IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

M Bhasker
03 Dec 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API's
  - Data Wrangling – To clean data
  - Exploratory Data Analysis – EDA
    - Exploratory Analysis with SQL
    - Exploratory Analysis with Visualization
    - Interactive Visual Analytics with Folium
    - Build interactive Dashboard with Ploty Dash
  - Machine Learning Predictions
- Summary of all results
  - Clean data final output
  - Exploratory Analysis Results
  - Screen shots – interactive analysis and Visual analytics
  - Prediction results

Introduction

- Project background and context

Commercial space voyage is a fast-growing sector. Existing providers launch costs upwards of $165m for a launch. One of the space company - SpaceX advertises $62m depending on the re-usability of first stage. If the first stage successfully lands, then we can determine the cost. The goal of the project to create a Machine Learning module to predict if the first stage will land successfully.

- Problems you want to find answers
    - ➢ Determine price of each launch
    - ➢ Determine if we can reuse the first stage
    - ➢ Train a Machine Learning model to predict

Section 1

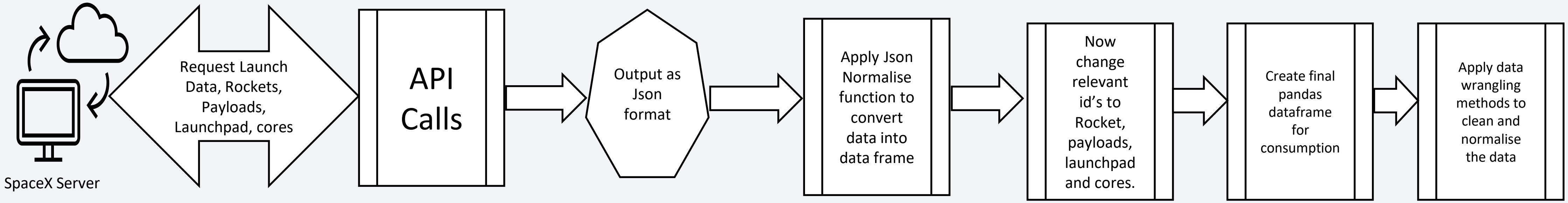
Methodology

Executive Summary

- Data collection methodology:
  - The data is accessed using API calls from SpaceX website.
- Perform data wrangling
  - The data is cleaned using Python's Pandas library and its function
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

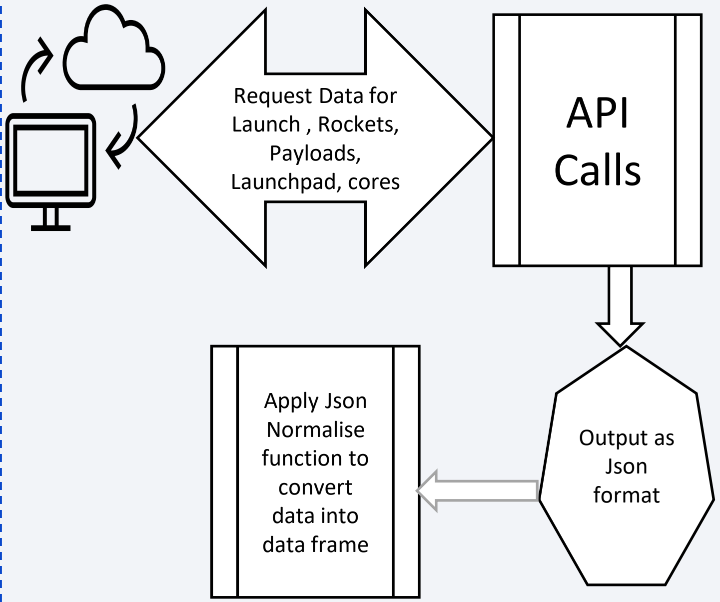
Data Collection

- Describe how data sets were collected.
  - The data is collected using Request python library which support manipulation of Json formatted source files. Once the file is successfully downloaded then it is converted into Pandas dataframe using Json normalize method.
- You need to present your data collection process use key phrases and flowcharts



Data Collection – SpaceX API

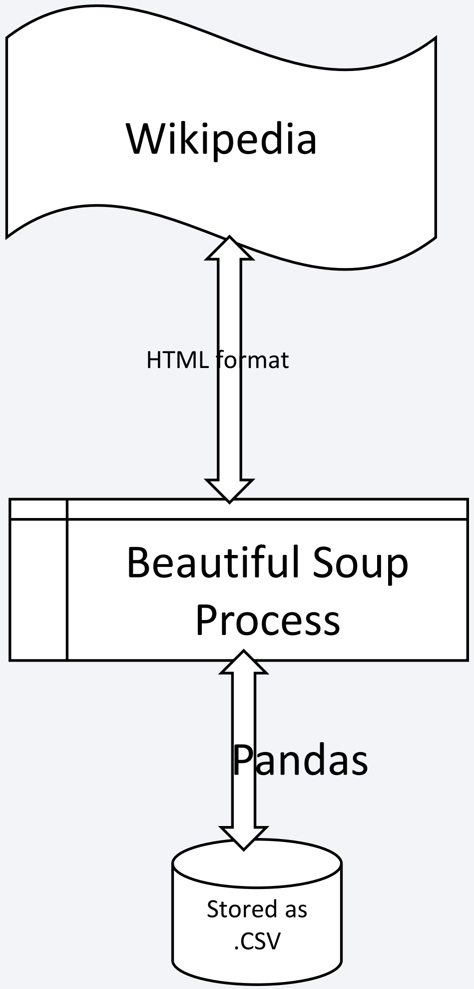
- Request library allows us to make HTTP requests to SpaceX using API calls.
- We also define helper function to extract information using identification numbers in the launch data. These helper function fetches rocket, launchpad, payload and cores information and append the data list.
- Once this information has successfully retrieved then we request rocket launch data from SpaceX which is received as .Json formatted file.
- Once the .Json formatted launch data is received then we use Pandas Json_normalize method the data into a Pandas dataframe.
- https://github.com/bhasker63/IBM-Data-Science/blob/e4f5e0e15ce5fa86ab9bb0b92e59aac9a10c91d9/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- We applied web scrapping to collect data of Falcon 9 launch records with BeautifulSoup library.
- Converted the html table and converted into a dataframe and then saved it as a .csv file for further processing.

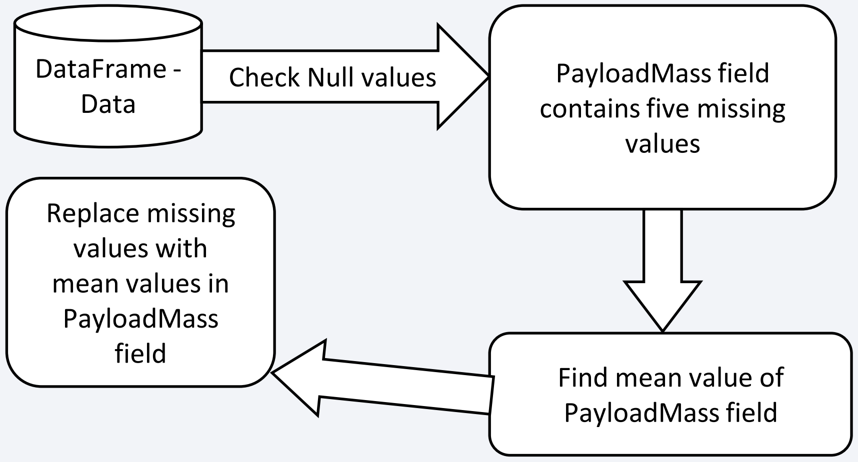
- https://github.com/bhasker63/IBM-Data-Science/blob/b35ccfe63943d26150027ae5dd3684315d6ad0fc/jupyter-labs-webscraping.ipynb

