Winning Space Race with Data Science

Ahad Abbaszadeh Azar

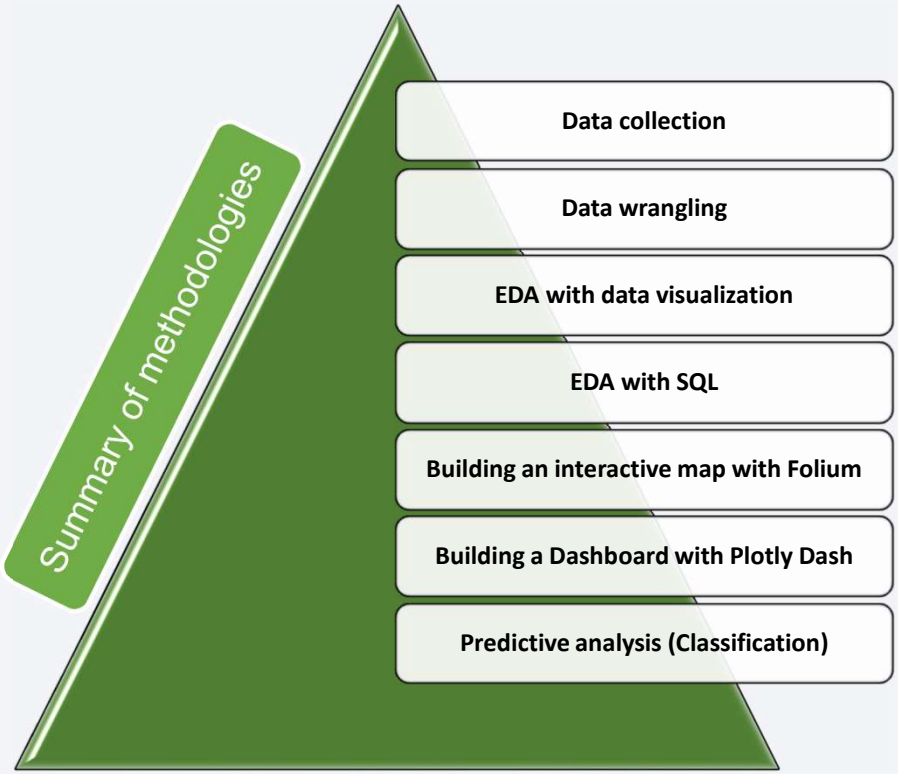
2023 - August

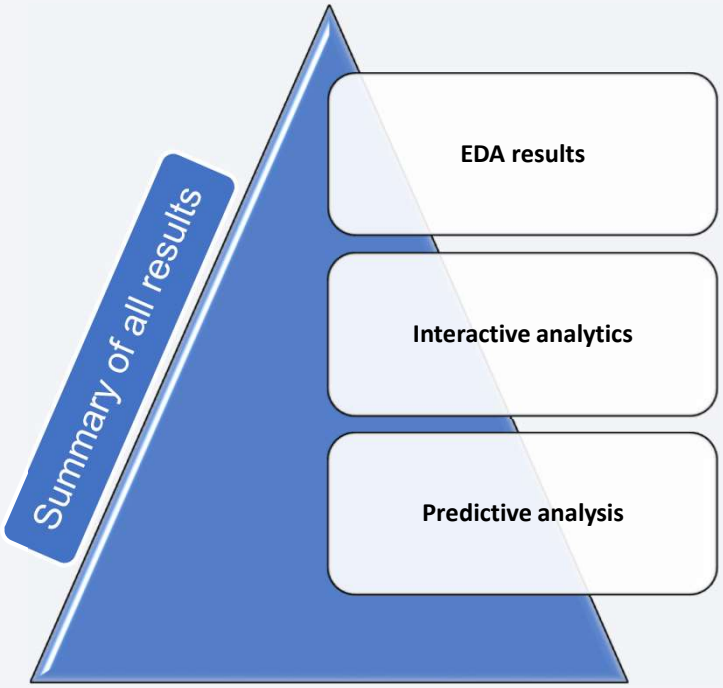
https://github.com/absze/Capstone

Outline

- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULTS
- CONCLUSION
- APPENDIX

Executive Summary





Introduction

Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Problems you want to find answers

The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully.

Section 1

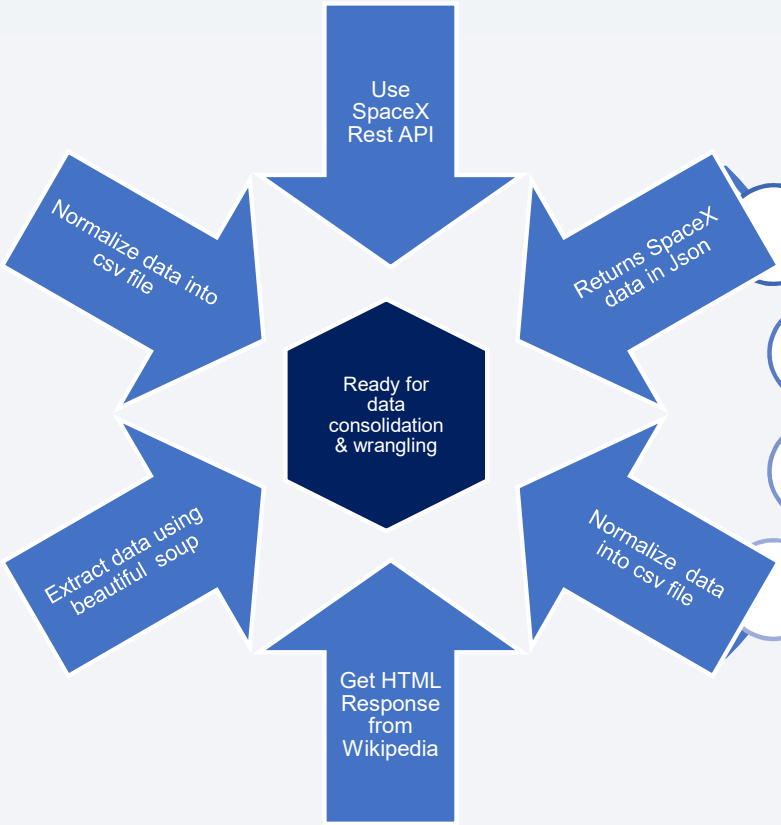
Methodology

Methodology

Executive Summary

- **Data collection methodology**
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- **Perform data wrangling**
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection



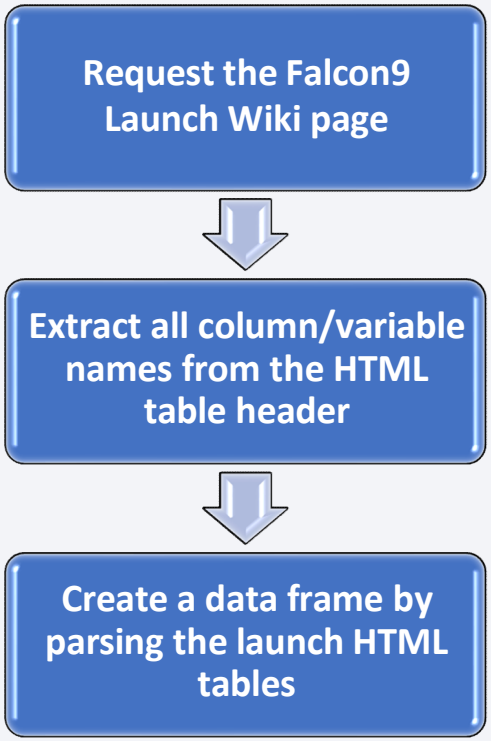
- SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com /v4
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using Beautiful Soup.

Data Collection – SpaceX API

- SpaceX offers a public API from where data can be obtained and then used;
- This API was used according to the flowchart beside and then data is persisted.

