IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ALWaseem M.Ali
04-March-2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis
- Modelling (Classification)
- Evaluation

Summary of all results

- EDA results
- Interactive analytics
- Predictive analysis

Introduction

- **Project background and context**
- SpaceX advertises Falcon9 rocket launches on its website with a cost of 62 million dollars ; other providers cost upwards of 165 million dollars because SpaceX can reuse the first stage .
- **Problems to find answers for :**
- Will the Falcon 9 first stage land successfully ?
- The need to determine the cost of a launch
- How far can we help any other company in case that company wants to bid against SpaceX for a rocket launch ?

Section 1

Methodology

Methodology
Executive Summary

- Data collection methodology:
- The SpaceX API REST
- Web scraping from Wikipedia
- Perform data wrangling
- Web scrap Falcon9 with BeautifulSoup
- Extract a Falcon 9 launch records HTML table from Wikipedia

  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Parse the table and convert it into a Pandas data frame
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly and jupyter-dash
- Perform predictive analysis using classification models
- create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

Data Collection

- Describe how data sets were collected.

Take datasets and use the Rocket , Launchpad ,Payloads ,Cores columns to call the API and append the data to the list

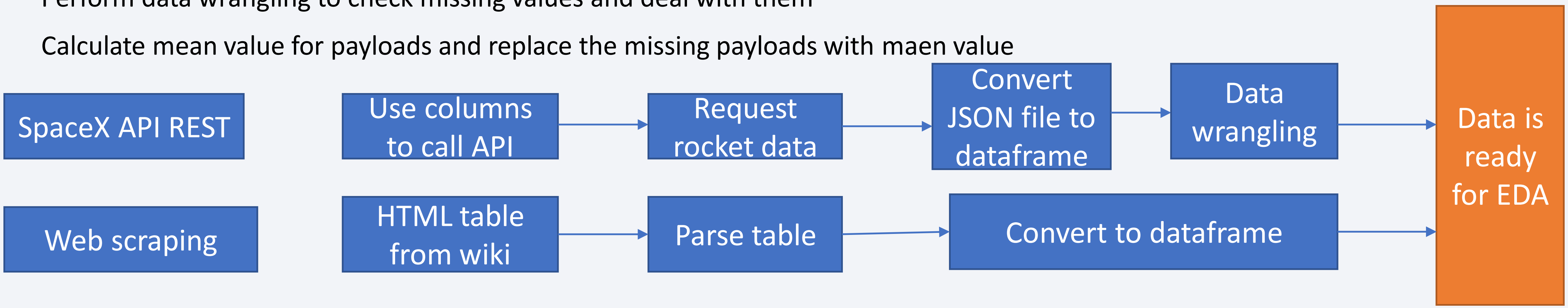
Request rocket launch data from SpaceX API calls using the following URL:

spacex_url=http://api.spacexdata.com/v4/launches/past

Use json_normalize method to convert JSON file to dataframe

Perform data wrangling to check missing values and deal with them

Calculate mean value for payloads and replace the missing payloads with maen value



Data Collection – SpaceX API

https://github.com/alwaseemali/ML-Cappro/blob/master/cappro_data_collection.ipynb

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

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

```
# Call getBoosterVersion
getBoosterVersion(data)
```

the list has now been update

```
BoosterVersion[0:5]
```

```
['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

https://github.com/alwaseemali/ML-Cappro/blob/master/cappro.ipynb

Extract data from wikipage

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

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

Convert to data frame

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

Data Wrangling

Load dataset and read csv file

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

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

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

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

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
# landing_class = 0 if bad_outcome
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
landing_class
```

https://github.com/alwaseemali/ML-Cappro/blob/master/cappro2.ipynb

EDA with Data Visualization

- Charts plotted :
- Flight number vs Payload mass kg (scatter plot)
- Flight number vs Launch sites (scatter plot)

Payload mass kg vs Launch sites (scatter plot)

Orbit vs Success Rate (bar chart)

Flight number vs Orbit (scatter plot)

Payload mass kg vs Orbit (scatter plot)

Years vs Success rate (line plot)

I used these plots and charts to check how far these factors are related to one another so that I be able to extract some beneficial findings

https://github.com/alwaseemali/ML-Cappro/blob/master/cappro4.ipynb

EDA with SQL

- SQL queries performed include:
  - 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.

https://github.com/alwaseemali/ML-Cappro/blob/master/capprosql.ipynb

Build an Interactive Map with Folium

Marked all Launch sites on the map and highlighted circle areas with specific text labels on every specific coordinate

Marked success/failed launches for each site

Calculated the distance between every launch site and its proximities

Drawn line between a launch site and its proximities

Those were added to make some findings about optimum locations for launch sites .

https://github.com/alwaseemali/ML-Cappro/blob/master/cappro5.ipynb

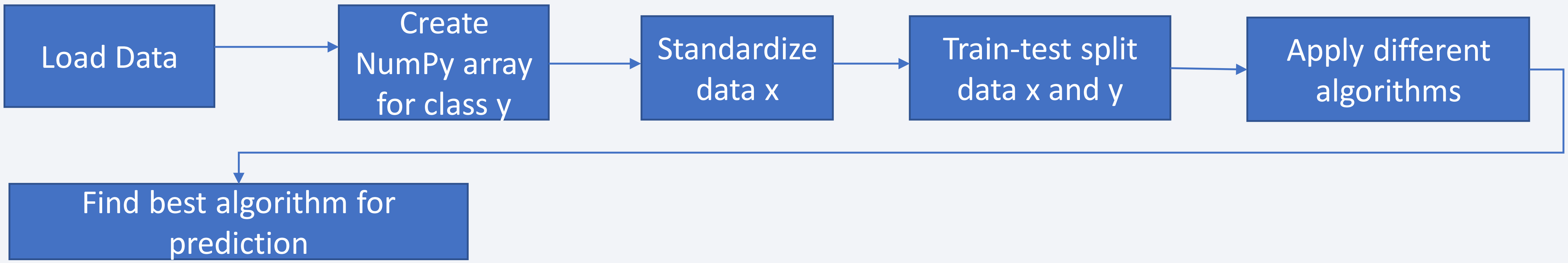
Build a Dashboard with Plotly Dash

- Dashboard built to show success rate for all sites related to one another and the ratio of success to failure for each site all in pie charts
- Also some scatter plots included for payload mass kg vs success rate for all sites and for each site in separate
- The dashboard created to predict which site is the most successful site concerning launch operations taking into consideration the payload mass in kgs .