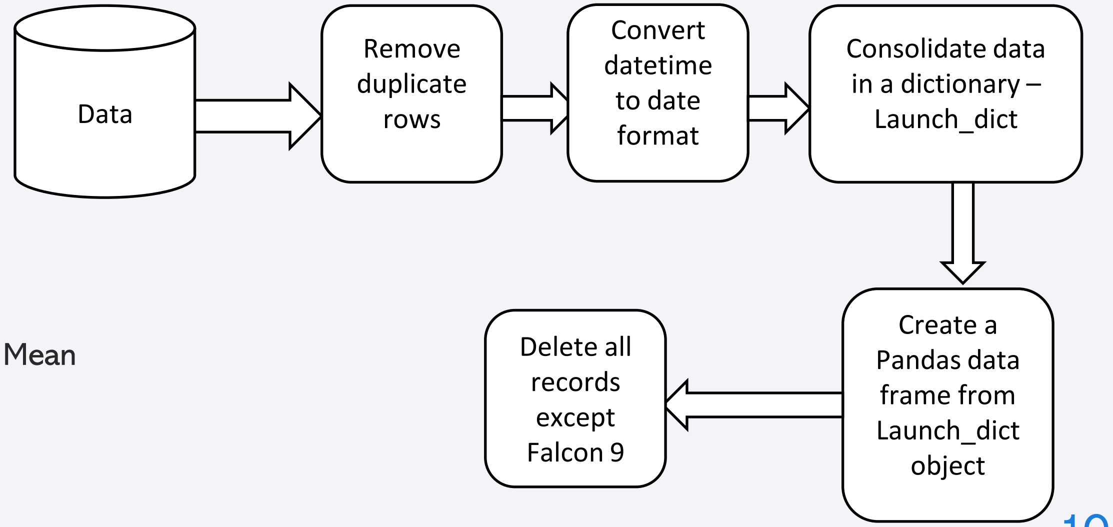


Data Wrangling

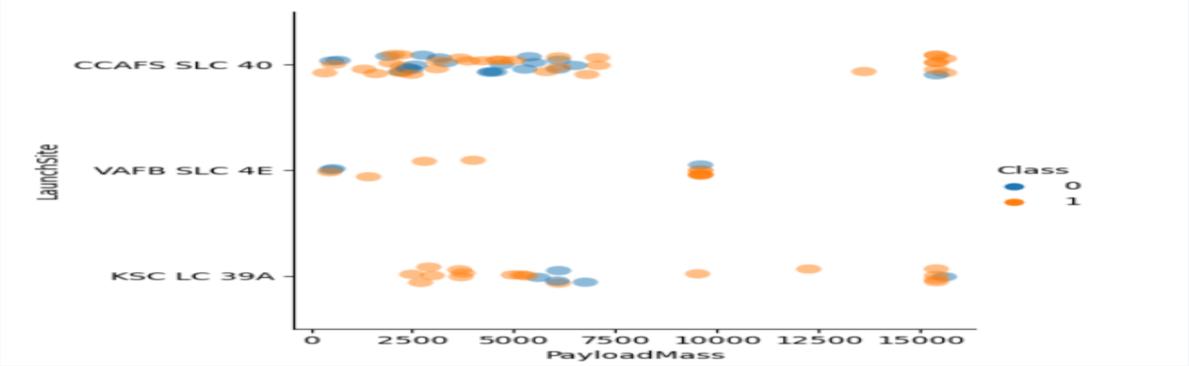
- We remove duplicate rows e.g., falcon rockets contain two extra rocket boosters and rows that have multiple Payloads.
- The Launchdata we received is normalized and contains IDs for rocket Payloads etc.
- We convert date to datetime data type using Pandas to_datetime() function to a date removing the time element. We also restrict the dates of the launches.
- We call functions like GetBooseterVersion to consolidate the data.
- We construct our dataset using the columns we have received. We combine the columns into Launch_dict.
- We created a Pandas data frame from the launch_dict.
- We keep only Falcon 9 launches deleting other launches.
- Data is processed using Pandas functions.
- We find columns with NULL values by using isNULL().SUM() Pandas function.
- We found PayloadMass filed contains five null values, we replaced these null values with Mean value of this column using .replace(np.nan, fieldname) function.

- https://github.com/bhasker63/IBM-Data-Science/blob/b35ccfe63943d26150027ae5dd3684315d6ad0fc/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



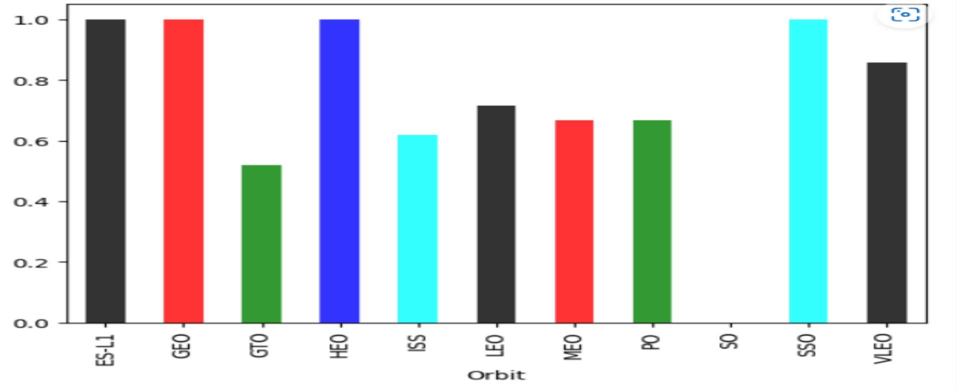


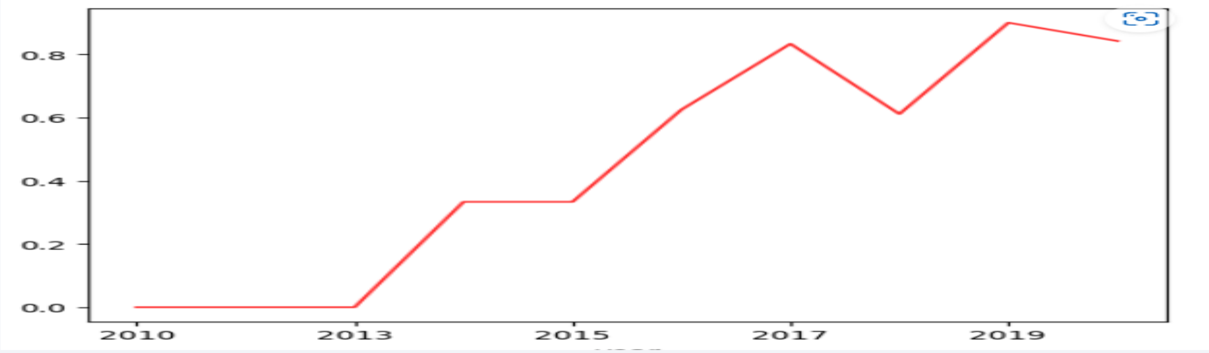
EDA with Data Visualization



Launch site : CCAFS ALC 40 has more flight with success indicated by class 1

CCAFS SLC 40 launch site has launched <7500 Kgs load than other launch sites also launched more 15k load then other launch sites





