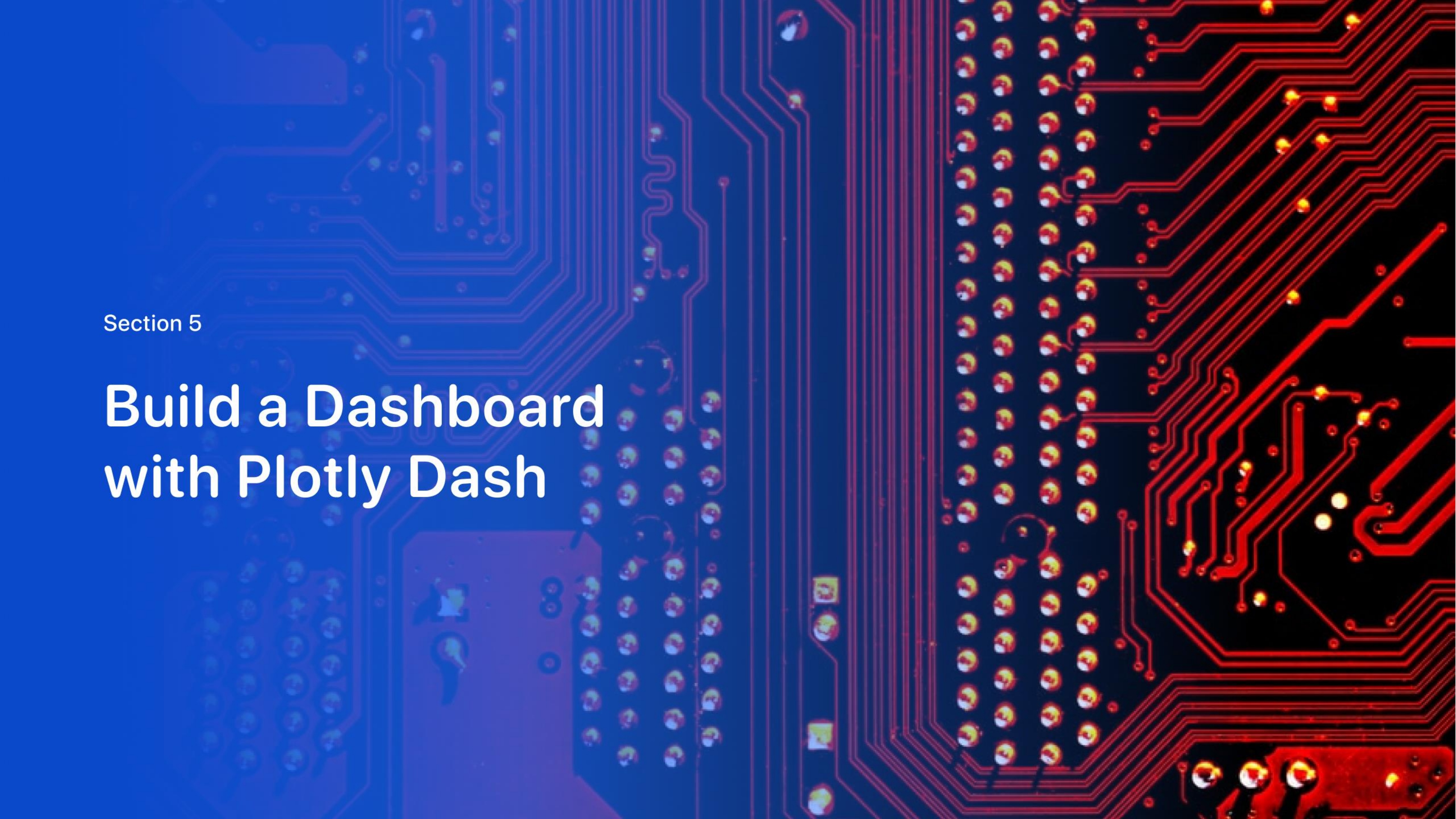
Launch site and its distance from proximities.