https://github.com/alwaseemali/ML-Cappro/blob/master/interactive_dashboard.ipynb

Predictive Analysis (Classification)

- Read data (csv file) using pandas
- Create NumPy array from the column class in the data
- Standardize date loaded earlier
- Use train-test split to to split date into train and test sets
- Then apply Logregression,Decision Tree , SVM ,and KNN Algorithms ; calculate accuracy using score and accuracy using confusion matrix to find the algorithm that perform the best



https://github.com/alwaseemali/ML-Cappro/blob/master/ML_cappro.ipynb

Results

- Exploratory data analysis results
- LEO orbit is related to fight number whereas no relationship between GTO and flight number
- CCAFS SLC 40 is the most successful launch site in terms of heavy pay loads
- KSC LC 39A preforms the best in terms of light payloads
- GTO orbit has the worst success rate whereas ES-L1, HEO and SSO have the best record of success rate

Predictive analysis results

Algorithms applied where equal in terms of accuracy in train-test split technique but when it comes to cross validation

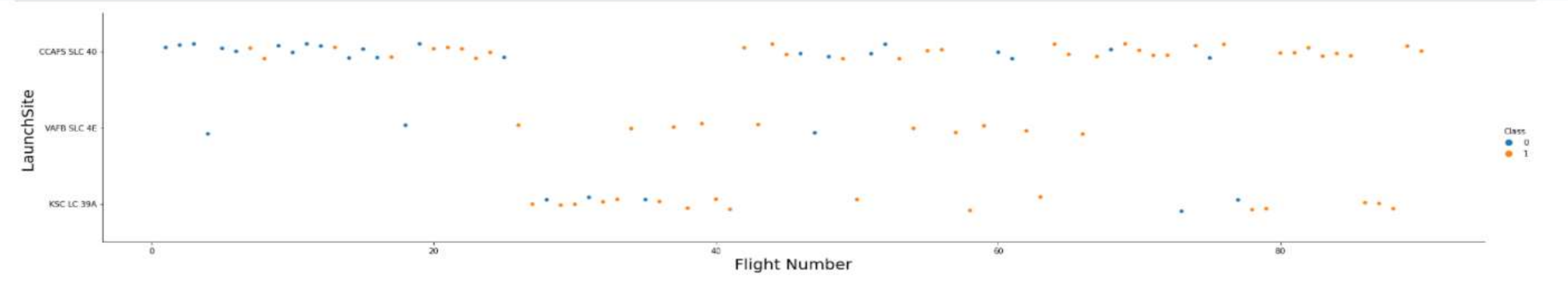
- We will find that Decision tree has the best accuracy among all .

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

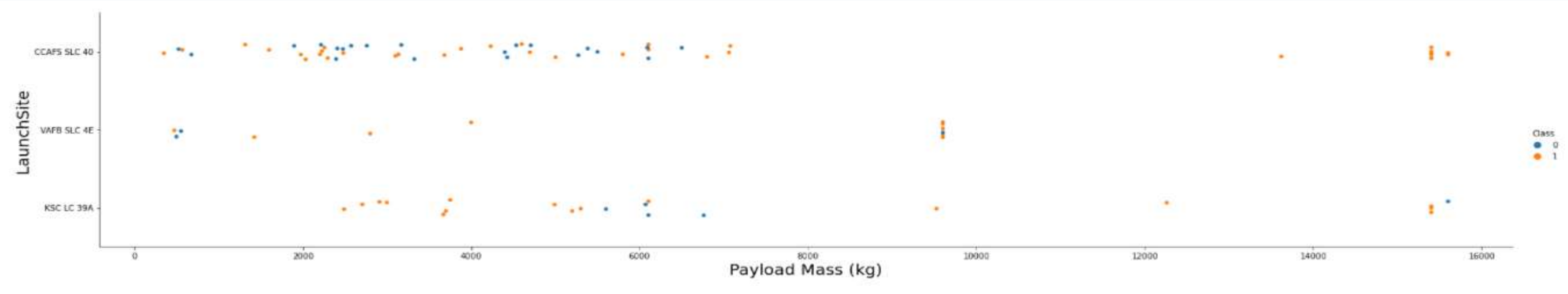
- A scatter plot of Flight Number vs. Launch Site



- Yellow dots refer to successful launching operations whereas blue dots refer to failed launching operations all as per launch site .

Payload vs. Launch Site

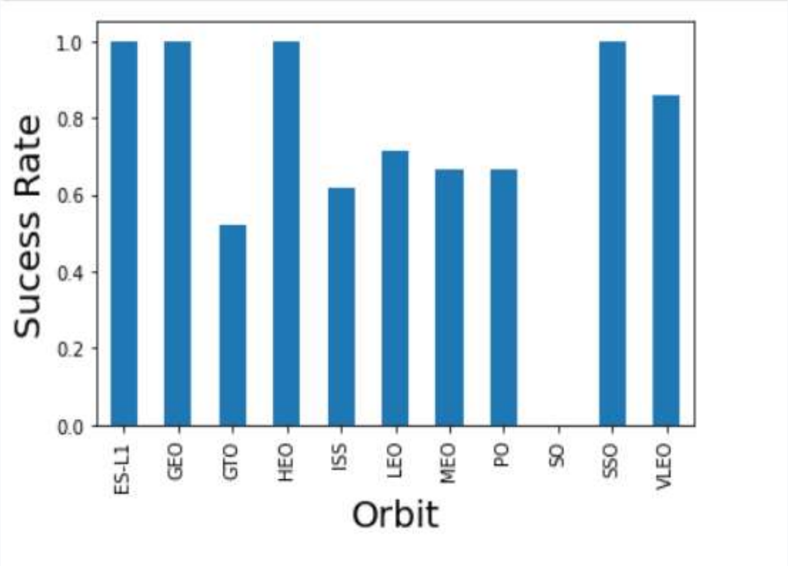
- A scatter plot of Payload vs. Launch Site



- Yellow dots refer to successful launching operations whereas blue dots refer to failed launching operations all as per launch site .