- Following orbit (ES-L1,GFO,HEO, and SSO) are more successful

Even though initial years the success were scarce but it picked up in later years

- https://github.com/bhasker63/IBM-Data-Science/blob/b35ccfe63943d26150027ae5dd3684315d6ad0fc/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Following SQL were performed to get the insights : (Due to constant crashing of Cognitive Class Labs site, I used Microsoft SQL to perform this operation)
  - TASK 1: Display the names of the unique launch sites in the space mission
  - TASK 2 – Display 5 records where launch sites begin with the string 'KSC'
  - TASK 3 - Display the total payload mass carried by boosters launched by NASA (CRS)
  - TASK 4 - Display average payload mass carried by booster version F9 v1.1
  - Task 5 - List the date where the successful landing outcome in drone ship was achieved
  - TASK 6 - List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - TASK 7 - List the total number of successful and failure mission outcomes
  - TASK 8 - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - TASK 9 - List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
  - TASK 10 - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

- https://github.com/bhasker63/IBM-Data-Science/blob/b35ccfe63943d26150027ae5dd3684315d6ad0fc/jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

- Using Folium library we added maps, objects such as markers , circles, lines to mark the success or failure of launches, distance to the landmarks, identify launching sites.
- We marked success/failures on each launch outcome for each sites.
- We calculated distance to nearest landmark like railways and displayed the results
- We were able to identify proximity of the launch sites to Equator.

- https://github.com/bhasker63/IBM-Data-Science/blob/9cbe99272d6f3fe5b4bf61a4606ff5bba8cbedae/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

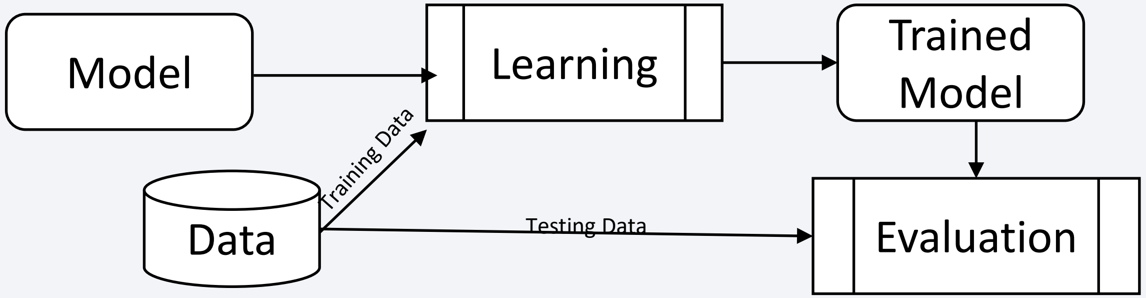
Build a Dashboard with Plotly Dash

- We built interactive dashboard with Plotly dash
- We plotted Pie chart with show successful/failed outcome of individual sites and on a high-level percentage outcome.
- We used slider to adjust the payload and display the results for different booster versions

- https://github.com/bhasker63/IBM-Data-Science/blob/9cbe99272d6f3fe5b4bf61a4606ff5bba8cbedae/Capstone%20Dash%20report.ipynb

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, split our data into testing and training data set.
- We built four machine learning models and found different the best parameters from the dictionary parameter.
- We found accuracy of the models using Score method and best performing classification model



- https://github.com/bhasker63/IBM-Data-Science/blob/9cbe99272d6f3fe5b4bf61a4606ff5bba8cbedae/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(3).ipynb

Results

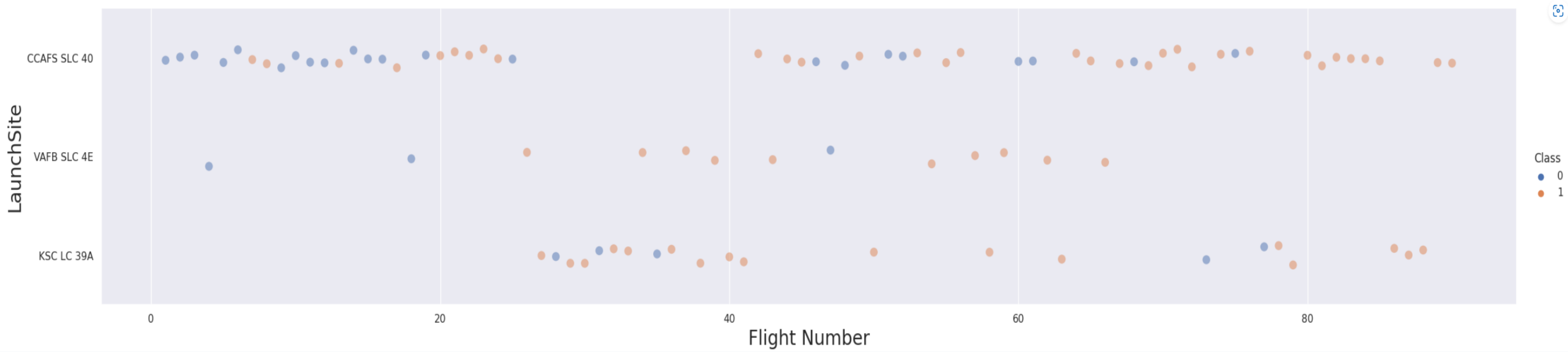
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

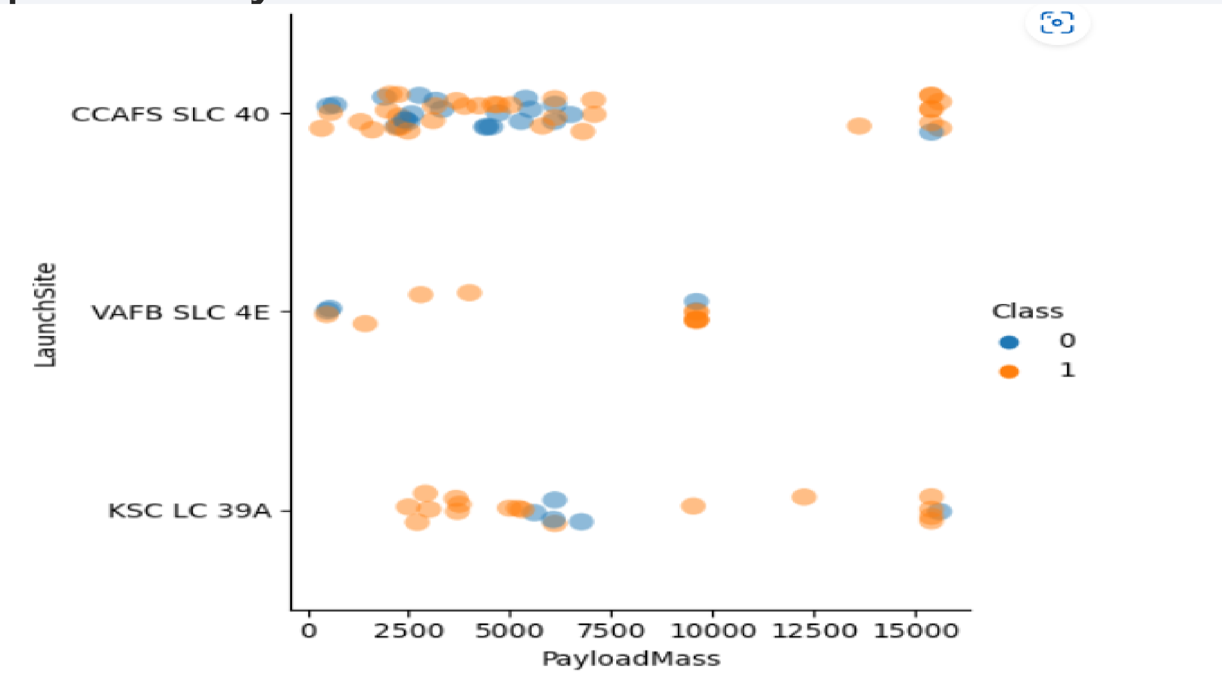
- Show a scatter plot of Flight Number vs. Launch Site



- Larger the number of flight launch greater the success.

