IBM Data Science Capstone Project

NALAN BAYRAKTAR

Outline

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

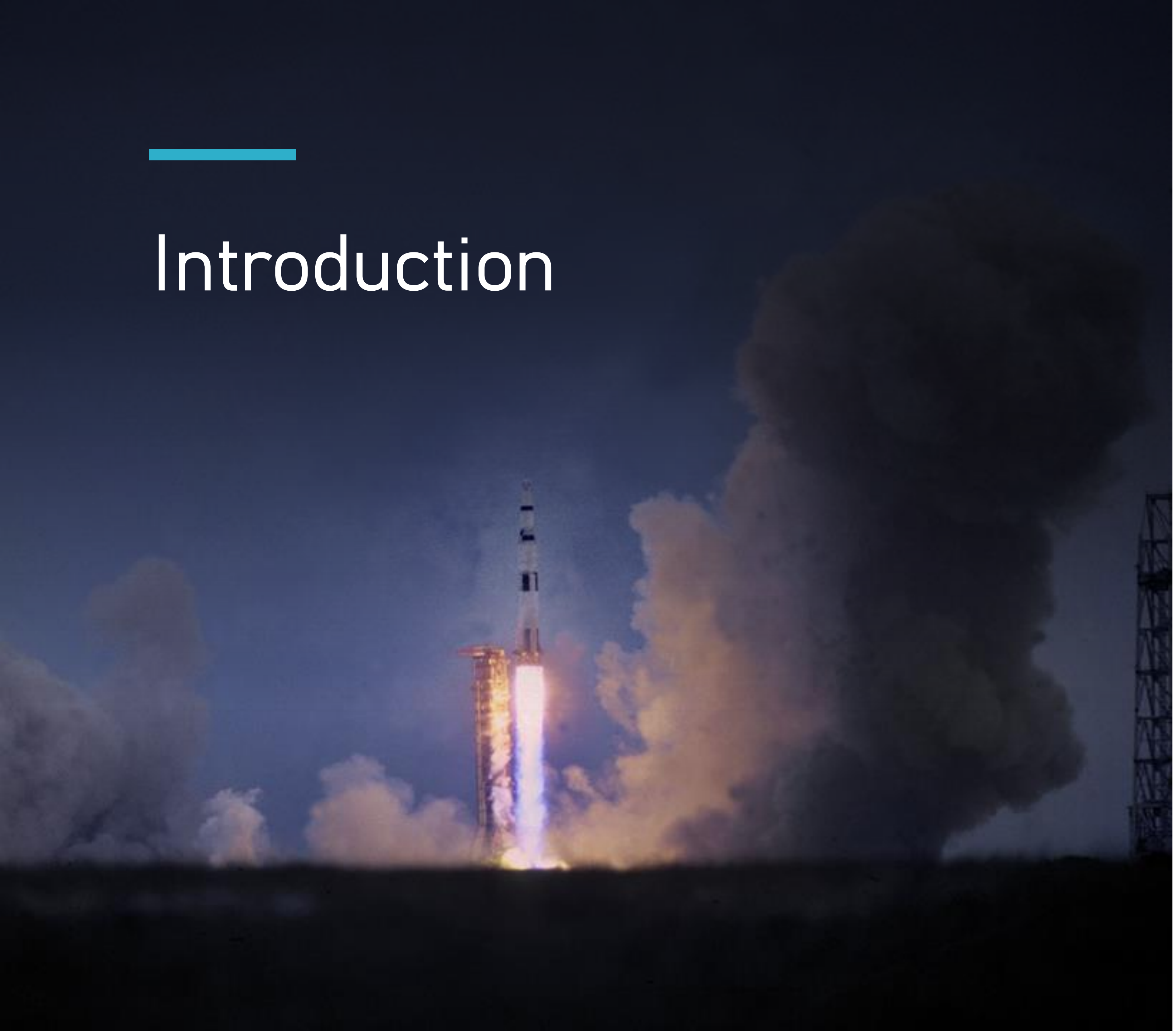
Executive Summary

- **Summary of Methodologies**

1. Data Collection through API
2. Data Collection with Web Scraping
3. Data Wrangling
4. Exploratory Data Analysis with SQL
5. Exploratory Data Analysis with Data Visualization
6. Interactive Visual Analytics with Folium
7. Machine Learning Prediction

- **Summary of All results**

1. Exploratory Data Analysis Results
2. Interactive analytics demo in screenshots
3. Predictive analysis results



Introduction

- ## Project Background and context

Commercial space age is here, companies are making space travel affordable for everyone. SpaceX has gained worldwide attention for a series of historic milestones. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.

- ## Problems you want to find answers

What factors determine if the rocket will land successfully?

