IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nadir Sezgin
4/23/2023

Outline

Executive Summary

- Summary of methodologies
  - Collecting Data through API or Web Scrapping
  - Cleaning Data
  - Expletory Data Analysis (EDA) by Data Visualization
  - EDA using SQL
  - EDA using Folium
  - EDA using Interactive Visualization
  - Predictive Analysis
- Summary of all results
  - Presenting EDA insights
  - Machine Learning insights

Introduction

- Project background and context
  - We analyze the Falcon-9 rocket launches of SpaceX company. SpaceX is the first rocket company to re-use the 1st stage of their rockets to reduce the launch cost. We use the data provided by SpaceX API as well as public data from Wikipedia page about Falcon-9 launches to gain insights. From this data we can determine if the Falcon-9 rocket first stages landed successfully or not and then make prediction if the Falcon-9 rocket first stage will land successfully in the future.
- Problems you want to find answers
  - We would like to answer if the 1st stage of a launched Falcon-9 rocket will land successfully. To be able to answer this question first we collect the data. Then we use data wrangling techniques to put the data in a format that is useful for us to analyze. We use Explanatory Data Analysis (EDA) to gain insight about the data. Finally we use Machine Learning classification models to predict if the 1st stage of a launched Falcon-9 rocket will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data about Falcon-9 launches is obtained from SpaceX API and scrapping Wikipedia web page about the SpaceX rocket launches.
- Perform data wrangling
  - The data obtained using API or web-page scrapping is not in the format we like. Therefore, we discard the information that we do not need and handle the missing values that are needed for our analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We use ML models to make predictions about Falcon-9 rocket usability.

Data Collection

- The Falcon-9 rocket launch data is obtained through the SpaceX REST API as well as using web scrapping of a Wikipedia page about Falcon-9 launches.
- In both cases we use Python. For the REST API we use Python's requests library to obtain data in JSON format. Then we use panda's json_normalize function to convert the data to a dataframe. We then select only the relevant data from the dataframe to construct a dictionary of the futures that we desire. Finally, we clean the data and save it into a csv file for future analysis.
- For the Wikipedia webpage scrapping, we use Python's BeautifulSoup library in addition to the requests library. We parse the requested webpage using html.parser. Again, constructing a dictionary of the futures we desire and then converting the dictionary to a data frame and then saving it in csv format.

Data Collection – SpaceX API

- Import libraries and define auxiliary functions.
- Request and parse SpaceX launch data using GET request.
- Use json_normalize() method to convert JSON result into a dataframe.
- Save desired data into a csv file.

We will import the following libraries into the lab

In [1]:
```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data mc
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, mu
import numpy as np
# Datetime is a library that allows us to represent dates
```

In [2]:
```
# Takes the dataset and uses the rocket column to call the API and append the data to tl
def getBoosterVersion(data):
    for x in data['rocket']:
        if x:
         response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
         BoosterVersion.append(response['name'])
```

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())
```

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20API.ipynb

Data Collection - Scraping

- Import necessary Python libraries and create auxiliary functions.
- Using requests.get() method to scrape Wikipedia webpage.
- Using BeautifulSoup html.parser parser on requested data.
- Obtain necessary data (in table) from the soup object to construct a dataframe with desired columns.
- Save the dataframe as a csv file.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20Web%20Scrape.ipynb

In [1]:
```
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

and we will provide some helper functions for you to process web scraped HTML table

In [2]:
```
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.s

def booster_version(table_cells):
```

In [3]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_ar
```

In [4]:
```
# use requests.get() method with the provided
# assign the response to a object
req = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [5]:
```
# Use BeautifulSoup() to create a BeautifulSou
soup = BeautifulSoup(req, 'html.parser')
```

Data Wrangling

- In this Lab we prepare data obtained in the previous Lab into a desired format.
- Identify percent of missing values in each attribute (column) of data.
- Identify the data types of each column.
- Identify the number of launches in each site.
- Calculate number of occurrence of each orbit.
- Find out mission outcome per orbit type.
- Create landing outcome label from outcome.
- Save dataframe in csv format.

```
df.isnull().sum()/df.count()*100

df.dtypes

# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts()

# Apply value_counts on Orbit column
df.Orbit.value_counts()

# landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()

# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

df.to_csv("dataset_part_2.csv", index=False)
```

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Visualize relationship between **Flight Number and Payload** in kg. We observe that initial flights have low payload, increasing gradually as flight number increases.
- Visualize relationships between **Flight Number and Launch Site**. Launch sites KSC and VAFB have the most success recovering Falcon-9 rockets. The success rate for launch site CCAFS is mixed.
- Visualize relationship between **Payload and Launch Site**. For the very heavy payloads (i.e. greater than 10,000 kg) the launch site VAFB is not used. Also success rate is higher in heavy payloads.
- Visualize relationship between **Orbit and Success Rate**. The recovery success rate for orbits ES-L1, GEO, HEO and SSO are 100%. For other orbits it is between 50% and 85%. Therefore the higher orbit flights are likely to be successful.
- Visualize relationship between **Payload and Orbit type**. Very heavy payloads are delivered to ISS, PO and VLO orbits. The recovery of the rockets for these is very successful.
- Visualize **Success Rate yearly trend**. We observe the success rate is improving throughout the years.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20EDA.ipynb

EDA with SQL

- Queries Performed Using SQL:
  - Display the names of unique launch sites.
  - Display 5 records where launch sites begin with string ‘CCA’
  - Display 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 boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 kg.
  - List 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, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Captone%20SQL.ipynb

Build an Interactive Map with Folium

- We have build an interactive map of Launch Sites using Folium.
- In order to demonstrate the failed and successful recovery of Falcon-9 rockets at a particular site we have build a marker_cluster child object. These object have red color if the recovery is failed and green color if the rocket recovery is successful.
- In order to find out the distance between given coordinates (such as between the launch site and railroad line), we build a distance_marker child object.Using the coordinates of the launch site and a landmark, distance marker shows how far the landmark is away from the launch site.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We have build an interactive dashboard for Falcon-9 launch sites using Plotly Dash.
- By using Dash, we can visualize the successful Falcon-9 rocket recovery rate at each launch site by selecting the site from a selection menu.
- By using Dash, we can also visualize successful and failed recoveries of Falcon-9 rockets according to their payloads during the launch. The payload values can be interactively chosen by a slider to visualize the results dynamically.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20Dash.py

Predictive Analysis (Classification)

- Using the Machine Learning classification models, Logistic Regression, Support Vector Machines (SVM), Decision Tree and K Nearest Neighbors (KNN) we predict if a Falcon-9 rocket can be successfully landed.
- The data is split into training and test sets. Each model is trained on the training set data and then model accuracy is calculated using the data in the test set.
- For each model, the prediction accuracy of Falcon-9 landing outcome and the related confusion matrix are calculated.

https://github.com/nsetgil/IBM-DS-Capstone/blob/c1097ce34cd8474003209e0ea6e574a89567ec7b/IBM%20DS%20Capstone%20ML.ipynb