Payload vs. Launch Site

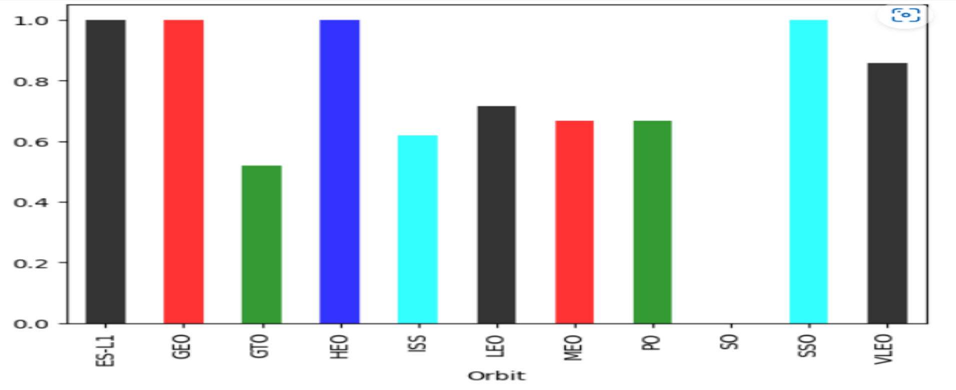
- Show a scatter plot of Payload vs. Launch Site



- The greater the payload mass for launch site CCAFS SLC 40, higher the rate of success.

Success Rate vs. Orbit Type

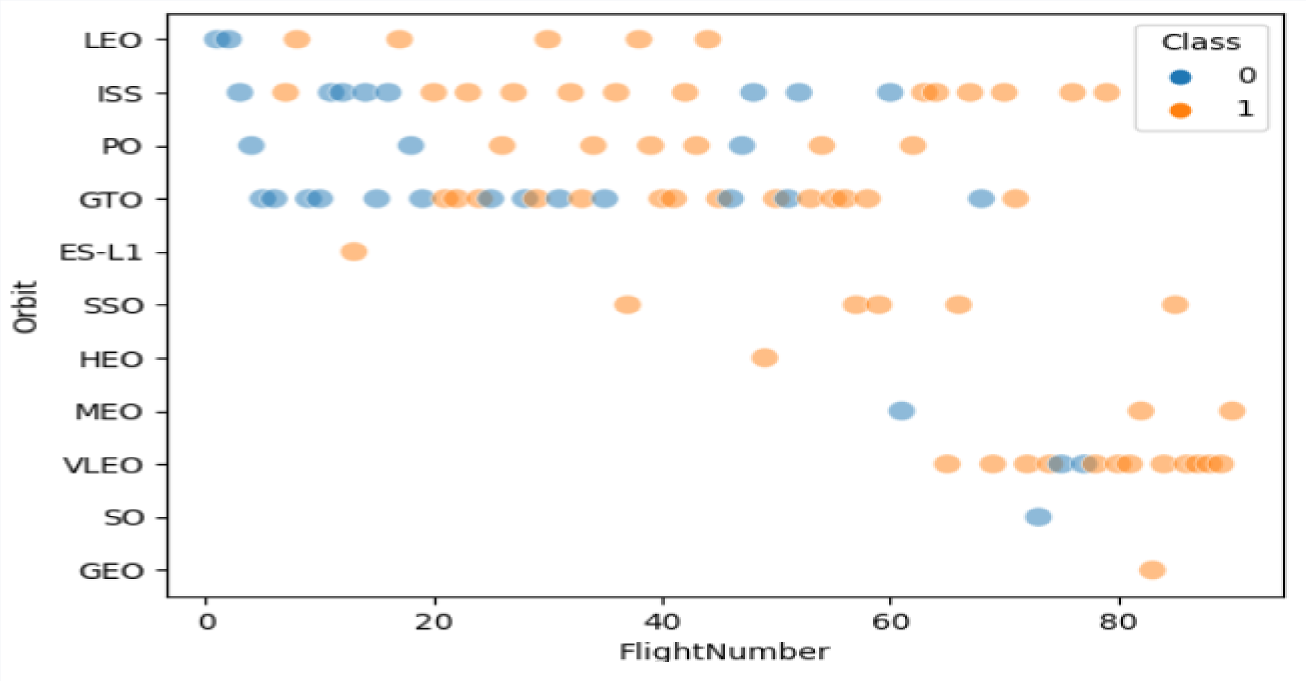
- Show a bar chart for the success rate of each orbit type



- Following orbit (ES-L1,GFO,HEO, and SSO) are more successful than others.

Flight Number vs. Orbit Type

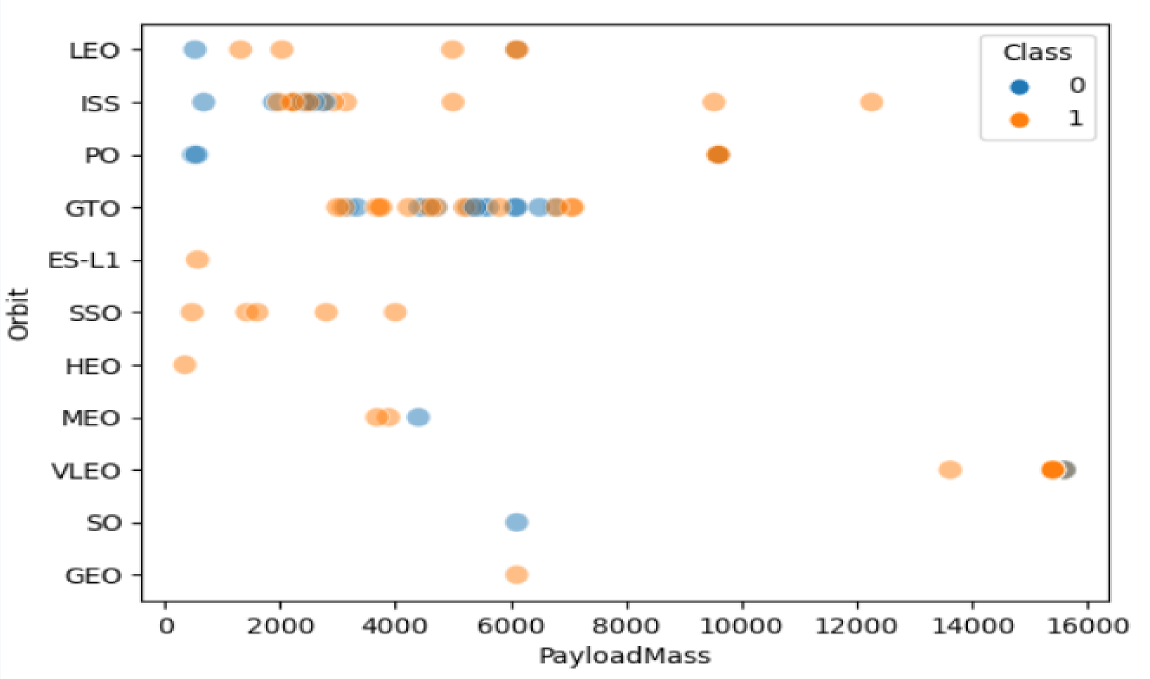
- Show a scatter point of Flight number vs. Orbit type



- VLEO has more launch more success, LEO orbit success is related to the number of flights .