Relationship between various features to determine the success rate of a successful landing.

What conditions need to be ready on SpaceX side in order to get best results?

Methodology

Data Collection

- Data collection was done getting a request to the SpaceX API.
- Next, we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize().
- In addition, we performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches. Our objective was to extract a Falcon 9 launch records HTML table from Wikipedia and parse the table and convert it into a Pandas data frame. Webscraping was done with BeautifulSoup.

Data Collection – SpaceX API

- We used get request to the SpaceX API to collect data and clean the requested data.

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

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

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

we can apply the rest of the functions here:

In [18]:
```
# Call getLaunchSite
getLaunchSite(data)
```

In [19]:
```
# Call getPayloadData
getPayloadData(data)
```

In [20]:
```
# Call getCoreData
getCoreData(data)
```

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

In [29]:
```
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}
```

Task 2: Filter the dataframe to only include `Falcon 9` launches

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

In [33]:
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=df[df["BoosterVersion"]!='Falcon 1']
```

In [35]:
```
data_falcon9
```

URL:
[Data Collection API]

Data Collection – WebScraping

- We performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled **List of Falcon 9 and Falcon Heavy launches.**

In [6]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [8]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data  = requests.get(static_url).text
```

In [9]:
```
soup = BeautifulSoup(data,"html.parser")
```

In [11]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

In [38]:
```
column_names = []
element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
            pass
```

In [41]:
```
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']=[]
```

URL:

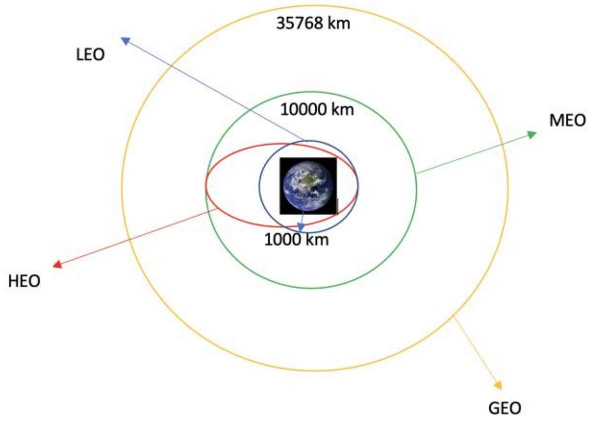
[Data Collection with Web Scraping]

Data Wrangling

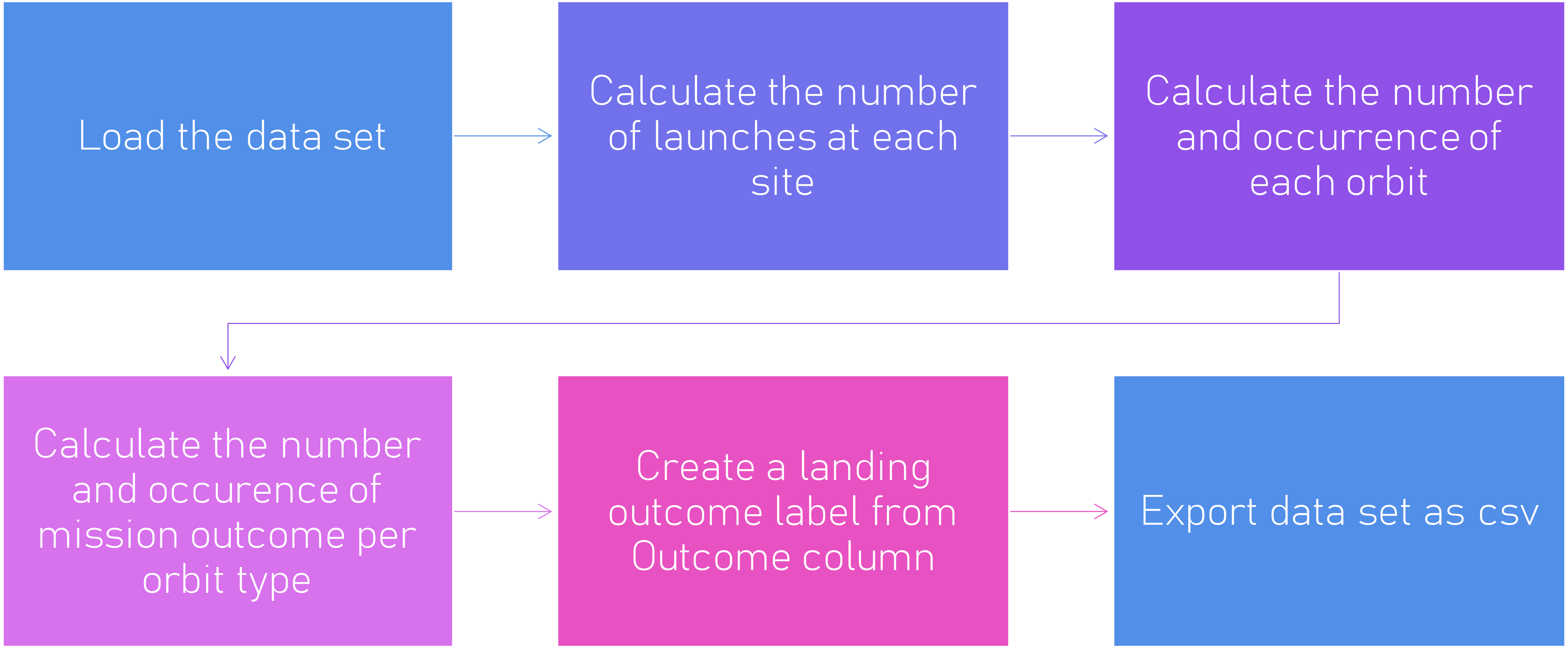
We performend some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