Results

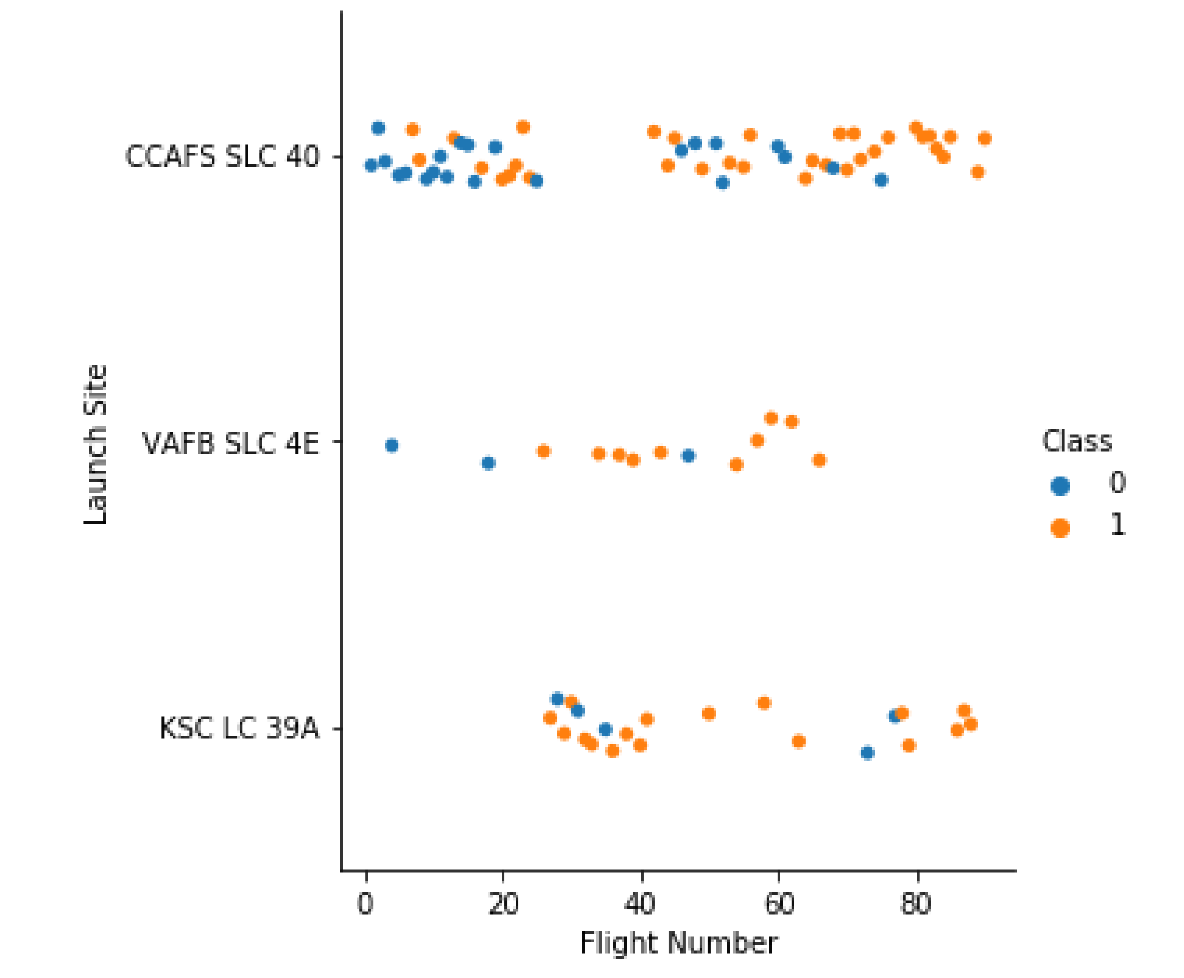
- From the Exploratory data analysis we obtained following results:
  - Payload of Falcon-9 rocket is increasing gradually throughout the years.
  - Success rate of landing outcome of Falcon-9 rockets is increasing throughout the years.
- From Interactive analytics demo we observe that:
  - Landing sites of Falcon-9 rockets are in California and Florida.
  - Florida site have the highest number of successful landing of Falcon-9 rockets.
- From Predictive analysis we observe that:
  - Decision Tree classification method has the highest predication accuracy, even though other models also have high prediction accuracy.

Section 2

Insights drawn from EDA

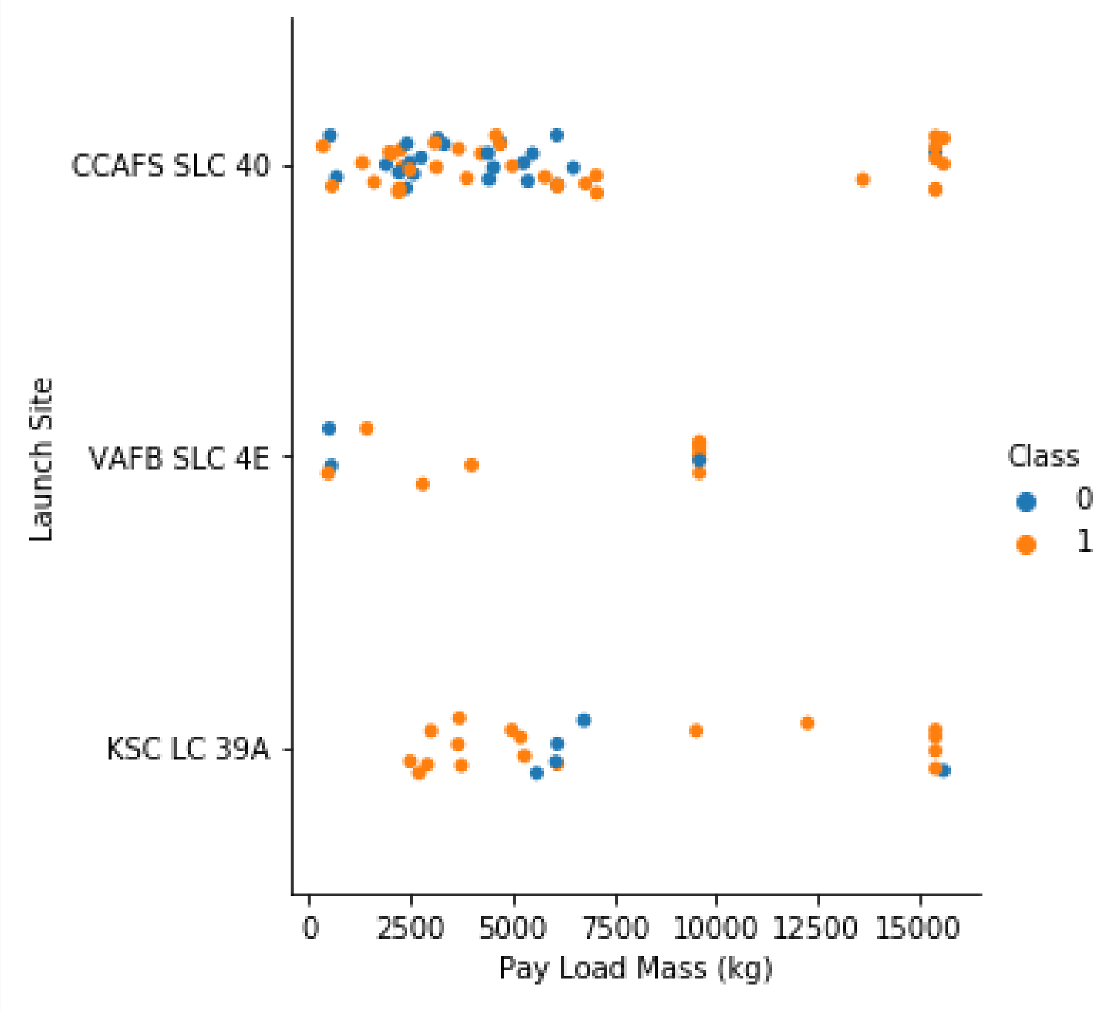
Flight Number vs. Launch Site

- Launches from all sites have better landing outcomes as flight numbers increases.
- Most of the launches take place in CCAFS launch site.