Data Collection - Scraping

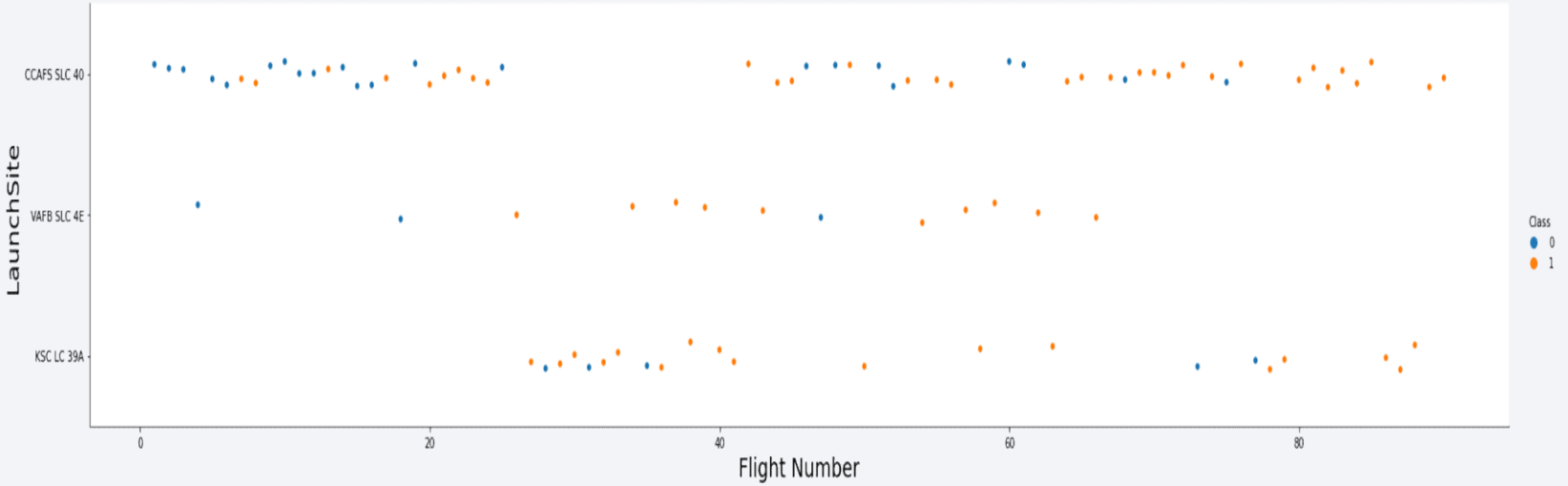
- Data from SpaceX launches can also be obtained from Wikipedia;
- Data are downloaded from Wikipedia according to the flowchart and then persisted.



Data Wrangling

- Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then the summaries launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
- Finally, the landing outcome label was created from Outcome column.

EDA with Data Visualization



- To explore data, scatterplots and barplots were used to visualize the relationship between pair of features:
  - Payload Mass X Flight Number, Launch Site X Flight Number, Launch Site X Payload Mass, Orbit and Flight Number, Payload and Orbit

EDA with SQL

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

Displaying 5 records where launch sites begin with the string 'KSC

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

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

Listing the date where the successful landing outcome in drone ship was achieved.

Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

Listing the total number of successful and failure mission outcomes

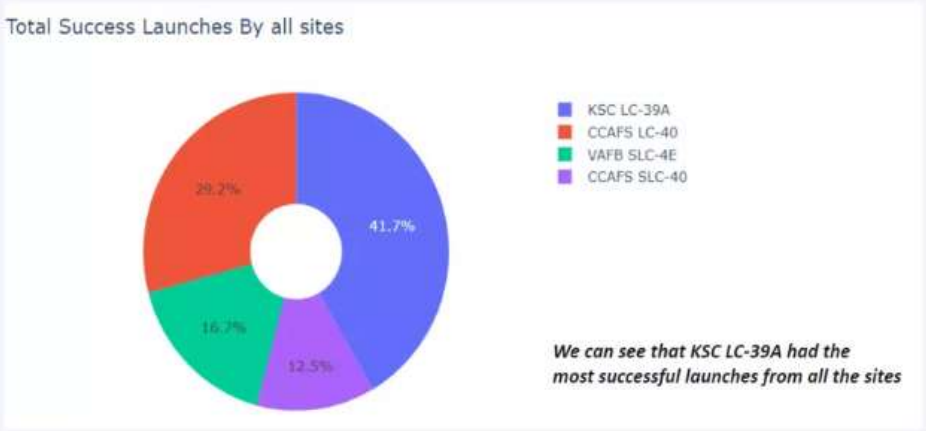
Listing the names of the booster versions which have carried the maximum payload mass

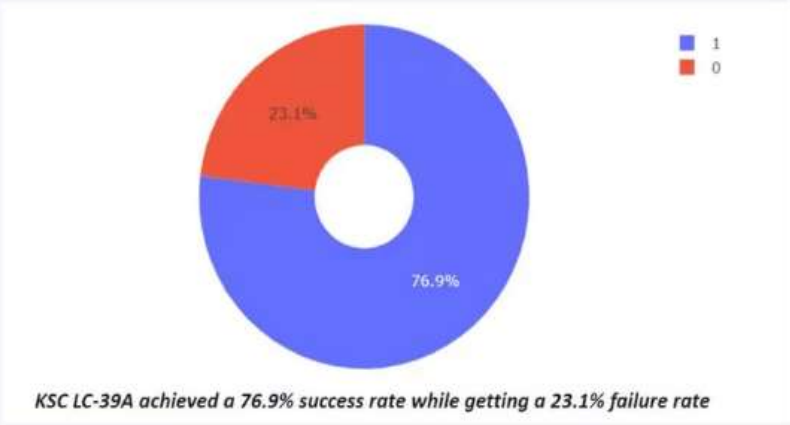
Listing the records which will display the month names, successful landing outcomes in ground pad booster versions, launch site for the months in year 2017

Ranking the count of successful landing outcomes between the date 2010 06 04 and 2017 03 20 in descending order.

Build an Interactive Map with Folium

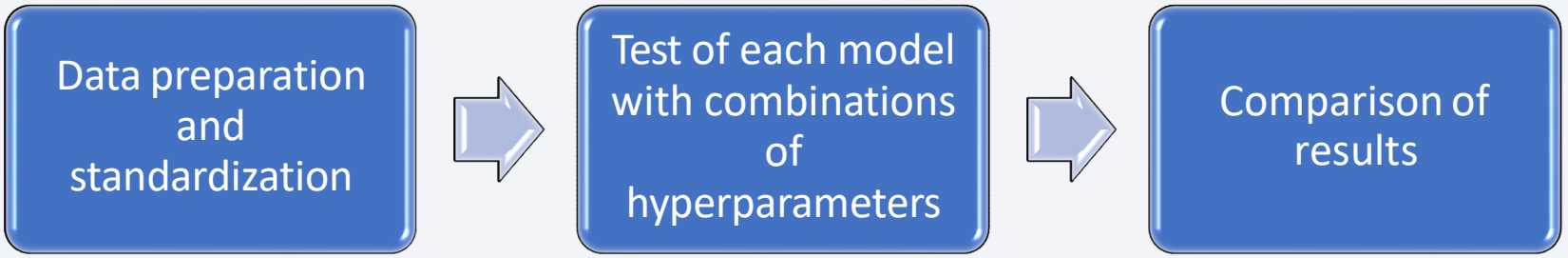
Dashboard with Plotly Dash

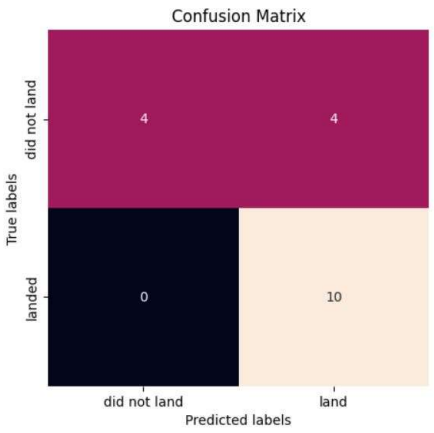


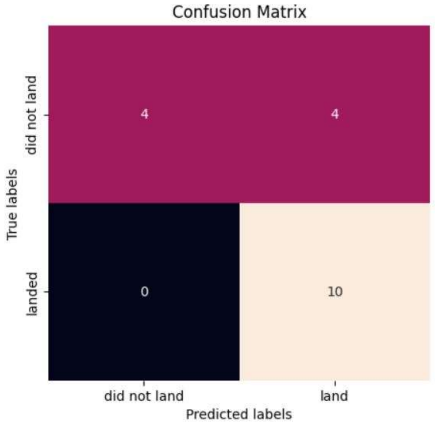


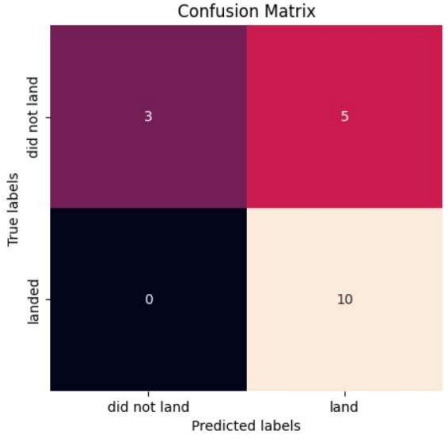
Predictive Analysis (Classification)

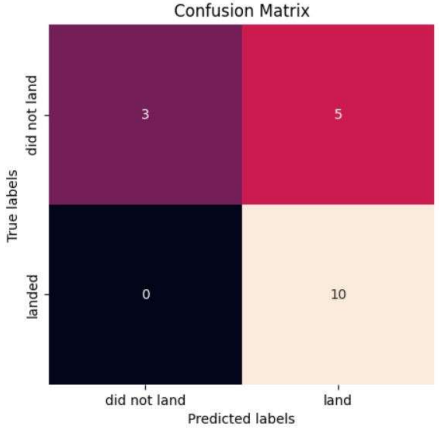
- Four classification models were compared: logistic regression, support vector machine, decision tree and k nearest neighbors.