Success Rate vs. Orbit Type

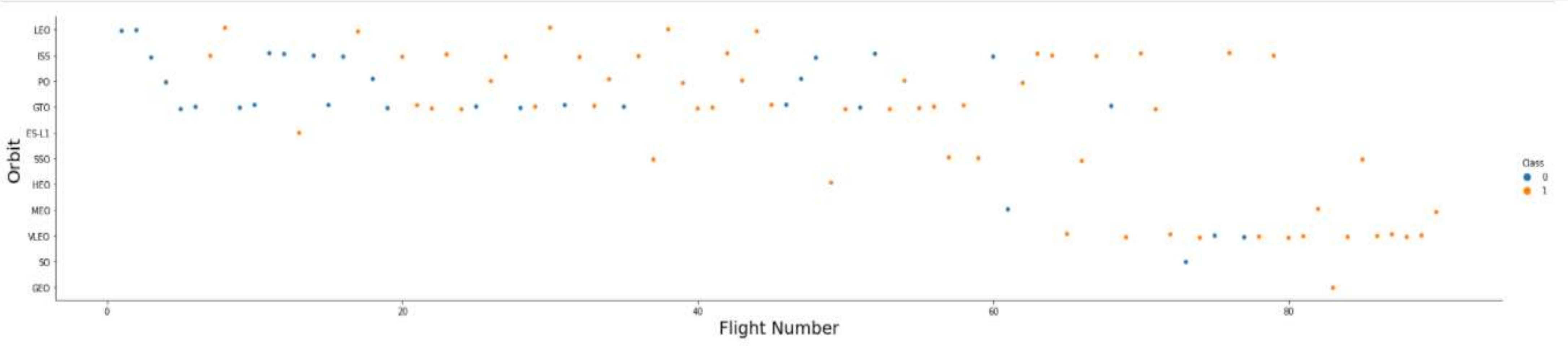
- A bar chart for the success rate of each orbit type



- It is clear that the longer the bar , the better the success rate for an orbit

Flight Number vs. Orbit Type

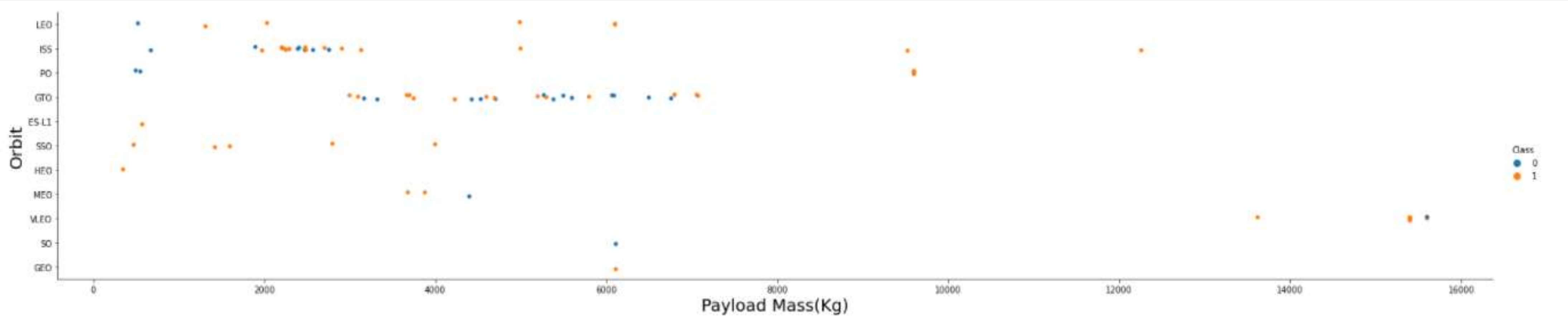
- A scatter point of Flight number vs. Orbit type



- Yellow dots refer to successful launching operations whereas blue dots refer to failed launching operations all as per orbit type. .

Payload vs. Orbit Type

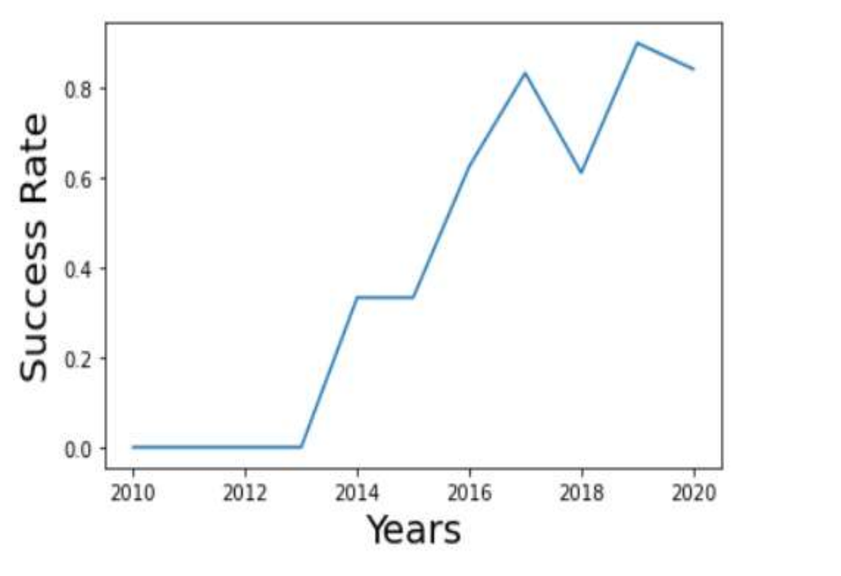
- A scatter point of payload vs. orbit type



- Yellow dots refer to successful launching operations whereas blue dots refer to failed launching operations all as per orbit type in relation to payload mass in kgs. .