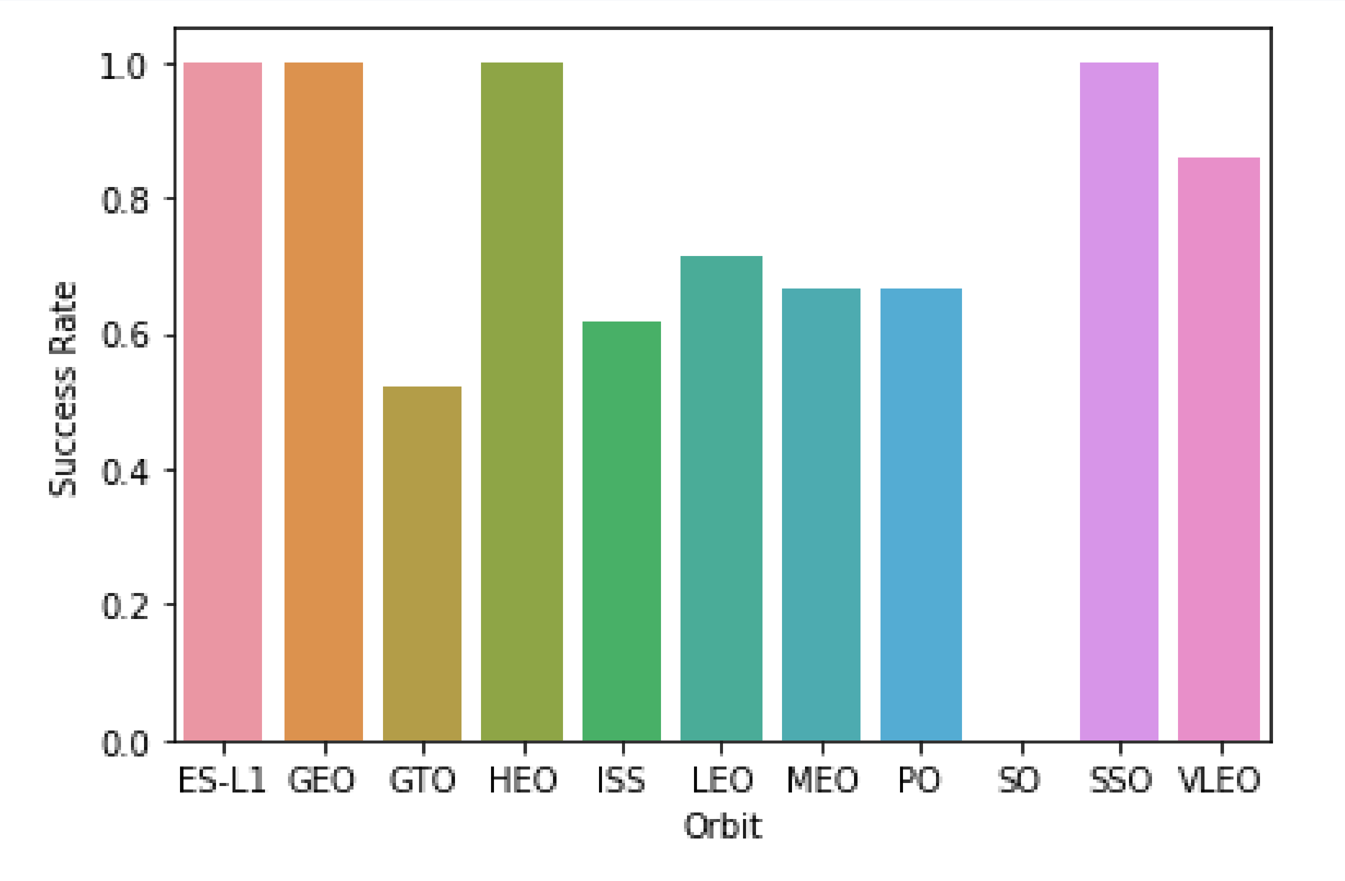
Payload vs. Launch Site

- The payload for majority of launches is less than 7,500 kg.
- Majority of launches from KSC LC 39-A launch site have successful landing outcome.



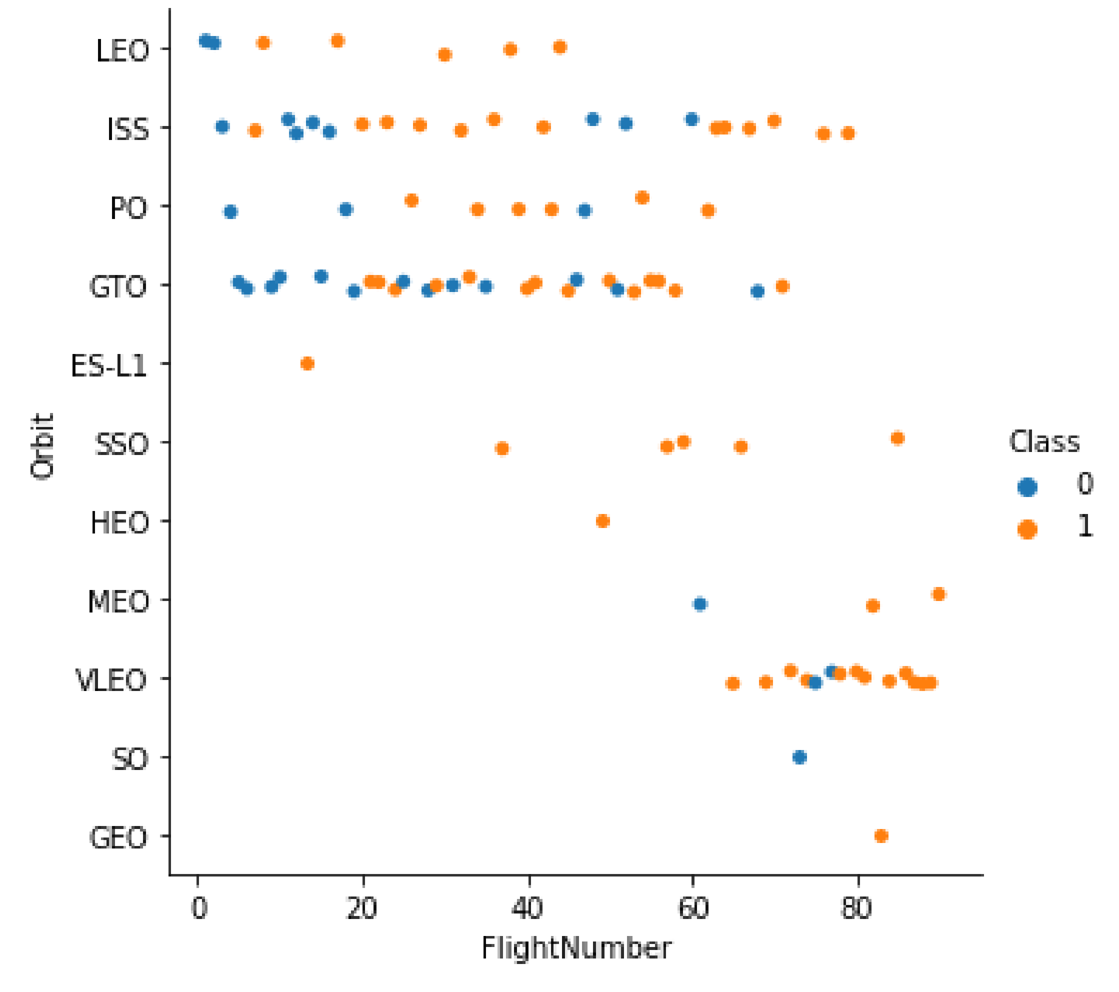
Success Rate vs. Orbit Type

- The Falcon-9 rocket launches to ES-L1, GEO, HEO and SSO orbits have landing success rate of 100%.
- The Falcon-9 rocket launches to other orbits have landing success rate between 50% and 85%