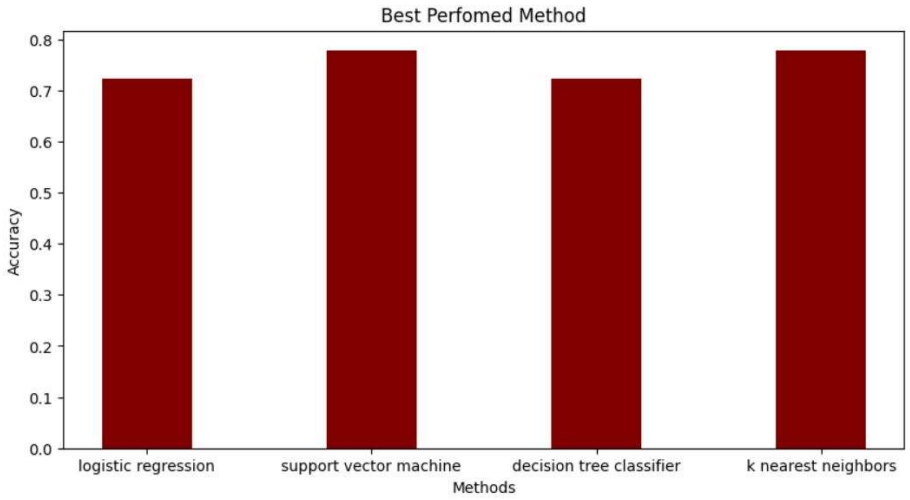












Predictive Analysis (Classification)

The SVM, KNN, and Logistic Regression model achieved the highest accuracy at 83.3%, while the SVM performs the best in terms of Area Under the Curve at 0.958.

Results

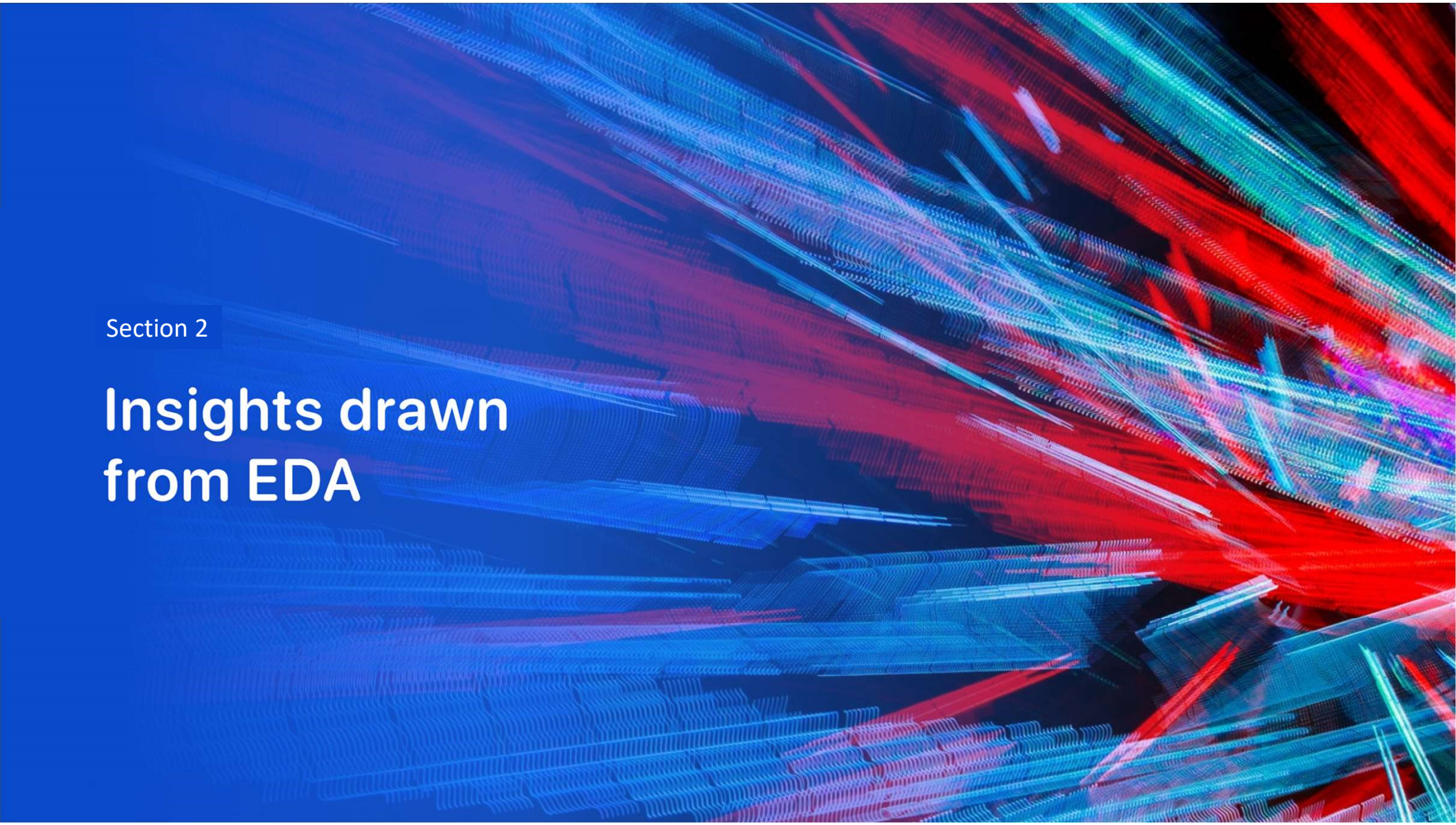
The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.

- Low weighted payloads perform better than the heavier payloads.

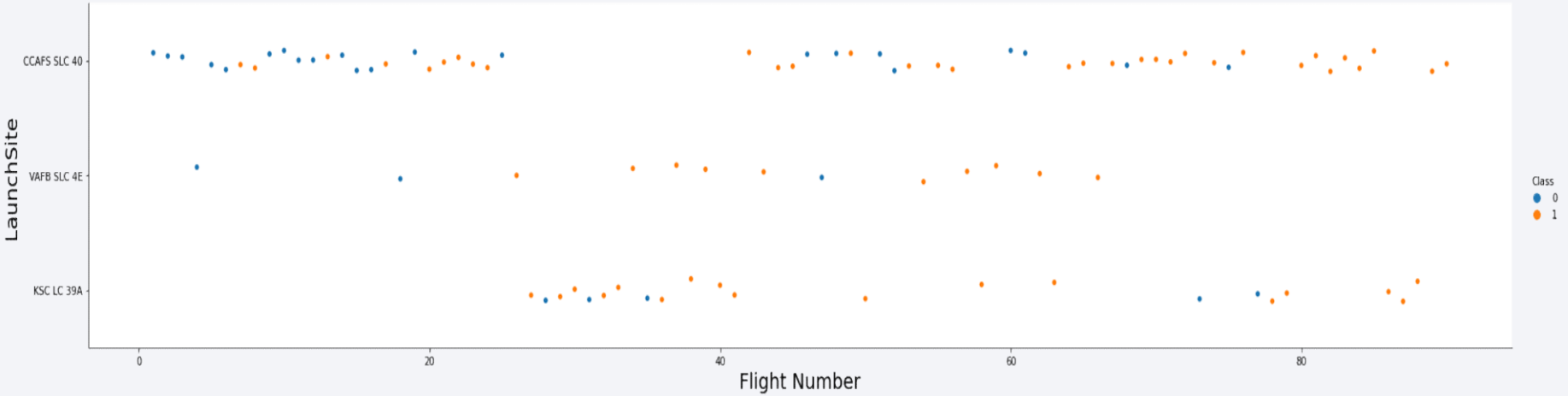
The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.