Launch Success Yearly Trend

- A line chart of yearly average success rate



- The line refers to the relationship between years and success rate which increases significantly since 2013

All Launch Site Names

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Launch Site Names Begin with 'KSC'

- 5 records where launch sites' names start with `KSC`

```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'KSC%' LIMIT 5
```

```
* ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

Total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
2928

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
101

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2015

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

month_name	landing__outcome	booster_version	launch_site
JANUARY	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
APRIL	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Ranking of the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

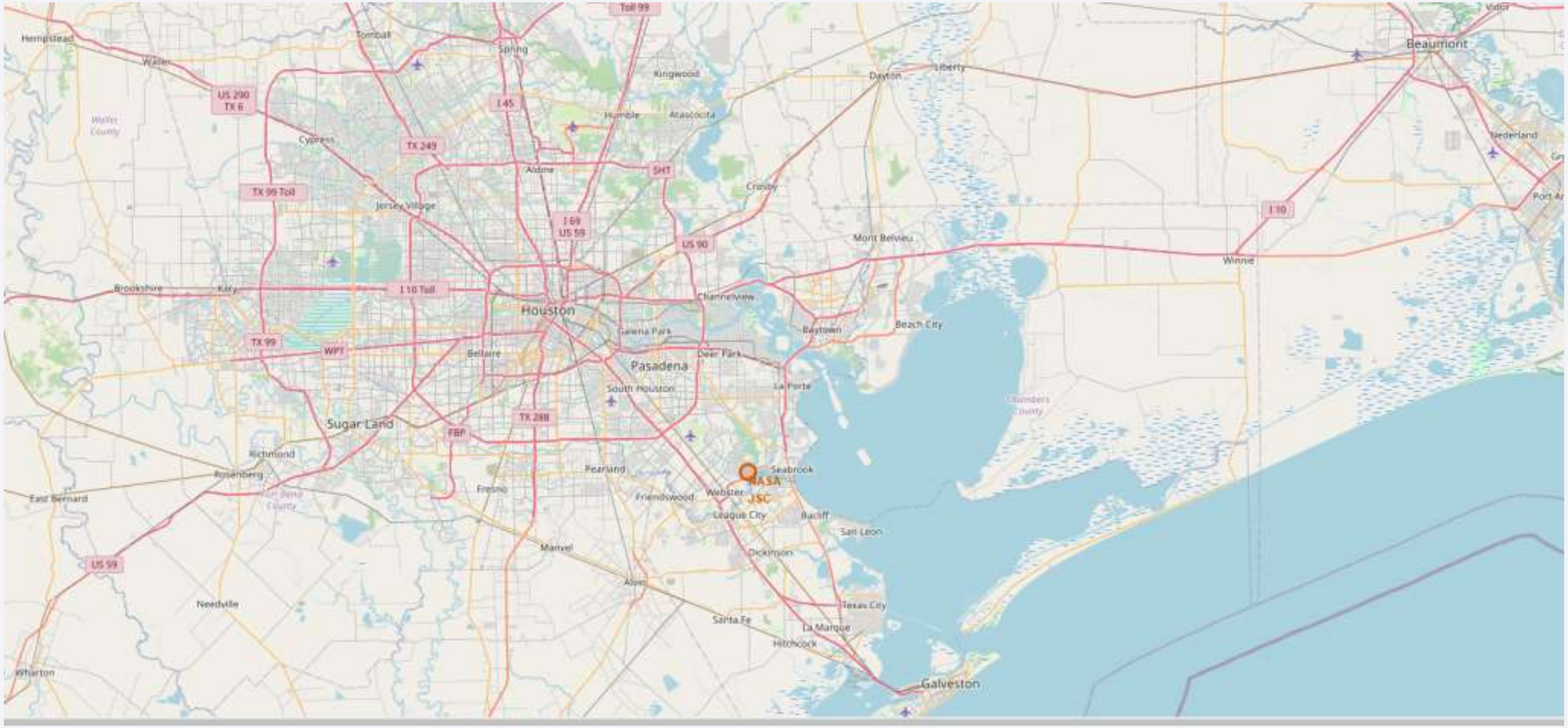
```
   ibm_db_sa://xhb09638:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
 * ibm_db_sa://xtp32429:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