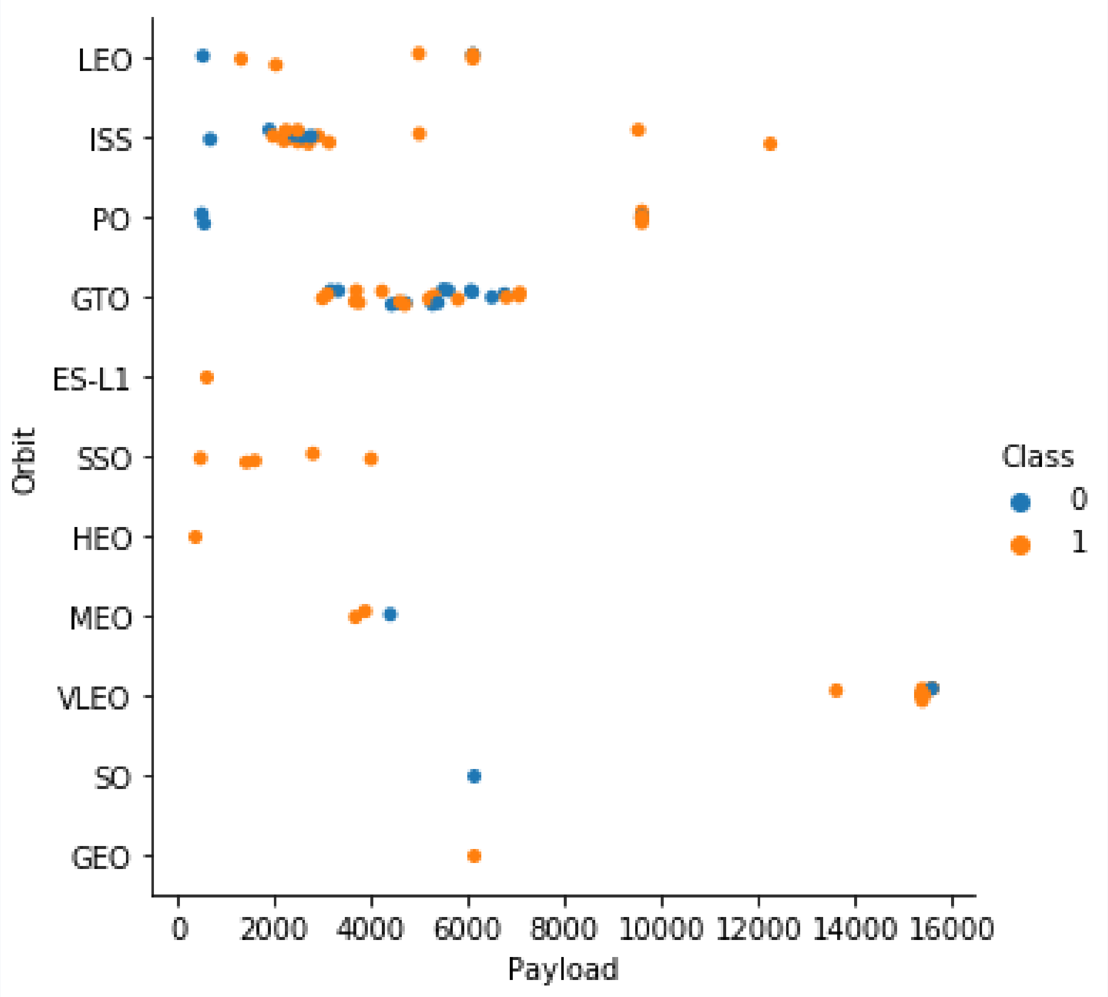
Flight Number vs. Orbit Type

- Launches to higher orbits (SSO, HEO, MEO, VLEO, SO and GEO) are becoming more common as flight number increases.
- Success rate of Falcon-9 landings is increasing with flight number for all orbits.



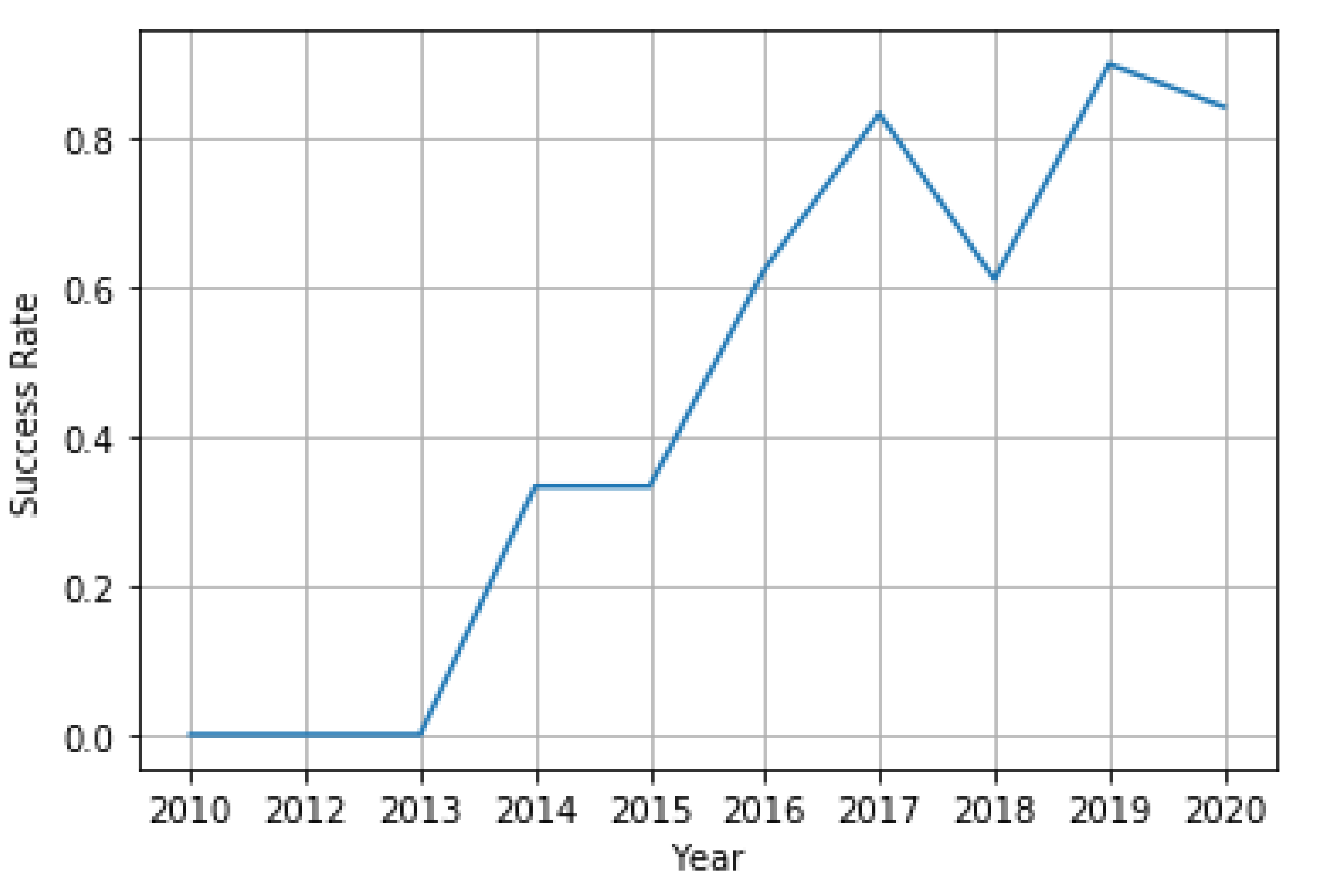
Payload vs. Orbit Type

- The payload to orbit VLEO us greater than 13,000 kg.
- The number of launches with payload greater than 9,000 is very low.
- Payload amount is not a good indicator if the rocket will land successfully.