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

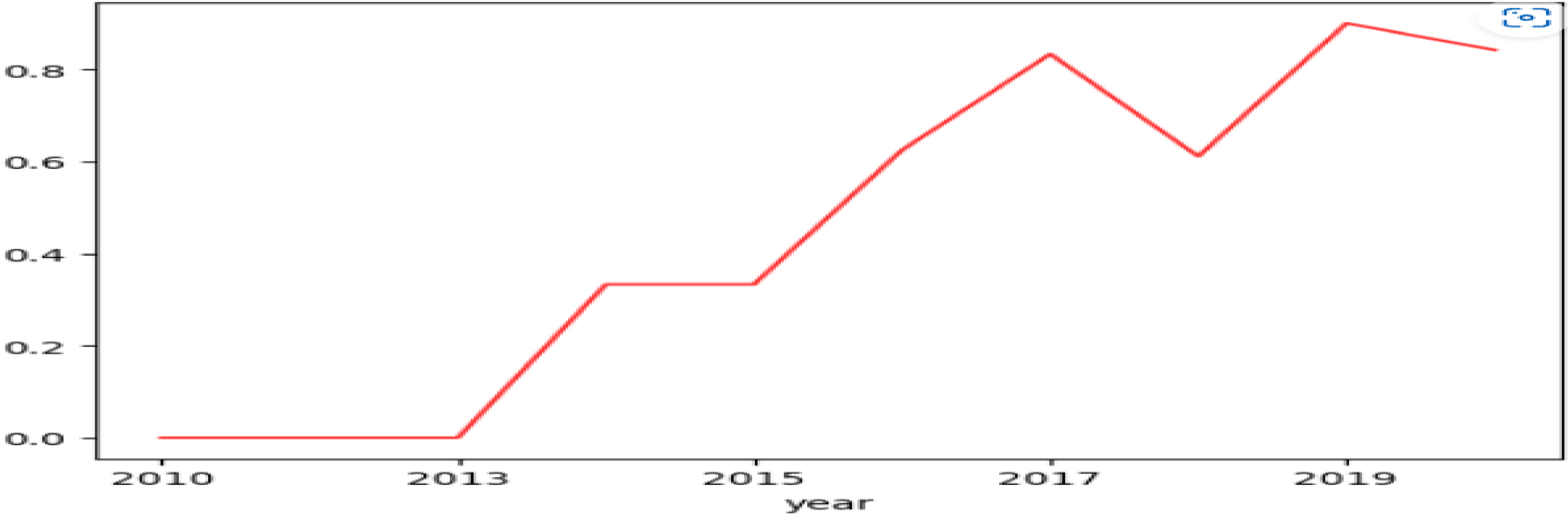


- PO, LEO and ISS orbits have successful landing for heavy payloads.

Launch Success Yearly Trend

Show a line chart of yearly average success rate

We can observe success rate were higher from year 2013.



All Launch Site Names

- Find the names of the unique launch sites
- Four launch sites were identified (CCAFS LC-40, VAFB SLC-4E, KSC LC-39A,CCAFS SLC-40)

Task 1

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

```
# 1Display the names of the unique launch sites in the space mission

rows = cur.execute("SELECT DISTINCT (Launch_site) FROM spacextbl").fetchall()
print(rows)

# to get column names
#rows = cur.execute("SELECT * FROM spacextbl")
#print(rows.description)
```

```
[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- We were able to fetch records successfully based on the request.

Task 2

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

```
#2 Display 5 records where launch sites begin with the string 'KSC'

rows = cur.execute("SELECT * FROM  spacextbl WHERE Launch_site LIKE 'KSC%' Limit 5").fetchall()
print(rows)
```

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- The result we got was 45,596 Kg.

Task 3

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

```
rows = cur.execute("SELECT SUM(PAYLOAD_MASS__KG_) FROM spacextbl WHERE CUSTOMER = 'NASA (CRS)'").fetchall()
print(rows)
```

```
[(45596,)]
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The average Payload mass in Kg is 2928.4kgs.

Task 4

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

```
rows = cur.execute("SELECT AVG(PAYLOAD_MASS__KG_)  FROM spacextbl WHERE Booster_Version = 'F9 v1.1'").fetchall()
print(rows)
```

```
[(2928.4,)]
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.
- The first successful launch data is 06-May-2016.

Task 5

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

```
rows = cur.execute("SELECT  MIN(DATE) FROM spacextbl WHERE [Landing _Outcome] = 'Success (drone ship)'").fetchall()
print(rows)
#names = list(map(lambda x: x[0], cur.description))
#print(names)
```

```
[('06-05-2016',)]
```

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
- Three booster version were identified.
- F9 FT B1032.1
- F9 B4 B1040.1
- F9 B4 B1043.1

Task 6

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

```
rows = cur.execute("SELECT Booster_Version FROM spacextbl WHERE [Landing _Outcome] = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000").fetchall()
print(rows)
```

```
[('F9 FT B1032.1',), ('F9 B4 B1040.1',), ('F9 B4 B1043.1',)]
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- 99 number of successful outcome, 1 failure, 1 payload status unclear but successful.

- 'Failure (in flight)', 1,
- 'Success', 98 ***
- 'Success ‘, 1 ***
- 'Success (payload status unclear)', 1)

Task 7

List the total number of successful and failure mission outcomes

```
rows = cur.execute("SELECT Mission_Outcome, Count(*)  FROM spacextbl GROUP BY Mission_Outcome").fetchall()
print(rows)
```

```
[('Failure (in flight)', 1), ('Success', 98), ('Success ', 1), ('Success (payload status unclear)', 1)]
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Retrieved 12 boosters with payload of 15,600kgs.

Task 8 ¶

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
rows = cur.execute("SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM spacextbl WHERE PAYLOAD_MASS__KG_ IN(SELECT MAX(PAYLOAD_MASS__KG_) FROM spacextbl )").fetchall()
print(rows)
```

```
[('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

- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Six months(Feb, May, Jun, Aug, Sept, Dec) had successful (ground pad) landing outcome.

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2017' for year.

