Winning Space Race with Data Science

JIAN WU (LInkedIn)

August 27, 2022

Contents

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

Executive Summary

- **Summary of methodologies**
  - Itantiated the IBM Data-Science Methodology in this Capstone project
  - Applied the public data to analyze and predict whether sucessfully recovering 1st-Stage Orbit Rocket of SpaceX
  - Developed a solution incl. Data Collection, Understanding, Preparation, Modeling, Evaluation and Deployment etc.
- **Summary of all results**
  - Conclusion: Four conventional classifiers performed predictions with expected results.
  - Discussion: 1) False possitive happend to all optimized classifiers, which shall be improved by further Feature Extraction or Advanced Classifiers; 2) DecisionTree has not performed reliably, yet its prediction reached better results mostly, whose GridSearchCV shall be traced and studied carefully in the furture.

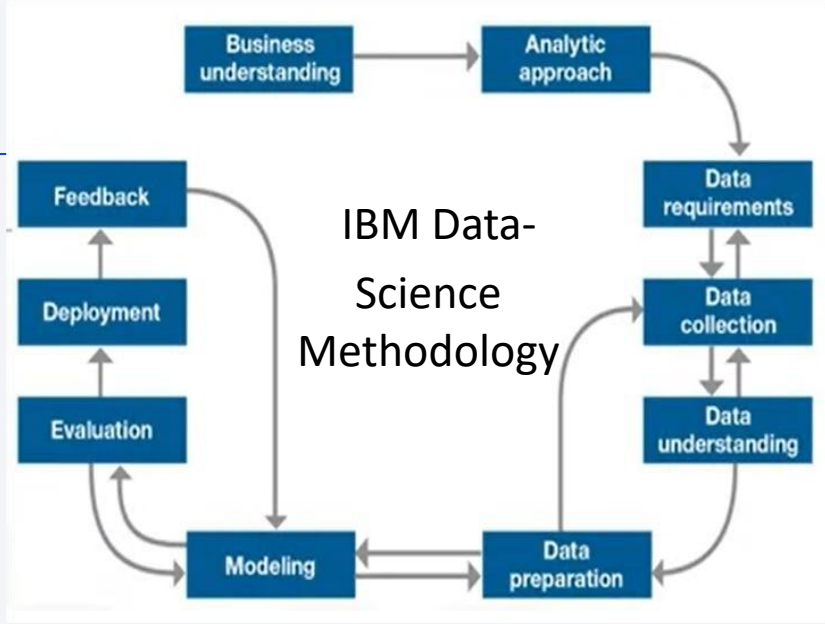
Introduction

- Project background
- Compared with 165 millions of dollars, around 65 millions is an attractive cost of orbit rocket launching, lying in that SpaceX acchieves recovering the first stage, thereby bringing unparalleled stimulation to the market and great commercial values to rocket users.
- Problem statements
- If it can be predicted whether or not the first stage of SpaceX rocket lands successfully, it means that cost of a launch can be estimated as well; meanwhie, those factors, features or conditions used for predictions can be introduced to the study of orbit rocket recovering. In this project, we develop a solution to predict landing results of one version of SpaceX rocket by public data sources.

Section 1

Methodology

Methodology - Executive Summary



Instantiated Methodology mainly from Step 3 to Step10

- Data collection methodology:
  - Collected and scrapped by SpaceX API and library of BS4 respectively
- Perform data wrangling
  - Replaced missing values by means, extracted festures, one-hot endoding, labelled outcomes, built and splited datasets for traing, testing and cross-validation etc.
- Perform exploratory data analysis using visualization library (Seanborn) and SQL (MySQL)
- Perform interactive visual analytics using library (Folium) and Plotly Dash
- Perform predictive analysis using classification models
  - Standerized scalers, build datasets, classifiers tuned by GridSearchCV, evaluated classification models by Accuracy, Jaccard score, Avg F1-score etc, and further analyzed results with confusion matrix.

Data Collection

- Key phrases and flowcharts of data collection process:

SpaceX API

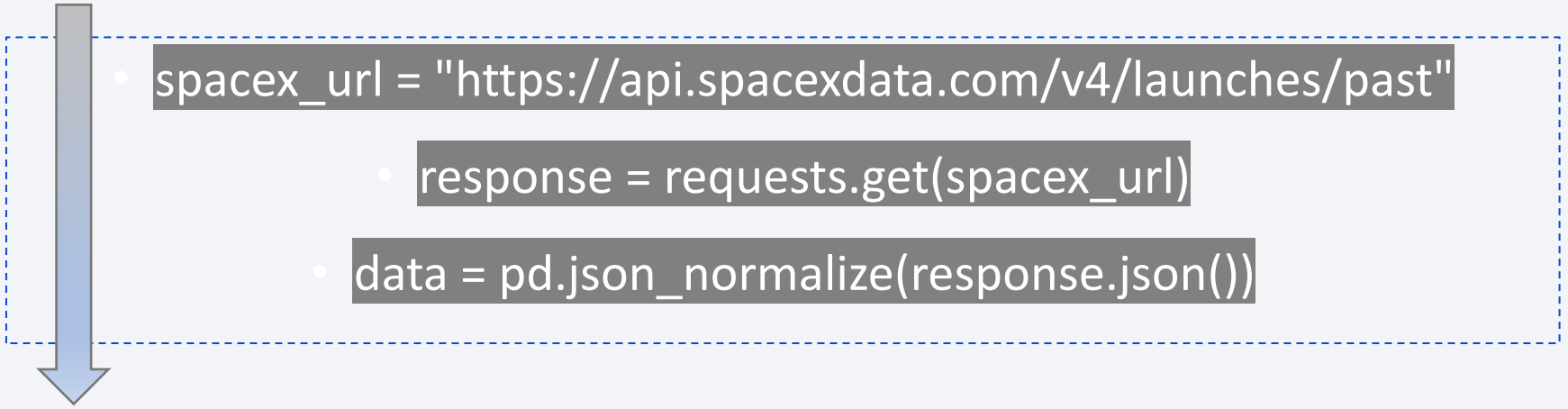
- Request rocket launch data from SpaceX API
- Parse and decode the reponse contents as JSON by .json()
- Turn ino IDs into Pandas dataframe by .json_normalized()
- Get Launch info (rockets, payloads, launchpads, core) into dict by parsed IDs and API again
- Filter and contruct the dataframe only include Falcon9 (dataset_part_1.csv)

Web Scraping

- Request Falcon 9 launch data from HTML page of wiki
- Parse HTML into Table
- Contruct dataframe and extract info (FlightNo, date, Site, Payload, Mass, Orbit, Customer, Outcomes)

Data Collection – SpaceX API

- Key phrases and flowcharts of data collection with SpaceX REST calls using

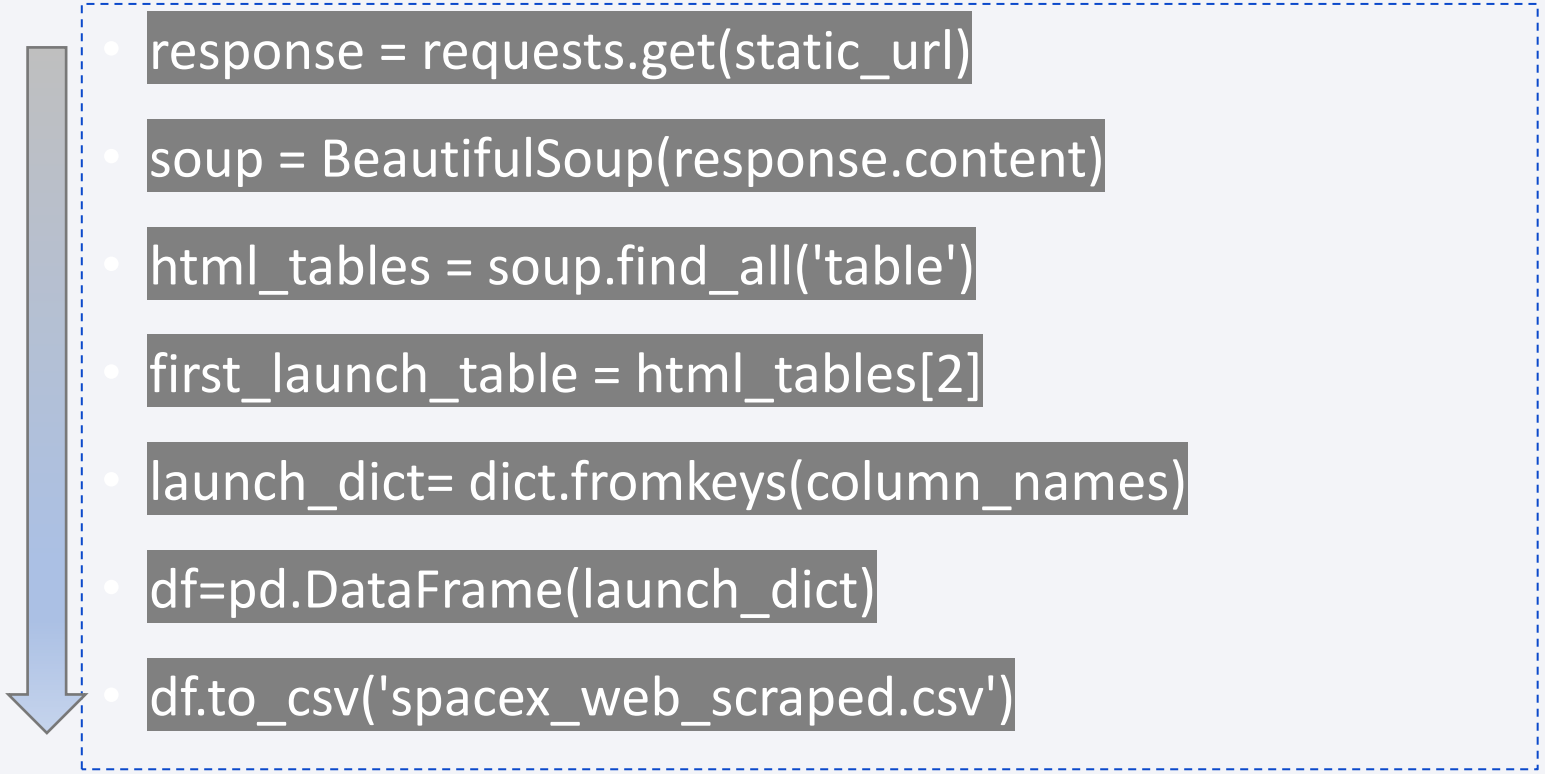


- GitHub URL of the completed SpaceX API calls notebook (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_01_DataCollection_API_150822_final.ipynb