DATE	COUNT
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

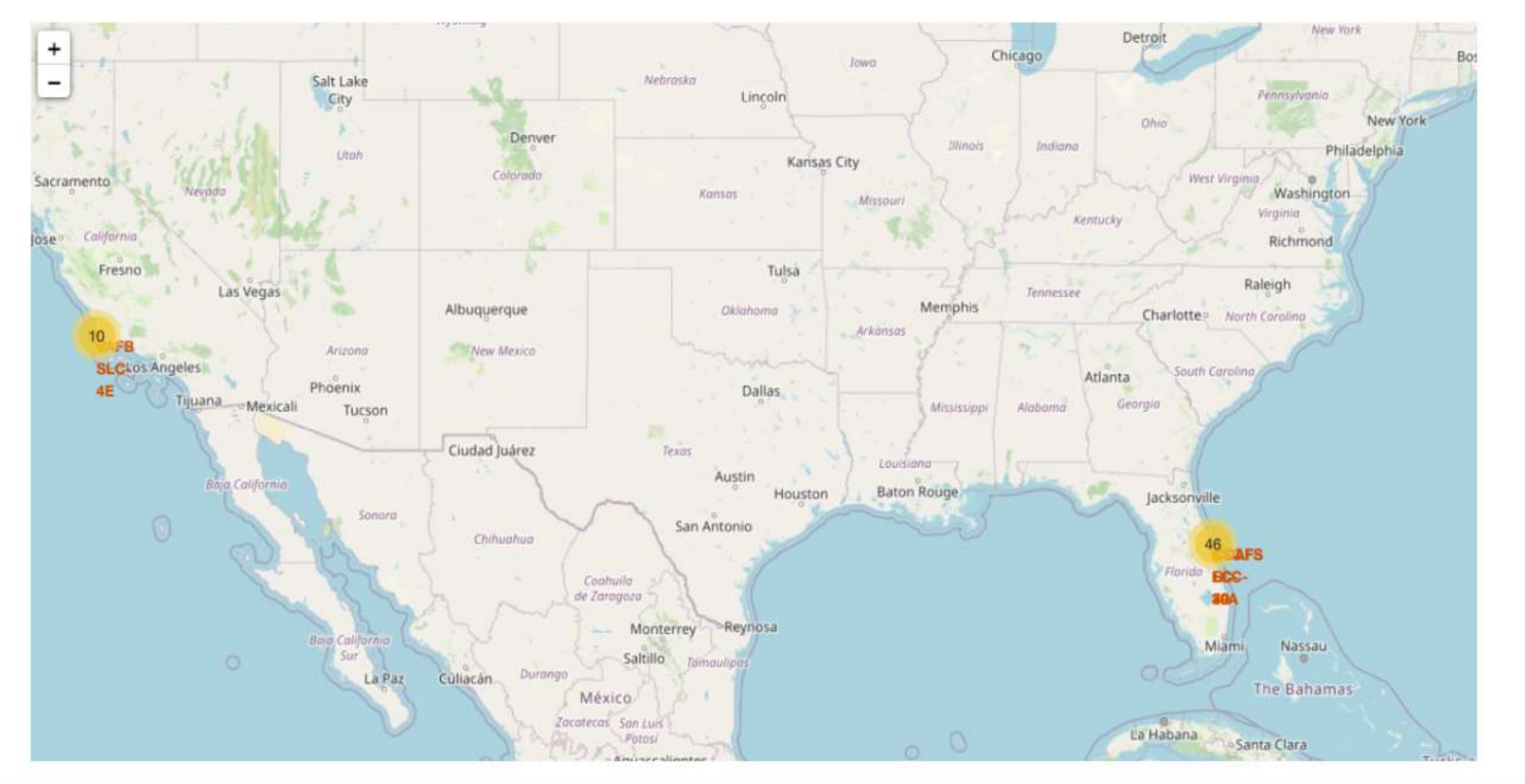
Section 3

Launch Sites Proximities Analysis

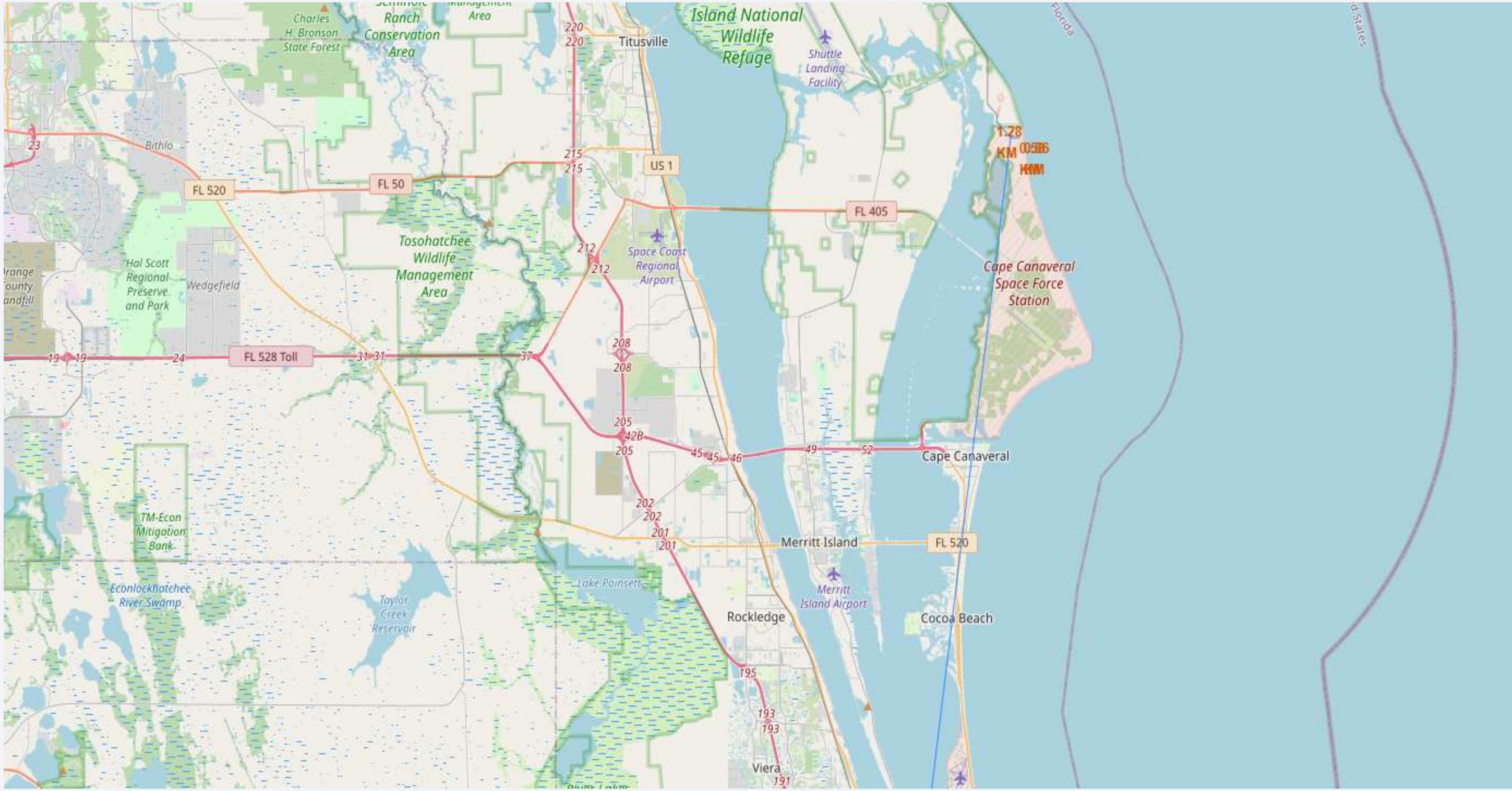
Launch sites marked on a map



Success/failed launches marked on the map



Distances between a launch site to its proximities