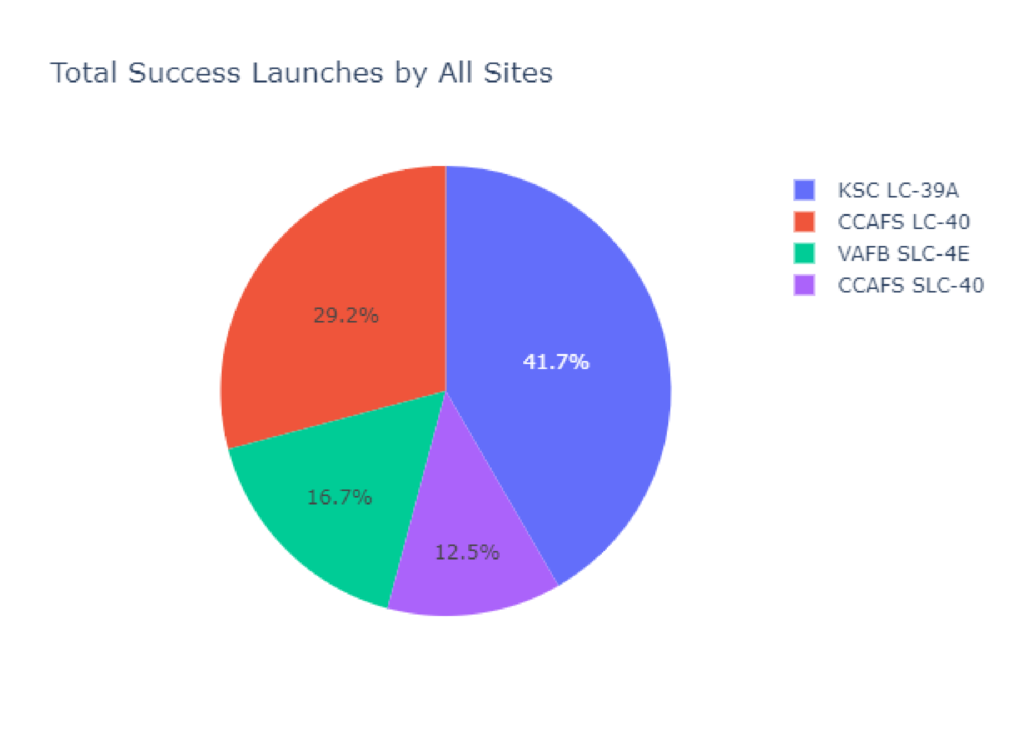
- As we can see there's a Highway nearby the location, a Railway line 0.89 Km away from the launch site and a coast 0.88 km away.