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.



Data Wrangling - Process



URL:

[Data Wrangling]

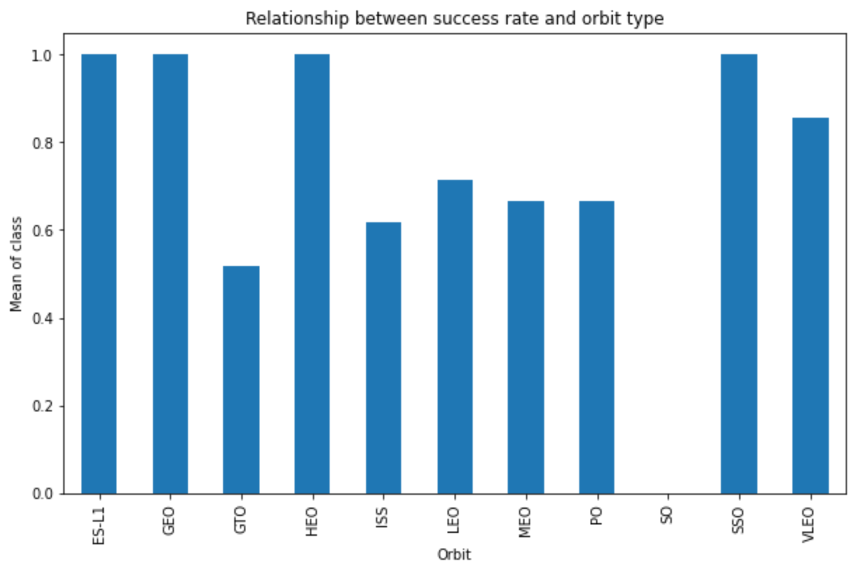
Exploratory Data Analysis with Data Visualization

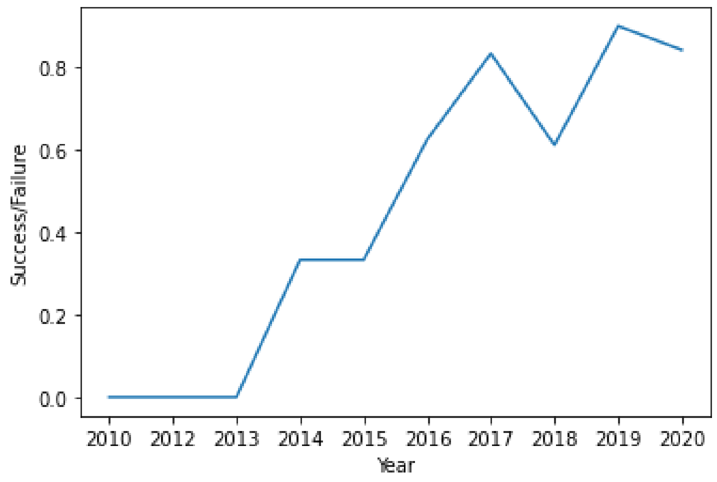
We explored the data by visualizing the relationship between:

- FlightNumber vs PayloadMass
- Flight Number vs Launch Site
- Payload and Launch Site
- Success rate of each orbit type
- FlightNumber and Orbit type
- Payload and Orbit type

URL:

EDA with Visualization





Exploratory Data Analysis with SQL

We performed SQL queries in order to extract meaningful answers to guide the modeling process. We loaded the SpaceX dataset into a PostgreSQL database. We found answers for below questions through SQL queries.

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was acheived.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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.

URL:

EDA with SQL

Build an Interactive Map with Folium

The launch success rate may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

We added each site's location on a map using site's latitude and longitude coordinates and added each a Circle marker around each launch site with a label of the name of the launch site.