Data Collection - Scraping

- Key phrases and flowcharts of Web scraping process

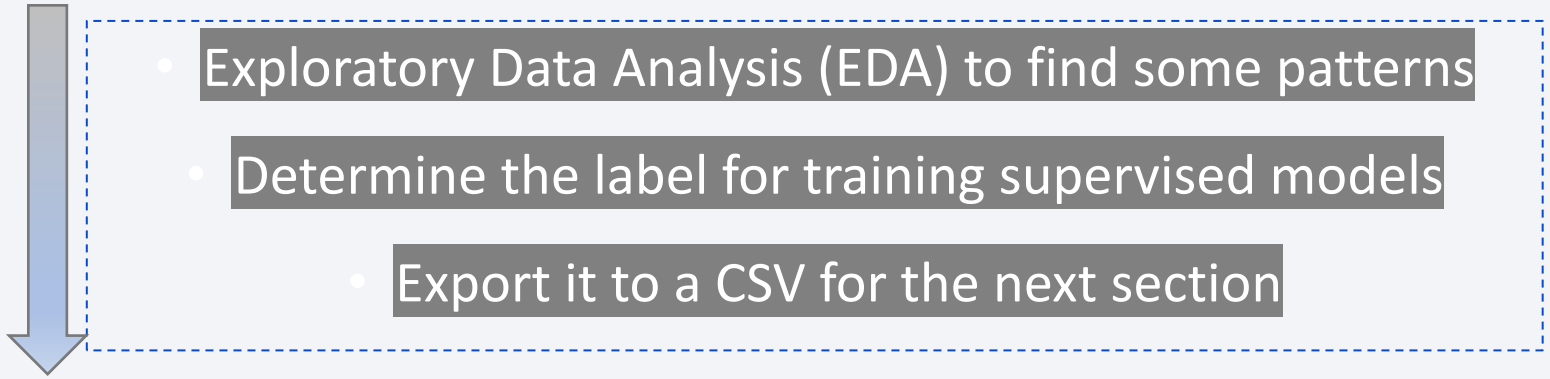


- GitHub URL of the completed web scraping notebook (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_02_Webscraping_bs4_160822_final.ipynb

Data Wrangling

- Key phrases and flowcharts of data wrangling process

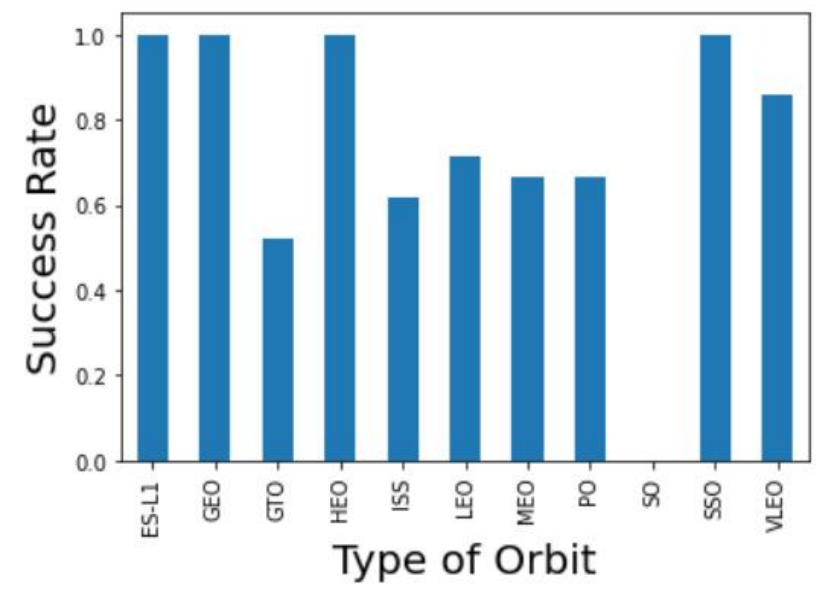


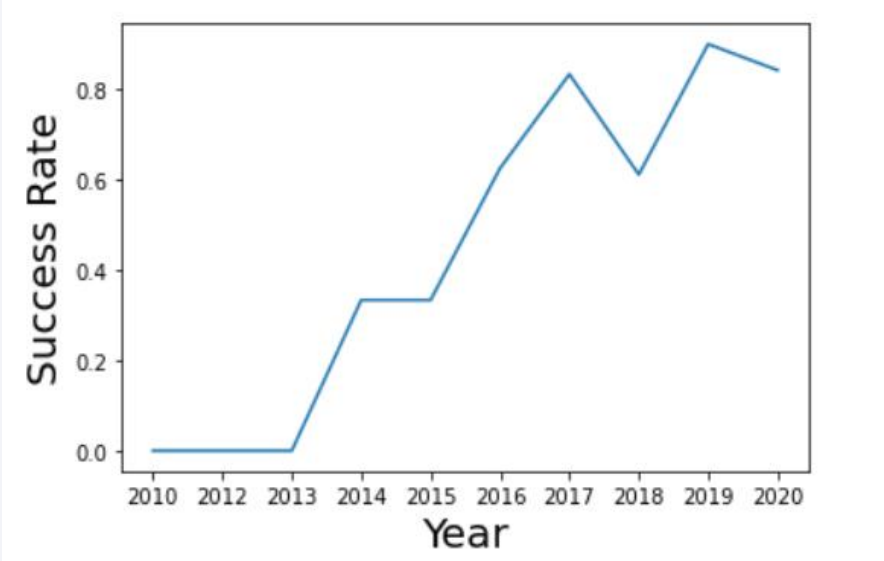
- Note: Training Labels w/ 1 means the booster successfully landed, 0 means unsuccessful.
- GitHub URL of completed data wrangling related notebooks (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_03_DataWrangling_170822_final.ipynb

EDA with Data Visualization

- EDA w/ Visualization help to find relationship of flight number vs. launch Site, payload vs. launch site, success rate of each orbit type, and flight number vs. orbit etc.
- Summarized charts showing significant trend and relations





- GitHub URL of completed EDA with data visualization notebook (Click) or Copy URL to your brower:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_05_EDA_DataViz_20082_final.ipynb

EDA with SQL

- **Summarize the SQL queries performed**
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the 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 succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass (4k, 6k)
  - List the 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 04-06-2010 and 20-03-2017 in descending order

- GitHub URL of completed EDA with SQL notebook (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_04_EDA_MySQL_200822_hide.ipynb

Build an Interactive Map with Folium

- Summarize map objects created and added to a folium map
- Added markers to all sites of lauhch geographically
- Group labelled outcomes of success and failure by Green and Red respectively
- Created the color-labeled markers in marker clusters, by which it is able to identify which launch sites have relatively high success rates
- Calculated the distances between a launch site to its proximities and analysis reasons
    - Close to railways, highways and coastlines for fluent transportation
    - Distant to cities in consideration of safety

- GitHub URL of completed interactive map with Folium map (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_06_LaunchSiteLocation_Solium_210822_final.ipynb

Build a Dashboard with Plotly Dash

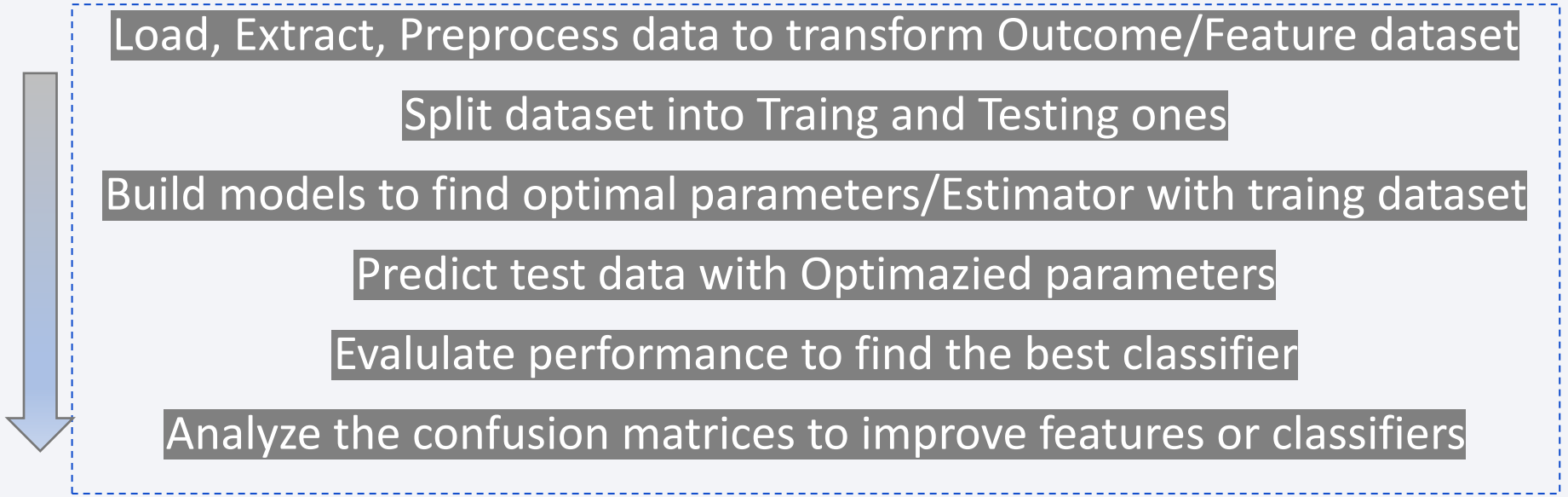
- Dash is the original low-code framework for rapidly building data Apps, which is here applied to show results of data exploratory or visual conclusion to stakholders interactively. There are mainly two components:
- A pie chart illustrates the success percentage of all launch sites and success rate of each site, switched by choosing options of Dropdown;
- A ScatterPlot shows the distribution of launch outcomes within specific range of Payload Mass (Kg) for each booster version.