Section 5

Build a Dashboard with Plotly Dash

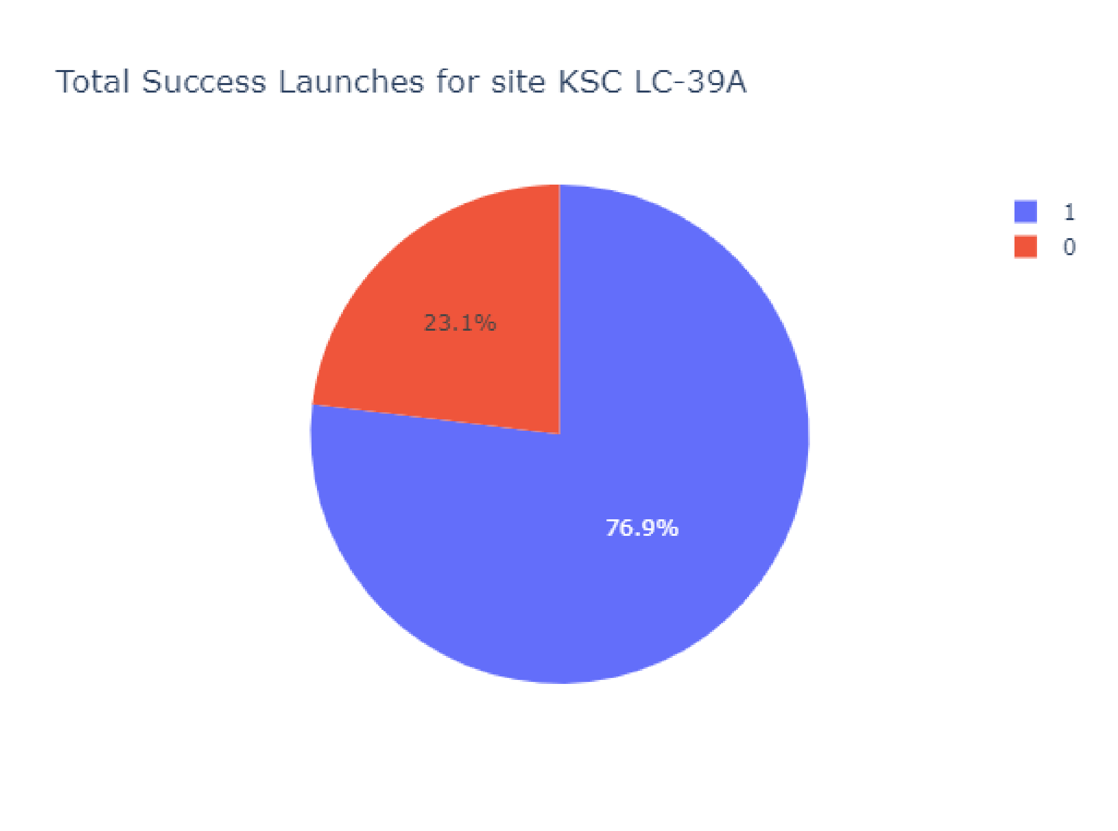


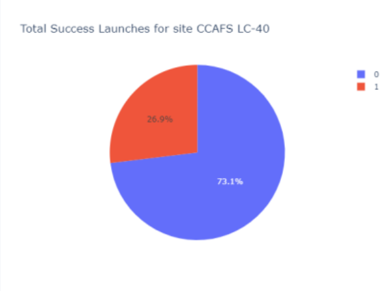
Total Successful Launches by All sites

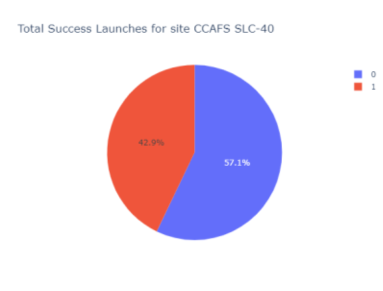


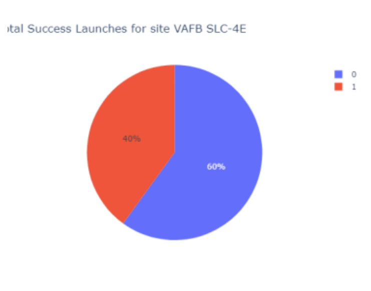
- As we can observe, 41.7% of the successful launches come from KSC LC-39A.
- CCAFS LC-40 contributed 29.2% to successful attempts. While VAFC SLC-4E contributed only 16.7% and CCAFS SLC-40 only 12.5%.