Launch Success Yearly Trend

- Success rate of Falcon-9 rocket landing is improving steadily since 2013.
- 80% of Falcon-9 rockets are landing successfully and hence can be re-used since 2019.



All Launch Site Names

- Unique launch site names are displayed using the distinct keyword

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

```
[6]: %sql select distinct Launch_Site from spacextbl

 * sqlite:///my_data1.db
Done.
```

[6]:

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

Launch Site Names Begin with 'CCA'

- Display 5 records where launch sites begin with `CCA`

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

[7]:
```
%%sql
select *
from spacextbl
where Launch_Site like 'CCA%'
limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

[7]:

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

Total Payload Mass

- Display the total payload carried by boosters from NASA

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

[8]:
```
%%sql
select sum(PAYLOAD_MASS__KG_)
from spacextbl
where Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

[8]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

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

```
[9]: %%sql
select avg(PAYLOAD_MASS__KG_), Booster_Version from spacextbl
where Booster_Version = 'F9 v1.1'

 * sqlite:///my_data1.db
Done.
```

[9]:

avg(PAYLOAD_MASS__KG_)	Booster_Version
2928.4	F9 v1.1

First Successful Ground Landing Date

- Display the date of the first successful landing outcome on ground pad.

```
[21]: %%sql
select Date
from spacextbl
where "Landing _Outcome" ='Success (ground pad)'
order by strftime('%s',Date)
limit 1

 * sqlite:///my_data1.db
Done.
```

[21]:

Date
22-12-2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- Display names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

[60]:
```
%%sql
SELECT Booster_Version, PAYLOAD_MASS__KG_, "Landing _Outcome"
from spacextbl
where "Landing _Outcome" = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000
```

```
 * sqlite:///my_data1.db
Done.
```

[60]:

Booster_Version	PAYLOAD_MASS__KG_	Landing _Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Display the total number of successful and failure mission outcomes.
- We observe that all of the mission outcomes of the Falcon-9 rocket are successful, except 1.

[22]:
```
%%sql
select COUNT(Mission_Outcome) as "# of successful missions"
from spacextbl
where Mission_Outcome LIKE 'Success%';
```

```
 * sqlite:///my_data1.db
Done.
```

[22]:

# of successful missions
100

[23]:
```
%%sql
select COUNT(Mission_Outcome) as "# of failed missions"
from spacextbl
where Mission_Outcome LIKE 'Failure%'
```

```
 * sqlite:///my_data1.db
Done.
```

[23]:

# of failed missions
1

Boosters Carried Maximum Payload

- Display the names of the booster which have carried the maximum payload mass.

[70]:
```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_
from spacextbl
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

```
 * sqlite:///my_data1.db
Done.
```

[70]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- Display the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- We observe that in months of January and April of 2015 the drone ship landings of Falcon-9 rockets have failed.

```
[79]: %%sql
select substr(Date,7,4) as Year, substr(Date, 4, 2) as Month
Booster_Version, Launch_Site, "Landing _Outcome"
from spacextbl
where "Landing _Outcome" = 'Failure (drone ship)'
and substr(Date,7,4) = '2015'

 * sqlite:///my_data1.db
Done.
```

[79]:

Year	Month	Booster_Version	Launch_Site	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We observe that most of the landings are successful between 2010-06-04 and 2017-03-20.

[57]:
```
%%sql
select "Landing _Outcome", count("Landing _Outcome") as Rank
from spacextbl
where Date between '04-06-2010' and '20-03-2017'
group by "Landing _Outcome"
order by rank desc
```