- GitHub URL of completed Plotly Dash lab (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_Spacex_07_DashBoard_PlotlyDash_230822_final.ipynb

Predictive Analysis (Classification)

- Key phrases and flowcharts of classification process



- 、
-

- GitHub URL of completed predictive analysis lab (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_SpaceX_08_MachineLearningPrediction_Scilearn_250822_final.ipynb

Results

- Exploratory data analysis results
- Columns of raw data have been chosen to build feature dataset, such as Payload, Launch site, Orbit type, Flighte Number, booster version etc, which were explored numerically, visually and interactivly
- Interactive analytics demo in screenshots
- KSC LC-39A with best succss rate occupied highest percentage of success among all launch sites
- Predictive analysis results
- Overall performace of 4 classifiers reached 80% accuracy and D.Tree sometimes has better although false possitive is a problem
- GitHub URL of completed predictive analysis lab (Click) or Copy URL to your browser:

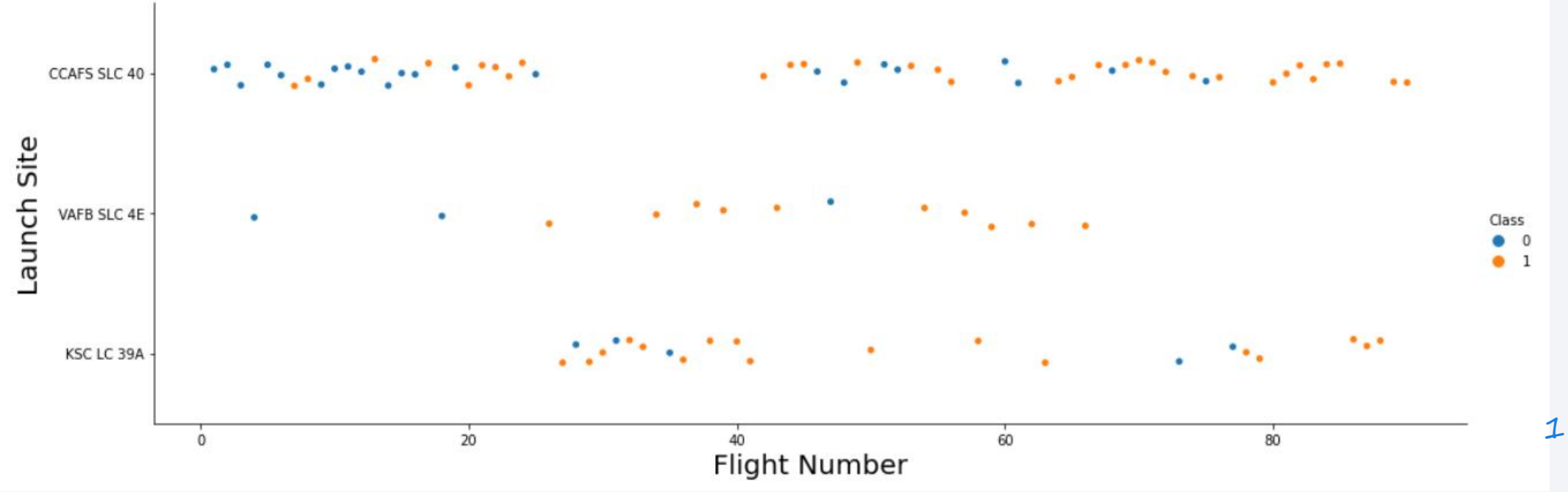
https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_SpaceX_09_MachineLearningPrediction_Scilearn_250822_discuss.ipynb

Section 2

Insights drawn from EDA

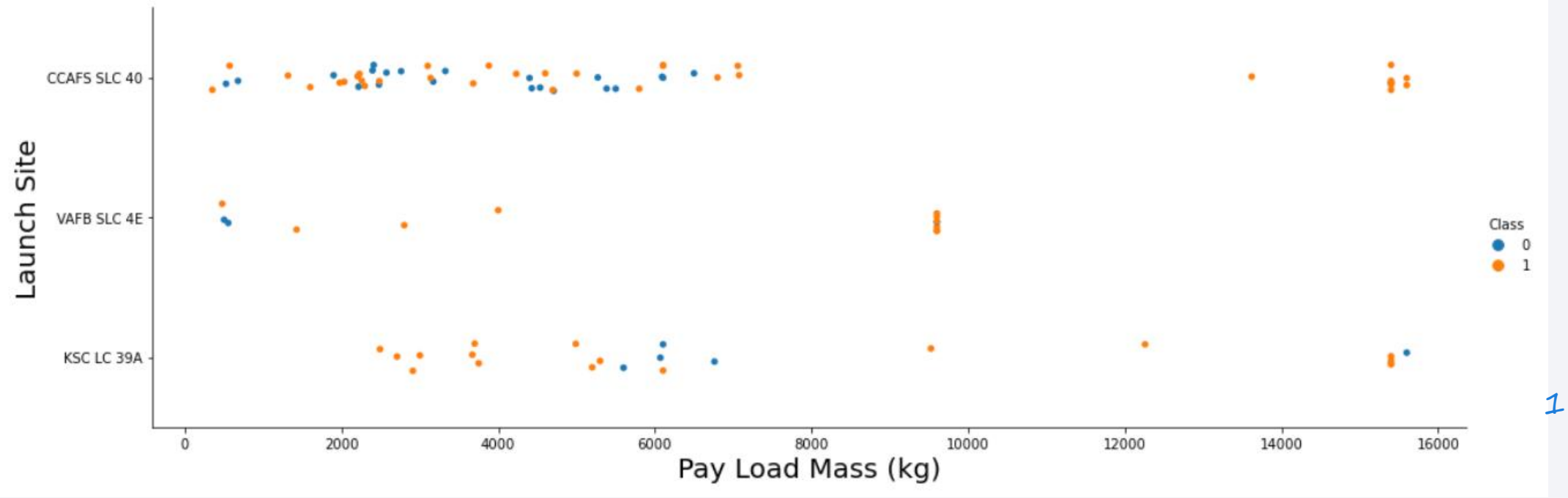
Flight Number vs. Launch Site

- ScatterPlot of Flight Number vs. Launch Site
- It is found that the bigger the Flight Number, the more success happens to each Launch site. Compared with KSC LC 39A, that of CCAFS SLC 40 performs more obvious, VAF8 SLC 4E in medium.
- ***Note: 1 in Orange (Success) / 0 in Blue (Failure), hereinafter***



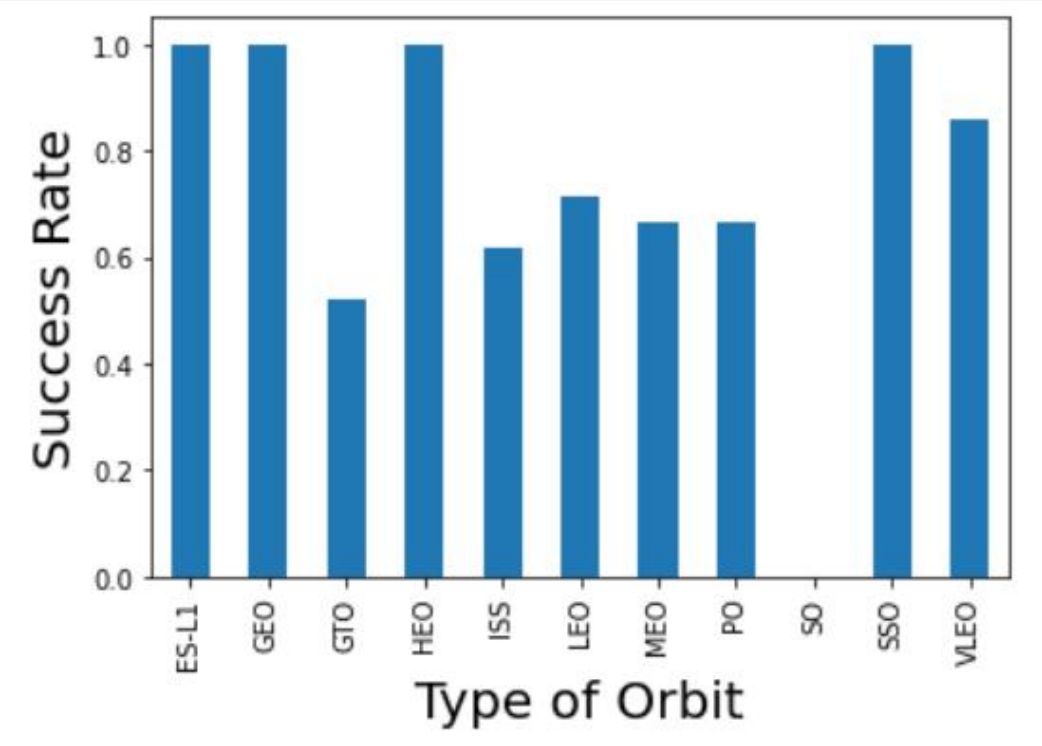
Payload vs. Launch Site

- ScatterPlot of Payload vs. Launch Site
- It is shown that 1) the heavier the PayloadMass, the more success to VAFB SLC 4E with a soft threshold around 6K kg; 2) success of VAFB SLC 4E performs more obvious, yet amount of samples seems not quite enough; 3) that of KSC LC39A has a failure range around 6K kg and a outlier around 16000 kg ... **All mentioned points can be summaried as knowledges for proven by further observations.**