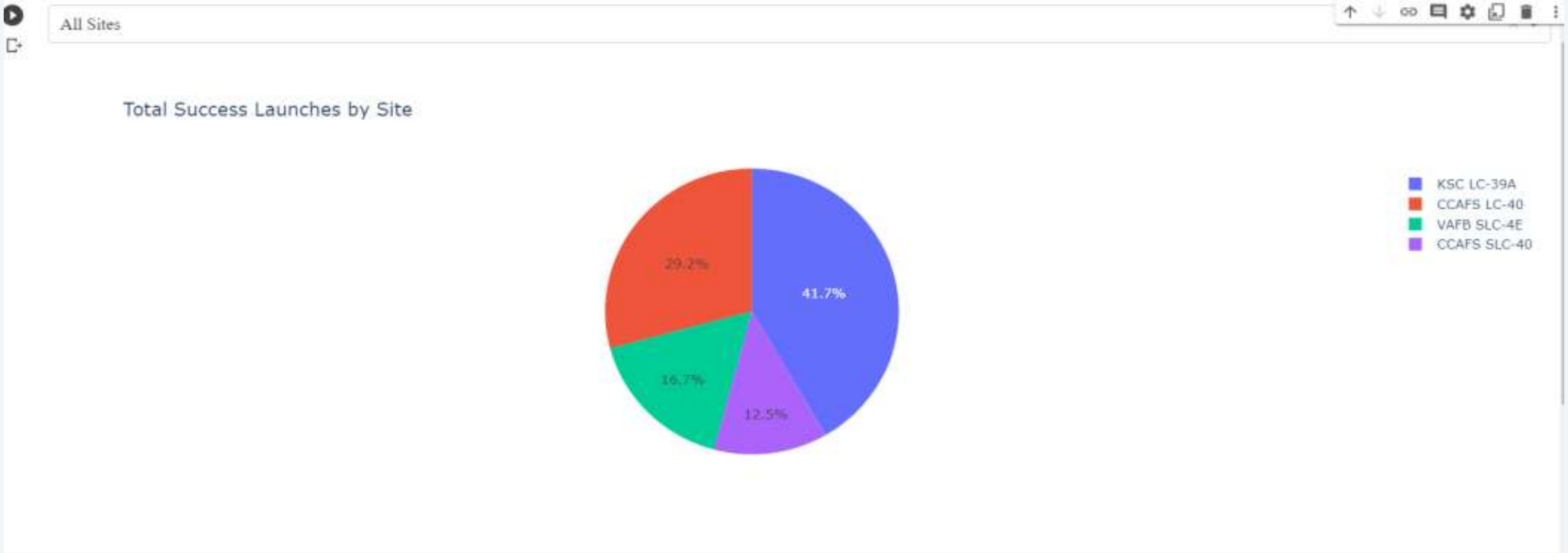


Section 4

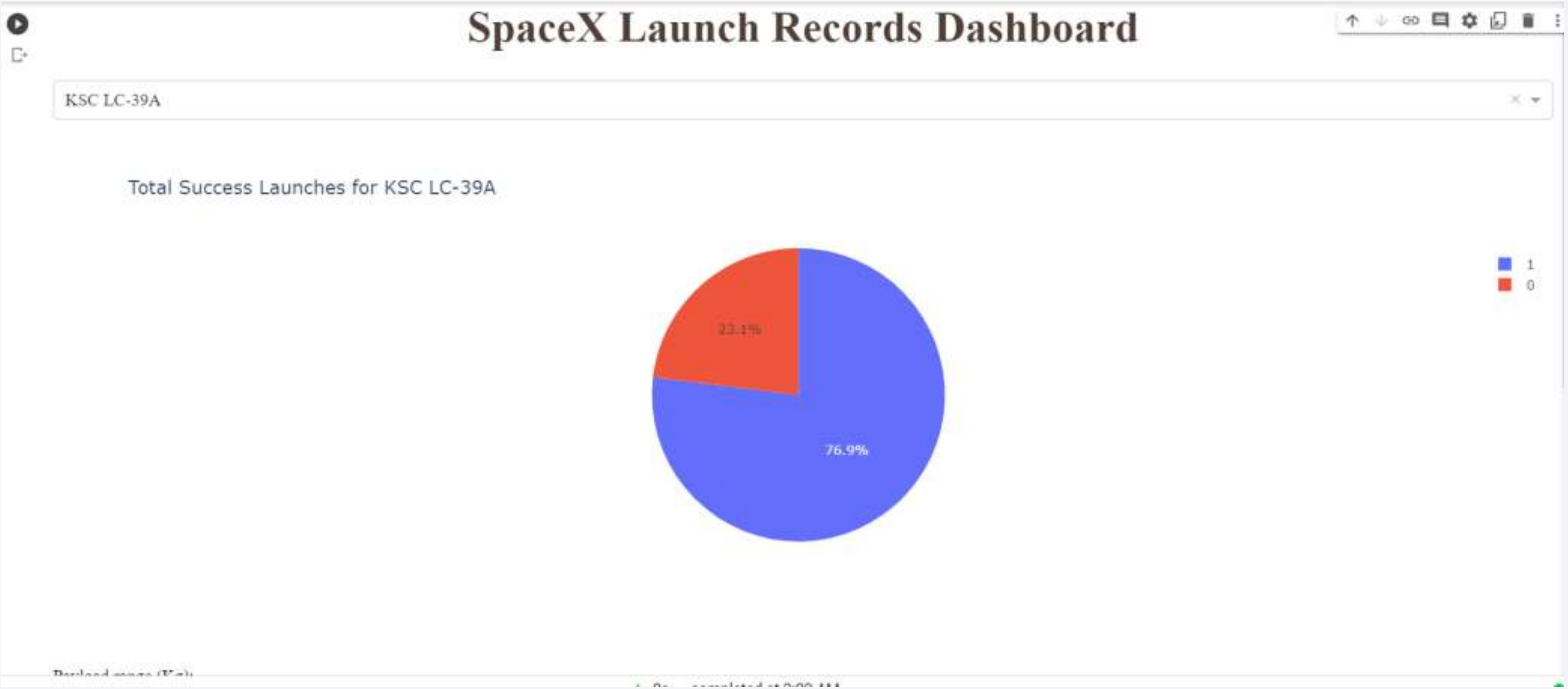
Build a Dashboard with Plotly Dash

Total success launches by all sites

- KSC LC 39A posses the highest success rate for launches with 41.7%

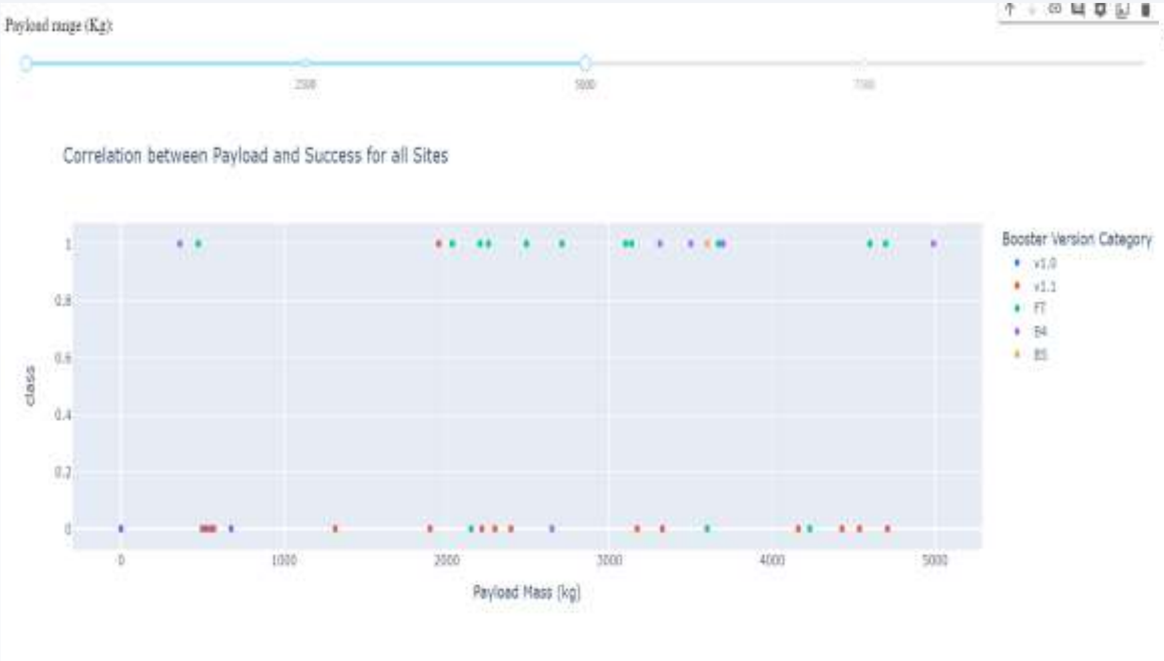