```
rows = cur.execute("SELECT substr(Date, 4, 2),[Landing _Outcome],Booster_Version,Launch_Site FROM spacextbl WHERE substr(Date,7,4)='2017' AND [Landing _Outcome] = 'Success (ground pad)'").fetchall()
print(rows)
```

```
[('02', 'Success (ground pad)', 'F9 FT B1031.1', 'KSC LC-39A'), ('05', 'Success (ground pad)', 'F9 FT B1032.1', 'KSC LC-39A'), ('06', 'Success (ground pad)', 'F9 FT B1035.1', 'KSC LC-39A'), ('08', 'Success (ground pad)', 'F9 B4 B1039.1', 'KSC LC-39A'), ('09', 'Success (ground pad)', 'F9 B4 B1040.1', 'KSC LC-39A'), ('12', 'Success (ground pad)', 'F9 FT  B
1035.2', 'CCAFS SLC-40')]
```

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

- Rank the count of successful landing_outcomes between the date 04-06-2010 and 02-03-2017 in descending order
- Total 34 outcomes were successful during the requested period.

Task 10

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

```
rows = cur.execute("SELECT [Landing _Outcome], Count(*) FROM spacextbl WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' AND [Landing _Outcome] LIKE '%Success%' GROUP BY [Landing _Outcome] ORDER BY Count(*) DESC").fetchall()
print(rows)
```