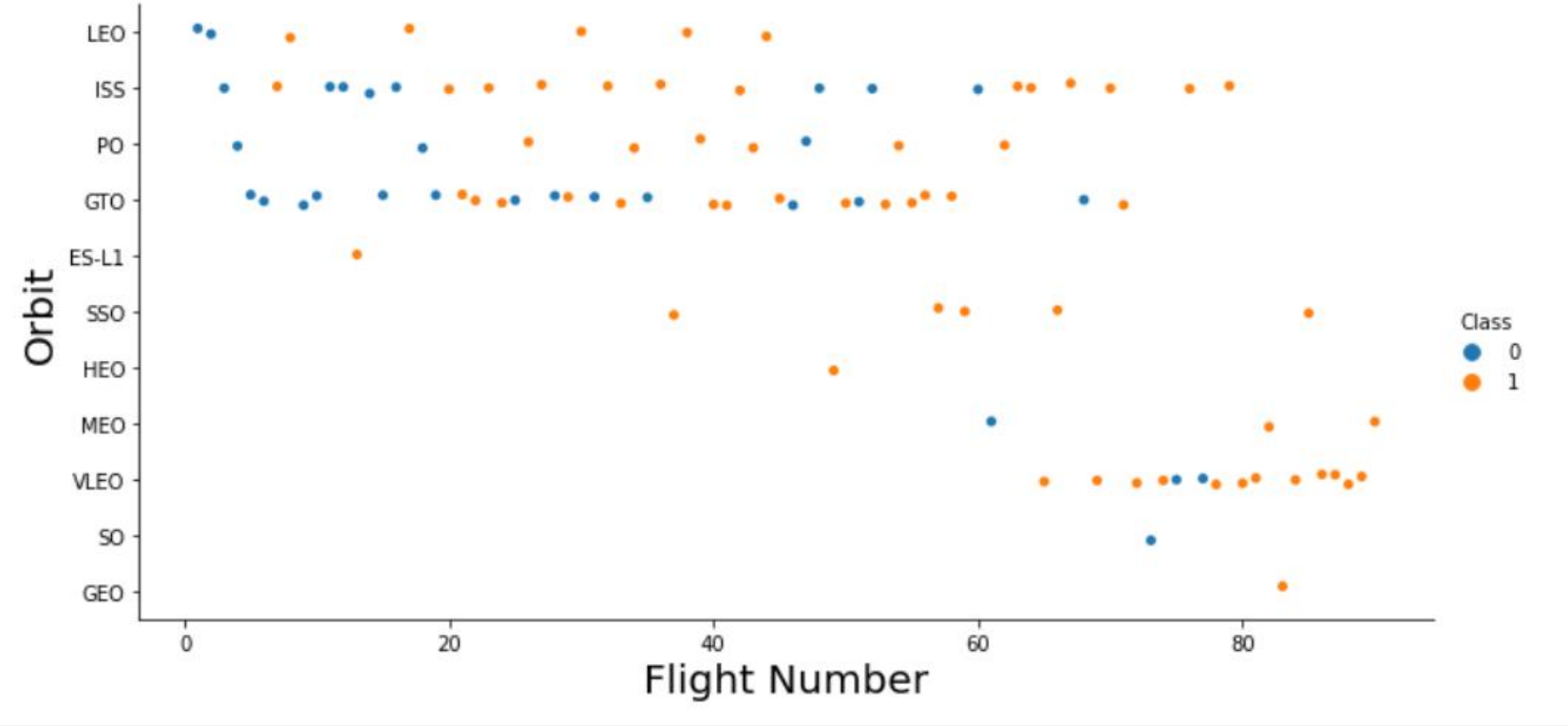
Success Rate vs. Orbit Type

- BarChart for the success rate of each orbit type
- It is shown that success rates vary by orbit types, and rates of ES-L1, GEO, HEO, SSO and VLEO reach 0.8 or 80%. **SO and GTO less than 50% shall be paid more attention.**



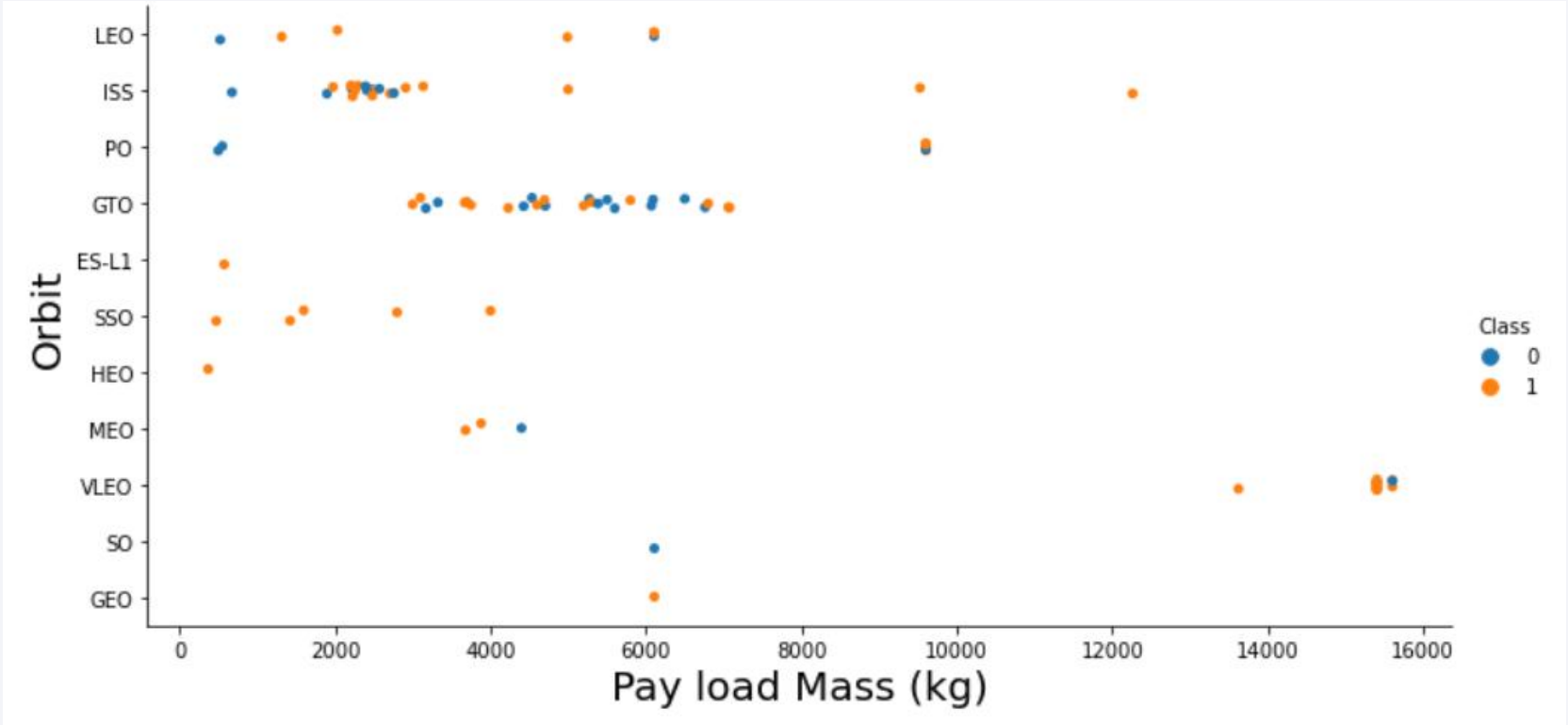
Flight Number vs. Orbit Type

- ScatterPlot of Flight number vs. Orbit type
- It is explored that Success of LEO orbit relates to the number of flights; however this sort of relationship is not obvious to others **with limited amount of samples**, say success is somewhat distributed randomly or disorderly.



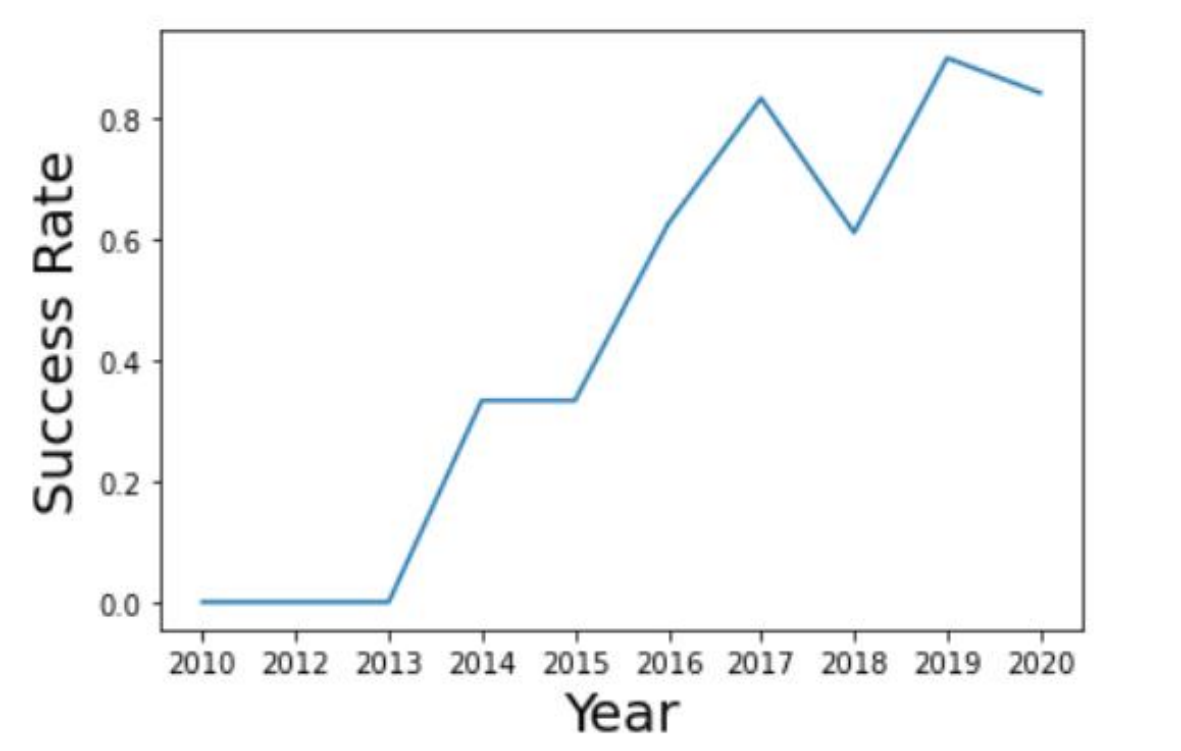
Payload vs. Orbit Type

- ScatterPlot of payload vs. orbit type
- It is observed that 1) PayloadMass and Success rate have a trend of possitive correlation in LEO and ISS, however the amout of samples for all types of orbits is limited, this trend shall be proven by more accumulated observations in the furture; **SSO is ideal orbit for less mass of payload (<4K kg)**