KSC LC 39A had the most successful launches from all the sites.

- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

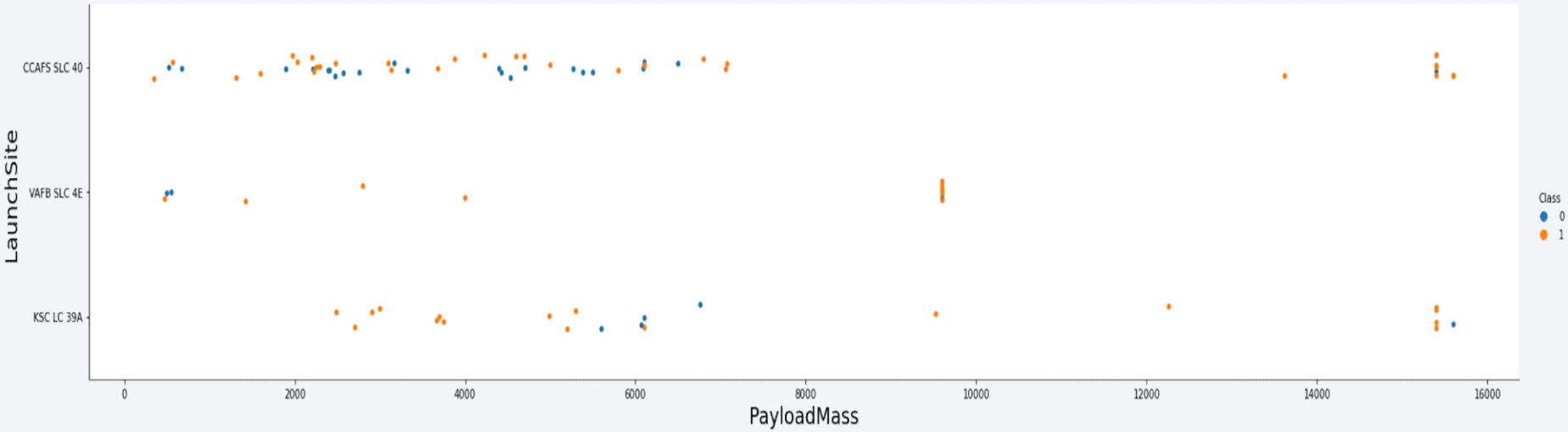

Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site



- According to the plot above, it's possible to verify that the best launch site nowadays is CCAF5 SLC 40, where most of recent launches were successful;
- In second place VAFB SLC 4E and third place KSC LC 39A;
- It's also possible to see that the general success rate improved over time.

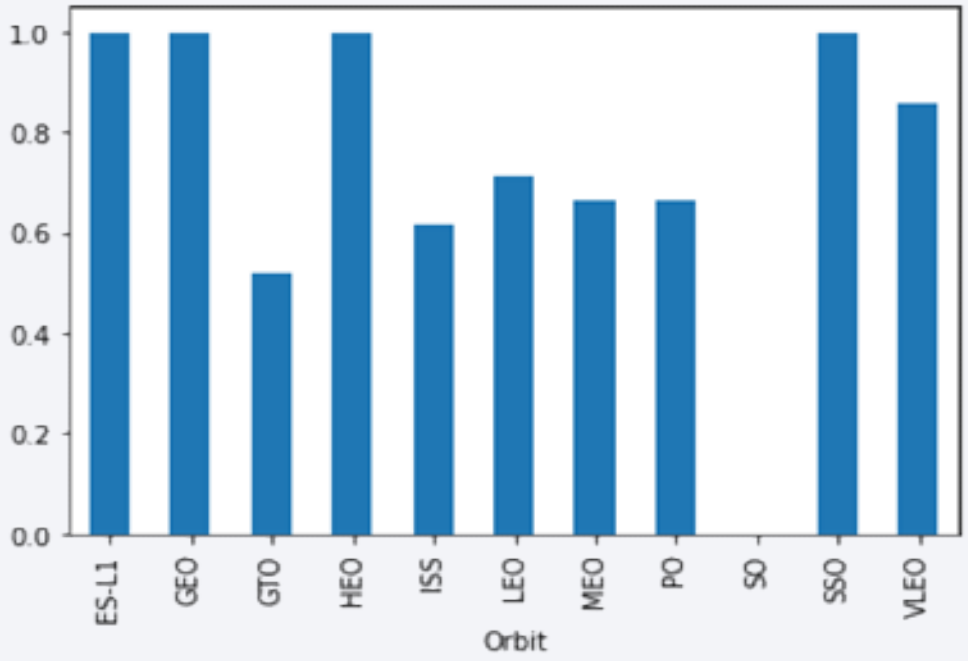
Payload vs. Launch Site



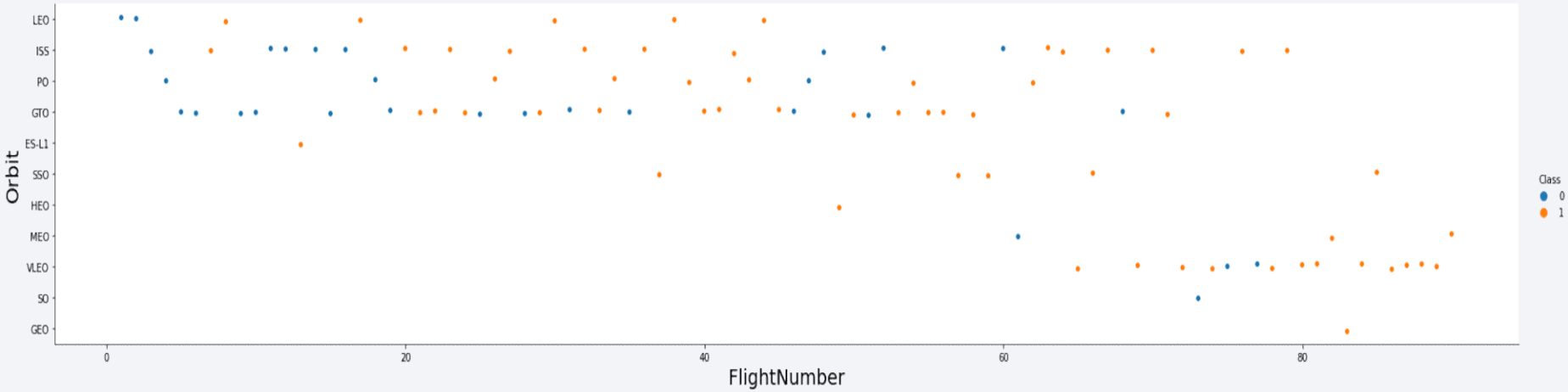
- Payloads over 9,000kg (about the weight of a school bus) have excellent success rate;
- Payloads over 12,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.

Success Rate vs. Orbit Type

- The biggest success rates happens to orbits:
  - ES-L1;
  - GEO;
  - HEO; and
  - SSO.
- Followed by:
  - VLEO (above 80%); and
  - LFO (above 70%).

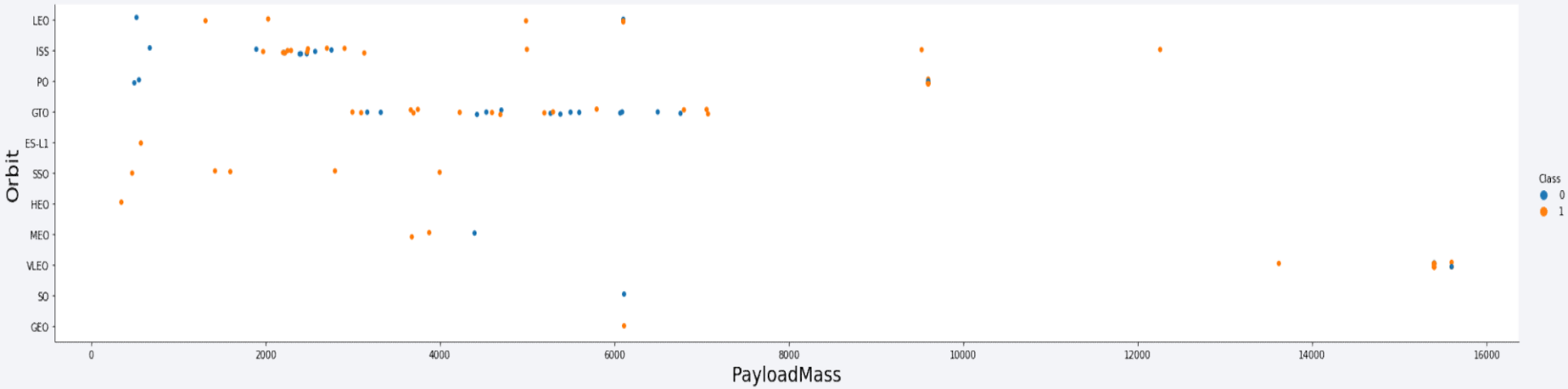


Flight Number vs. Orbit Type



- Apparently, success rate improved over time to all orbits;
- VLEO orbit seems a new business opportunity, due to recent increase of its frequency.