```
[('Success', 20), ('Success (drone ship)', 8), ('Success (ground pad)', 6)]
```

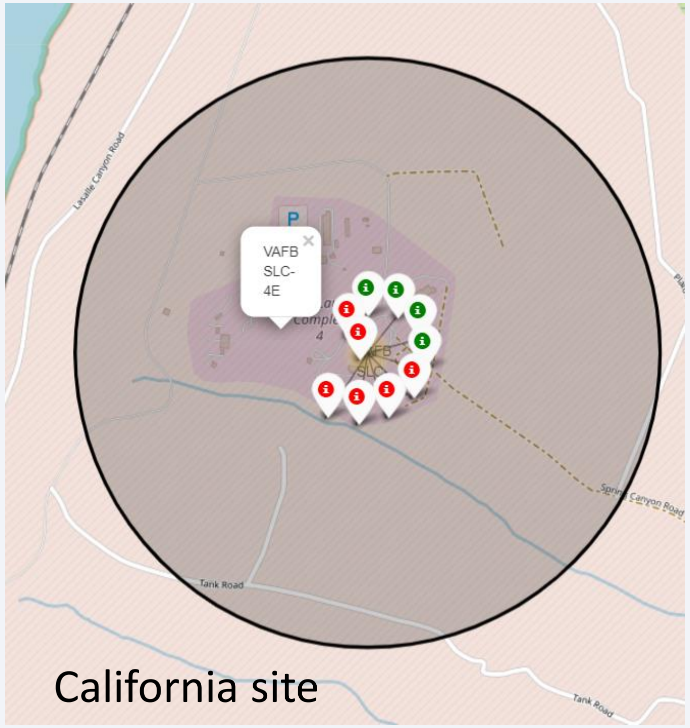
Section 3

Launch Sites Proximities Analysis

SpaceX Site Map Markers

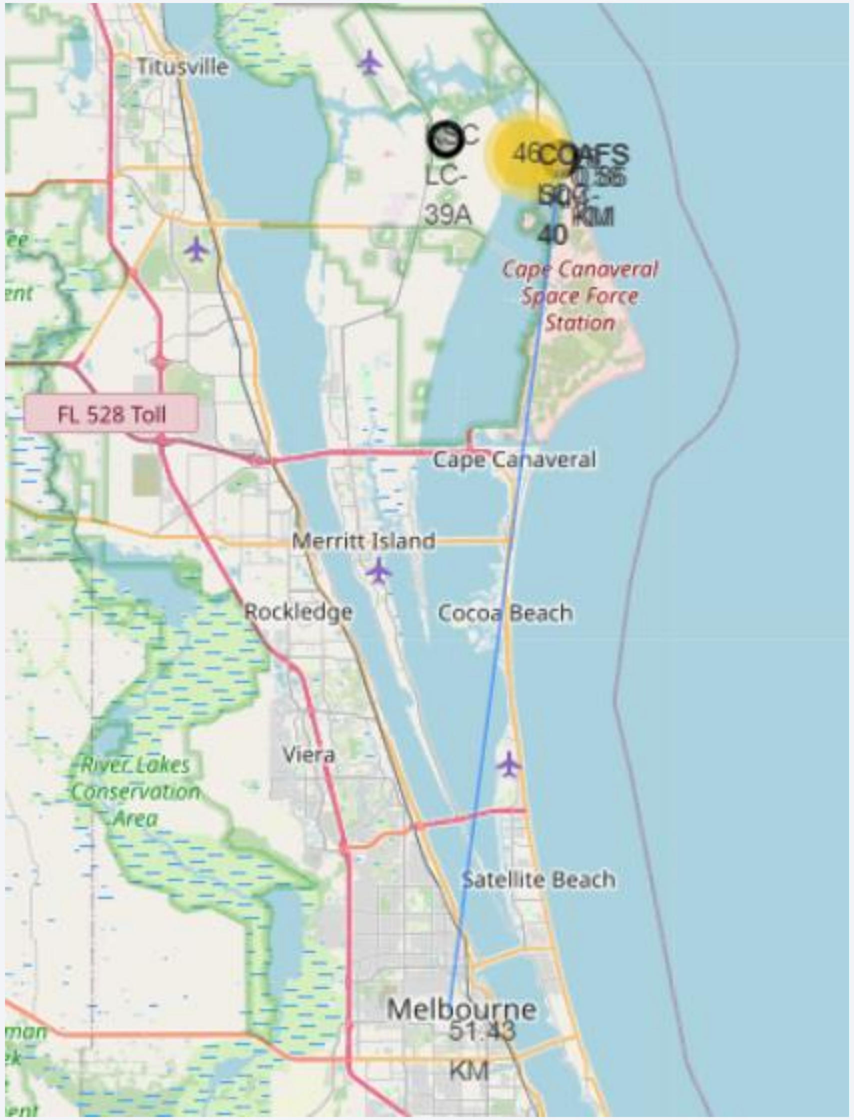
- We can see that the launch site are in USA's costal states of California and Florida.

Launch site with color markers



Green Markers show successful launch and Red Markers show unsuccessful.

Launch site distance to Landmarks



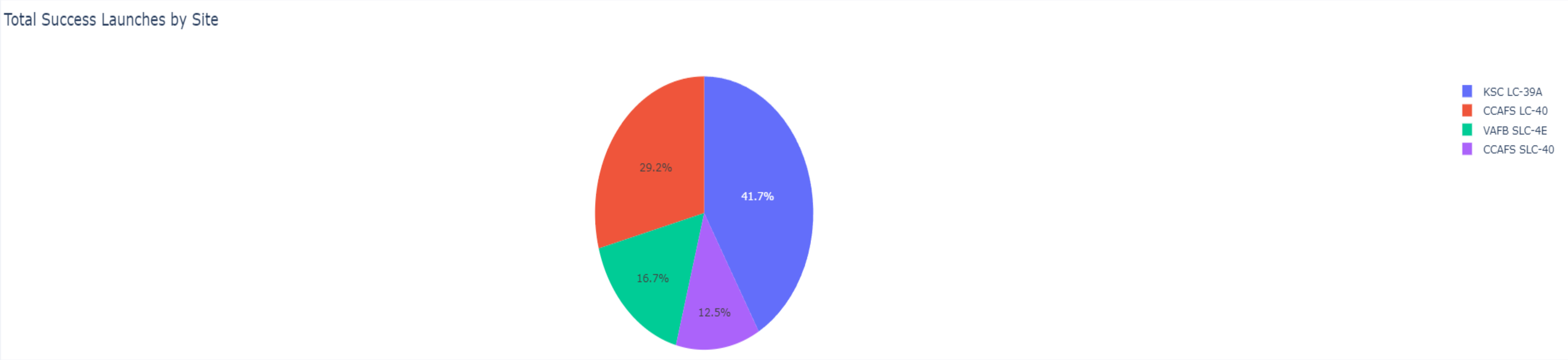
- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities Yes
- Distance_highway = 0.5834695366934144 km
- Distance_railroad = 1.2845344718142522 km
- Distance_city = 51.434169995172326 km

Section 4

Build a Dashboard with Plotly Dash

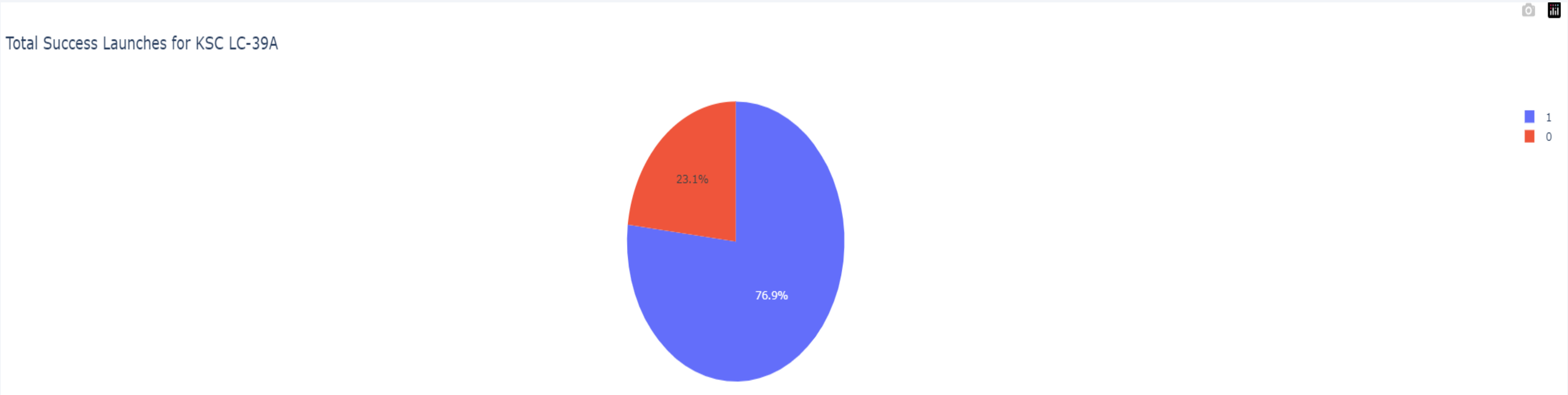
Total success launches by site

- KSC LC – 39A has the highest percentage of 41.7% success rate.



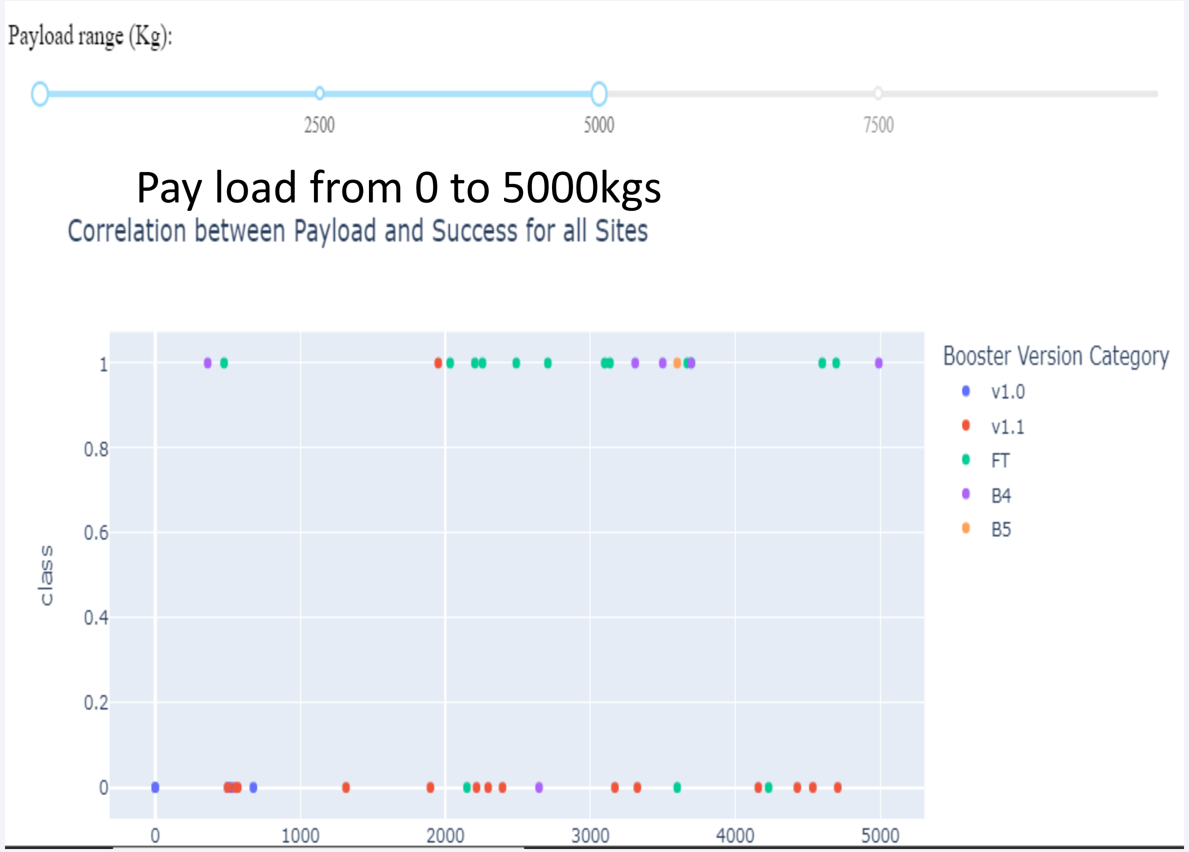
Launch site with highest launch success rate

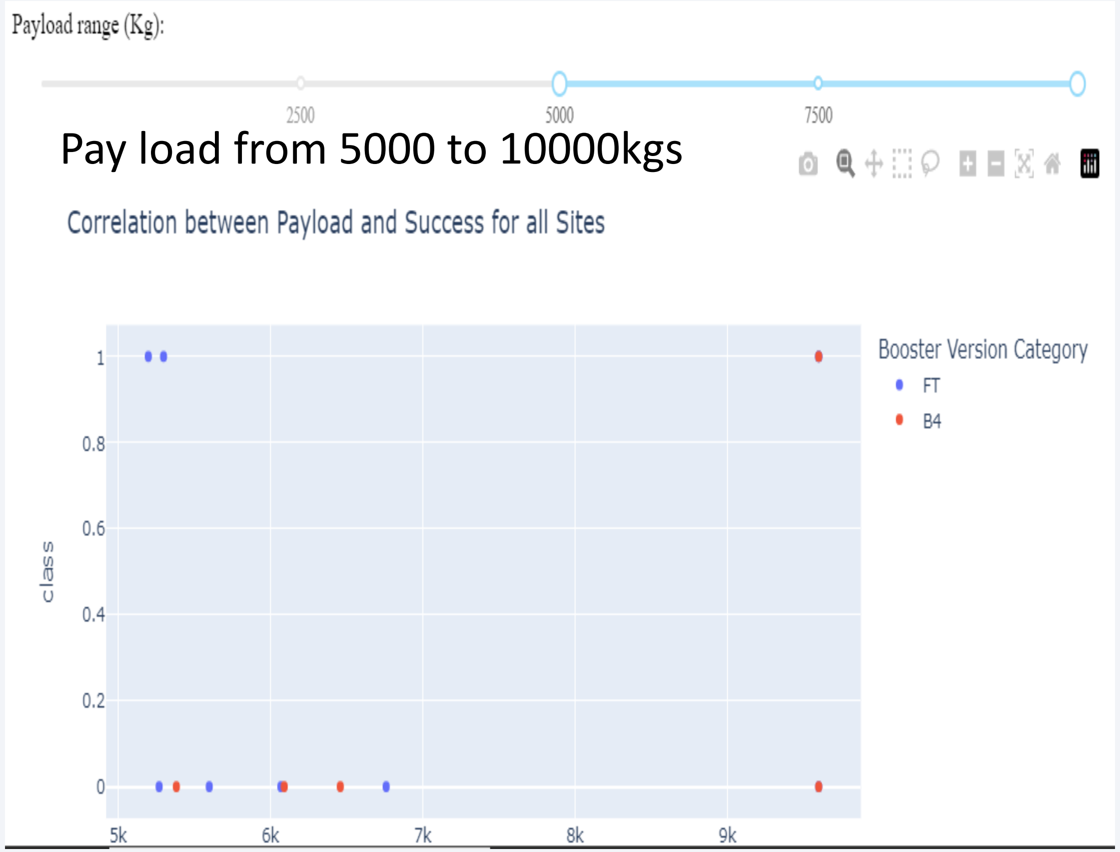
- KSC LC-39A had 76.9% success rate and 23.1% Failure rate



Payload Vs Launch outcome with different payloads selected using a slider

- Lower payload got more success than higher payloads





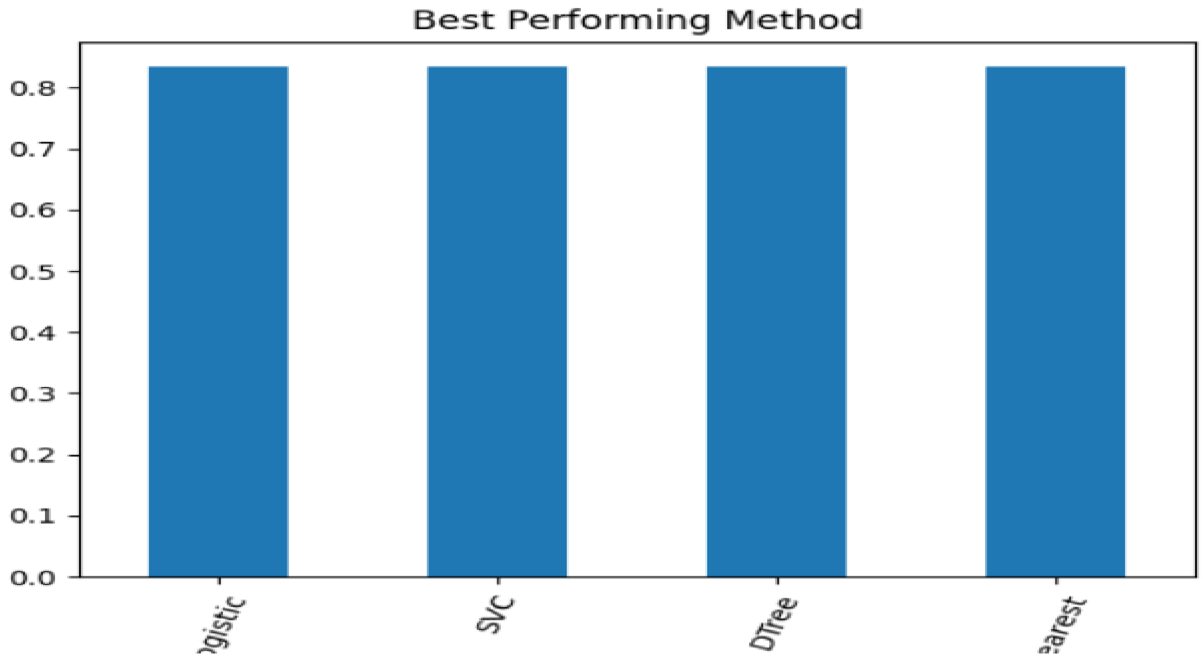