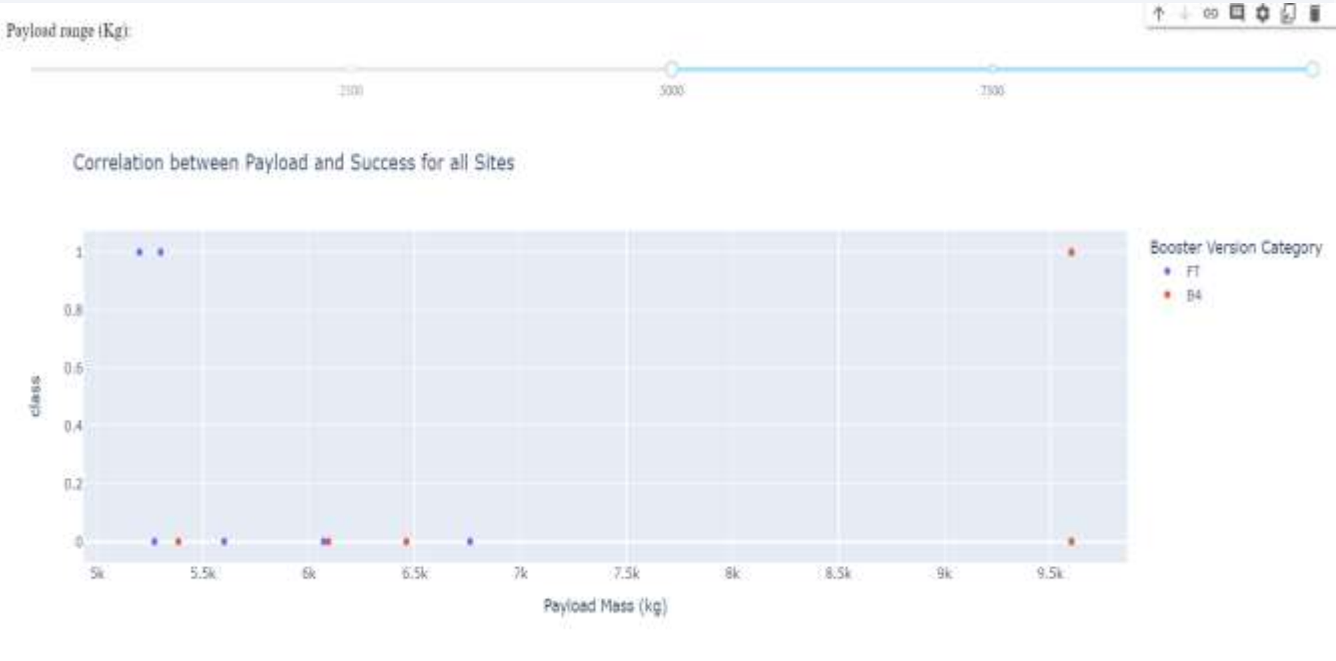
KSC LC-39A most successful launch site



Different payloads ranges vs success launches per site

As for high payload mass more than 5000 kg ; B4 booster version is the most successful then FT is runner up whereas again FT performs the best for low payload mass below 5000 kg