```
 * sqlite:///my_data1.db
Done.
```

[57]:

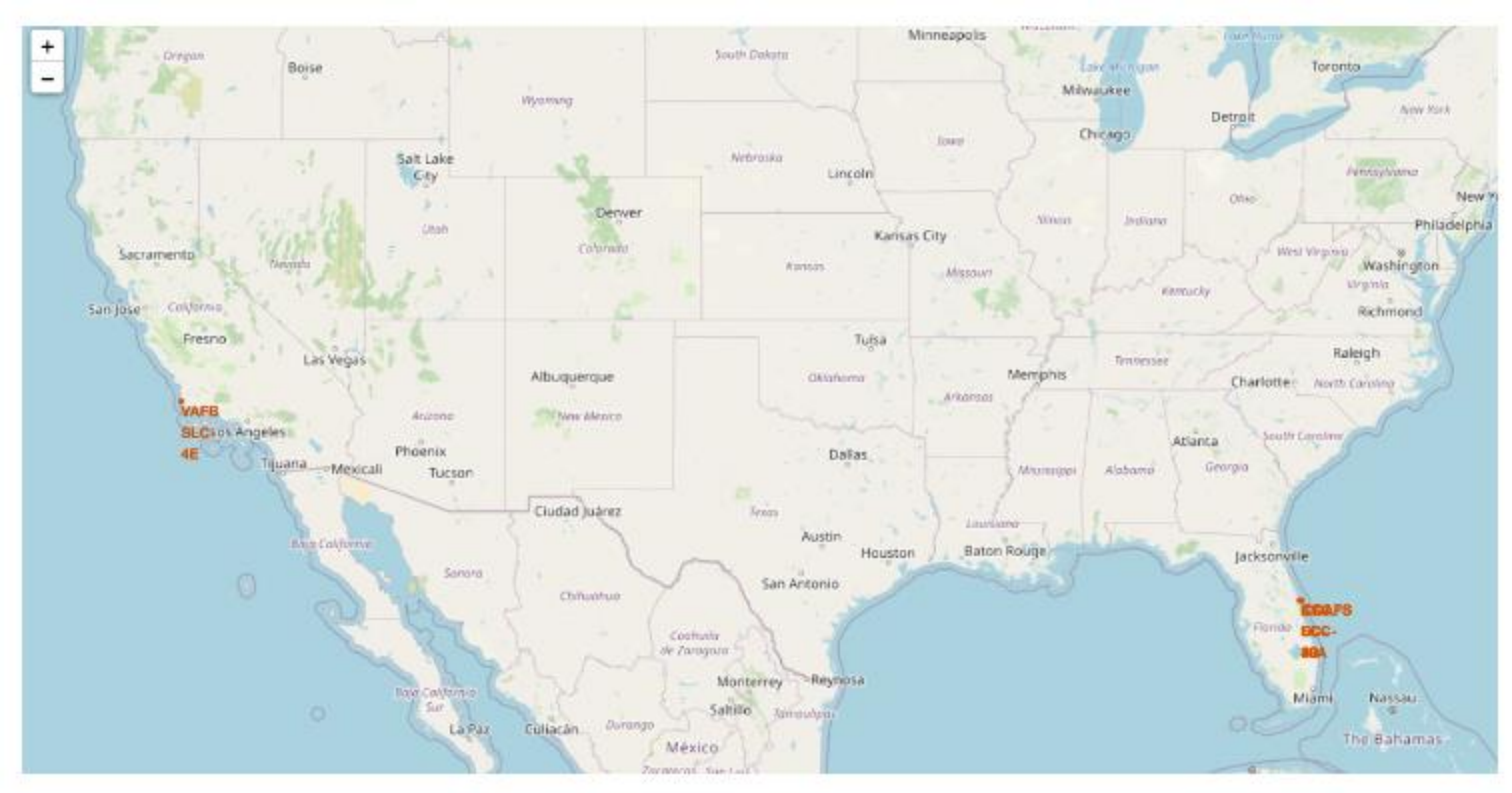
Landing _Outcome	Rank
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

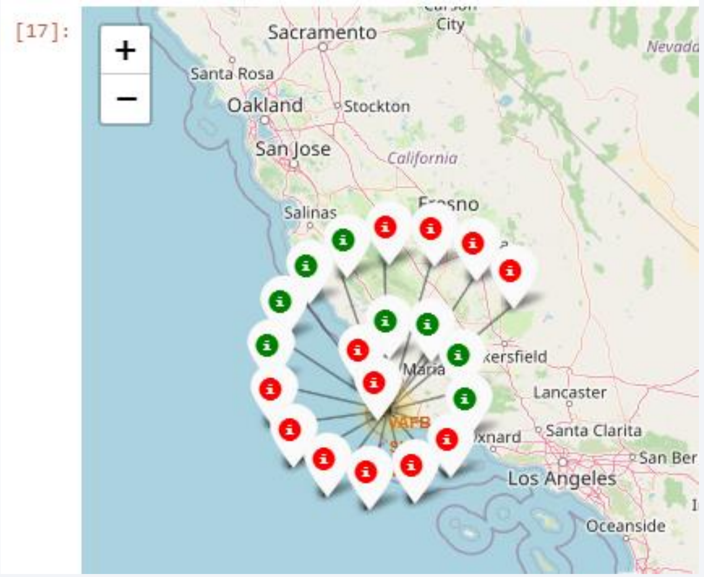
All Launch Sites Locations

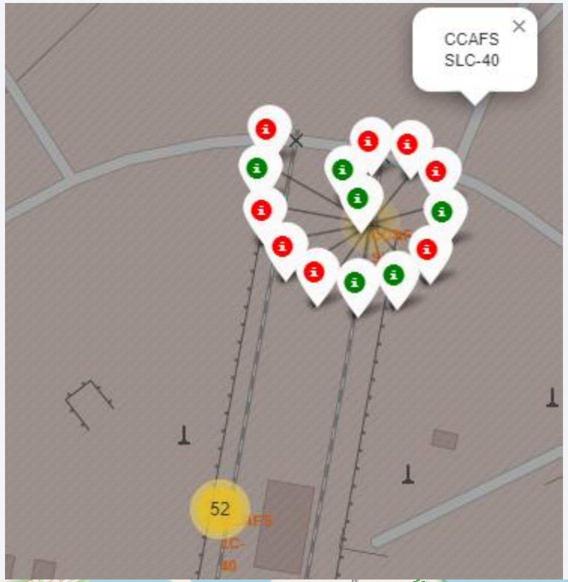
- The SpaceX Falcon-9 launch sites are located in Florida and California

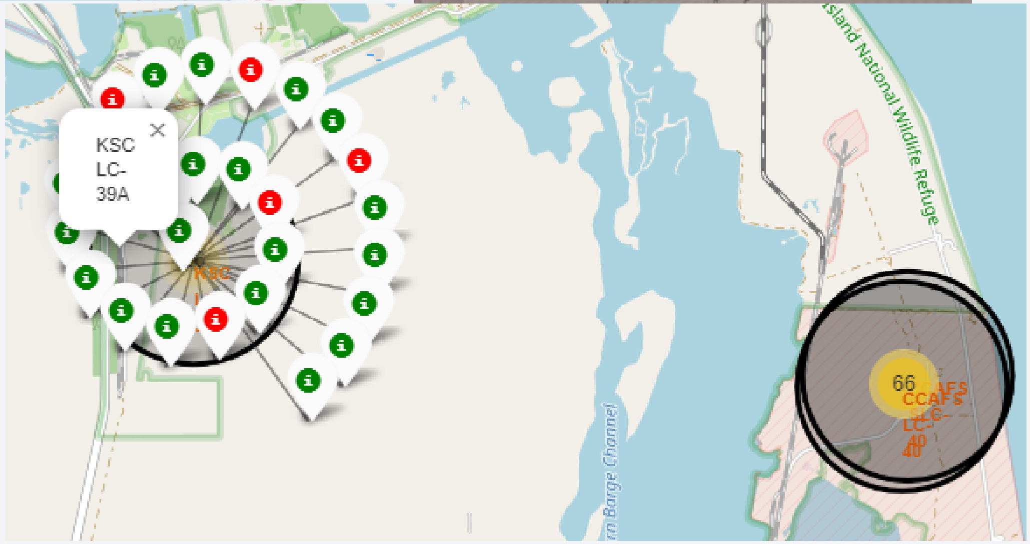


Color Labeled Launch Outcomes at Launch Sites

- Green markers show successful landings. Red markers show failed landings.







Distance to Railroad, Highway and Coastline

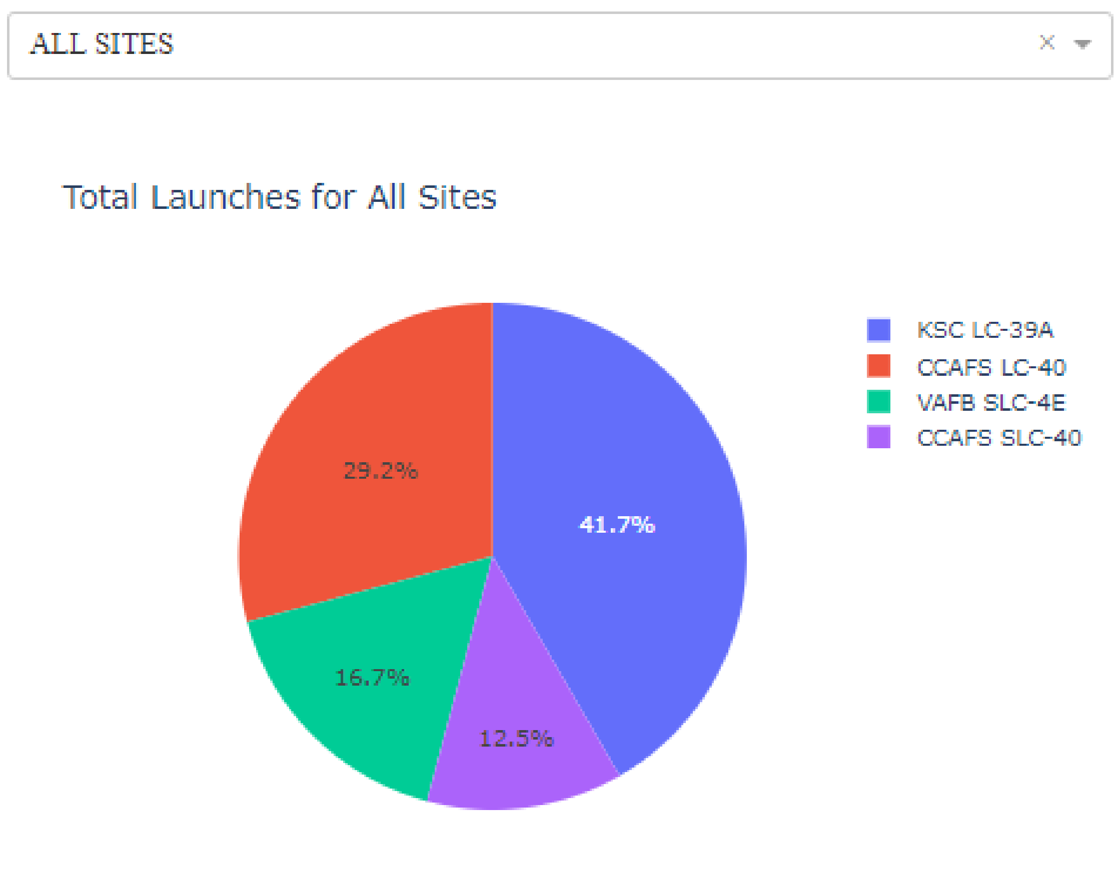
- It is observed that CCAFS SLC-40 launch site is 0.86 km to Florida coastline, 0.58 km to the nearest highway and 1.28 km to the nearest railway line.

Section 4

Build a Dashboard with Plotly Dash

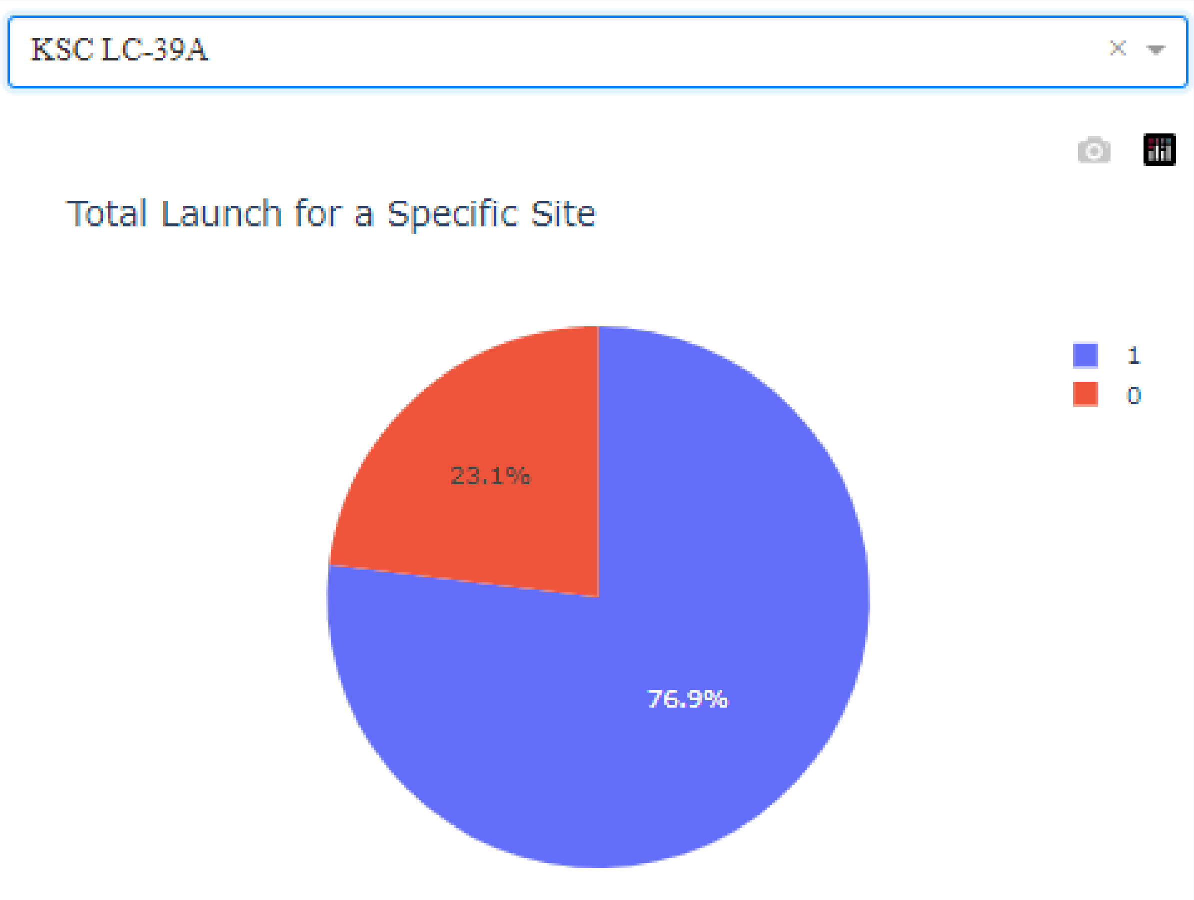
Launch Successes for All Sites