After we plot distance lines to the proximities, you can answer the following questions easily:

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?

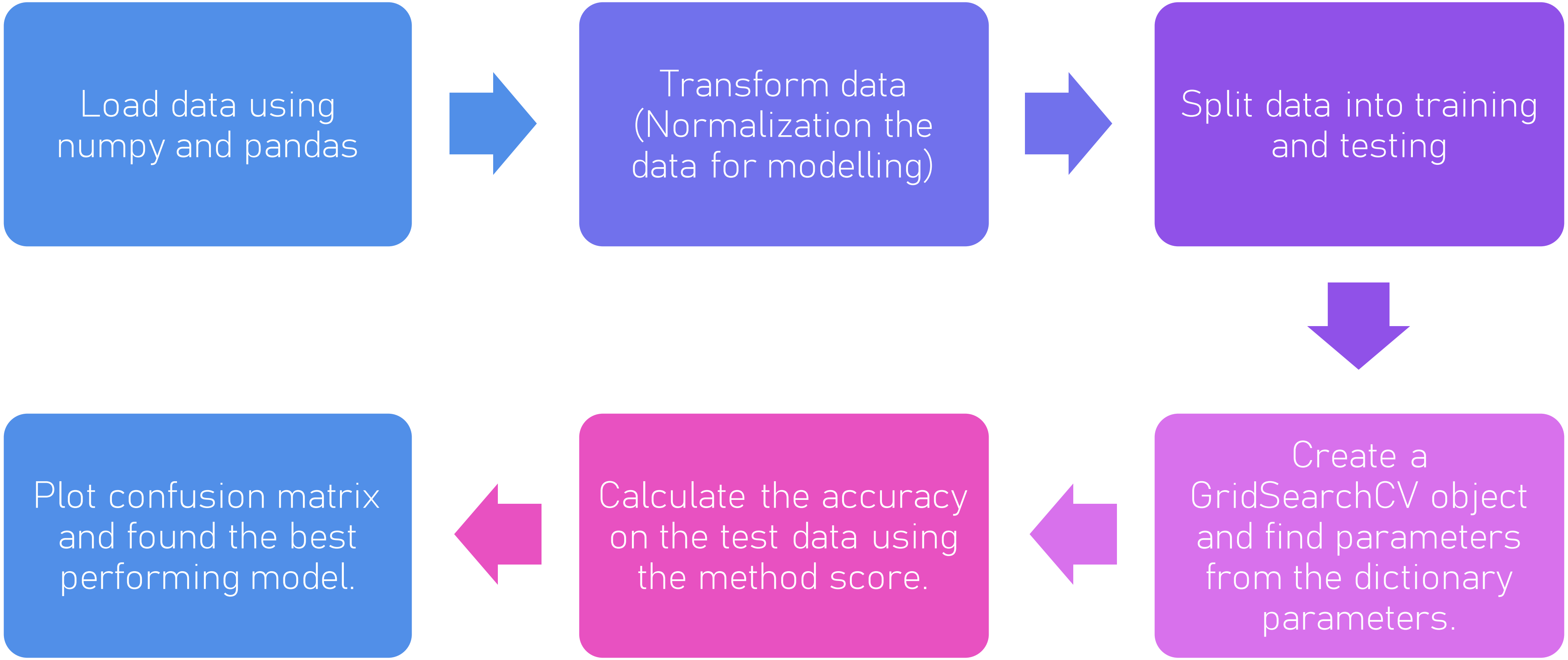
URL:

Data Visualization with Folium

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

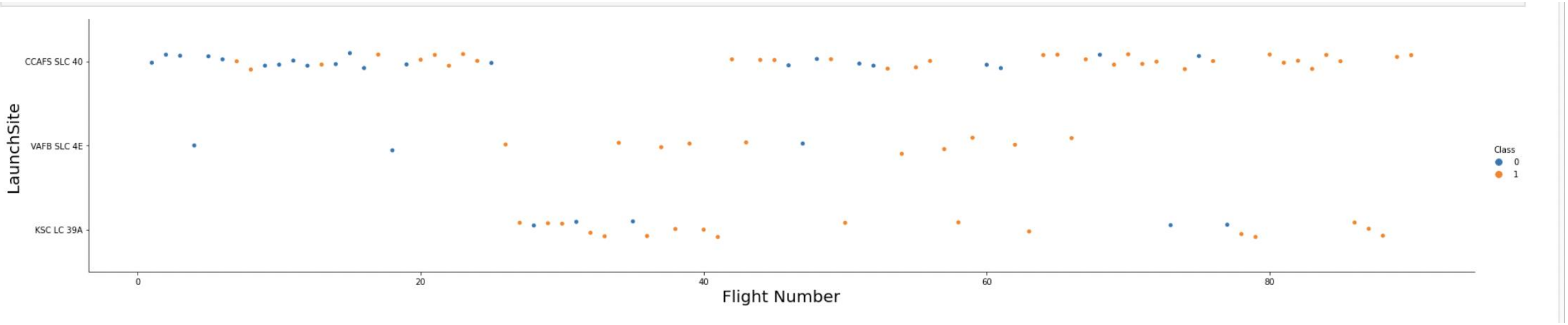
Predictive Analysis (Classification)