Launch Success Yearly Trend

- LineChart of yearly average success rate
- It is shown that Success rate has been increasing till 2017 dramaticly and droping in 2018, then keep inclining in 2019 then droping a little in 2020; **however, samples in 2018 are more valuable of study**.



All Launch Site Names

- Find the names of the unique launch sites by SQL for further feature exploration
- Keyword “DISTINCT” applies to query a list of unique launch sites effectively

```
In [15]: task="SELECT DISTINCT LAUNCH_SITE FROM SPACEX_NT"
         sql_task=pd.read_sql_query(task,conn)
         sql_task
```

Out[15]:

	LAUNCH_SITE
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`, whose success rate has strong correlation with Mass of Payload
- Keyword “LIKE” is useful for String process, and more info or columns can be listed under the same condition after “where”, if necessary later

```
In [16]: task = "SELECT LAUNCH_SITE from SPACEX_NT where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;"
         sql_task=pd.read_sql_query(task,conn)
         sql_task
```

Out[16]:

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS LC-40
2	CCAFS LC-40
3	CCAFS LC-40
4	CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA, any similar calcation can be done for the aim of comparision
- The query result is a great success

In [17]:
```
task = "SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX_NT WHERE Customer = 'NASA (CRS)';"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[17]:

	SUM(PAYLOAD_MASS__KG_)
0	45596.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- The query result, the average of payload mass implies that it is the value with more reliabe performance of success rate, which can be proved by further data exploration, and similarly, other statitics, say the mode, 1sd, 2sd, shall be calculated as well by SQL or Python.

In [18]:
```
task = "SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEX_NT WHERE Booster_Version LIKE 'F9 v1.1%';"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[18]:

	AVG(PAYLOAD_MASS__KG_)
0	2534.6667

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Date or Time is an objective factor which shall be considered and analyzed; apparenlty, 2015-12-22 is a significant date to SpaceX.

In [19]:
```
task = "SELECT MIN(Date) FROM SPACEX_NT WHERE Landing_Outcome = 'Success (ground pad)';"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[19]:

	MIN(Date)
0	2015-12-22

Successful Drone Ship Landing with Payload between 4,000 and 6,000 Kg

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4K but less than 6K kg
- It is useful query method by “AND” to extract info with range of payloadMass and groupby landingMethod to specific boosters

In [20]:
```
task = "SELECT BOOSTER_VERSION,LANDING_OUTCOME,PAYLOAD_MASS__KG_ \
            FROM SPACEX_NT \
            WHERE LANDING_OUTCOME='Success (drone ship)' AND PAYLOAD_MASS__KG_<6000 AND PAYLOAD_MASS__KG_>4000;"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[20]:

	BOOSTER_VERSION	LANDING_OUTCOME	PAYLOAD_MASS__KG_
0	F9 FT B1022	Success (drone ship)	4696
1	F9 FT B1026	Success (drone ship)	4600
2	F9 FT B1021.2	Success (drone ship)	5300
3	F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Keyword “group by” applies to extract categorious data, obviously the query result shows an ideal success rate of mission

In [21]:
```
task = "SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM SPACEX_NT GROUP BY MISSION_OUTCOME;"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[21]:

	MISSION_OUTCOME	TOTAL_NUMBER
0	Failure (in flight)	1
1	Success	99
2	Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- The subquery applies to list outcome of caculation which cannot be used under “where” conventinally

```
In [22]: task = "SELECT DISTINCT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEX_NT \
                WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_)\
                FROM SPACEX_NT);"
         sq_task = pd.read_sql_query(task,conn)
         sq_task
```

Out[22]:

	BOOSTER_VERSION	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1049.4	15600
2	F9 B5 B1051.3	15600
3	F9 B5 B1056.4	15600
4	F9 B5 B1048.5	15600
5	F9 B5 B1051.4	15600
6	F9 B5 B1049.5	15600
7	F9 B5 B1060.2	15600
8	F9 B5 B1058.3	15600
9	F9 B5 B1051.6	15600
10	F9 B5 B1060.3	15600
11	F9 B5 B1049.7	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Keyword “SUBSTR” is a method to extract parts of String

In [23]:
```
#SUBSTR(string, start_position, number_of_characters)
task = "SELECT SUBSTR(DATE,6,2) AS MONTHS, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, DATE AS NOTE\
            FROM SPACEX_NT\
            WHERE SUBSTR(DATE,1,4)='2015' AND LANDING_OUTCOME='Failure (drone ship)';"
sql_task = pd.read_sql_query(task,conn)
sql_task
```

Out[23]:

	MONTHS	LANDING_OUTCOME	BOOSTER_VERSION	LAUNCH_SITE	NOTE
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- 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
- The query is implemented by the combination of Keyword “GROUP BY” ,“HAVING”, and ”ORDER BY”