Success Rate percentage



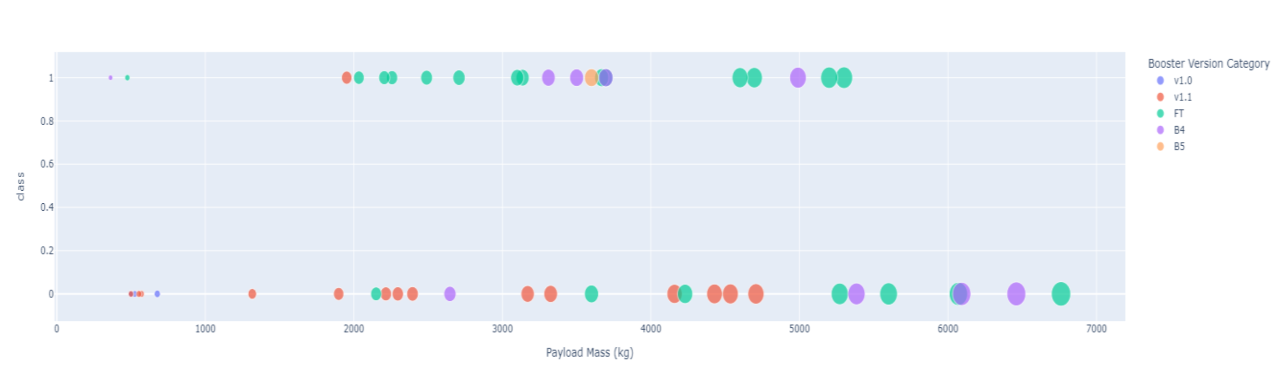


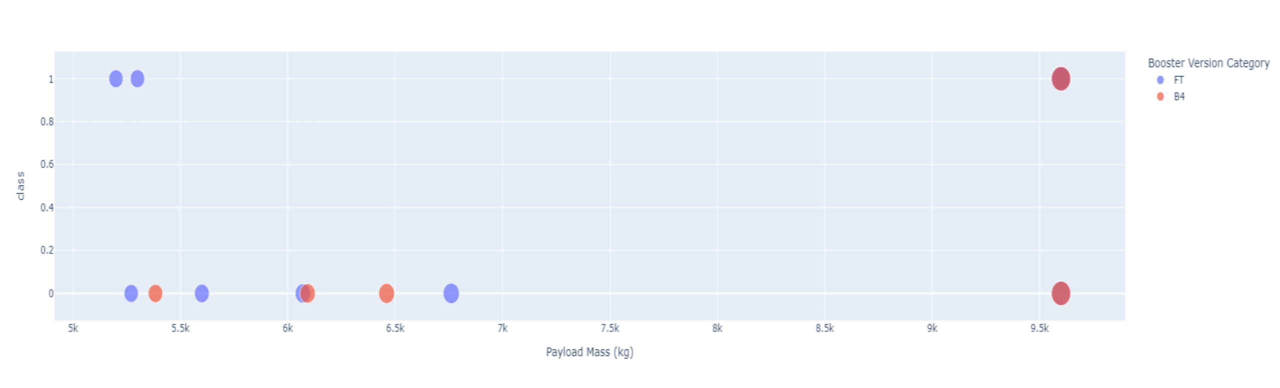




- From the Pie-chart of KSC LC 39A we can observe that the site has a success rate of 76.9%.
- KSC LC-39A has the highest success rate out of the four followed by CCAFS LC-40 at 73.1%, VAFB SLC-4E at 60% and CCAFS SLC-40 at 57.1%.
- (zoom in to view the breakdown of launches from CCAFS LC-40, VAFB SLC-4E, CCAFS SLC-40)

Payload Mass and Launch Outcome





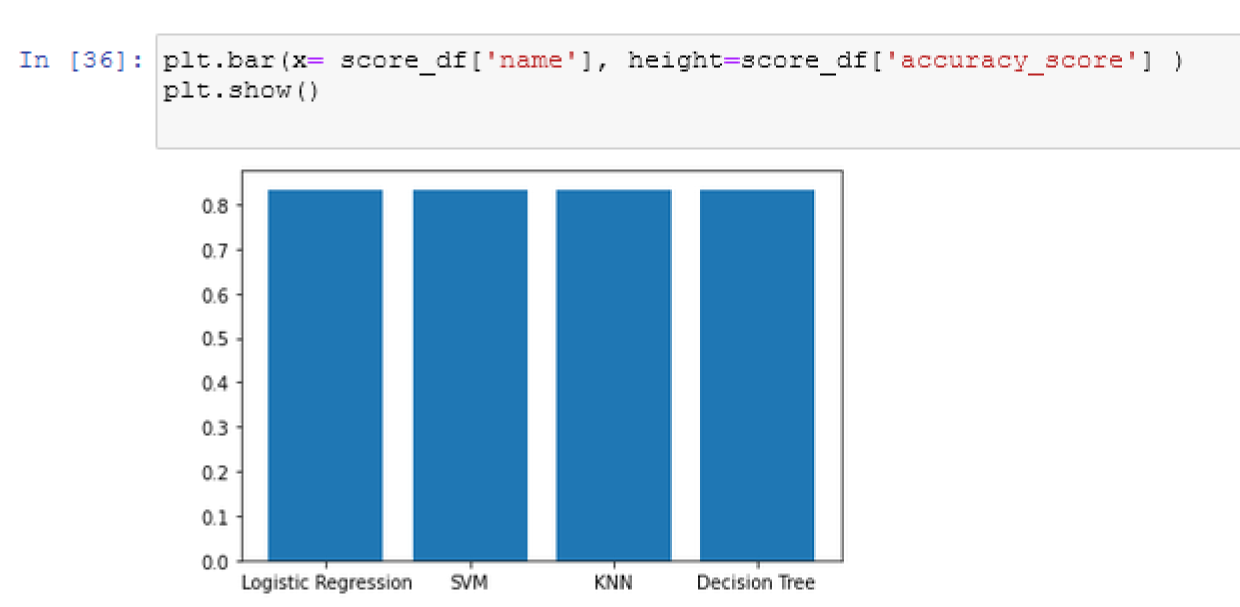
- Launches with lighter payload mass are more frequent.
- Most of the launches have payload mass in the range of 2000kg to 8000kg.
- Launches with the largest payload mass are of 9600kg.