Results

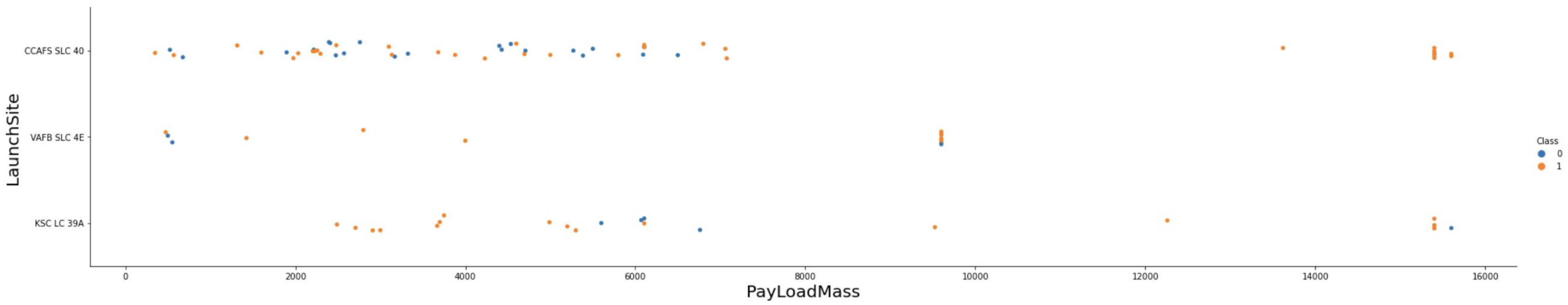
Results
Insights drawn from EDA visualization

Flight Number vs. Launch Site



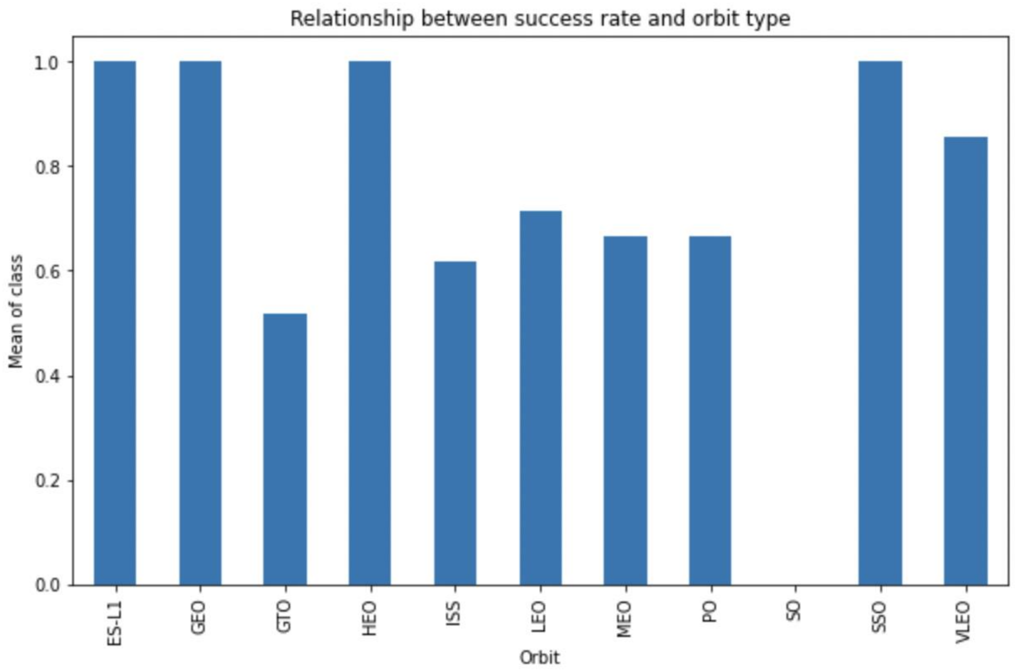
- From the plot we saw that, the more amount flights realized the greater success rate at a launch site.