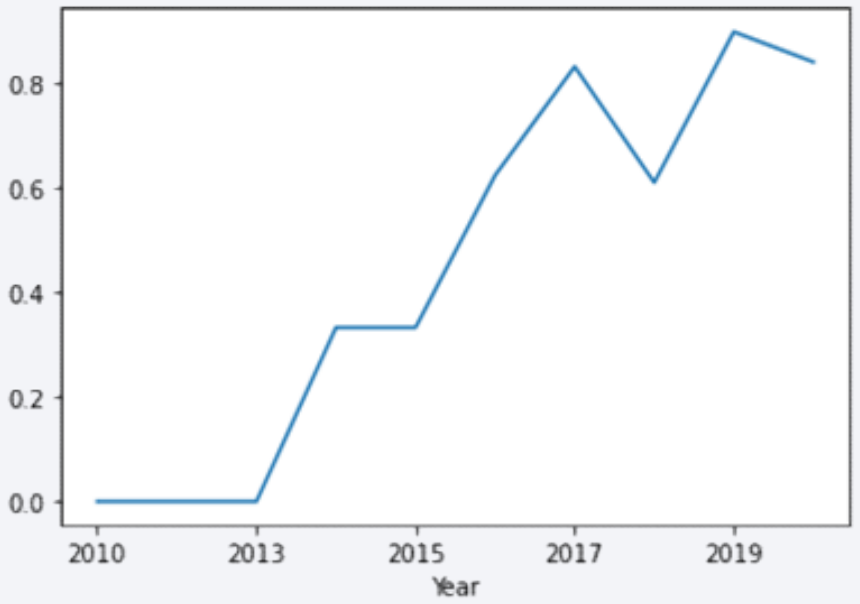
Payload vs. Orbit Type



- Apparently, there is no relation between payload and success rate to orbit GTO;
- ISS orbit has the widest range of payload and a good rate of success;
- There are few launches to the orbits SO and GEO.

Launch Success Yearly Trend

- Success rate started increasing in 2013 and kept until 2020;
- It seems that the first three years were a period of adjusts and improvement of technology.



All Launch Site Names

- There are three different LaunchSite here

Out[5]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
0	1	6/4/2010	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
1	2	5/22/2012	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
2	3	3/1/2013	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1
3	4	9/29/2013	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1
4	5	12/3/2013	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1

```
In [6]: df['LaunchSite'].unique()
Out[6]: array(['CCAFS SLC 40', 'VAFB SLC 4E', 'KSC LC 39A'], dtype=object)
```

Launch Site Names Begin with 'CCA'

- The following code presented in the table displaying five rows from the LaunchSite column that start with 'CCA' by using the associated code.

```
In [9]: df[df['LaunchSite'].str.startswith('CCA')].head(5)
```

Out[9]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Se
0	1	6/4/2010	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1	0	B0
1	2	5/22/2012	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1	0	B0
2	3	3/1/2013	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1	0	B0
4	5	12/3/2013	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1	0	B1
5	6	1/6/2014	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1	0	B1

Total Payload Mass

- From related calculation, the total PayLoadMass carried by boosters from NASA was zero

```
In [10]: total_payload_nasa = df[df['BoosterVersion'] == 'NASA']['PayloadMass'].sum()
         print("Total payload carried by NASA boosters:", total_payload_nasa)

         Total payload carried by NASA boosters: 0.0
```

Average Payload Mass by F9 v1.1

- From related calculation, the total PayLoadMass carried by boosters from NASA was 549446.3470600001

```
In [16]: total_payload_nasa = df[df['BoosterVersion'] == 'Falcon 9']['PayloadMass'].sum()
         print("Total payload carried by NASA boosters:", total_payload_nasa)

         Total payload carried by NASA boosters: 549446.3470600001
```

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad were shown here:

```
In [17]: success_df = df[df['Class'] == 1]
         success_landing_dates = success_df.groupby('LandingPad')['Date'].min()
         print("Dates of the first successful landing pad outcomes:")
         print(success_landing_dates)

         Dates of the first successful landing pad outcomes:
         LandingPad
         5e9e3032383ecb267a34e7c7     1/8/2018
         5e9e3032383ecb554034e7c9    10/8/2018
         5e9e3032383ecb6bb234e7ca    1/29/2020
         5e9e3033383ecbb9e534e7cc    1/11/2019
         Name: Date, dtype: object
```

Successful Drone Ship Landing with Payload between 4000 and 6000

```
# Find unique values in the 'BoosterVersion' column
unique_boosters = df['BoosterVersion'].unique()

# Filter the DataFrame for boosters with successful landing outcomes, payload mas
filtered_boosters = df[(df['LandingPad'] == 'Success') & (df['PayloadMass'] > 40(

# Count the occurrences of each booster
booster_counts = filtered_boosters['BoosterVersion'].value_counts()

# Print the number of unique boosters and the booster names along with their resp
print("Number of unique boosters:", len(unique_boosters))
print("\nCount of boosters that successfully landed on drone ship with payload ma
print(booster_counts)
```

```
Number of unique boosters: 1

Count of boosters that successfully landed on drone ship with payload mass betw
een 4000 and 6000:
Series([], Name: BoosterVersion, dtype: int64)
```

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes is placed in the Class column which has 60 success and 30 failure

```
In [21]: df['Class'].value_counts()

Out[21]: 1    60
         0    30
         Name: Class, dtype: int64
```

Boosters Carried Maximum Payload

In [38]:
```
max_payloads = df.groupby('BoosterVersion')['PayloadMass'].max()

# Find the booster(s) with the maximum payload mass
max_payload_boosters = max_payloads[max_payloads == max_payloads.max()]

# Print the names of boosters with the maximum payload mass
print("Booster(s) with the maximum payload mass:")
for booster in max_payload_boosters.index:
    print(booster)
```

```
Booster(s) with the maximum payload mass:
Falcon 9
```

2015 Launch Records

- Here there is a list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [32]:
```
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y')

# Filter the DataFrame for dates with year 2015
dates_2015 = df[df['Date'].dt.year == 2015]

# Print the results
print("Dates from the year 2015:")
print(dates_2015)
```

```
Dates from the year 2015:
    FlightNumber       Date BoosterVersion  PayloadMass  Orbit    LaunchSite  \
11            12 2015-01-10       Falcon 9       2395.0    ISS  CCAFS SLC 40
12            13 2015-02-11       Falcon 9        570.0  ES-L1  CCAFS SLC 40
13            14 2015-04-14       Falcon 9       1898.0    ISS  CCAFS SLC 40
14            15 2015-04-27       Falcon 9       4707.0    GTO  CCAFS SLC 40
15            16 2015-06-28       Falcon 9       2477.0    ISS  CCAFS SLC 40
16            17 2015-12-22       Falcon 9       2034.0    LEO  CCAFS SLC 40

       Outcome  Flights  GridFins  Reused   Legs               LandingPad  \
11  False ASDS        1      True   False   True  5e9e3032383ecb761634e7cb
12  True Ocean        1      True   False   True                      NaN
13  False ASDS        1      True   False   True  5e9e3032383ecb761634e7cb
14   None None        1     False   False  False                      NaN
15   None ASDS        1      True   False   True  5e9e3032383ecb6bb234e7ca
16   True RTLS        1      True   False   True  5e9e3032383ecb267a34e7c7

    Block  ReusedCount Serial  Longitude   Latitude  Class
11      1            0  B1012 -80.577366  28.561857      0
12      1            0  B1013 -80.577366  28.561857      1
13      1            0  B1015 -80.577366  28.561857      0
14      1            0  B1016 -80.577366  28.561857      0
15      1            0  B1018 -80.577366  28.561857      0
16      1            0  B1019 -80.577366  28.561857      1
```