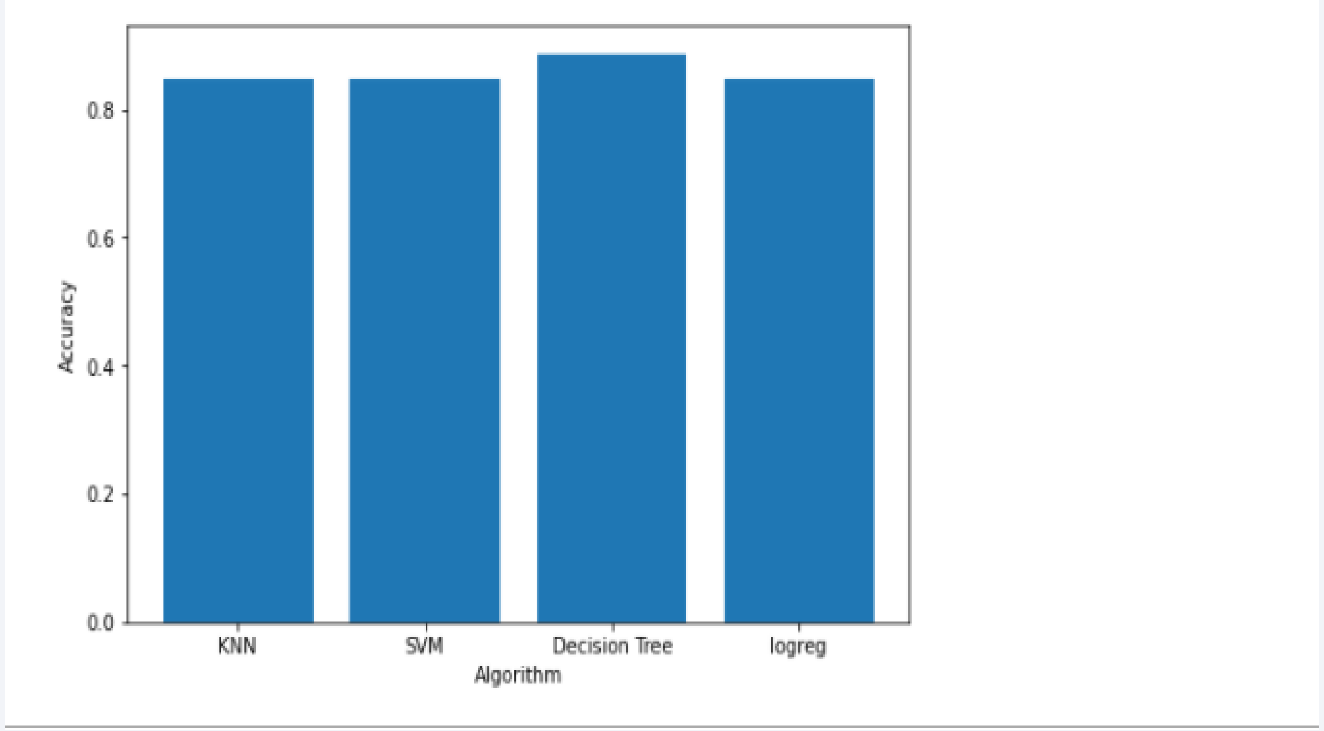


Section 5

Predictive Analysis (Classification)

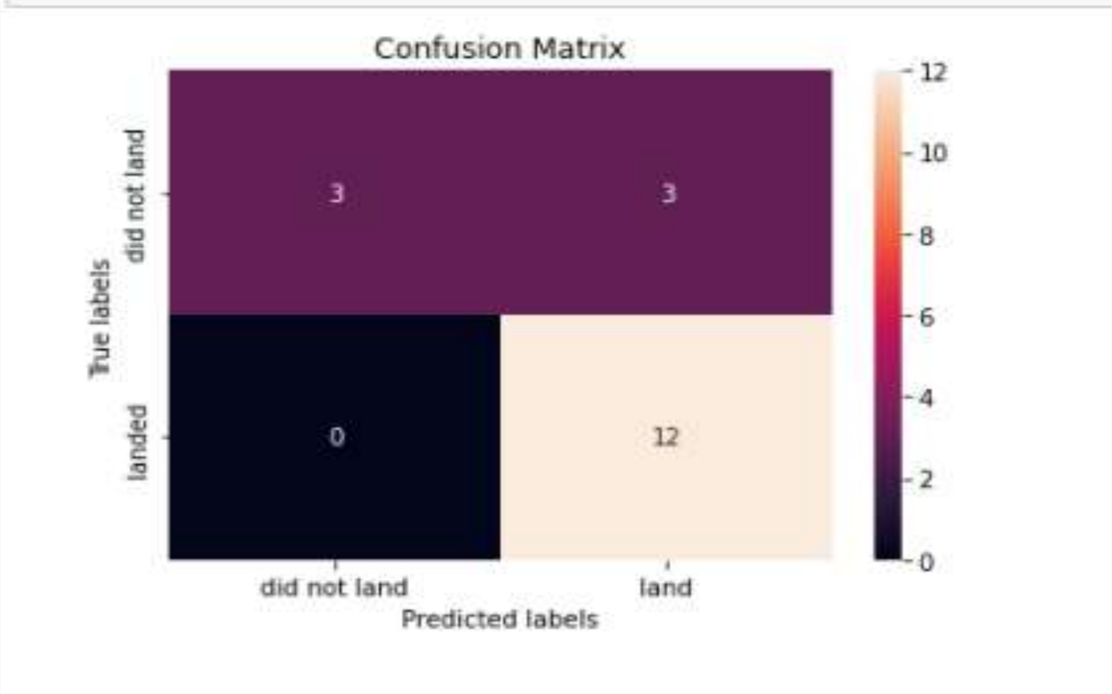
Classification Accuracy

- As shown , **Decision Tree** performs the best .All other algorithms performs well though



Confusion Matrix

Decision Tree as said before performs the best



Conclusions

1-LEO orbit is related to flight number whereas no relationship between GTO and flight number

2-CCAFS SLC 40 is the most successful launch site in terms of heavy pay loads

3-KSC LC 39A performs the best in terms of light payloads

4-KSC LC 39A posses the highest success rate for launches with 41.7%

5-GTO orbit has the worst success rate whereas ES-L1, HEO and SSO have the best record of success rate

6-As for high payload mass more than 5000 kg ; B4 booster version is the most successful then FT is runner up whereas again FT performs the best for low payload mass below 5000 kg

7-Launch success rate increased significantly after 2013

8-With heavy payloads the successful landing or positive landing rate are more of Polar for LEO and ISS.

9-Launch sites in close proximity to railways, highways, and coastline and this helps facilitate people and heavy cargo transportation to and from launch sites and good in terms of safety regulations for proposes of landing on water attempts be it success or failure besides these launch sites are far enough from cities which is good for the safety of population.

10-Algorithms applied where equal in terms of accuracy in train-test split technique and all perform well but when it comes to cross validation technique we find that Decision tree has the best accuracy among all .

Appendix

- All codes performed on Watson studio and uploaded to GitHub (refer to GitHub links attached into this presentation) except for building an interactive dashboard which was performed on colab google research.
- As for interactive dashboard , jupyter-dash installed and imported instead of dash

https://github.com/alwaseemali/ML-Cappro

Thank you!