Payload vs. Launch Site



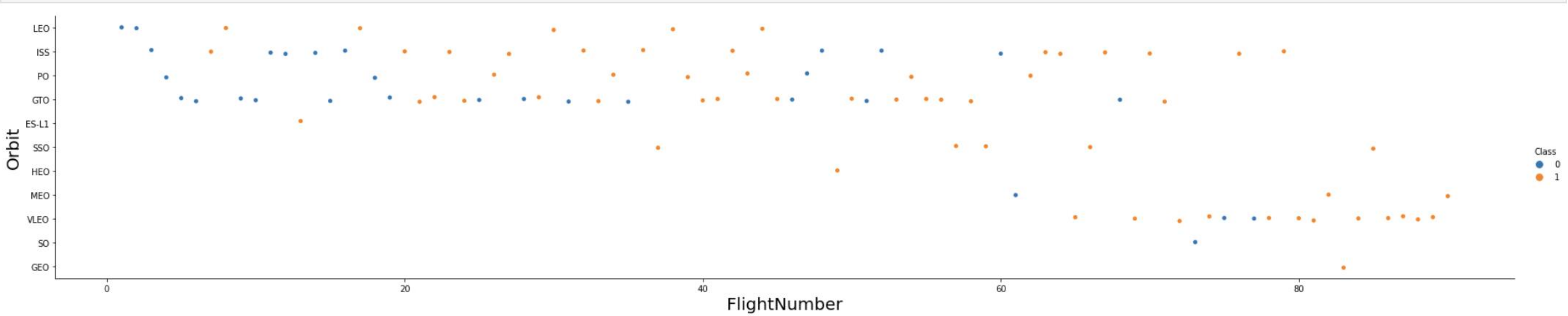
- From the plot, we observed that here are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



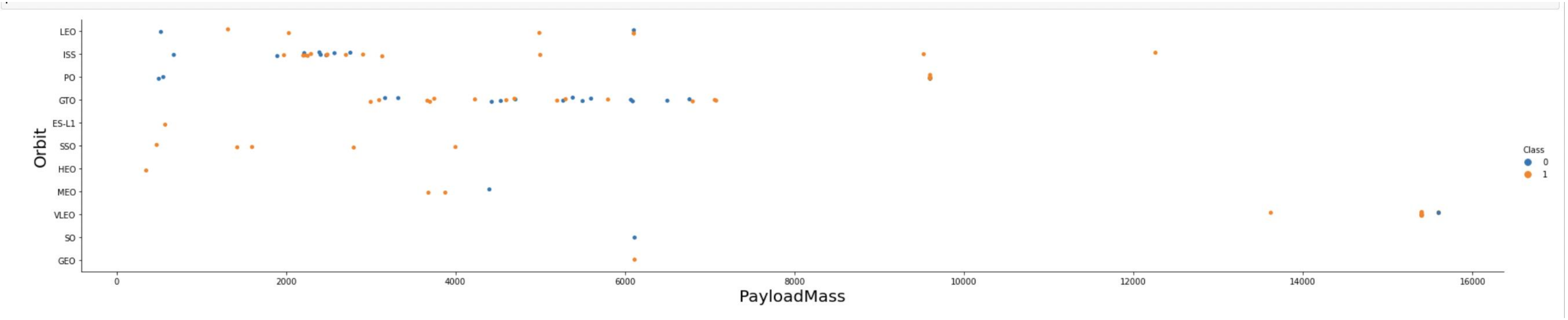
- Orbit ESL-1, GEO, HEO and SSO has the highest success rate.

Flight Number vs. Orbit Type



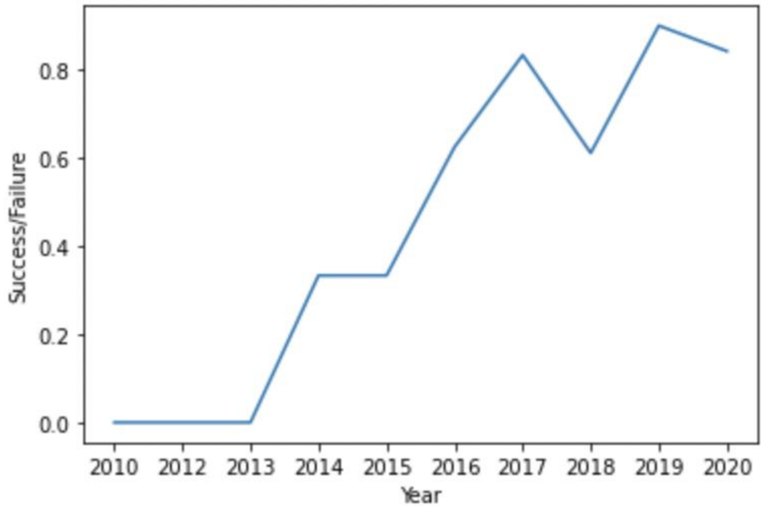
- We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- 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 observe that the sucess rate since 2013 kept increasing till 2020.