In [35]:
```
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y')

# Filter the DataFrame for failed landing outcomes (Class 0) and the year 2015
failed_landings_2015 = df[(df['Class'] == 0) & (df['Date'].dt.year == 2015)]

# Print the results
for index, row in failed_landings_2015.iterrows():
    print("Booster Version:", row['BoosterVersion'])
    print("Launch Site:", row['LaunchSite'])
    print("------------------------------")
```

```
Booster Version: Falcon 9
Launch Site: CCAFS SLC 40
------------------------------
Booster Version: Falcon 9
Launch Site: CCAFS SLC 40
------------------------------
Booster Version: Falcon 9
Launch Site: CCAFS SLC 40
------------------------------
Booster Version: Falcon 9
Launch Site: CCAFS SLC 40
------------------------------
```

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

- Here, the 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 is presented

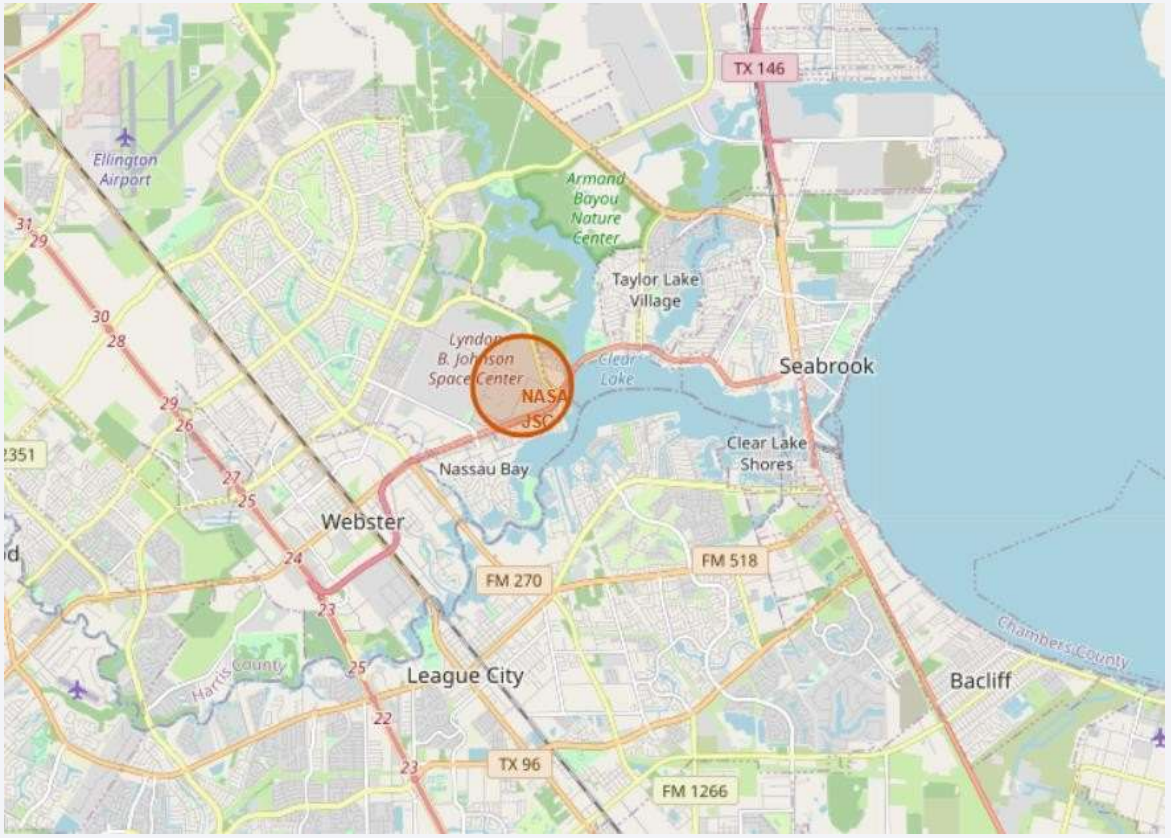
In [37]:

```
# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y')

# Filter the DataFrame for the specified date range
start_date = '2010-06-04'
end_date = '2017-03-20'
filtered_data = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]

# Count the landing outcomes based on 'LandingPad', 'Class', and 'Date'
landing_outcome_counts = filtered_data.groupby(['LandingPad', 'Class'])['Class']

# Sort the counts in descending order
sorted_counts = landing_outcome_counts.sort_values(ascending=False)

# Print the results
print("Ranking of landing outcomes between", start_date, "and", end_date, ":\n")
print(sorted_counts)
```

```
Ranking of landing outcomes between 2010-06-04 and 2017-03-20 :

LandingPad                  Class
5e9e3032383ecb6bb234e7ca    1        4
5e9e3032383ecb267a34e7c7    1        3
5e9e3032383ecb6bb234e7ca    0        3
5e9e3032383ecb761634e7cb    0        2
5e9e3033383ecbb9e534e7cc    0        1
                            1        1
Name: Class, dtype: int64
```