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All the model's accuracy were similar.

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

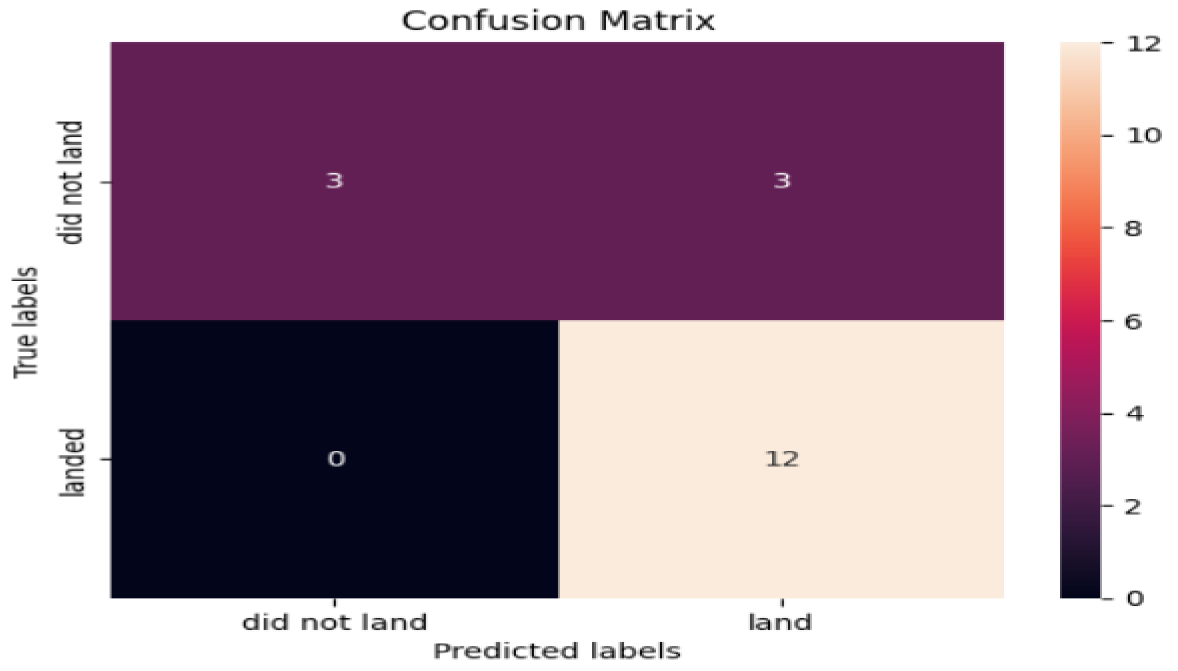



```
<Figure size 640x480 with 0 Axes>
```

Confusion Matrix

- All the model's prediction accuracy was same. Here I am displaying SVM related confusion matrix. The problem is with the false positives. It is showing unsuccessful landing is marked as successful landing by the classifier.

```
[53]: yhat=svm_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- We can observe success rate were higher from the year 2013.
- PO, LEO and ISS orbits have successful landing for heavy payloads
- KSC LC-39A had the highest success rate of 76.9% and 23.1% Failure rate
- All the classifiers performed similar with an accuracy of .83
- Lower payload got more success than higher payloads
- KSC LC – 39A has total success launches with the highest percentage of 41.7% success rate.
- VLEO has more launch more success, LEO orbit success is related to the number of flights
- ES-L1,GFO,HEO, and SSO orbit are more successful for landing.
- Above factors should help us to predicting the price of launch.

Appendix

- All code and data available in Github directory
- https://github.com/bhasker63/IBM-Data-Science.git

Thank you!