Results
Insights drawn from EDA with SQL

All Launch Site Names

Display the names of the unique launch sites in the space mission

```
]: %sql select DISTINCT(launch_site) from spacextbl
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

]:

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

- DISTINCT show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from spacextbl  where launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	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

- We use LIMIT for certain number of records. The **`LIKE`** operator is used in a **`WHERE`** clause to search for a specified pattern in a column. Finds any values that start with "CCA"

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(payload_mass__kg_) AS total_payload_mass from spacextbl where customer='NASA (CRS)'
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

total_payload_mass
45596

- SUM function calculates total payload_mass_kg column. WHERE clause to search for a specified customer which is "NASA (CRS)".

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(payload_mass__kg_) AS avg_payload_mass from spacextbl where booster_version='F9 v1.1'
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

avg_payload_mass
2928

- AVG function calculates average payload_mass_kg column. WHERE clause filter for a specified booster_version which is "F9 v1.1".

First Successful Ground Landing Date

```
%sql select MIN(DATE) from spacextbl where landing__outcome='Success (ground pad)'
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

1
2015-12-22

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [60]: %sql select booster_version from spacextbl where landing__outcome='Success (drone ship)' and payload_mass__kg_>4000 and payload_mass__kg_<6000

 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[60]:

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

- WHERE clause filter landing outcome of Success (drone ship) and payload mass between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [28]:
```
%sql SELECT count(mission_outcome) as SuccessOutcome from spacextbl where mission_outcome LIKE 'Success%'
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[28]:

successoutcome
100

In [30]:
```
%sql SELECT count(mission_outcome) as FailureOutcome from spacextbl where mission_outcome LIKE 'Failure%'
```

```
 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[30]:

failureoutcome
1

- We observe that out of 101 mission outcomes 100 outcomes is successful and only 1 outcome resulted as failure.

Boosters Carried Maximum Payload

```
In [31]: %sql select booster_version from spacextbl  where payload_mass__kg_=(select max(payload_mass__kg_) from spacextbl)

 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[31]:

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

- We listed the names of the booster_versions which have carried the maximum payload mass. We used subquery for maximum payload_mass_kg

2015 Launch Records

```
In [76]: %sql select landing__outcome, booster_version,launch_site from spacextbl WHERE landing__outcome='Failure (drone ship)' and year(DATE)=2015

 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[76]:

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

- We listed the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015. In order to do this we used WHERE clause for filtering.