- Most of the landing successes (41.7%) are at KSC LC-39 launch site.



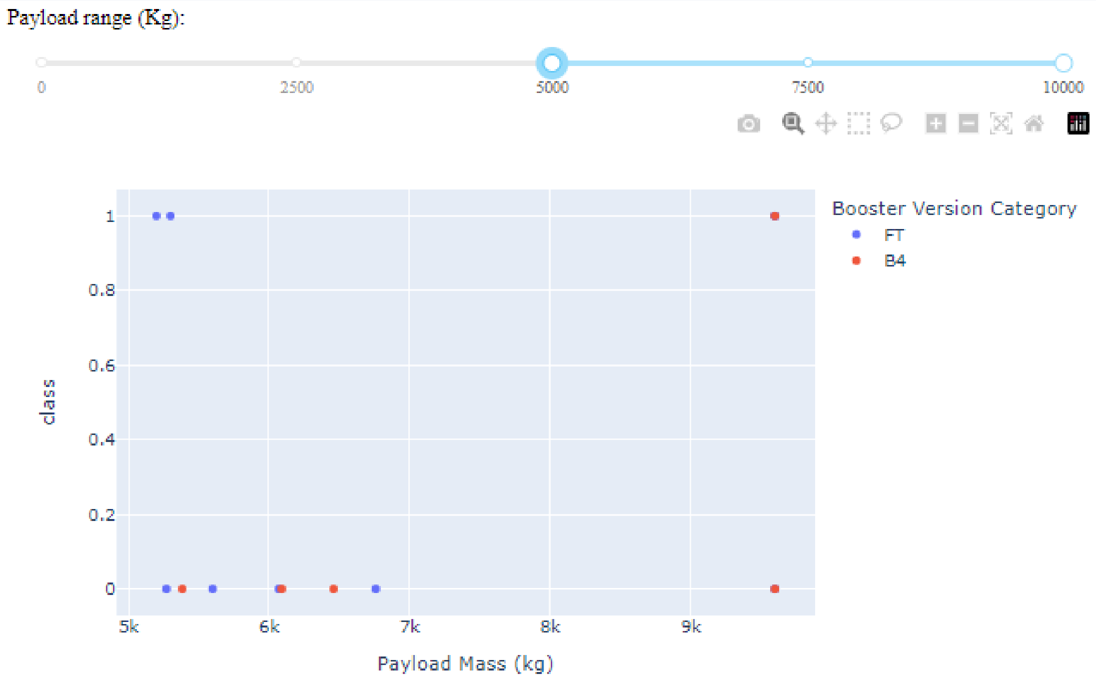
Launch Site with Highest Launch Success Ratio

- KSC LC-39A is the launch site with the highest success ratio. That is for the 76.9% of the launches from this site, the first stage of the Falcon-9 rocket landed successfully.



Payload vs. Launch Outcome for All Sites

- When payload is less than 5,000 kg all booster versions are used.
- When payload is greater than 5,000 kg only the booster versions FT and B4 are used.
- Each booster version launched can successfully land or fail to land. There is no clear choice.

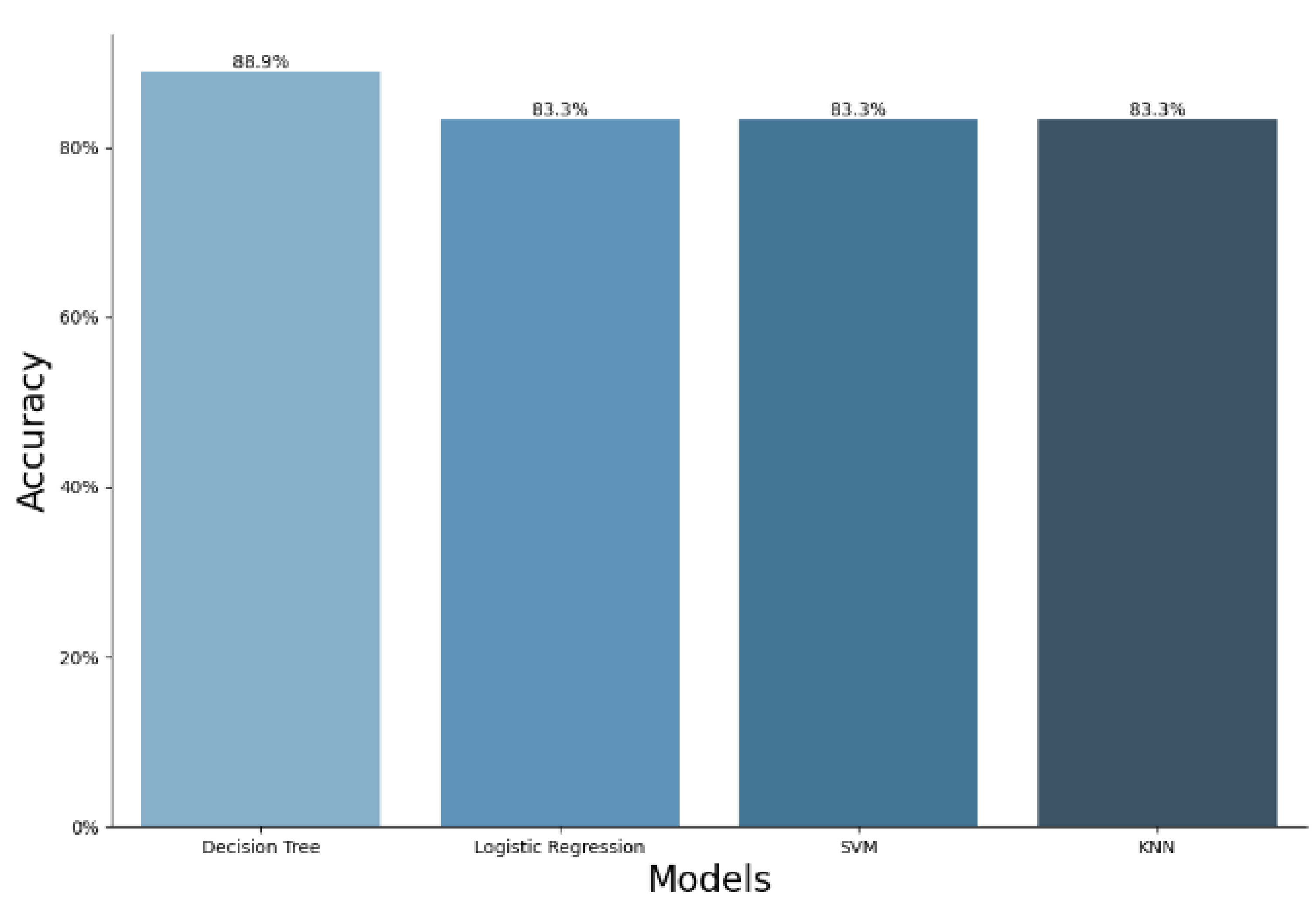


Section 5

Predictive Analysis (Classification)

Classification Model Accuracy

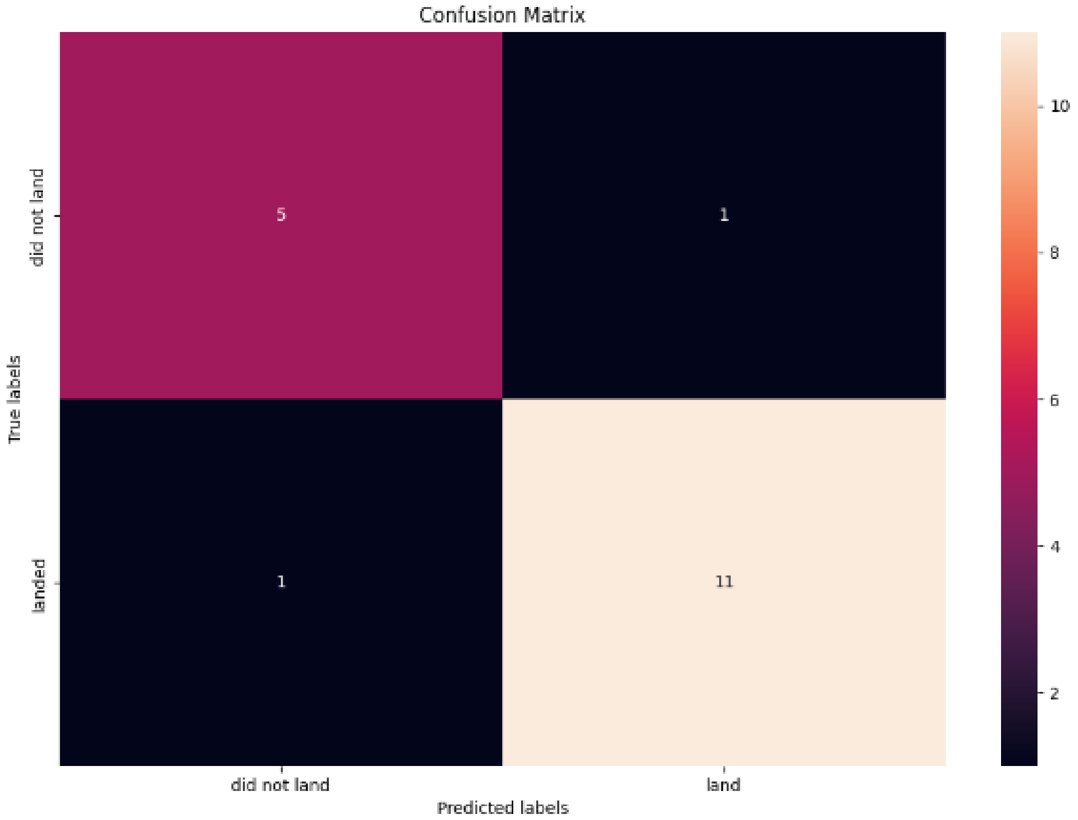
- Decision Tree model has the highest accuracy with 88.9% to predict a launch outcome.
- Other classification models have 83.3% accuracy to predict the launch outcome.



Confusion Matrix of Best Performing Model

- Decision Tree classification model has the confusion matrix from the test samples, such that there are 5 true negatives and 11 true positives. There is only 1 false negative and 1 false positive.

```
[70]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- We observed that the 1st stage of Falcon-9 rocket landing success improved significantly over the years.
- We also observed that ML classification models can be trained with a high accuracy to predict the landing outcome of Falcon-9 rocket.
- In this capstone we have used API requests, webpage scrapping, SQL queries, Folio maps and Plotly Dash to obtain data, query data and visualize data to have insights. We have also used ML classification models to predict the landing outcome of a Falcon-9 rocket with high accuracy.

Thank you!