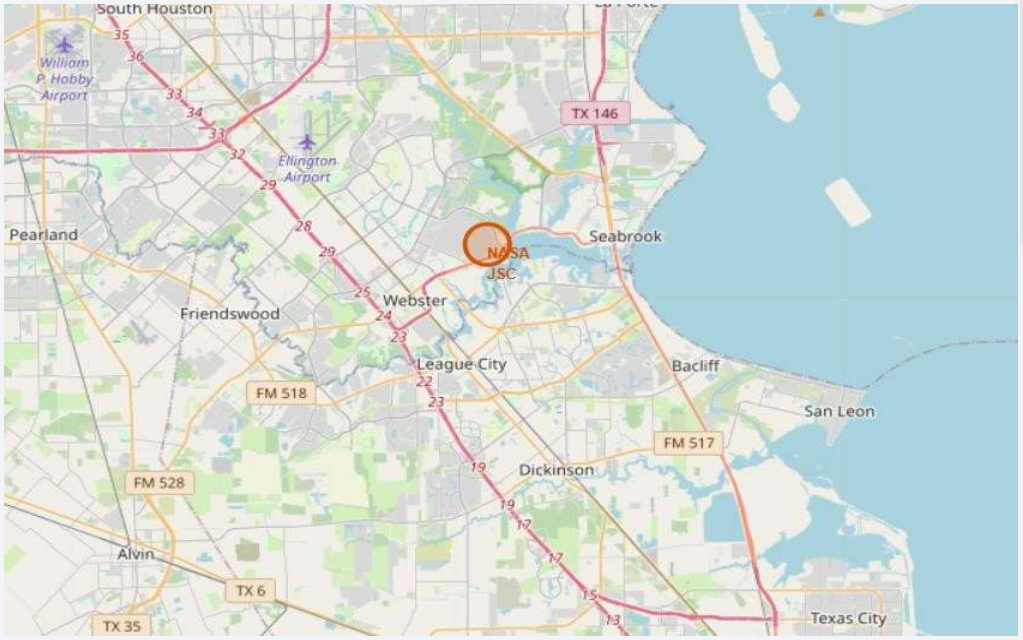

Section 3
Launch Sites
Proximities Analysis

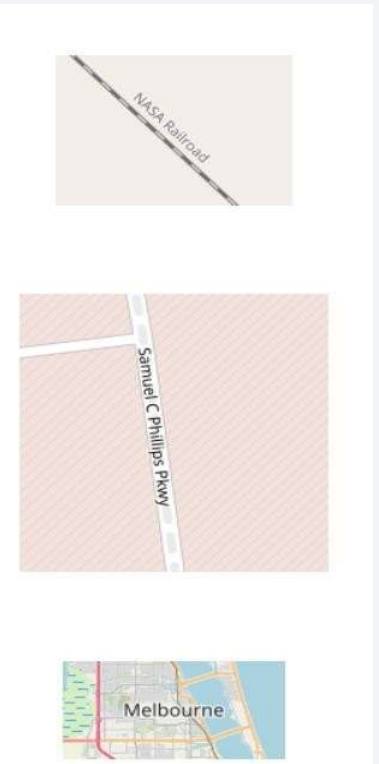
Folium Map Screenshot 1

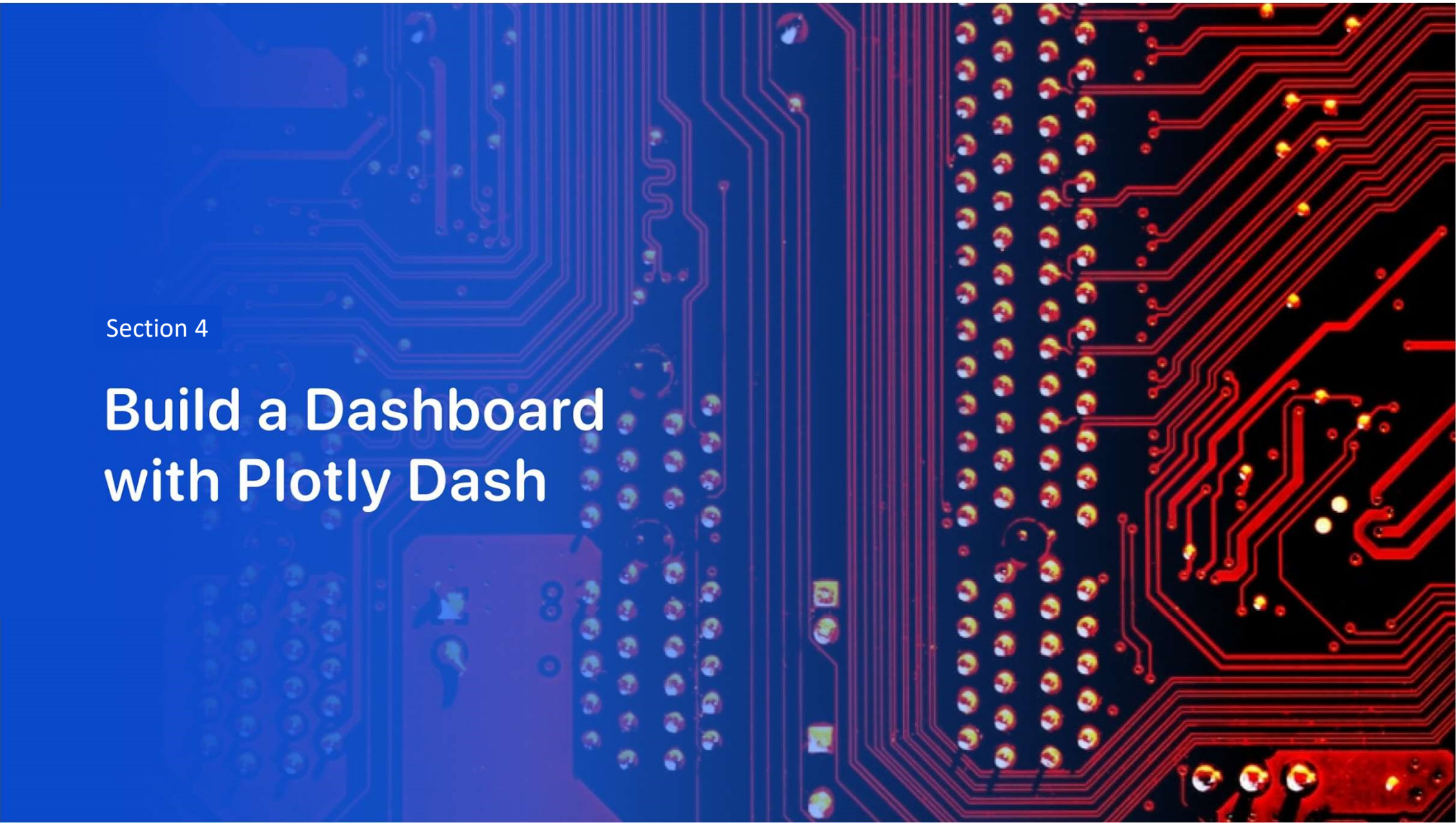


Folium Map Screenshot 2

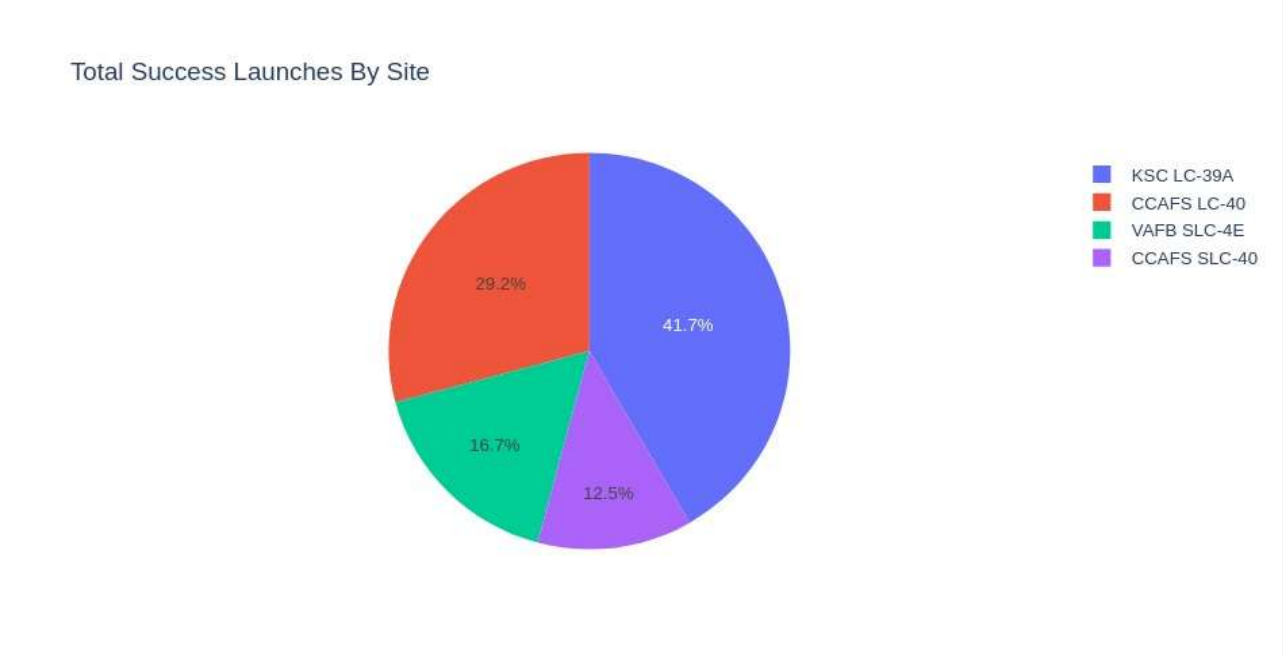
Folium Map Screenshot 3





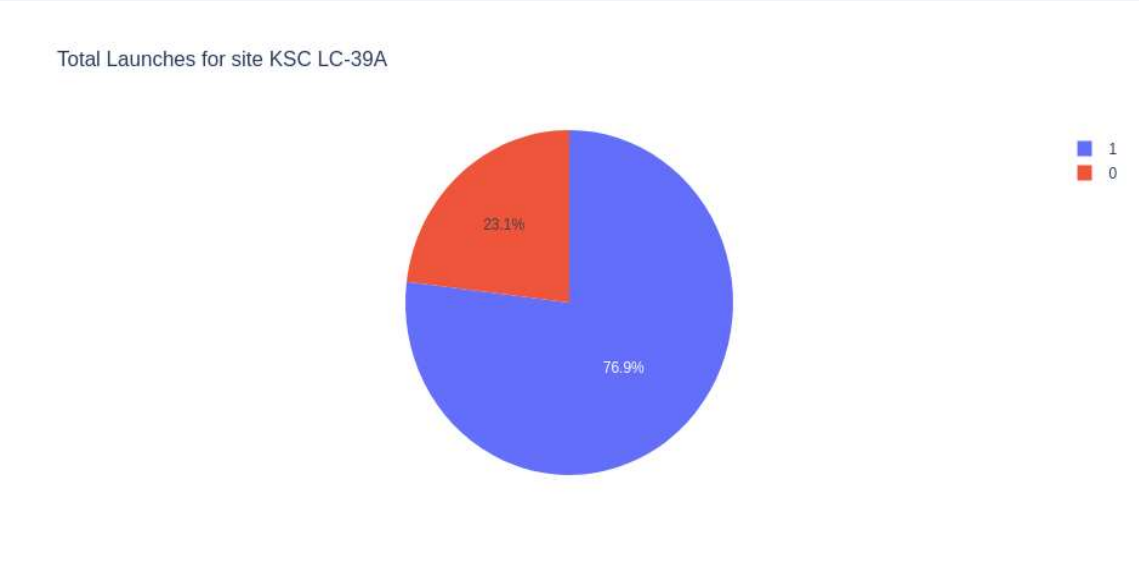

Section 4
Build a Dashboard with Plotly Dash

Dashboard Screenshot 1



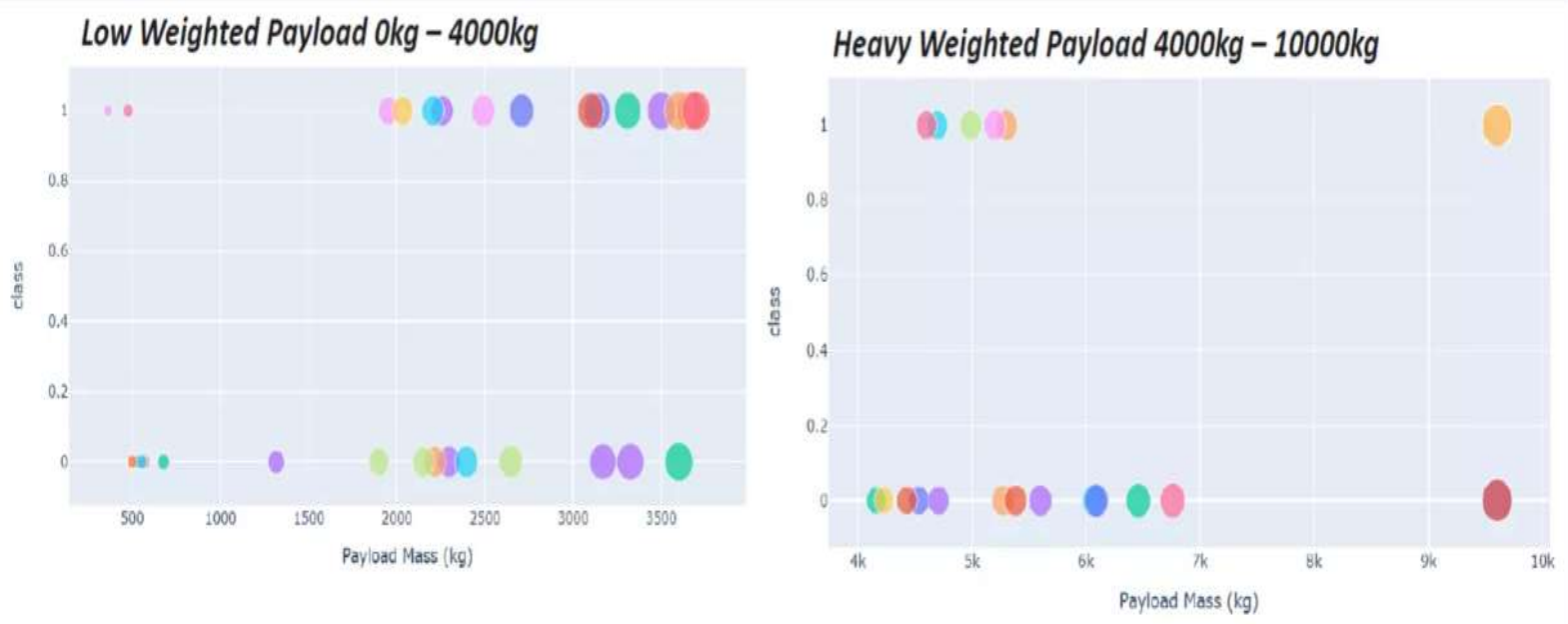
- The place from where launches are done seems to be a very important factor of success of missions.

Dashboard Screenshot 2



- 76.9% of launches are successful in this site.

Dashboard Screenshot 3



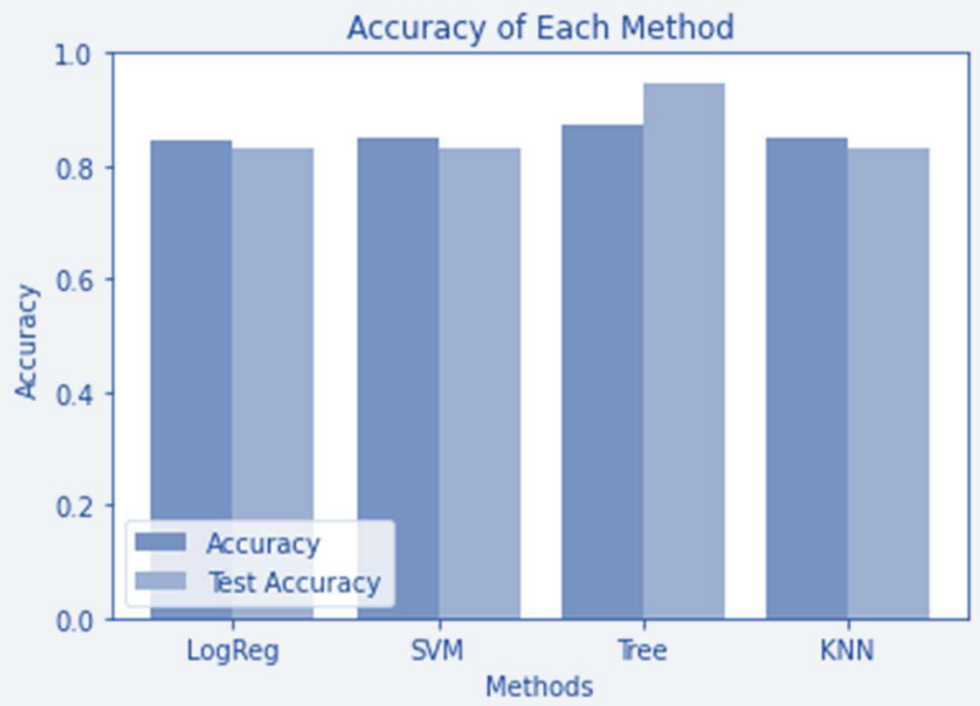
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