```
In [24]: task = "SELECT LANDING_OUTCOME,COUNT(LANDING_OUTCOME) AS TOTAL \
                  FROM SPACEX_NT \
                  WHERE DATE BETWEEN '2010-06-04' AND '2017-03-30' \
                  GROUP BY LANDING_OUTCOME \
                  HAVING LANDING_OUTCOME LIKE 'SUCCESS%' \
                  ORDER BY TOTAL DESC;"
         sql_task = pd.read_sql_query(task,conn)
         sql_task
```

Out[24]:

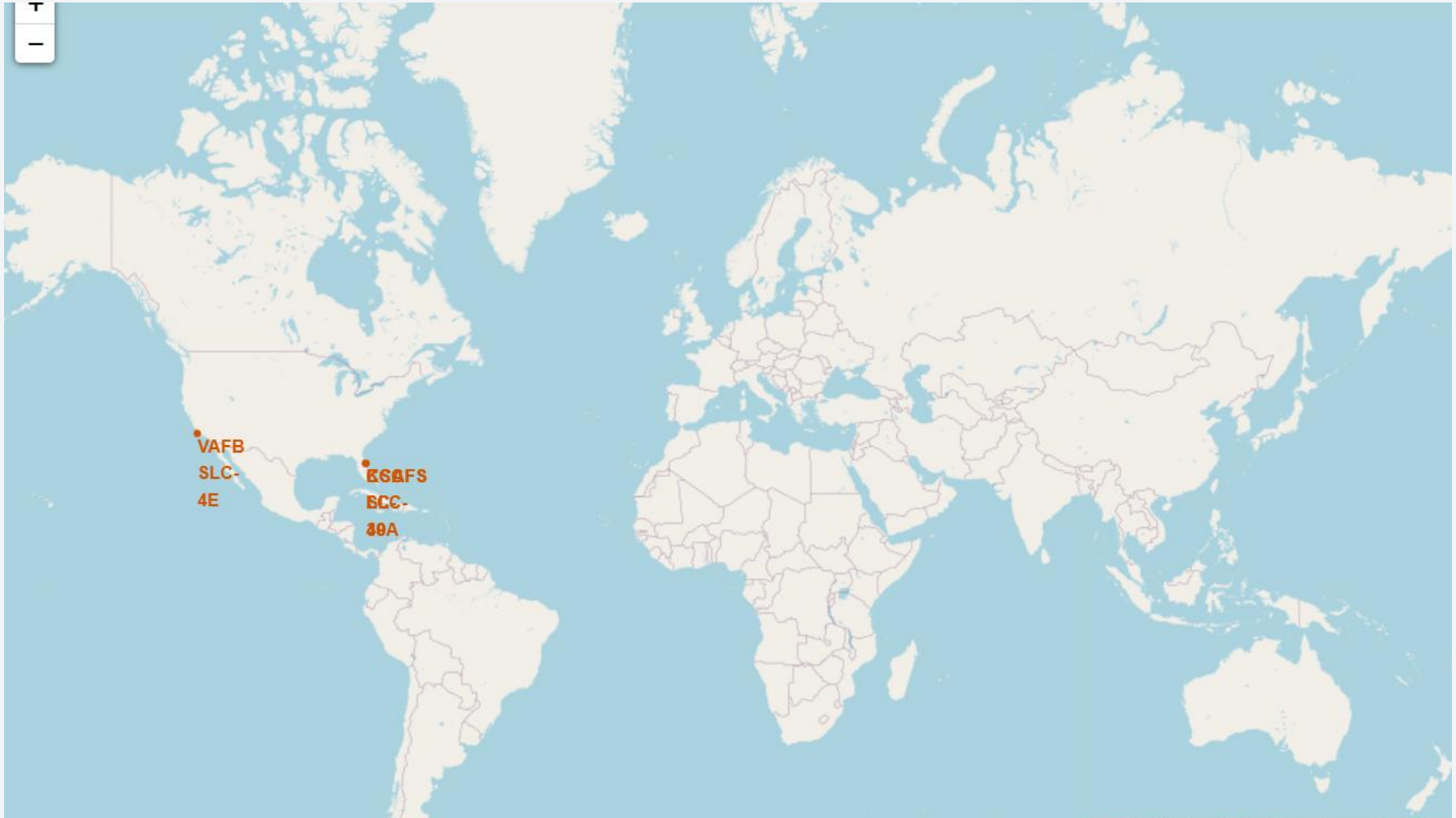
	LANDING_OUTCOME	TOTAL
0	Success (drone ship)	6
1	Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

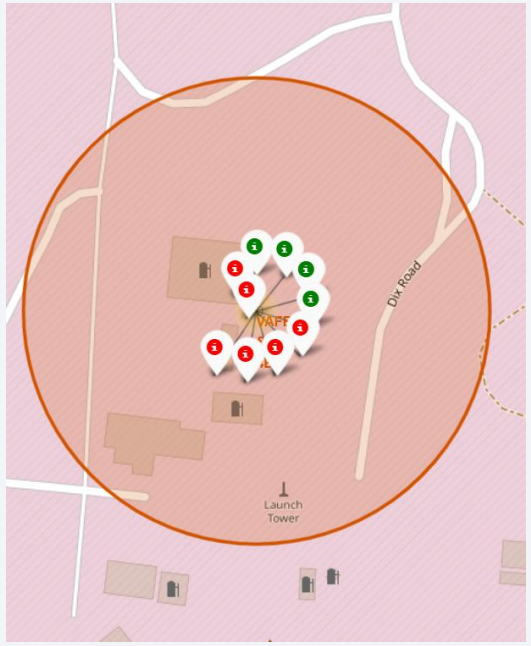
Location of LaunchSites markers on Global Map

- Lauch sites locate at both sites of the continent and close to the ocean, meaning that safety, transportation, climate and geography are all factors in the choice of launch site.

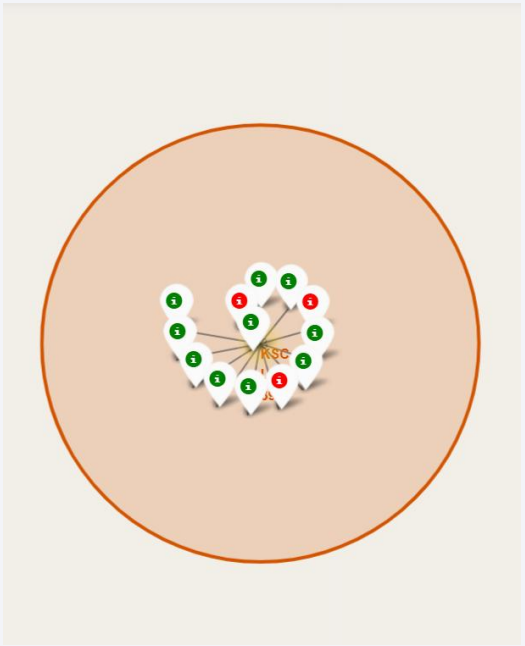


Color-labeled Launch Outcomes

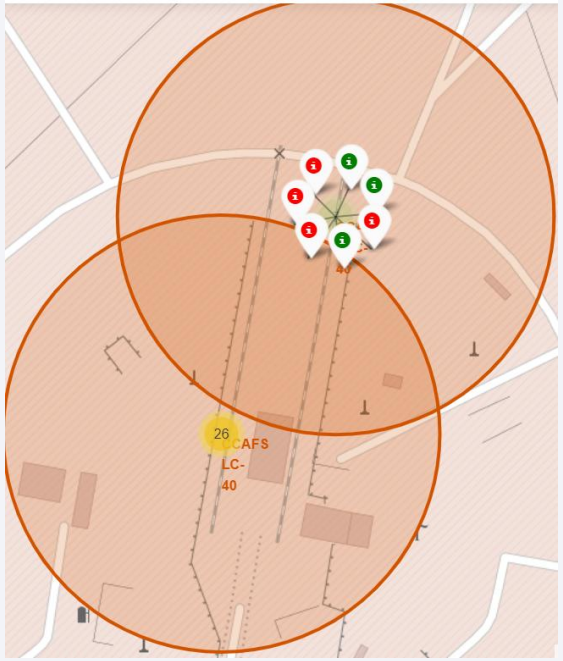
- KSC LC-39A has the highest Succss rate and CCAFS LC-40 has the most launch missions
- Note: Success marked in Green and Failure in Red



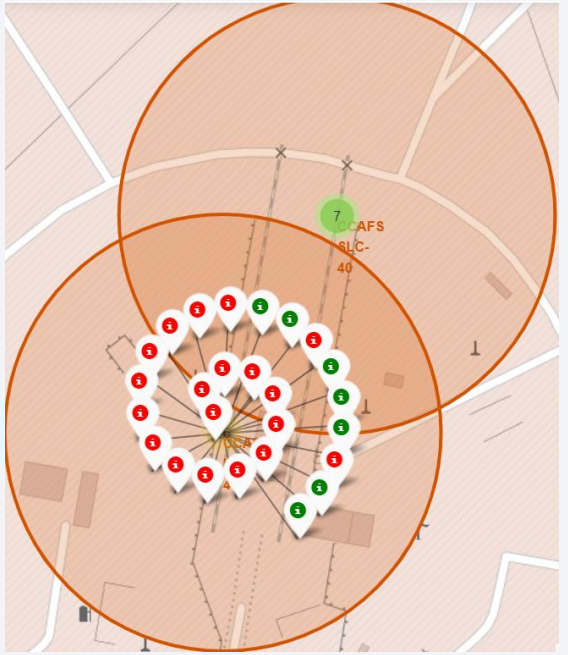
VAFB SLC-4E



KSC LC-39A



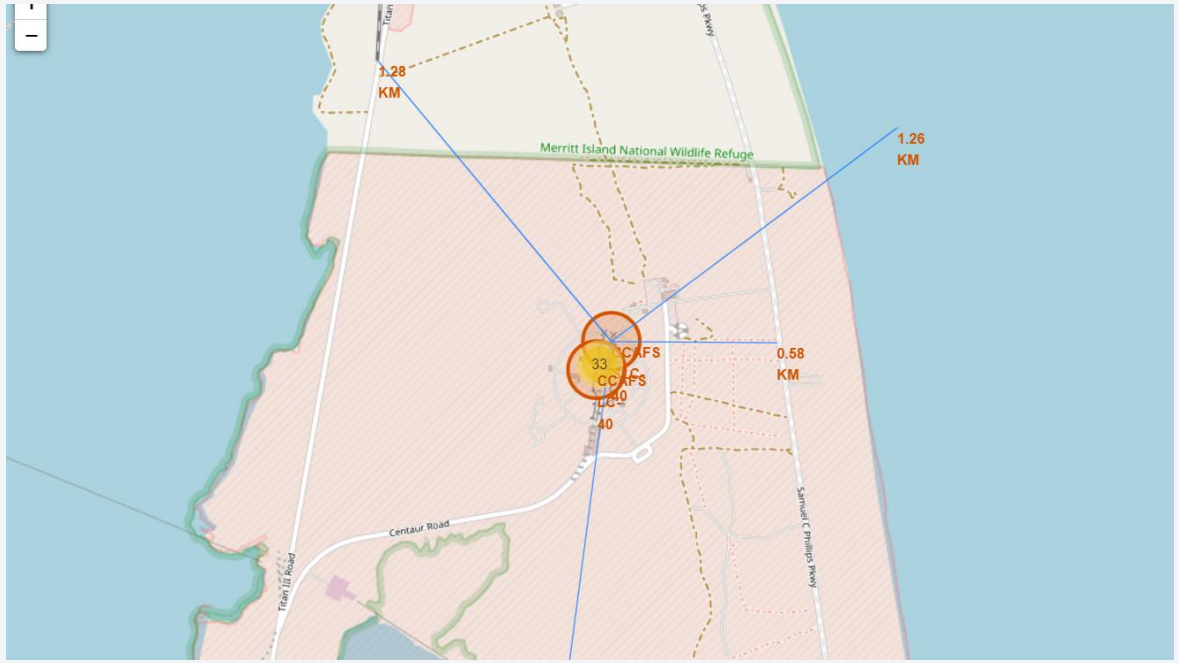
CCAFS SLC-40

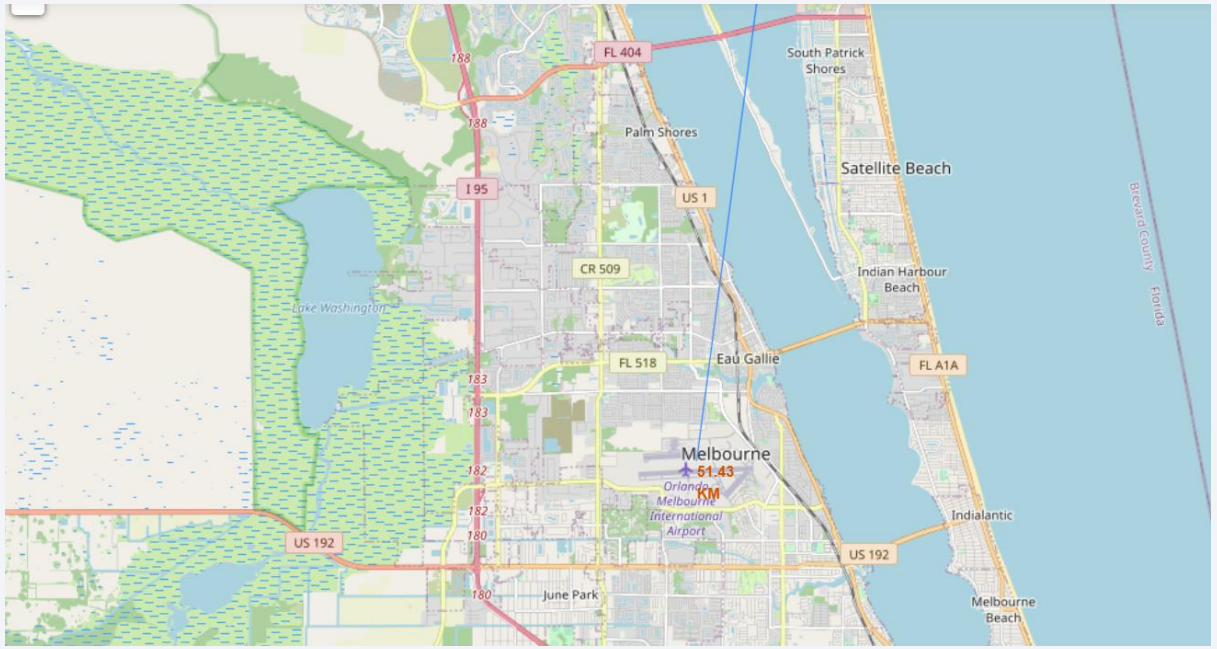


CCAFS LC-40

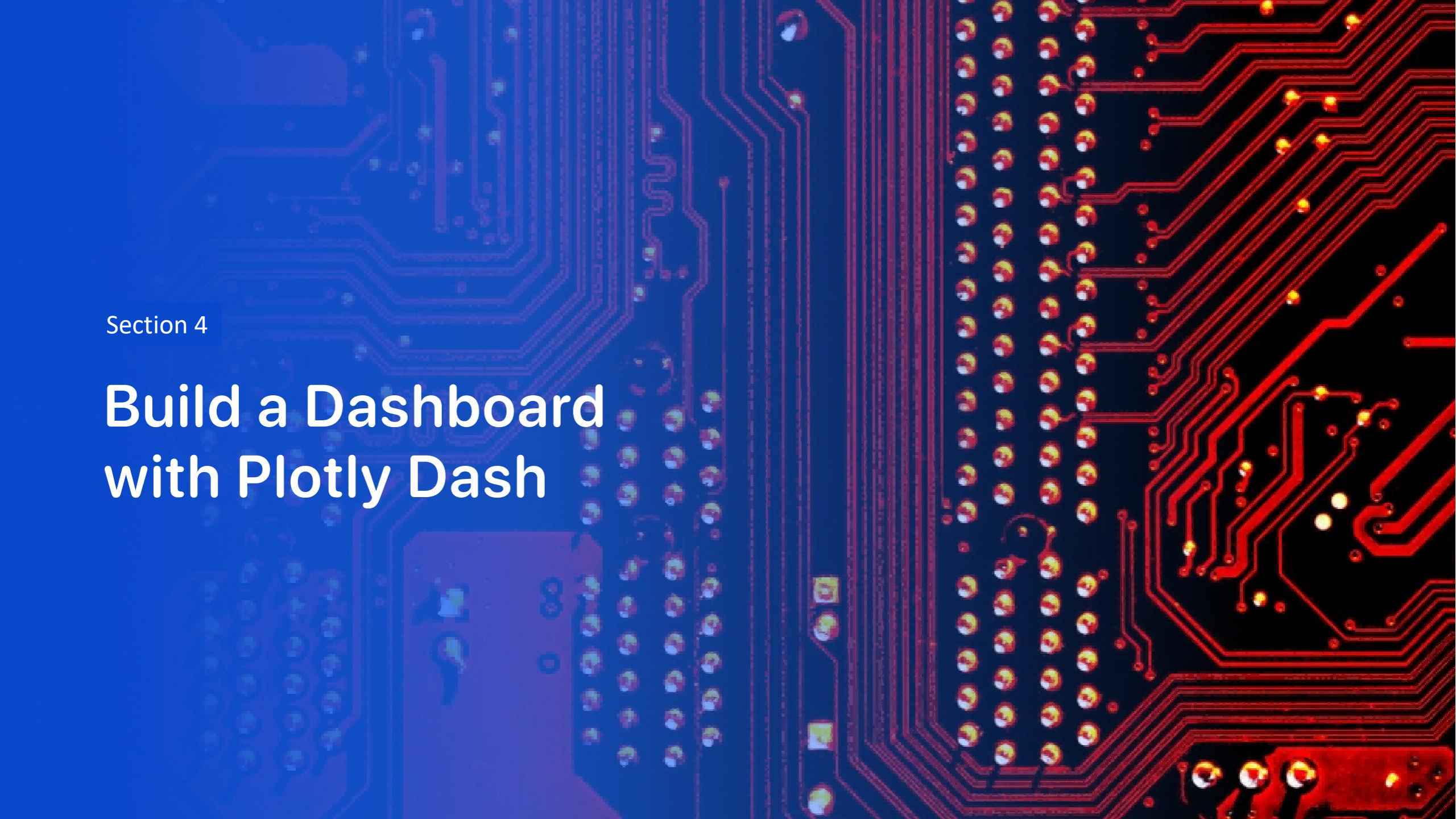
LaunchSite to its Proximities

- CCAFS SLC-40 is close to its transportation stations (highway, railway and coastline) and far from the city, Melbourn