Section 6

Predictive Analysis (Classification)

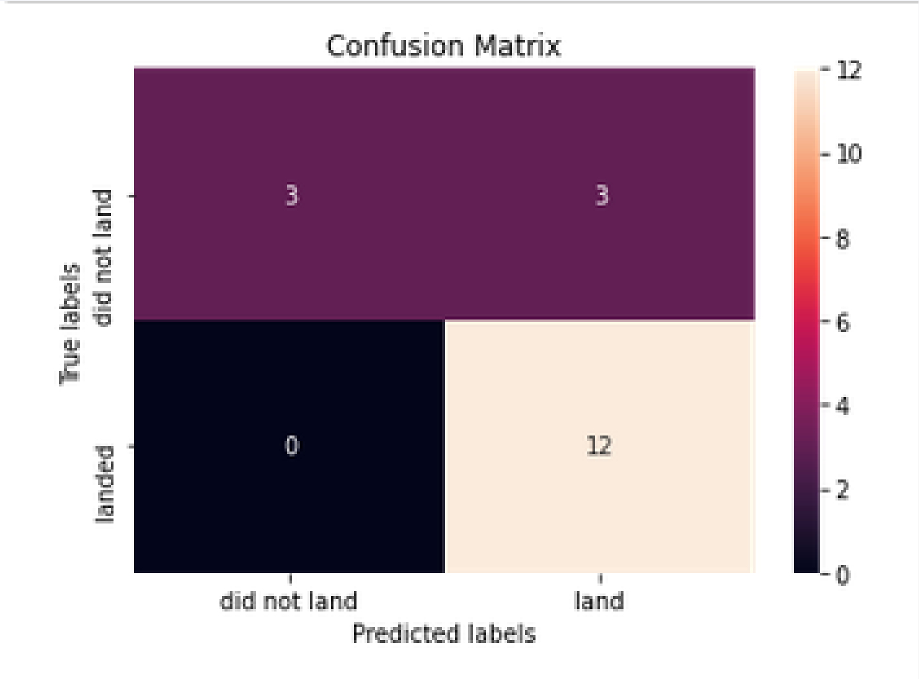
Classification Accuracy

All the plots have similar score so there’s no one model with better accuracy score.



```
In [36]: plt.bar(x= score_df['name'], height=score_df['accuracy_score'] )
         plt.show()
```

Confusion Matrix



- As we can see the best model predicted the following:
- It successful predicted the all actual landing labels.
- There was some error when it predicted that the first stage would land but according to the true labels it did not.
- It had zero error in labelling did not land as landed.

Conclusions

- Factors such as Payload Mass, Orbit Type, and Launch site showed effects on the outcome.
- Decreasing Payload mass has a positive impact on the success rate.
- Launch Site KSC LC-39A has the highest success rate among other launch sites.
- The mean accuracy score of models was: 0.833333
- Best parameters varied from model to model.

Thank you!