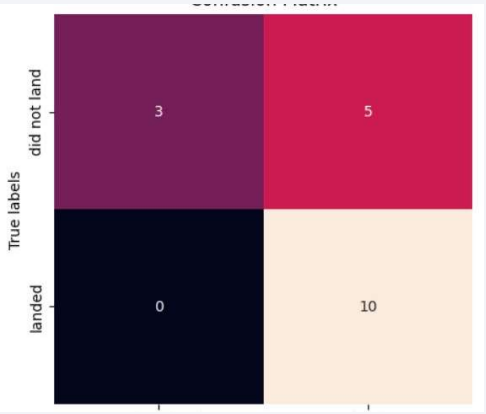
Predictive Analysis (Classification)

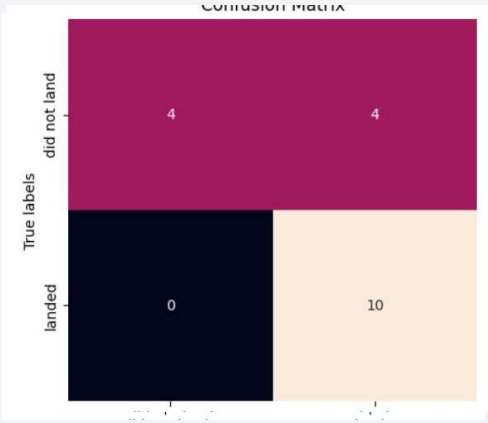
Classification Accuracy

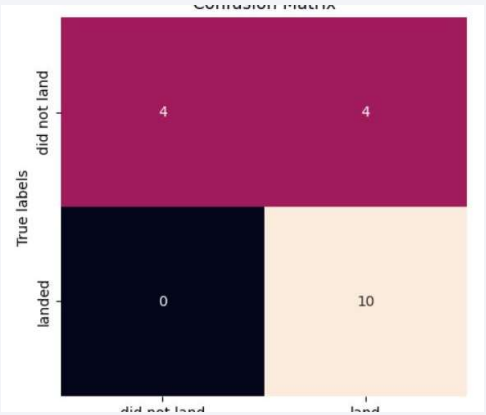
- Four classification models were tested, and their accuracies are plotted beside;
- The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.

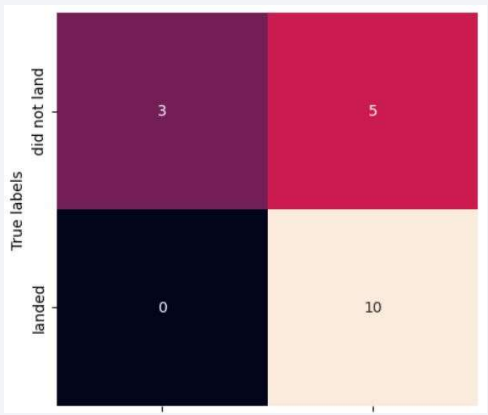


Confusion Matrix









Conclusions

The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.

Low weighted payloads perform better than the heavier payloads.

The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.

KSC LC 39A had the most successful launches from all the sites.

Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

Thank you!