- distance_highway = 0.5832563397764914 km
- distance_railroad = 1.2845344718142522 km
- distance_city = 51.43416999517233 km



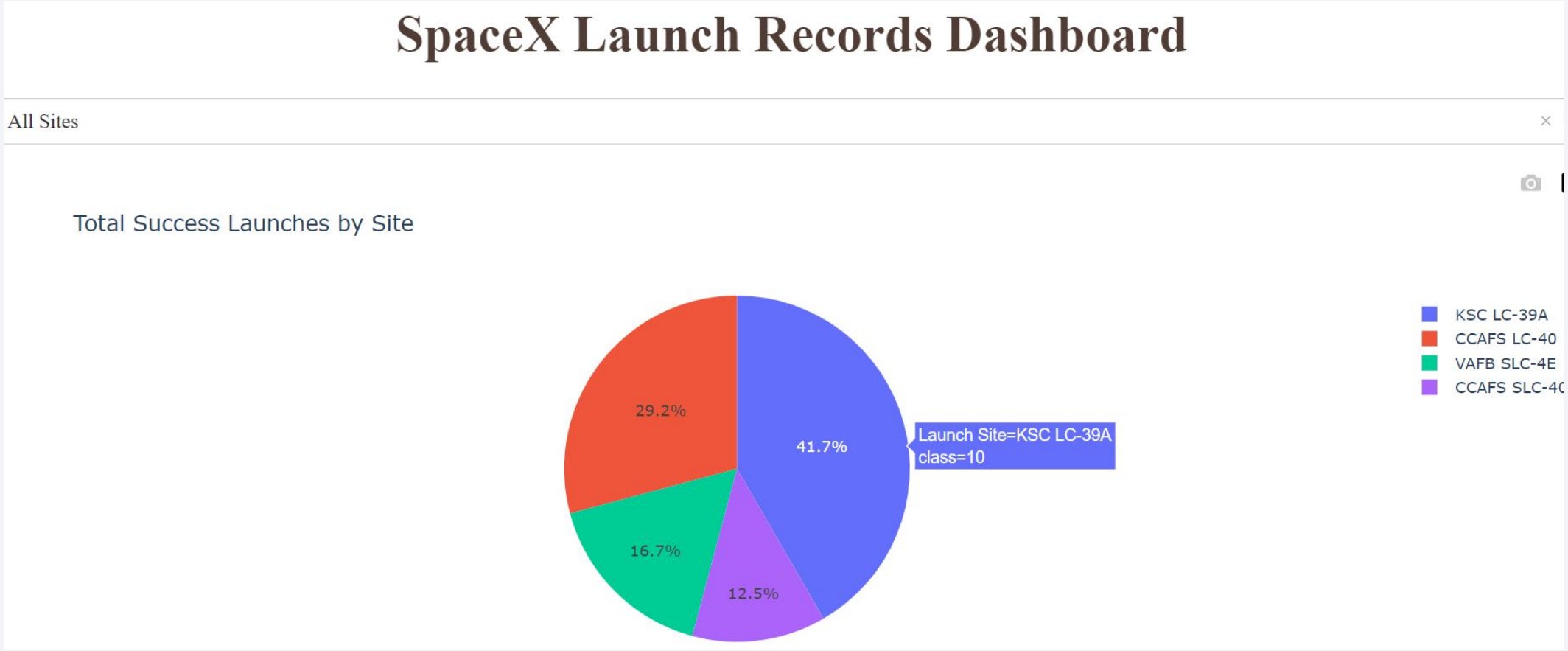


LauchSite: CCAFS SLC-40


Section 4
Build a Dashboard with Plotly Dash

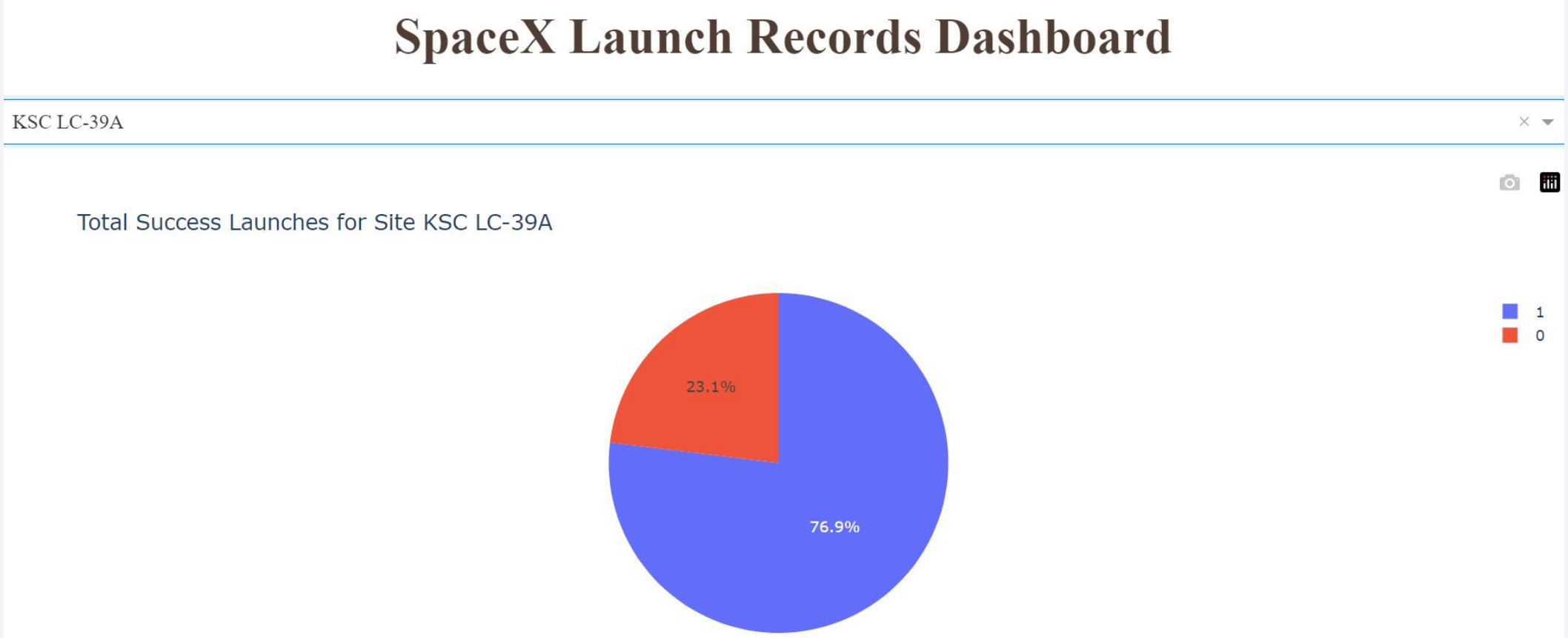
PieChart w/ Success Percentage of Each Site

- KSC LC-39A with the highest success percentage is an expected launch site

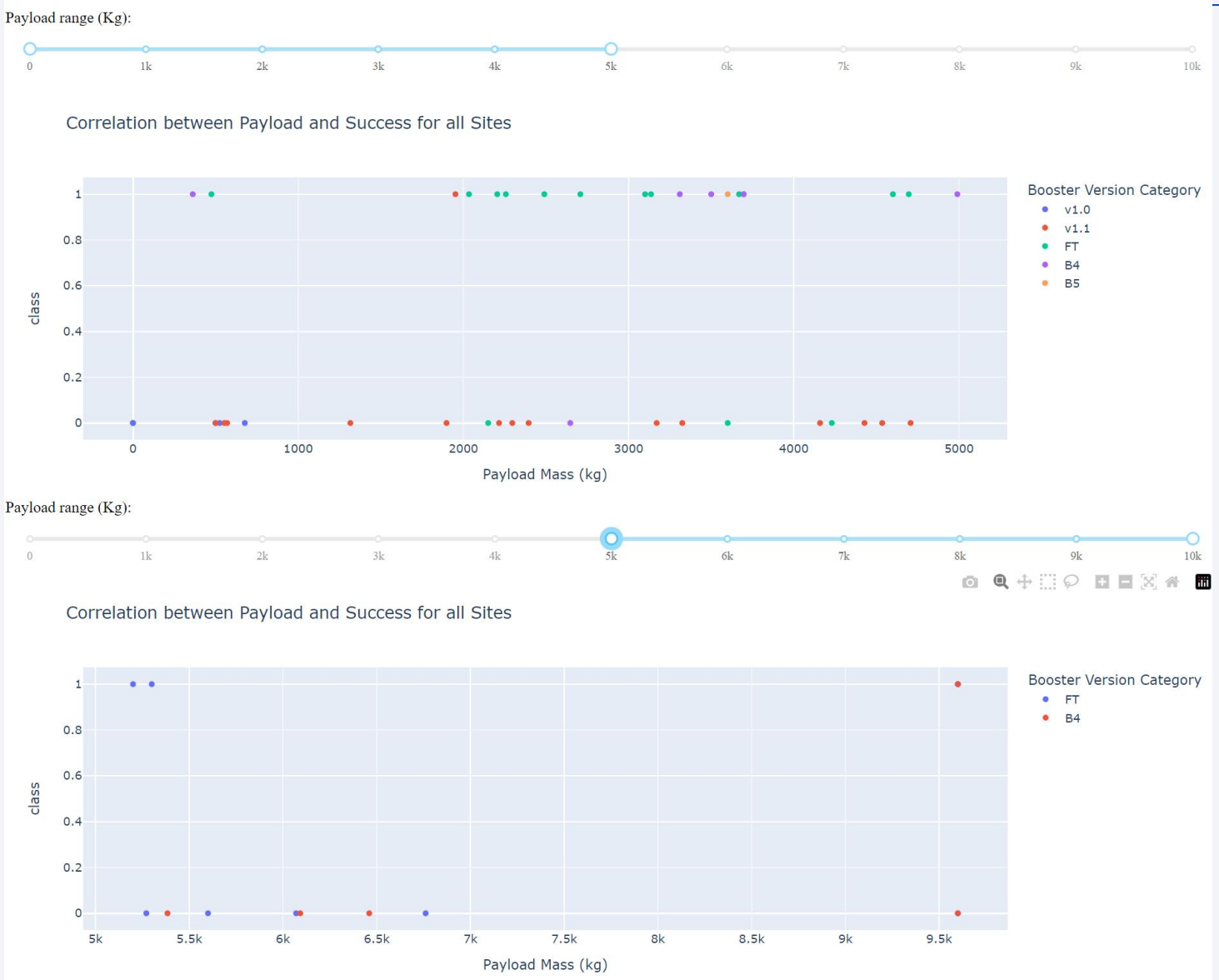


PieChart w/ the Highest Success Rate

- KSC LC-39A reaches 76.9%, the highest success rate.



ScatterPlot of Payload vs Outcomes



- LauchSites bearing Payload less than 5K kg (the upper figure) visually shows higher success rate than those sites with Payload more than 5K kg (the lower figure)

Section 5

Predictive Analysis (Classification)

Classification Accuracy

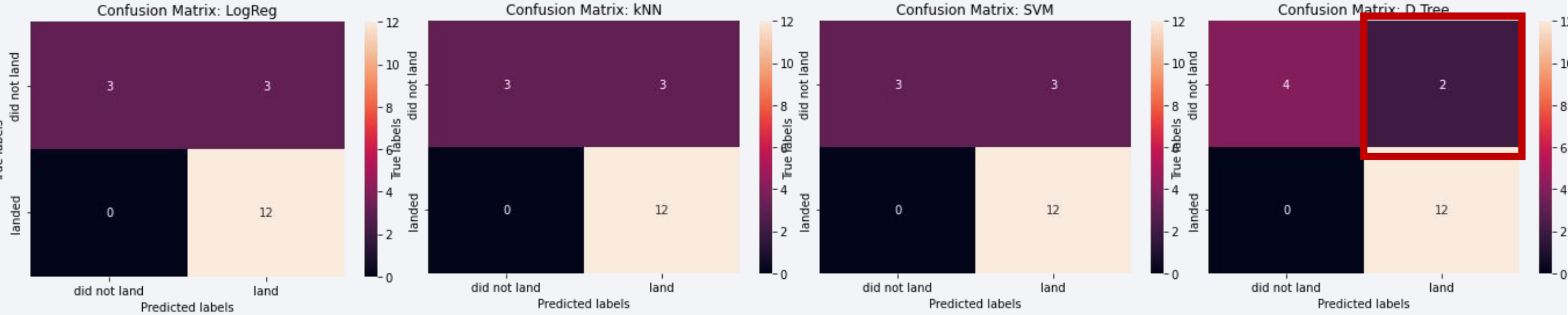
- Decision Tree performed the best valuated by Accuracy, J.Score and Avg F1-Score;
- Others performed identically

	Methods	Accuracy	J.Score	A.F1 Score
1	Logreg	0.8333	0.8000	0.8148
2	SVM	0.8333	0.8000	0.8148
3	D.Tree	0.8889	0.8571	0.8821
4	kNN	0.8333	0.8000	0.8148

	Methods	Accuracy	J.Score	A.F1 Score
1	Logreg	0.8333	0.8	0.8148
2	SVM	0.8333	0.8	0.8148
3	D.Tree	0.8333	0.8	0.8148
4	kNN	0.8333	0.8	0.8148

Confusion Matrix

- Confusion matrices of 4 models shows that:
- identical performance of Logreg, SVM and kNN with worse problem of false possitive of prediction, meaning that those classifiers or features of “Land” and “DidNotLand” can not distinguish two categories very well;
- there is no false negtive, lying in less amount of “DidNotLand” samples: 6, compared with “Land”:12
- D.Tree performance not reliably and the latest has less false possitive marked in RED



Conclusions & Discussion

- **Conclusions**
- Decision Tree performed the best valuated by Accuracy, J.Score and Avg F1-Score; others performed identically; and
- Prediction with false possitive meaning that those classifiers or features can not distinguish significantly two categories very well.
- **Discussion**
- Compared with classifiers, Feature engineering still give us more space explore so that to improve overall performance;
- There is no false negtive, lying in less amount of “DidNotLand” samples: 6, compared with “Land”:12, which should be proved by more samples; and
- Decision Tree’s performance is not reliable and the latest test has less false possitive than that of others. Its GridSearchCV shall be studied further to avoid unconsistency.
- GitHub URL predictive discussion (Click) or Copy URL to your browser:

https://github.com/bgo2002/IBM_DS_CAPSTONE_PROJECT/blob/main/Capstone_SpaceX_09_MachineLearning Prediction_Scilearn_250822_discuss.ipynb

Appendix

- Useful programming resources free to download:
- Toolkits of Python
- GUI of MySQL
- R Studio

Thank you!