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

```
In [94]: %sql Select * from spacextbl where DATE BETWEEN '2010-06-04' AND '2017-03-20' AND landing__outcome IN ('Failure (drone ship)','Success (drone ship)')

 * ibm_db_sa://sdq41011:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

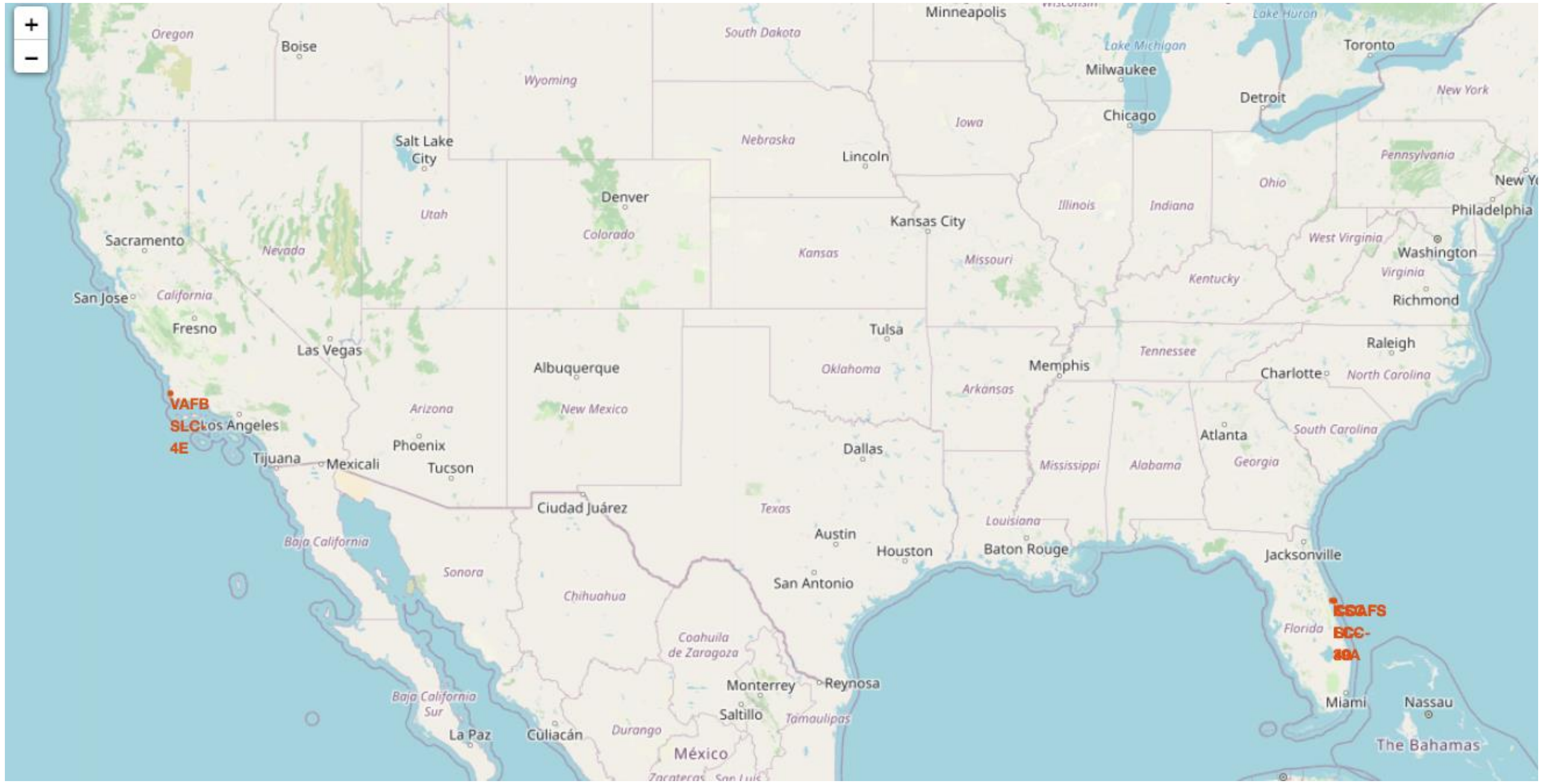
Out[94]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)

- We 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.

Results
Launch Sites Proximities Analysis

Folium Map



- SpaceX launch sites are located in United States of America coasts; Florida and California.

Results
Predictive Analysis (Classification)

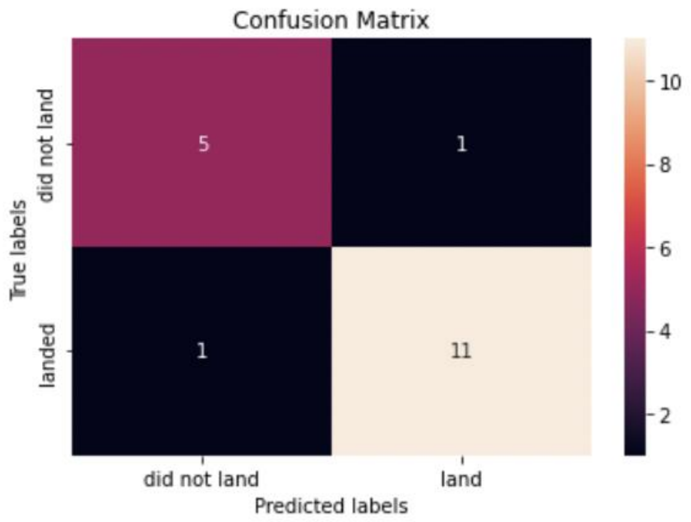
Classification Accuracy

```
In [32]: models = {'KNeighbors':knn_cv.best_score_,
                   'DecisionTree':tree_cv.best_score_,
                   'LogisticRegression':logreg_cv.best_score_,
                   'SupportVector':svm_cv.best_score_}
```

```
In [39]: bestalgorithm = max(models,key=models.get)

         print('Best model is', bestalgorithm,'with a score of',models[bestalgorithm])

         if bestalgorithm=='DecisionTree':
             print('Best params is:', tree_cv.best_params_)
         if bestalgorithm=='KNeighbors':
             print('Best params is:', knn_cv.best_params_)
         if bestalgorithm=='LogisticRegression':
             print('Best params is:' ,logreg_cv.best_params_)
         if bestalgorithm=='SupportVector':
             print('Best params is:' ,svm_cv.best_params_)

         Best model is DecisionTree with a score of 0.875
         Best params is: {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}
```

- According the model, the decision tree classifier with the highest classification accuracy.

Confusion Matrix



- In confusion matrix, we see that decision tree can distinguish between different classes.

Conclusion

Conclusion

- The larger the flight amount a launch site, the greate the success rate.
- Decision Tree classifier is the best algorithm with the highest accuracy rate.
- Orbit ESL-1, GEO, HEO and SSO has the highest success rate.
- We observe that the sucess rate since 2013 kept increasing till 2020.

Thank you!